UNIVERSITA' DEGLI STUDI DI CAGLIARI

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Informatica

Visual Engine for Reading On Network In Comprehensive Acceptation

Docenti di riferimento:

Prof. Massimo Bartoletti
Prof.ssa Silvia Corso
Prof. Gianni Fenu
Prof.ssa Barbara Pes
Prof. Riccardo Scateni

Candidati:

44782 Simone Barbieri
45190 Andrea Loddo
44829 Emanuele Mameli
44820 Alessandro Muntoni
45046 Livio Pompianu

ANNO ACCADEMICO 2011-2012

Indice

1. Introduzione

1.1. Nascita del progetto

Il progetto, sviluppato in accordo con il patrocinio della Regione Autonoma della Sardegna, è nato dall'esigenza di offrire un utile strumento a chi presenta problemi di dislessia e permette di offrire un servizio di sintesi vocale per libri e documenti, fornendo anche la possibilità di conservarli in remoto in modo da non doverli caricare più volte. Esistono numerosi programmi di sintesi vocale, tuttavia pochi riescono ad offrire il servizio in maniera facile e intuitiva: ne deriva la necessità di affiancare all'utente un tutor o una persona che già conosca il programma.
Il progetto vuole mettere a disposizione i propri servizi nel più semplice dei modi, così da offrire un utilizzo semplice anche ai meno esperti: gli utenti non necessitano di manuali e possono da subito immergersi nella lettura di libri o documenti già condivisi nel sistema oppure caricarne di propri nell'area privata.
È stata fatta una proposta di tesi verso il Professor Gianni Fenu riguardo la progettazione di un sistema che richiedesse la collaborazione di cinque individui. Il Professore ha individuato nel team le giuste modalità per realizzare un idea sua e della Dott.ssa Federica Loi (direttrice del sistema informativo nel reparto sanitario della Regione Sardegna). Il progetto ha inoltre il supporto della Dott.ssa Veronica Fadda che, attualmente, aiuta una bambina dislessica il cui nome è appunto Veronica.
L'ottica del progetto è maturata osservando come la bambina, tramite strumenti informatici, stia migliorando le proprie capacità molto rapidamente, affrontando i problemi precedentemente elencati con serenità e determinazione.
Da subito si è vista una suddivisione in non meno di cinque moduli: database, iterazione uomo-macchina, sicurezza, networking e diritto. I dettagli si sono evoluti nel corso del tempo, dando origine al progetto ora presentato.

V.E.R.O.N.I.C.A.

L'acronimo del sistema ha una stretta relazione con quanto il team si è preposto di realizzare. L'obiettivo fondamentale della tesi consiste nella realizzazione un sistema che offra, agli utenti dislessici, un servizio di lettura facilitata di un qualunque testo. Tale sistema è realizzato all'interno di una rete con l'intento di essere quanto più disponibile e comprensibile possibile nei confronti degli utenti dislessici.
V.E.R.O.N.I.C.A. nasce, infatti, con le prerogative di semplicità d'uso, chiarezza dei contenuti e garanzia di accessibilità verso qualsiasi utente dislessico, senza alcun tipo di distinzione legata al suo disturbo.
Per tali ragioni, "Visual Engine for Reading On Network In Comprehensive Acceptation" rappresenta il sistema realizzato ed i servizi offerti, con un importante occhio di riguardo verso gli utenti che lo utilizzeranno, nel rispetto dei requisiti richiesti per l'utilizzo del sistema.

1.2. Definizione di dislessia

Per avere un'idea generale del termine, è importante comprenderne la derivazione; la parola "dislessia" deriva dal greco e significa "difficoltà nella capacità di leggere le parole". Termini simili, ma con diverso significato, sono: "alessia", che comporta la perdita totale della capacità di leggere o "paralessia", laddove la perdita di tale capacità è solo parziale. La dislessia viene distinta in due principali categorie: dislessia evolutiva quando appare legata a

fattori maturazionali; dislessia specifica quando la sua presenza non altera le capacità generali di apprendimento (anche se, inevitabilmente, provoca una diminuzione del rendimento, ad esempio scolastico). Un disturbo che può presentarsi in concomitanza con la dislessia è la discalculia, che riguarda la debolezza della strutturazione cognitiva delle componenti numeriche fino alle compromissioni a livello procedurale di calcolo.
Più specificatamente, la Dislessia è un Disturbo Specifico dell'Apprendimento (DSA). Con questo termine ci si riferisce ai soli disturbi delle abilità scolastiche ed in particolare a:

- DISLESSIA : difficoltà nel leggere e scrivere;
- DISORTOGRAFIA : difficoltà di tipo ortografico;
- DISGRAFIA : difficoltà nel movimento fino-motorio della scrittura, cioè una resa formale inefficiente;
- DISCALCULIA : difficoltà a trattare i numeri o a fare calcoli.

Non sono, tuttora, ben chiare le cause della dislessia ed essendo sempre in corso un "dibattito" fra specialisti del settore, non tutti la ritengono una malattia.
Tuttavia, la visione moderna che si ha sulla dislessia è notevolmente mutata rispetto al passato: le sue manifestazioni, infatti, venivano considerate espressioni di scarsa intelligenza, pigrizia, mancanza di concentrazione. Tali considerazioni, ritenute al giorno d'oggi estremamente superficiali nel giudicare la malattia, comportano la necessità di un'attenzione minuziosa sia nella cura del disturbo, sia, più in generale, nel sostegno psicologico che deve essere garantito al dislessico per cercare di ovviare al suo problema.

Sintomi

I dislessici assolvono tutte le funzioni vitali svolte di norma dai non affetti da tale disturbo, sebbene alcune di esse siano svolte con maggiore difficoltà: in primis si trovano la lettura e la scrittura. I sintomi iniziali prevedono forte difficoltà di apprendimento, con inversioni di lettere ("la" anziché "al"), scambio di lettere simmetriche ("p" anziché "q"), troncamenti di simboli grafici, ritmo adagio nella lettura e osticità nell'utilizzo della memoria a breve termine. Lo sforzo di una agevole lettura può comportare anche un ritardo linguistico o una certa insicurezza nel comportamento.
Volendo riepilogare, le difficoltà principali causate dalla dislessia risultano essere:

- difficoltà di lettura e di scrittura;
- difficoltà di comprensione linguistica nella lettura;
- difficoltà di organizzazione personale;
- difficoltà di memoria e concentrazione;
- difficoltà di organizzare i pensieri in modo chiaro;
- scarsa immagine di sé.

Conseguenze

Nonostante i problemi visti, le persone dislessiche sviluppano metodi alternativi di apprendimento e comportamento, per questo nella maggior parte dei casi diventano delle persone creative, intuitive, con uno stile di pensiero e di comportamento piuttosto autonomo.
L'insuccesso scolastico determina una serie di fallimenti, che (inevitabilmente) compromettono l'autostima del soggetto. Tale pessimismo arreca un disagio psicologico quale timidezza, fobia sociale e depressione. Le sollecitazioni provenienti dal mondo adulto possono far sentire il soggetto dislessico ancora più incapace ed incompetente, ma soprattutto colpevole di non applicarsi allo studio come si dovrebbe fare o, peggio, come fanno gli altri. Il non sentirsi all'altezza delle aspettative, gli insuccessi scolastici, le difficoltà relazionali, la scarsa autostima possono indurre il soggetto dislessico ad un forte disimpegno scolastico, all'apatia, all'isolamento sociale.

Quando fare la diagnosi

La diagnosi viene solitamente effettuata alla fine del 2° anno della scuola primaria. Tuttavia, già alla fine del 1° anno delle elementari, profili funzionali compromessi e presenza di altri specifici indicatori diagnostici (ritardo del linguaggio e anamnesi familiare positiva per DSA) possono anticipare i termini della formulazione diagnostica. Un'ulteriore strumento per la rilevazione di queste difficoltà è lo screening, inteso come ricerca-azione da condurre direttamente nelle scuole, da parte di insegnanti formati con la consulenza di professionisti sanitari. Esso andrebbe condotto all'inizio dell'ultimo anno della scuola dell'infanzia con l'obiettivo di realizzare attività didattiche-pedagogiche mirate a potenziare le abilità deficitarie. Nel caso in cui alla fine dell'anno permangano significativi segnali di rischio è opportuna la segnalazione ai servizi sanitari per l'età evolutiva.
La diagnosi viene effettuata da un equipe multidisciplinare composta da Neuropsichiatria Infantile, Psicologo e Logopedista.
Un dislessico si stanca più facilmente ed ha perciò bisogno di molta più concentrazione

- Può leggere un brano correttamente e non cogliere il significato
- Può avere grosse difficoltà con le cifre (tabelline), la notazione musicale o qualsiasi cosa che necessita di simboli da interpretare
- Può avere difficoltà nella lettura e/o scrittura di lingue straniere (es. inglese, latino, greco, ecc..)
- Può scrivere una parola due volte o non scriverla
- Può avere difficoltà nel memorizzare termini specifici, non di uso comune
- Può avere difficoltà nello studio (storia, geografia, scienze, letteratura, problemi aritmetici) quando questo è veicolato dalla lettura e si giova invece dell'ascolto (es. registratori, adulto che legge, libri digitali)
- Non prende bene gli appunti perché non riesce ad ascoltare e scrivere contemporaneamente
- Quando si distrae da ciò che sta leggendo o scrivendo ha grosse difficoltà a ritrovare il punto.

Un dislessico lavora lentamente a causa delle sue difficoltà, perciò è sempre pressato dal tempo. È importante dire che pur avendo un modo differente di farlo, essi apprendono ugualmente ciò che studiano.

Diffusione

In Italia, secondo gli esiti tratti dall’allegato della Consensus Conference DSA (vedi Sitografia Allegato CC DSA), i disturbi specifici dell'apprendimento interessano il 2,5 - 3,5% della popolazione Italiana.

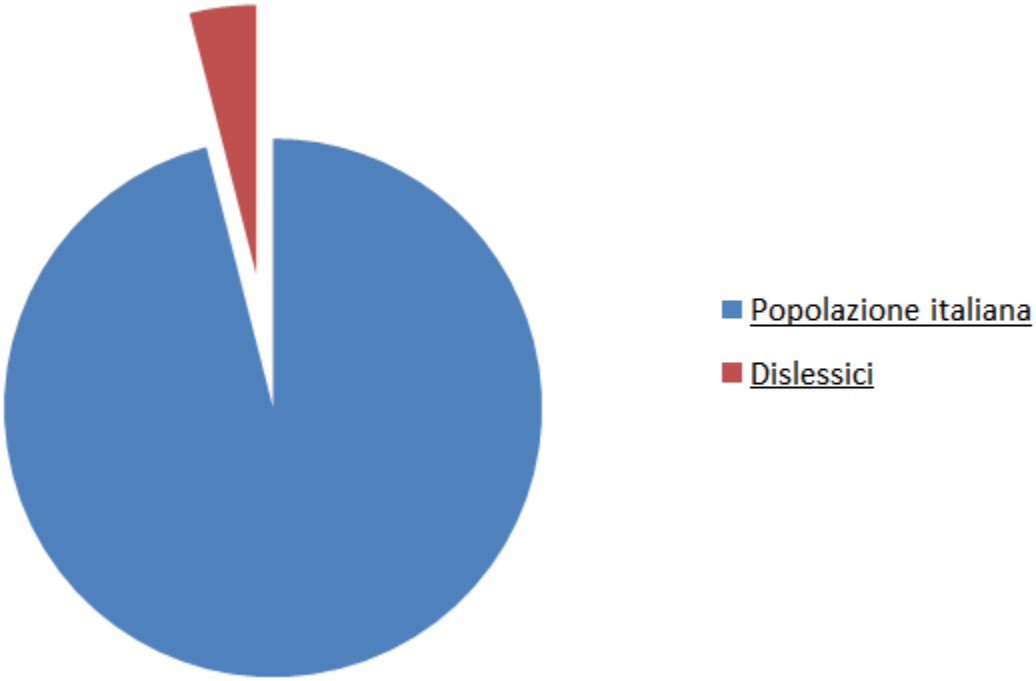

Immagine 1.1: Diffusione della dislessia in Italia

"Di fatto, anche se ancora non esiste uno specifico osservatorio epidemiologico nazionale, le informazioni che provengono dai Servizi di neuropsichiatria infantile indicano che i DSA rappresentano quasi il 30% degli utenti di questi servizi in età scolare e il 50% circa degli individui che effettuano un intervento riabilitativo." (vedi Sitografia Consensus Conference Disturbi specifici dell'apprendimento, del Ministero della Salute)

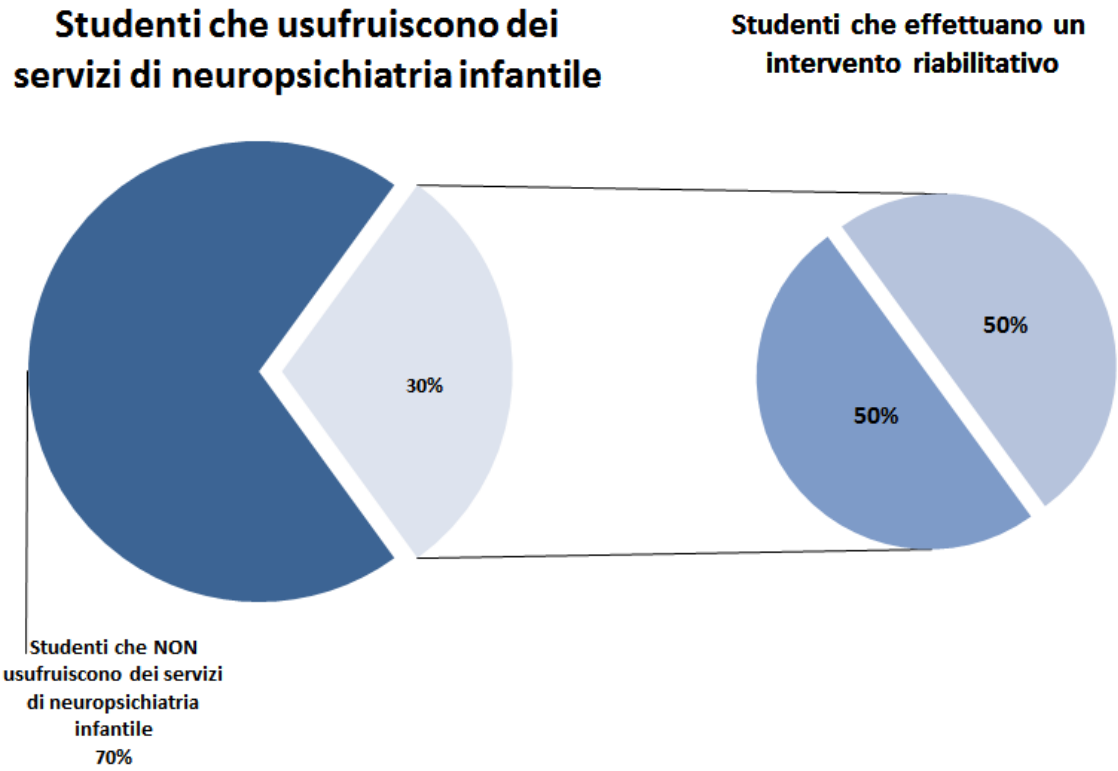

Immagine 1.2: Studenti che usufruiscono dei servizi di neuropsichiatria infantile

In Italia ne soffre almeno un milione e mezzo di persone, circa il 3% della popolazione, sebbene si tratti di stime piuttosto prudenti. Gran parte dei dislessici ha, infatti, avuto una carriera scolastica costellata di insuccessi, con abbandoni precoci e con conseguenze sociali a volte molto pesanti. In età scolastica la percentuale sale al 4-5%, su 7 milioni e 760 mila studenti, sono dislessici tra i 350 e i 400 mila (ma c'è chi pensa che siano cifre in difetto e che i bambini dislessici arrivino almeno a mezzo milione). In pratica un bambino o ragazzo per ogni classe (di 25 alunni). Ogni anno ci sono 25 mila nuovi casi e 6 volte su 10 alla dislessia si associa la difficoltà di scrittura (disgrafia e disortografia) e di calcolo (discalculia), anche se questi ultimi disturbi possono presentarsi da soli.
Nell'anno scolastico 2008-2009 gli iscritti nelle scuole della Sardegna, di ogni ordine e grado sono circa 220 mila . Facendo delle stime, ipotizzando 2 studenti ogni 20 o 25 (cioè quasi il 10% degli studenti), gli affetti da DSA sono circa 22 mila, solo nell'isola.

Immagine 1.3: Diffusione della dislessia in Sardegna

Trattamento

Non sempre gli insegnanti sono in grado o possono effettivamente risolvere il problema dell'alunno dislessico attraverso metodi di insegnamento convenzionali. La dislessia si cura con opportuni interventi, individuali o di gruppo. I trattamenti più efficaci prevedono l'utilizzo di tecnologia informatica, ma è importante intervenire nel sostegno psicologico, in quanto nessun obiettivo didattico potrà essere raggiunto se non vi è sufficiente stima di sé e motivazione al successo del soggetto interessato. Per questo, l'uso di software specifici permette di affrontare il problema con serenità. Vi sono dei software che permettono di esercitare le capacità di comprensione del testo scritto, o anche di migliorare gli aspetti metacognitivi.
Non è possibile prevenire la dislessia, ma riconoscerla al suo primo manifestarsi può consentire un migliore recupero.

2. Obiettivi

La tesi si pone come obiettivo lo sviluppo di un sistema che fornisca un supporto alle persone affette dalla DSA, ovvero Disturbo Specifico dell'Apprendimento, comunemente noto come dislessia. Si intende realizzare una piattaforma web-based che permetta all'utente dislessico, in linea generale, di poter effettuare l'upload di documenti personali e poterli leggere attraverso il sintetizzatore vocale presente all'interno del sistema.
Nel capitolo 3, Specifica dei requisiti, vengono approfonditi gli obiettivi della tesi attraverso una dettagliata descrizione delle specifiche individuate per la realizzazione di V.E.R.O.N.I.C.A.
Prima di affrontare i dettagli del progetto è bene dare una definizione di documento pubblico e privato, in quanto il termine pubblico è relativo solamente agli utenti che possono accedere al sistema, e quindi non necessariamente accessibile a chiunque; per questo sarebbe appropriato definirlo come "documento condiviso". In questa sede verranno usati i termini "documento condiviso" e "documento pubblico" come sinonimi. Per quanto concerne i documenti privati, la consultazione è consentita ai soli proprietari di tali documenti.

3. Specifica dei Requisiti

Si illustrano ora, con un ulteriore livello di dettaglio, i servizi offerti dal sistema. Tali servizi sono ricavati direttamente dalle linee guida definite al capitolo Obiettivi.

3.1. Definizione dei Requisiti

3.1.2. Requisiti Funzionali

Codice	R00
Nome	Accesso alla fascia di appartenenza assegnata
Descrizione	Ad ogni utente dislessico, previa registrazione, si assegna una fascia che lo contraddistingue secondo parametri tipici legati all'età ed alla dislessia. Tale fascia viene utilizzata per scopi prettamente statistici legati alla sindrome e per individuare un'interfaccia grafica gradevole all'utente

Codice	R01
Nome	Possibilità di effettuare un test
Descrizione	Ogni utente che accede a V.E.R.O.N.I.C.A. ha la possibilità di effettuare un test, di valenza statistica e non medica, riguardante la dislessia, allo scopo di ottenere una valutazione che lo possa, eventualmente, collocare in una delle fasce previste dal sistema

Codice	R02
Nome	Inserimento di un nuovo documento nell'area privata
Descrizione	Gli utenti dislessici hanno la possibilità di effettuare l'upload di documenti, quali libri e testi, all'interno della loro area personale privata. Tali documenti sono accessibili unicamente dall'utente che ne ha effettuato l'upload

Codice	R03
Nome	Inserimento di un nuovo documento nell'area pubblica
Descrizione	L'utente dislessico ha la possibilità di richiedere il caricamento di uno o più files, presenti all'interno della sua area privata, nell'area pubblica, liberamente accessibile da tutti gli utenti registrati a V.E.R.O.N.I.C.A. Tale richiesta deve essere in primo luogo verificata dagli amministratori, allo scopo di evitare problemi legali e di contenuti non sicuri, infine viene validata o rifiutata

Codice	R04
Nome	Utilizzo del sintetizzatore
Descrizione	Il sintetizzatore, pur essendo un modulo esterno non sviluppato dal team, costituisce l'aspetto più importante di tutto il sistema: l'utente utilizza tale strumento per ascoltare la lettura dei documenti caricati o quanto digitato nell'area di testo a sua disposizione. L'obiettivo primario del progetto infatti è alleviare il carico di lavoro intrapreso dall'utente nella lettura di un testo, indipendentemente dalle caratteristiche del testo caricato

Codice	R05
Nome	Inserimento o rimozione di un test per dislessia
Descrizione	Al personale medico viene data la possibilità di caricare o rimuovere un test per la dislessia (test senza valenza medica). Come qualsiasi altro contenuto del portale, il test dovrà essere validato da parte di un amministratore del sistema

Codice	R06
Nome	Visualizzazione delle schede risultati dei test
Descrizione	Il personale medico può visionare (per fini medici o statistici) il risultato dei test effettuati. Le modalità di visualizzazione dei contenuti devono rispettare i principi normativi in materia, con particolare riguardo alla normativa sulla privacy. Per maggiori informazioni è possibile visionare il capitolo 12, Aspetti Legali.

Codice	R07
Nome	Registrazione di un nuovo utente
Descrizione	Per ragioni definite nel paragrafo 11.4, Scelte progettuali orientate alla sicurezza, la registrazione al sistema non è libera per l'utente ma è delegata esclusivamente ai responsabili del portale, quindi gli amministratori devono poter registrare nuovi utenti

Codice	R08
Nome	Rimozione utente registrato
Descrizione	Gli amministratori del portale devono avere la possibilità di rimuovere utenti, qualora adottino comportamenti dannosi verso il sistema, verso gli altri utenti, oppure, più semplicemente, non desiderino più far parte del sistema V.E.R.O.N.I.C.A

Codice	R09
Nome	Lettura di un documento
Descrizione	Gli utenti possono leggere i documenti pubblici e i documenti privati da loro caricati

3.1.3. Requisiti Non Funzionali

Codice	R50
Nome	Accesso al sistema tramite autenticazione
Descrizione	Ogni utente deve accedere al proprio account tramite autenticazione. I dati per l'autenticazione, ovvero login e password, vengono forniti esclusivamente dal sistema

Codice	R51
Nome	Scambio dei dati cifrato
Descrizione	Lo scambio di dati sensibili tra client e server, ad esempio dati che riguardano l'autenticazione o dati sull'utente, devono essere cifrati prima dell'invio nel canale

Codice	R52
Nome	Verifica autenticazione utente
Descrizione	Il sistema deve accertarsi che l'utente che tenti di accedere ai diversi servizi del sistema, abbia il grado di autorizzazione richiesto

Codice	R53
Nome	Cambio di tema
Descrizione	Il sistema offre diversi temi utilizzabili, vista la ampia fascia d'età degli utilizzatori finali di V.E.R.O.N.I.C.A. All'utente deve essere concessa la possibilità di cambiare il proprio tema

Codice	R54
Nome	Logout
Descrizione	Il sistema, per motivi di sicurezza, deve gestire le sessioni utente. Assicurare il logout automatico dopo un tempo determinato di inattività e offrire all'utente stesso la possibilità di eseguire il logout al termine della propria attività

Codice	R55
Nome	Pagine web da implementare in HTML5, JavaScript e PHP
Descrizione	Il linguaggio utilizzato per implementare le pagine web deve essere HTML5, con il supporto dI PHP per la programmazione lato server ed, eventualmente, JavaScript per la programmazione lato client

Codice	R56
Nome	Interfaccia utente da implementare in CSS3
Descrizione	L'interfaccia utente, indipendentemente dalla fascia d'età trattata o il tipo di utente che utilizzi il sistema, deve essere implementata mediante CSS3

Codice	R57
Nome	DBMS MySQL
Descrizione	Il Data Base Management System per la gestione della base di dati deve essere MySQL

Codice	R58
Nome	Salvataggio dati utente
Descrizione	I dati degli utenti devono essere salvati sul database, siano essi anagrafici o medici.

Codice	R59
Nome	Salvataggio dati documento
Descrizione	I dati dei diversi documenti caricati sul sistema devono essere salvati all'interno del database

Codice	R60
Nome	Salvataggio dati login
Descrizione	I dati di tutti i login effettuati devono essere salvati sul database

3.2. Architettura del sistema

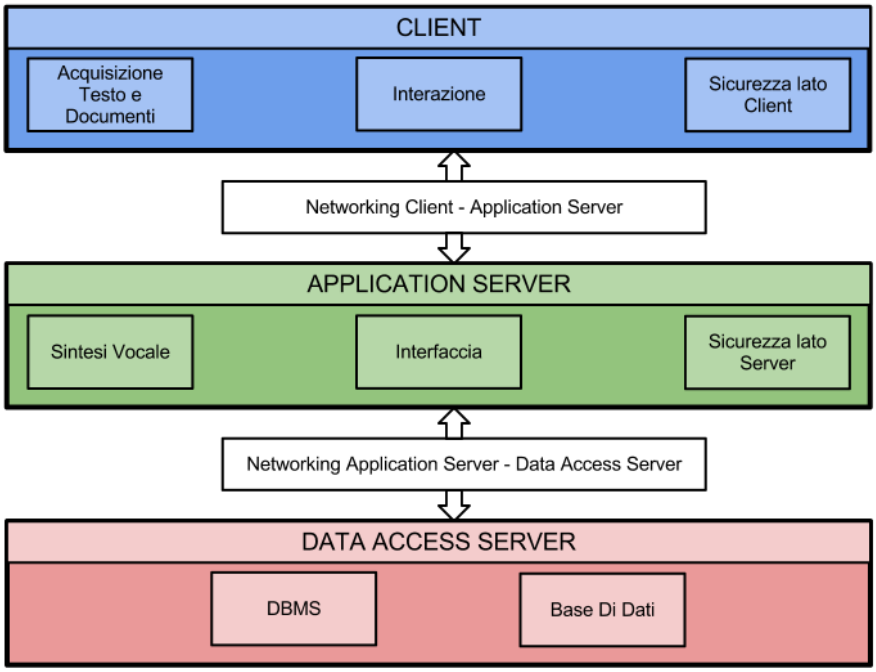

Immagine 3.1: Architettura del Sistema

Il sistema, come rappresentato nella Figura 3.1 è basato su un'architettura a tre livelli, o strati:

- Client;
- Application Server;
- Data Access Server.

I diversi livelli sono costituiti da uno o più moduli, che soddisfano le funzionalità del sistema previste: nello specifico, si dispone di moduli realizzati dal team o moduli pronti importati all'interno del sistema, di cui un esempio è il software di sintesi vocale.
Il Networking controlla la comunicazione degli strati: per tale motivazione il modulo non è stato collocato all'interno di nessuno dei livelli ma tra i diversi strati.
Nella Tabella 3.1 sono rappresentati i moduli individuati in V.E.R.O.N.I.C.A.

Modulo	Livello di Appartenenza	Requisiti soddisfatti
Acquisizione Testo e Documenti	Client	R02, R03, R05
Interazione	Client	R00, R01, R06, R09, R50, R53
Sicurezza lato client	Client	R51
Sintesi Vocale	Application Server	R04
Interfaccia	Application Server	R55, R56
Sicurezza lato server	Application Server	R07, R08, R51, R52
DBMS	Data Access Server	R57
Base di Dati	Data Access Server	R58, R59, R60
Networking	Interfacciamento strati	R5

Tabella 3.1: Moduli del sistema

3.3. Modello del sistema

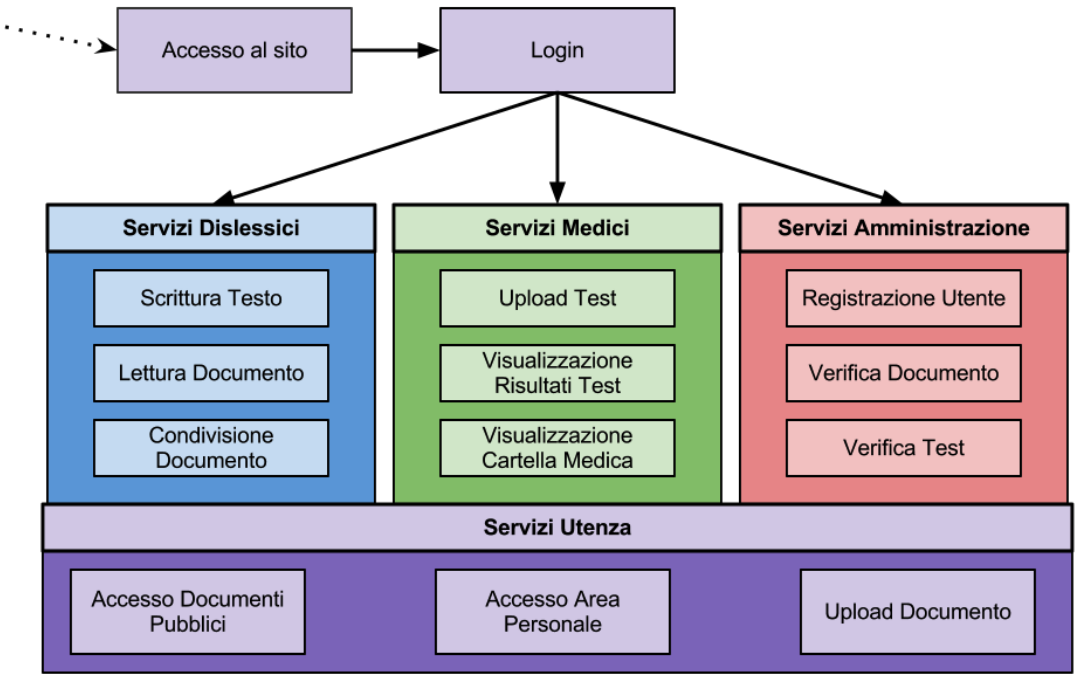

Figura 3.2: Modello concettuale del sistema

La Figura 3.2 mostra, in maniera schematica e gerarchica, l'elenco delle funzionalità offerte da V.E.R.O.N.I.C.A..

Si precisa, in primo luogo, che qualsiasi tipologia di servizi è accessibile previa registrazione al sistema ed inserimento delle proprie credenziali di accesso nell'apposita pagina di Login.
A tal proposito, sono stati volutamente indicati solamente i servizi accessibili dalle tipologie di utenti registrati, dal momento che ai visitatori del sito è offerta unicamente la possibilità di effettuare un test.
Ogni utente registrato, indipendentemente dalla classificazione all'interno del sistema, dispone di una serie di servizi, indicati nel modello come "Servizi Utenza"; elencati e descritti di seguito:

- Accesso Documenti Pubblici: si tratta dell'area in cui sono contenuti i documenti resi pubblici mediante condivisione da parte di un utente, ed in seguito ad approvazione effettuata da un amministratore;
- Accesso Area Personale: ogni utente dispone di una propria area personale, nella quale vengono memorizzati i documenti personali precedentemente caricati sul sito, in modo che se ne possa disporre in qualsiasi momento senza rieffettuare l'upload;
- Upload Documento: è offerta la possibilità di effettuare l'upload di un documento, sia esso un libro od un semplice testo di appunti personali, all'interno dello spazio web personale riservato all'utente.

Allo stesso modo, è noto che gli utenti registrati siano distinti in base alla tipologia di account da essi registrato: Dislessico, Medico o Amministratore; ogni tipologia di utente possiede servizi specifici.
Come mostra il modello, i servizi principali offerti all'utente dislessico sono:

- Scrittura Testo: ai dislessici è fornita una text-box nella quale è possibile scrivere del testo, senza dover affrontare la procedura di upload di un documento. Il testo scritto all'interno di tale area è immediatamente leggibile dal sintetizzatore vocale presente all'interno di V.E.R.O.N.I.C.A.;
- Lettura Documento: ogni documento presente all'interno di V.E.R.O.N.I.C.A. può essere letto dal sintetizzatore vocale presente; in questo senso, gli utenti dislessici possono utilizzare tale funzionalità per la lettura dei documenti ai quali hanno accesso, siano essi dei documenti privati o pubblici;
- Condivisione Documento: esiste, per ogni utente dislessico, la possibilità di richiedere la condivisione alla comunità di V.E.R.O.N.I.C.A. dei propri documenti privati. Tale richiesta viene immediatamente inoltrata agli amministratori di V.E.R.O.N.I.C.A., è essenziale che, prima di pubblicare un qualsiasi documento, ne venga verificato il suo contenuto ed approvata o rifiutata la sua pubblicazione;

Allo stesso modo, gli utenti medici dispongono di servizi specifici:

- Upload Test: ogni medico ha la possibilità di effettuare l'upload di test, dedicati agli utenti dislessici. I test sono pensati per avere una diagnosi indicativa sul grado della dislessia dell'utente, ma non hanno alcuna valenza medica specifica. Come per la Condivisione Documenti, anche questo servizio prevede una verifica del contenuto del test ed una conseguente approvazione o rifiuto della pubblicazione;
- Visualizzazione Risultati Test: gli utenti medici possono verificare i dati dei test effettuati dagli utenti dislessici. Lo scopo principale di tale servizio è la fornitura di dati statistici relativi al disturbo. Va specificato che tali dati non sono comprensivi di informazioni sensibili degli utenti che hanno svolto i test;
- Visualizzazione Cartella Medica: agli utenti medici è offerta la possibilità di visualizzare le informazioni sanitarie di utenti specifici, previo consenso fornito dagli stessi;

Infine, per la categoria di utenti amministratori, sono forniti i seguenti servizi:

- Registrazione Utente: gli amministratori di V.E.R.O.N.I.C.A., una volta completate le pratiche di richiesta di registrazione previste hanno la possibilità di registrare un nuovo utente, inserendo negli appositi campi, i dati identificativi richiesti. È compito del sistema fornire, mediante e-mail, le credenziali di accesso all'utente. In questo modo è garantita totale trasparenza verso gli utenti registrati;
- Verifica Documento: come definito in precedenza, è compito degli amministratori di V.E.R.O.N.I.C.A. la verifica del contenuto di un documento, di cui è fatta richiesta di pubblicazione da parte di un utente; tale servizio garantisce in primo luogo la sicurezza dei contenuti resi pubblici, ed, infine, evita problemi di natura legale, maggiormente approfonditi nel capitolo Aspetti Legali (cap. 12, Aspetti Legali);
- Verifica Test: essenzialmente, il funzionamento è lo stesso del servizio Verifica Documento, ma si riferisce alla verifica del contenuto dei test caricati dai medici, da sottoporre agli utenti dislessici.

3.4. Requisiti Base di Dati

Il sistema deve memorizzare i dati degli utenti, che in generale possiedono i seguenti attributi:

- Username (unico);
- Password;
- Nome;
- Cognome;
- Mail (unica);
- Data;
- Sesso.

Vi sono tre tipi di utenti: Dislessico, Medico e Amministratore. Oltre agli attributi elencati sopra, l'utente dislessico dovrà avere un attributo che caratterizza il grado di dislessia, mentre per quanto riguarda il medico andrà memorizzata la specializzazione. Per ogni utente, inoltre, si dovrà conoscere l'amministratore che ha creato il suo account e se è attivo (può essere un account rimosso).
Il sistema permette agli utenti di caricare dei documenti. Ogni documento è caratterizzato dai seguenti attributi:

- ID (unico);
- Titolo;
- Autore;
- Proprietario (Utente che ha caricato il documento).

Di ogni documento dovranno essere memorizzati il testo e le immagini che lo compongono. Inoltre, ogni documento può essere pubblico o privato, e, nel caso sia pubblico, è necessario memorizzare l'amministratore che ha approvato il documento. Ogni documento può essere un Libro o un Testo generico. Nel caso sia un libro, è necessario memorizzare:

- Copertina (tenendo traccia del formato);
- Anno di uscita;
- Editore;
- Genere;
- Descrizione.

Mentre, nel caso sia un testo generico, si memorizzano:

- Tipologia;
- Materia;
- Argomento.

Un generico utente, anche non registrato, può effettuare un test (caricato da un medico) che gli permette di avere un feedback sul suo grado di dislessia. Il Test è caratterizzato da:

- ID (unico);
- un testo da leggere;
- numero di sillabe del testo;
- Grado di difficoltà del test.

Inoltre, per ogni test, è necessario memorizzare il medico che l'ha caricato e l'amministratore che approva il test.
Per ogni accesso di ogni utente al sistema, è necessario memorizzare data e ora di accesso.

3.5. Diagramma UML

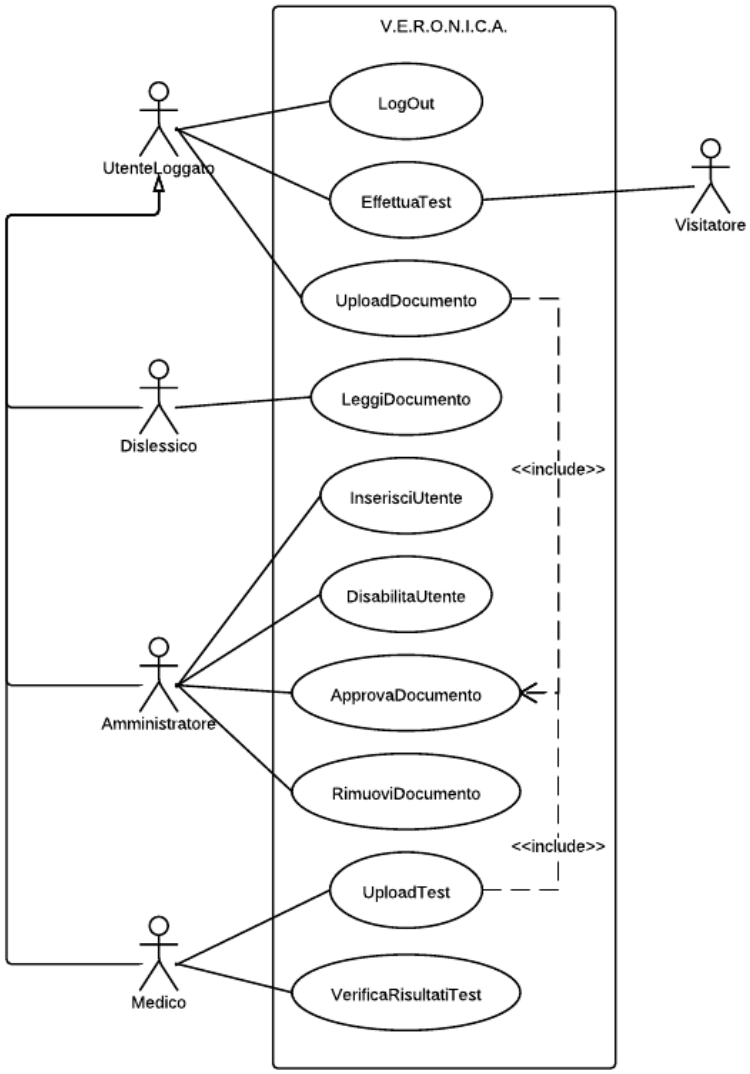

Immagine 3.3: Diagramma UML

3.5.1. Descrizione dei casi d'uso

ID	U1
Nome	LogOut
Descrizione Breve	Permette di chiudere la sessione.
Attori	UtenteLoggato
Precondizioni	L'utente deve essere registrato e aver effettuato il login.
Sequenza Principale	1. L'Utente attiva il caso d'uso Logout. 2. La sessione attuale viene terminata e non è più possibile visualizzare le pagine del sito.
Postcondizioni	L'utente viene reindirizzato alla pagina di login

ID	U2
Nome	EffettuaTest
Descrizione Breve	Permette di effettuare un test senza valenza medica, che indica quante sillabe al secondo l'utente riesce a leggere.
Attori	Utente, Visitatore
Precondizioni	Nessuna
Sequenza Principale	1. L'Utente o il Visitatore attiva il caso d'uso EffettuaTest. 2. Viene effettuato il test.
Postcondizioni	Viene visualizzato il risultato del test

ID	U3
Nome	UploadDocumento
Descrizione Breve	Viene caricato un documento sul server.
Attori	UtenteLoggato
Precondizioni	L'utente deve essere registrato e aver effettuato il login.
Sequenza Principale	1. L'Utente attiva il caso d'uso UploadDocumento. 2. L'Utente sceglie se rendere il documento pubblico o privato. 3. L'Utente inserisce i dati del documento. 4. Il documento viene caricato. Se il documento viene caricato come pubblico: 5. Include il caso d'uso ApprovaDocumento.
Postcondizioni	Viene visualizzato l'esito del caricamento

ID	U4
Nome	LeggiDocumento
Descrizione Breve	Viene aperto un documento per la lettura.
Attori	Dislessico
Precondizioni	L'utente deve essere registrato e aver effettuato il login.
Sequenza Principale	1. L'Utente attiva il caso d'uso LeggiDocumento. 2. L'Utente effettua una ricerca, visualizza la lista di tutti i documenti pubblici oppure visualizza la lista dei documenti privati. 3. L'Utente sceglie un documento dalla lista.
Postcondizioni	Il documento scelto viene mostrato all'Utente

ID	U5
Nome	InserisciUtente
Descrizione Breve	Un utente viene registrato.
Attori	Amministratore
Precondizioni	L'Amministratore deve aver effettuato il login.
Sequenza Principale	1. L'Amministratore attiva il caso d'uso InserisciUtente. 2. L'Amministratore inserisce i dati dell'utente che vuole registrare. 3. L'utente viene aggiunto al sistema.
Postcondizioni	Viene visualizzato l'esito dell'operazione

ID	U6
Nome	DisabilitaUtente
Descrizione Breve	L'account di un utente viene disabilitato.
Attori	Amministratore
Precondizioni	L'Amministratore deve aver effettuato il login.
Sequenza Principale	1. L'Amministratore attiva il caso d'uso DisabilitaUtente. 2. L'Amministratore visualizza la lista degli utenti. 3. L'Amministratore sceglie l'utente che vuole disabilitare. 4. L'account dell'utente scelto viene disabilitato.
Postcondizioni	Viene visualizzato l'esito dell'operazione

ID	U7
Nome	ApprovaDocumenti
Descrizione Breve	Un documento caricato come pubblico viene autorizzato e reso disponibile per tutti.
Attori	Amministratore
Precondizioni	L'Amministratore deve aver effettuato il login e il documento deve essere stato condiviso come pubblico da parte di un altro utente.
Sequenza Principale	1. L'Amministratore attiva il caso d'uso ApprovaDocumenti. 2. Un documento è caricato come pubblico. 3. L'Amministratore visualizza la lista dei documenti non ancora approvati. 4. L'Amministratore controlla il documento. 5. L'Amministratore approva o non approva il documento.
Postcondizioni	Il documento, se approvato, può essere visualizzato da tutti gli utenti

ID	U8
Nome	RimuoviDocumento
Descrizione Breve	Un documento pubblico viene eliminato dal sistema.
Attori	Amministratore
Precondizioni	L'Amministratore deve aver effettuato il login.
Sequenza Principale	1. L'Amministratore attiva il caso d'uso RimuoviDocumento. 2. L'Amministratore visualizza la lista di tutti i documenti caricati. 3. L'Amministratore seleziona il documento che vuole rimuovere. 4. Il documento selezionato viene rimosso.
Postcondizioni	Viene visualizzato l'esito dell'operazione

ID	U9
Nome	UploadTest
Descrizione Breve	Viene caricato un test sul server.
Attori	Medico
Precondizioni	Il Medico deve aver effettuato il login.
Sequenza Principale	1. Il Medico attiva il caso d'uso UploadTest. 2. Il Medico inserisce i dati del test. 3. Il test viene caricato.
Postcondizioni	Viene visualizzato l'esito dell'operazione

ID	U10
Nome	VerificaRisultatiTest
Descrizione Breve	Il Medico visualizza i risultati dei test dei suoi pazienti.
Attori	Medico
Precondizioni	Il Medico deve aver effettuato il login.
Sequenza Principale	1. Il Medico attiva il caso d'uso VerificaRisultatiTest. 2. Il Medico visualizza la lista dei propri pazienti. 3. Il Medico seleziona il paziente di cui vuole vedere i risultati.
Postcondizioni	I risultati del paziente scelto vengono visualizzati

4. Piano di Progetto

4.1. Modello del processo

Considerata la vastità del progetto affrontato, è stato necessario garantire un'organizzazione molto precisa del lavoro da svolgere e dei compiti all'interno del team: in primo luogo si discute il modello del processo scelto per sviluppare il sistema V.E.R.O.N.I.C.A..
Viene scartata l'ipotesi di un semplice modello a cascata, principalmente perché non si ha una visione iniziale precisa dell'insieme dei requisiti. È necessario utilizzare un modello ciclico, che permetta di rivisitare le fasi progettuali precedenti in base alle esigenze del cliente e del team. Nello specifico, si è optato per l'utilizzo di un modello di sviluppo evolutivo prototipale: in tale modello si producono dei prototipi attraverso un percorso riguardante differenti fasi, nel quale ogni singola fase può essere reiterata un numero arbitrario di volte e in maniera indipendente dalla fase di progetto attuale.
Ad esempio, se si sta svolgendo la validazione del prototipo attuale, la fase successiva potrebbe, nuovamente, essere l'analisi dei requisiti in quanto si potrebbe dover fornire una nuova interpretazione del sistema; oppure, direttamente la fase di implementazione se si vogliono implementare nuovi aspetti o aspetti già esistenti. Ancora, si potrebbe decidere di passare alla progettazione del sistema o approfondire la fase di test.
La particolarità di tale modello è proprio non avere un ordine specifico per affrontare le fasi, poiché si procede verso la fase che più conviene in base alle esigenze di sviluppo attuali; il modello è rappresentato nella Figura 4.1. Inoltre, la produzione di prototipi permette al team di avere dei feedback costanti prima della realizzazione del prodotto finito.

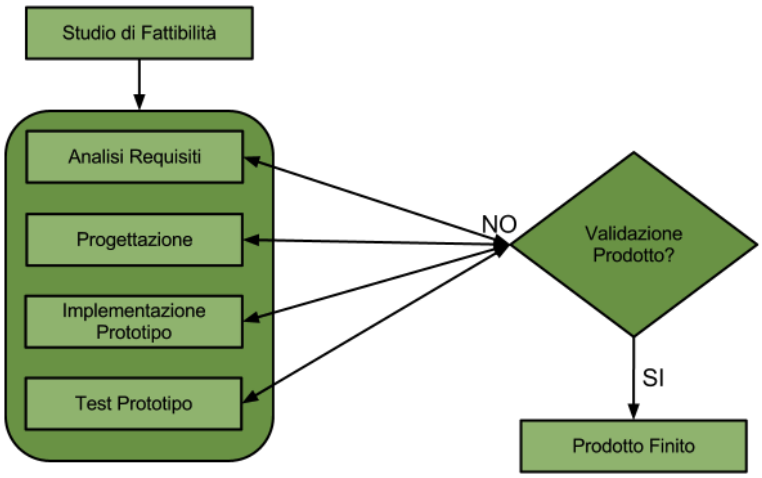

Immagine 4.1: Modello del Processo

4.2. Struttura organizzativa, moduli e aree tematiche

Il sistema V.E.R.O.N.I.C.A. è organizzato in diversi moduli, riguardanti aspetti legati all'implementazione del sistema stesso:

- modulo di interazione uomo-macchina: è il modulo in cui viene gestita la parte del sistema rivolta all'utente: interfaccia grafica, intuitività, ecc;
- modulo di basi di dati: è il modulo in cui si trattano tutte le informazioni che è necessario memorizzare per far si che il sistema possa fornire determinate funzionalità e rispettare i requisiti;

- modulo di internetworking: è il modulo che gestisce le connessioni e le sessioni, protocolli e architetture utilizzate dal sistema;
- modulo di sicurezza: è il modulo che garantisce un trasferimento e un immagazzinamento di dati sicuro;
- modulo di diritto: gestisce tutti gli aspetti legali legati al sistema.

4.3. Organizzazione del team

Il progetto, come già accennato e come è naturale che sia, prevede la creazione di diversi moduli, posti in relazione tra loro. Tali moduli devono essere realizzati secondo principi di coerenza ed integrazione, in quanto, in questo modo, si può ottenere il massimo dai singoli moduli e dall'intero progetto. Ad ogni modo, ogni modulo è gestito, principalmente, da un membro del team, anche se in alcuni casi sono state necessarie delle collaborazioni in vista della quantità di lavoro da sviluppare.
Procedendo con la fase di sviluppo si è giunti all'assegnazione delle seguenti responsabilità:

- responsabile modulo DataBase : Alessandro Muntoni, Andrea Loddo;
- responsabile modulo Sicurezza : Livio Pompianu;
- responsabile modulo interazione : Emanuele Mameli e Simone Barbieri;
- responsabile modulo interfaccia : Simone Barbieri e Emanuele Mameli;
- responsabile modulo Networking : Andrea Loddo;
- responsabile modulo Aspetti Legali : Team.

Affinché il progetto arrivasse ad un alto livello, ogni responsabile, oltre a gestire il proprio settore, ha valutato e discusso le problematiche e le modalità, con cui ha affrontato le diverse tematiche, con il resto del team. Tale aspetto ha permesso un'alta efficienze nella stesura dei contenuti ed ulteriori riflessioni su problematiche, eventualmente non sviluppate, dal responsabile del modulo analizzato.
È stato necessario mettere a disposizione dei relatori gli sviluppi dei rispettivi moduli assegnati. Ognuno, nel proprio campo, ha valutato quali potevano essere le scelte corrette che il team avrebbe potuto adottare affinché V.E.R.O.N.I.C.A. fosse adatta alle future esigenze.
In particolare, vengono di seguito indicati i relatori ed i relativi moduli sottoposti:

- Gianni Fenu : Networking e la direzione dell'intero progetto;
- Riccardo Scateni : Interazione Uomo-Macchina;
- Massimo Bartoletti : Sicurezza;
- Barbara Pes : Base di dati;
- Silvia Corso : Aspetti legali.

Lo sviluppo di ogni parte ha previsto oltre allo studio della materia sottoposta, uno studio della dislessia allo scopo di individuarne problematiche e possibili soluzioni. A questo punto, è stato affrontato in merito alle diverse problematiche, modellando il progetto affinché potesse essere estendibile e integrabile. Infine le fasi di testing, essenziali in quanto hanno permesso di scoprire quali problemi potessero essere risolti nelle diverse fasi affrontate.
L'ottimo rapporto tra le varie parti ha permesso la progettazione dell'intero sistema, in breve tempo. Il primo prototipo, sebbene non comprenda tutte le funzionalità, permette di avere un'idea ampia delle funzionalità che il Team intende fornire in futuro.

5. Tecnologie utilizzate

5.1. Server

Il server è dotato di Sistema Operativo Ubuntu, nella sua variante Xubuntu, con motore grafico XFCE, a causa della potenza ridotta della macchina a disposizione.
I principali componenti software utilizzati sono:

- Apache;
- MySQL;
- PHP.

Il server utilizza differenti porte per la comunicazione con l'esterno. Precisamente si tratta di:

- SSH;
- FTP;
- HTTP.

Per maggiori informazioni sui software, nonché sulla gestione lato server, le porte e i protocolli utilizzati, si veda quanto definito al capitolo 10, Modalità di accesso.
È importante evidenziare che anche il sintetizzatore utilizzato rientra tra i software utilizzati nel server, in quanto è prevista un'unica installazione, fornita sul server: non si possiede, infatti, una copia per ogni computer client che utilizzerà V.E.R.O.N.I.C.A.
Per maggiori informazioni sul software di sintesi vocale è possibile consultare il capitolo dedicato all'interazione, paragrafo Approfondimenti sul sintetizzatore.

5.2. Client

Per lo sviluppo del sistema, e in particolare dell'interfaccia, sono stati utilizzati i seguenti linguaggi:

- HTML5;
- CSS3;
- JavaScript.

La scelta del team è ricaduta su HTML5, l'ultima versione del linguaggio di markup ormai supportata dai principali browser e già ampiamente utilizzato (vedi statistiche). Tale linguaggio è utilizzato per definire la struttura della pagina web, lasciando il compito di modellare, colorare e posizionare ai fogli di stile, o CSS; essi hanno permesso la realizzazione di temi, animazioni e altri effetti grafici che rendono il sistema più gradevole da utilizzare. Si è scelto di utilizzare l'ultima versione di HTML e CSS per motivi legati a sicurezza, semplicità del codice, potenzialità e perché integrano funzioni che sarebbero altrimenti dovute essere scritte da zero; inoltre, permette alcuni facili adattamenti tra i vari dispositivi, ad esempio tablet o smartphones, svincolando il team da queste scelte e permettendo di concentrarsi su aspetti più importanti.
Altro fattore che ha contribuito alla scelta di versioni molto recenti, è il fatto che i migliori e più utilizzati browser supportano ampiamente le tecnologie descritte. Il sito html5test.com permette di testare il grado di supporto all'HTML5 del proprio browser, tramite un punteggio in scala da 0 e 500 (più eventuali bonus): vengono evidenziate quante e quali funzioni sono supportate da ogni browser.
Visualizzando le statistiche presenti sul sito, cliccando sulla voce altri browser, si può notare quali sono le funzioni supportate dai principali browser e i loro punteggi (anche quelli adottati da smartphones, tablet e tv).

Per meglio comprendere viene riportato il seguente grafico:

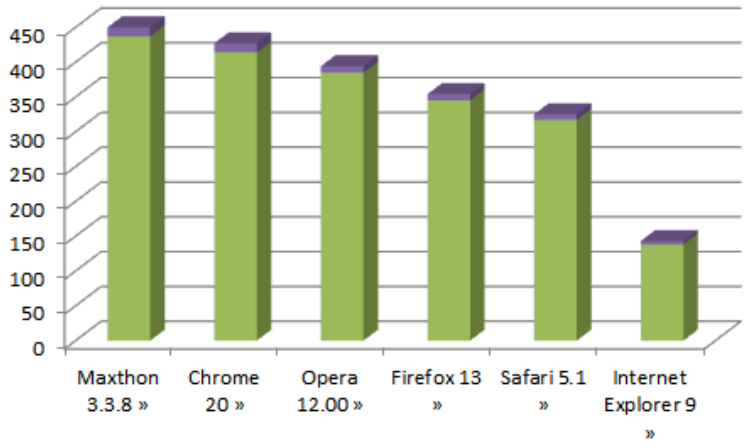

Immagine 5.1: Supporto dei browser per HTML5

Il supporto dei browser, anche nelle piattaforme mobili e altri supporti come le televisioni, è alto ed è destinato a crescere. Tra i motivi per cui HTML5 sta riscuotendo un successo tanto grande si ha certamente il supporto ai contenuti multimediali, in particolare audio e video; questo è anche uno dei motivi principali per cui il team ha adottato questo linguaggio. Infatti, il team può ritenersi ampiamente soddisfatto di non richiedere all'utente alcun plugin aggiuntivo, come per esempio Adobe® Flash Player.
Infine, JavaScript è stato utilizzato per realizzare animazioni o effetti che non è possibile realizzare con HTML e CSS, oppure per aggiungere funzionalità più complesse. Il suo supporto è essenziale per elaborare alcune informazioni lato client, ad esempio è grazie a JavaScript che è possibile ottenere il testo da inviare al sintetizzatore. Ricopre anche un ruolo fondamentale per quanto riguarda la sicurezza.

6. Studio dell'interfaccia e dell'interazione

6.1. Contesto ed Obiettivi

Affinchè il sistema rimanga impresso all'utente e lo invogli ad un utilizzo costante non è sufficiente che sia funzionale. Uno degli aspetti maggiormente esposti al giudizio dell'utente è l'usabilità del sistema e con essa anche l'estetica; è quindi necessario un meticoloso lavoro per renderla usabile e gradevole, senza perdere di funzionalità. Un altro aspetto importante è la facilità di utilizzo del sistema, ovvero fare in modo che l'utente sia in grado di utilizzare tutte le funzionalità senza dover ricorrere a una documentazione o ad altri aiuti.
V.E.R.O.N.I.C.A. è un sistema realizzato per utenti affetti da dislessia, quindi l'interfaccia è l'aspetto da considerare maggiormente affinché il sistema venga adottato e utilizzato con piacevolezza. L'interfaccia, quindi, deve essere soggetta ad un'attenta analisi, allo scopo di renderla usabile ed al fine di avere una serie di servizi che diano all'utente una web-application completa e un punto di riferimento per lo sviluppo delle capacità di lettura di chi possiede tale svantaggio.
Nel presente capitolo si presenta un'analisi sull'interfaccia del sistema e sull'interazione dell'utente con essa.

6.2. Obiettivi principali

Per la realizzazione del sistema, si sono posti i seguenti obiettivi principali:

- minimalità: realizzare un'interfaccia intuitiva e facile da usare;
- adatta a tutti: realizzare un'interfaccia per ogni fascia d'età;
- personalizzabile: dare all'utente la possibilità di utilizzare temi diversi.

Nonostante gli obiettivi elencati siano un piccolo numero, essi racchiudono al loro interno una quantità ingente di aspetti sul quale focalizzare l'attenzione e basare lo sviluppo.

6.3. Scelte di Design

Inizialmente, per quanto riguarda lo sviluppo del sistema, la scelta del team, guidato da persone esperte nel settore, è ricaduta su Drupal, un framework che avrebbe dovuto semplificare il lavoro di numerose parti, compreso lo sviluppo dell'interfaccia, in quanto ne offre una già pronta, ma completamente modificabile.
Il team ha dapprima provato a fare uso e studiato dei moduli che lo compongono, inoltre ne ha cercato di altri che rispondessero meglio alle proprie esigenze.
La scelta di abbandonare questo sistema, come descritto nel capitolo 8.7.2, si è concretizzata dopo qualche settimana di utilizzo e di studio. I motivi dell'abbandono del framework legati all'interfaccia sono principalmente due:

- la creazione di una nuova interfaccia, che si è evoluta in quella utilizzata nel prototipo, resa modulare grazie all'uso del PHP;
- l'uso, da parte di Drupal, di tabelle pronte (nel database) che sarebbero state poco utili per come è stato concepito il sistema e per gli obiettivi predisposti.

Il team, sviluppando l'intero sistema dalle fondamenta, senza l'utilizzo di framework, si è reso conto che questo tipo di sviluppo sarebbe stato più semplice per la realizzazione di V.E.R.O.N.I.C.A.
Vengono quindi effettuate delle scelte legate all'implementazione della web-application: quali linguaggi utilizzare e per quali tipi di client deve essere pensato (per esempio tablet o altri dispositivi, vedi il paragrafo 5.2 Client per maggiori approfondimenti).

6.4. I colori

La scelta dei colori è uno dei parametri a volte più difficili, ma consente di modellare l'apparenza mettendo in evidenza gli aspetti ritenuti più importanti, allo stesso tempo si cerca di non limitare l'uso di un unico gradiente per non limitare le personalizzazioni (ad esempio un bambino potrebbe apprezzare il blu, mentre una bambina preferirebbe il rosa). Si è scelto, quindi, di realizzate diverse interfacce. Per dare maggiore libertà agli utenti, si possono unire diversi colori ad ogni interfaccia.
I colori scelti, non sono in tinta unita; è stato impostato un gradiente per ogni colore, in modo da rendere l'interfaccia più leggera e gradevole. Inoltre, le parti centrali della pagina hanno un colore più chiaro, per dare un senso di distacco dalle intestazioni.
Di default, il tema è di una tonalità blu. È stato quindi utilizzato un gradiente (nel tema di default si va da #13486C a #2D72A0) per le intestazioni, #BEDAF5 per la parte centrale della pagina e #DBEAF9 per i menu laterali.

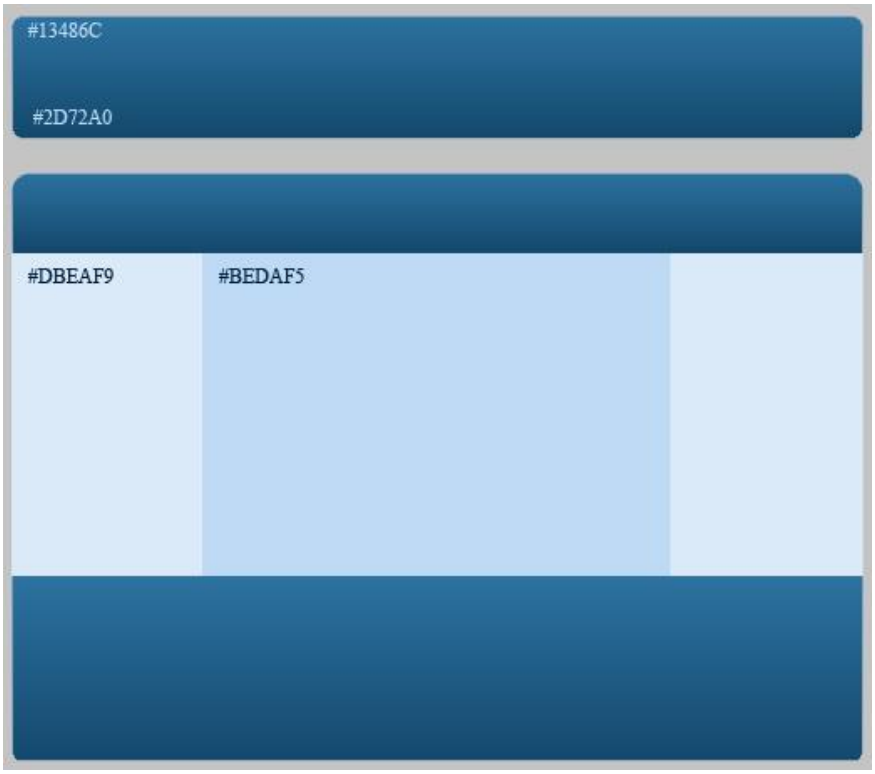

Immagine 6.1: Colori

Come colore di background, invece, è stato scelto un colore scuro, per dare risalto all'applicazione, dandole un secondo livello che la porta in primo piano, differenziandola dal solito sito web.

6.5. Il testo, caratteri per dislessici

Per agevolare la lettura dei testi su un monitor si punta ai caratteri senza grazie (Sans-Serif); per agevolarla ulteriormente sono stati creati caratteri ad alta leggibilità che vengono studiati in modo tale da avere un ulteriore distinzione tra lettere molto simili (nella stragrande maggioranza dei font), come ad esempio la "b" con la "d", la "q" con la "p" oppure la "l" (elle minuscola) con la "I" (i maiuscola) . Per aumentare la leggibilità vengono quindi studiati dei font che enfatizzano certe zone del carattere in modo da crearne una distinzione maggiore ed accentuarne le differenze.

Un esempio di font ad alta leggibilità è il font biancoenero#, che adotta i metodi sopra descritti utilizzando, appunto, una differenziazione netta nelle lettere, vedi Figura 6.2: Esempio font alta-leggibilità (font biancoenero).

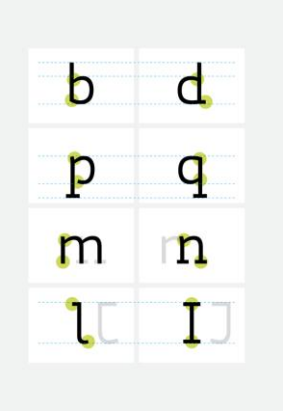

Immagine 6.2: Particolarità del font biancoenero

L'utente dislessico approfitterà di questo vantaggio leggendo un testo con minor sforzo, in quanto diminuendo gli sforzi cognitivi potrà concentrarsi sui contenuti.
Il testo, durante lo sviluppo del sistema, ha subito numerosi cambiamenti. Inizialmente, si è optato per il font Calibri, un font di tipo Sans-Serif di proprietà di Microsoft, molto semplice e bello da vedere. In seguito, tuttavia, il team di sviluppo è venuto a conoscenza di un font gratuito progettato per utenti dislessici, chiamato biancoenero, che dovrebbe agevolare la lettura grazie ad alcuni elementi che permettono di distinguere meglio le lettere. È stata quindi contattata la casa editrice biancoenero®, che detiene i diritti del font, per ottenere la licenza di utilizzo per il sistema V.E.R.O.N.I.C.A.
A seguito della concessione della licenza, il font viene utilizzato per visualizzare i testi dei documenti e dei libri. Tuttavia, per le altre sezioni del sito, è stato scelto come font il Sassoon, che è simile al font bianconero, ma più aggraziato.

6.6. Le immagini

Per un bambino come per un adulto, durante la lettura di un testo, è d'aiuto all'immaginazione ed alla comprensione, l'uso di immagini. Nei libri caricati, e quindi in molti epub, sono presenti immagini che nella fase di lettura verranno visualizzate dall'interfaccia e saltate dal processo di lettura del sintetizzatore, che analizza solamente il testo. Di conseguenza, si è scelto di mantenere le immagini dei libri caricati per non perdere quelle caratteristiche che, soprattutto per un bambino, rendono un libro, piacevole accattivante e bello.
Il team ha preferito lasciare le immagini senza alcuna cornice per un fattore estetico. In futuro non sarebbe mal visto un leggero effetto d'ombra dietro ogni immagine.
Le gestione della dimensione è un altro punto cruciale, poiché se non viene gestita può portare ad una visualizzazione poco ordinata, o ad immagini sgranate per via di esagerati ingrandimenti. A causa di queste problematiche si sceglierà di mantenere la dimensione originale ma anche di dare un limite sulla dimensione; limite imposto dall'interfaccia per la porzione di schermo utilizzabile in quel momento.

6.7. I temi

Uno dei punti cruciali che fin dall'analisi si è scelto di affrontare è la creazione dei temi, soprattutto perché V.E.R.O.N.I.C.A. è un sistema pensato dapprima per i bambini.
Avere un tema personalizzato, permette di dare all'interfaccia oltre ai colori anche delle immagini, ad esempio facendo uso di cartoni della Walt Disney®# (o altri). Ad ogni modo un tema rende all'utilizzatore l'impressione che l'applicazione in uso sia una sua creazione, portandolo ad utilizzarla con una maggiore piacevolezza.
I bambini, però, non sono gli unici per cui il sistema è stato ideato, infatti non esistono limitazioni. Si è cercato di creare un sistema bello e funzionale che soddisfasse bambini e adulti, maschi o femmine, indipendentemente dal grado di dislessia o le capacità informatiche. Inoltre l'utilizzo dei temi porta con se il vantaggio di dare all'utente un motivo in più per continuare ad utilizzare questo potentissimo strumento, che può cambiare interfaccia e quindi annoiare di meno.
Tra i temi, anche se non è stato ancora affrontato il problema, sarà facile costruirne uno adatto ai daltonici. La creazione di un tema è in effetti la scrittura di un foglio di stile, poiché in questo modo si prescinde da una serie di complicazioni dovute a dipendenze tra altri file. Verranno salvati in una cartella chiamata appunto "Temi", che conterrà i fogli di stile. Ognuno di questi all'interno riporta il codice organizzato per tipo di selettore: id, tag, classe, misto oppure con dei commenti che ne identificano un'apposita sezione, ad esempio la sezione menu: è possibile identificare con maggior rapidità il selettore corretto, in caso di modifica da parte di diversi programmatori.

6.8. La visione dell'interfaccia come web-application

La fase di realizzazione dell'interfaccia finale è stata un susseguirsi di critiche volte a creare l'interfaccia di default, interfaccia che cerca di far apparire V.E.R.O.N.I.C.A. come un'applicazione anziché un sito web. Si è cercato di distaccare il sistema dal background e dividere in sezioni anche il codice. In questo modo, anche grazie all'uso del PHP, è stata creata un'architettura modulare che pone il basso accoppiamento e l'alta coesione in primo piano. La modifica di un file relativo all'interfaccia non comporta modifiche a cascata: infatti, il file che andrà a disegnare l'header (come il footer) sarà sempre lo stesso per ogni pagina. Riprendendo il discorso sulla visione dell'interfaccia da parte dell'utilizzatore, si è pensato di dare all'utente un'applicazione, più che un sito-web; un'applicazione in cloud accessibile ovunque ci sia un browser e una connessione ad internet, con i propri documenti subito disponibili.

6.9. L'interfaccia per tablet, internet tv & smartphones

La scelta di utilizzare i fogli di stile CSS, e sopratutto adottare la versione terza di questi ultimi, rende anche possibile creare fogli di stile (o modificare quelli già presenti) adattandoli a varie piattaforme con dimensione dello schermo differente. Grazie a questo, l'interfaccia viene adattata dagli sviluppatori a tablet, tv, smartphones, senza troppe difficoltà, lasciando quindi all'utente un ampia gamma di dispositivi in cui potersi immergere nella lettura. In ogni piattaforma all'utente viene presentata un'interfaccia a tema simile che gli consentirà di sapere fin da subito come orientarsi, sviluppando una veloce familiarità con il sistema.
Per strutturare meglio l'interfaccia, si è scelto di utilizzare una suddivisione in 3 parti:

1. Header: con (semplicemente) un logo, un titolo e la barra di ricerca dei libri;
2. Corpo: che contiene eventuali menu e il cuore della web-application;
3. Footer: che potrà contenere eventualmente il logo della società che gestirà il sistema e informazioni generali sul progetto come di contatto.

Tale suddivisione consente di avere un carico modulare e di concentrarsi sulla parte centrale, che è la più interessante per chi usa il sistema, ma anche di avere la possibilità di cercare un libro partendo da una qualunque pagina.

6.10. L'interazione

Un altro importante capitolo che si è cercato di rimarcare tra gli obiettivi è l'interazione e la comunicazione dei messaggi all'utente. Lo scopo è fornire all'utente ciò di cui necessita senza difficoltà, in maniera veloce e senza preoccupazioni, anche in caso di errori.

Impostazione generale e di Design

Perché questo fosse possibile, si è scelto di dare ai menù pochi ed essenziali elementi e di rimarcare la pagina attuale; inoltre, per la comunicazione degli errori all'utente si è preferito l'uso di popup in HTML e CSS. Nell'interfaccia relativa ai form, invece, si è scelto di evidenziare la cella errata con un colore più "amichevole" del rosso, ad esempio il blu o il verde. Nel caso di errori del sistema, come ad esempio in caso di "pagina non trovata" oppure "attivare i cookies", si è scelto di riportare i messaggi utilizzando immagini e colori che non richiamino segnali di allarme, come riportato nella seguente immagine.

Immagine 6.3: Pagine di errore nel caso di cookies non attivi

L'aspetto principale, su cui si è focalizzata l'attenzione, è stato l'evitare ogni tipo di preoccupazione all'utente, una volta posto dinnanzi ad una qualsiasi situazione generata, anche dal sistema.

Pagina di lettura

L'iterazione con l'utente dislessico è estremamente importante nella pagina di lettura del testo. Per la progettazione di questa pagina si è pensato dapprima di affrontare il problema utilizzando pochi elementi:

- il testo da leggere;
- la parte di testo letta in quel momento (evidenziata);
- il sintetizzatore che "legge" dando la possibilità di modificare la velocità di lettura.

Questo pensiero è rimasto identico durante le fasi di sviluppo, ma si è cercato di studiare il modo in cui il libro viene letto dall'utente, favorendo il processo di visione. Le principali idee sono ricadute su:

- un testo scorrevole (verticalmente) in maniera automatica;
- un testo scorrevole (orizzontalmente), suddividendolo in pagine, e utilizzando un'interfaccia con rappresentato un libro sullo sfondo;

quest'ultima scartata perché rendeva pesante la visione e utilizzava inutilmente i processi cognitivi. Attualmente il prototipo utilizza un testo scorrevole orizzontalmente mediante appositi programmi.

Gestione della lettura

Durante la fase di lettura di un testo, nonostante, per il futuro, si pensi di adottare un sistema di riconoscimento vocale automatico, per ora ci si è accontentati di un sistema che permette di regolare la velocità di lettura manualmente. Per far questo, si è pensato di leggere il testo parola per parola (dando ad ognuna un numero), in modo da poter determinare la parola letta attualmente ed evidenziarla (o comunque portarla in primissimo piano). In seguito, si è adottato questo processo in quanto il sintetizzatore vocale adottato per queste prime versioni di V.E.R.O.N.I.C.A. non consente di avere output alcuno, circa la parola attualmente letta.
Dopo aver applicato questo processo si pensi a come viene evidenziata. Evidenziando parola per parola (o sillaba per sillaba) si pensa che la lettura possa essere semplificata, come in un karaoke; tale processo procura, però, uno sforzo non indifferente, portando ad un affaticamento che potrebbe causare un abbandono della lettura. Per via di questo motivo, un altro metodo candidato che ha portato a lunghe riflessioni è il seguente:

Immagine 6.4: Ipotesi della pagina di lettura

il testo scorre verticalmente dal basso verso l'alto, molto linearmente, attraversando una zona in cui una maschera ne evidenzia le (poche) righe attualmente da leggere, che per semplicità chiameremo "zona centrale", e un piccolo pallino (blu, verde o giallo) scorre sotto il testo, orizzontalmente, seguendo la parola attualmente letta. Per portare in evidenza la zona centrale, si pensa di utilizzare una maschera di sfocatura, in quelle righe già lette e in quelle da leggere, esterne alla zona centrale.

Attualmente, il sistema supporta gli epub e semplice testo. Il formato epub ha permesso di mantenere la formattazione già presente nel documento, ma ha anche creato problemi legati al sintetizzatore e all'utilizzo dei caratteri. Ciò è dovuto alla struttura degli epub, che prevede tag annidati, ignorati dal processo di suddivisione del testo in parole o frasi, per la lettura da parte del sintetizzatore.

6.11. Approfondimenti sul sintetizzatore

In questo paragrafo verranno affrontati alcuni punti chiave relativi al processo di sintesi vocale. Principalmente: quali sono state le scelte del team, quali sono state le problematiche e come sono state affrontate.

6.11.1. Ricerca e analisi del sintetizzatore

Dopo giorni di ricerche di software open source, che offrissero la sintesi vocale, si è capito che in futuro il sistema V.E.R.O.N.I.C.A. non avrebbe potuto adempiere agli obiettivi preposti, in quanto, i sintetizzatori vocali trovati, non possiedono sufficiente dizione e ignorano la punteggiatura. Estendendo le ricerche ai software proprietari, il team ha subito capito che questi ultimi avrebbero rappresentato l'obiettivo da raggiungere perché V.E.R.O.N.I.C.A. soddisfi a pieno le esigenze di un utente dislessico.
Ad ogni modo, nonostante le sue limitazioni, per lo sviluppo di questo primo prototipo, il team ha optato per l'utilizzo di un software di sintesi vocale open source. La scelta iniziale è caduta tra eSpeak e Festival; si è deciso, però, di ricorrere al primo, per motivi di semplicità e facilità di utilizzo.

6.11.2. Il sintetizzatore eSpeak

eSpeak funziona da linea di comando, come descritto di seguito, e permette l'inserimento dei seguenti parametri:

```
espeak -a 100 -p 56 -s 200 -g 2 -v it+f2 "testo da leggere "
        -a numero: indica il volume
        -p numero: indica la tonalità di voce
        -s numero: indica la velocità di lettura
        -f file: indica un file in input
        -g numero: indica la lunghezza della pausa tra le parole
        -v it+f2        : indica la lingua: italiano variante femminile 2
        -w file.wav: esporta il risultato in un file wav
        "testo da leggere": stringa sottoposta al sintetizzatore
```

Per far uso del comando, il team ha creato un apposito file bash, che verrà eseguito dal server e che generarà il file audio contenente il testo letto. Quest'ultimo verrà poi caricato nel client facendo uso dell'apposito tag <AUDIO> fornito da HTML5. Questo complesso processo di gestione del testo è descritto, dettagliatamente, nel paragrafo 6.11.4, La divisione del testo.

6.11.3. Integrazione di una nuova voce

Purtroppo i sintetizzatori open source, che il team ha provato, non funzionano in modo egregio quanto quelli a pagamento, come, per esempio, Loquendo. La voce, infatti, è molto artificiosa e la comprensione del parlato è ardua. Questo rischia di rendere più complessa la lettura di documenti, andando in contrasto con gli obiettivi del progetto. Tuttavia, il team si è imbattuto in una voce che si è rivelata migliore di quella di default fornita in dotazione con eSpeak. Tale voce, che fa parte del progetto MBROLA, possiede una dizione più chiara e comprensibile. Grazie ad essa, il team è fiero di poter affermare di essere riuscito a creare un sistema completo e funzionale utilizzando solamente software open source, anche se si

riconosce che il processo di lettura sarebbe semplificato ulteriormente se si facesse utilizzo di voci migliori, come quelle offerte da altri software di sintesi vocale a pagamento, descritti in precedenza.

6.11.4. La divisione del testo

Il team si è posto il problema di come gestire l'invio del testo dal sito al server affinché possa essere elaborato dal sintetizzatore. Il problema è stato risolto dividendo il testo in parti più piccole. Inizialmente, per una questione di semplicità, si è effettuata una divisione parola per parola, in modo tale da rendere ogni singolo elemento cliccabile. Una volta effettuato il click, il sistema pone tutte le parole, da quella cliccata in poi, in una lista, che verrà inviata al server; esso si occuperà di riunire il testo e di inviarlo al sintetizzatore. Il sintetizzatore esporta il risultato in un file wave, che verrà messo in una cartella dedicata per ogni utente; un frame sulla pagina del sito web, posto sotto il testo del libro, si occupa di prelevarlo e di riprodurlo.
Questo metodo ha un grave difetto: essendo il lavoro di divisione svolto tramite JavaScript, e quindi lato client, nel caso di testi molto lunghi, l'elaborazione risulta particolarmente lunga, e il sistema diventa quasi inutilizzabile. Per questo motivo, si è scelto di effettuare la divisione per periodo, anziché per parole. Tale aspetto presenta lo svantaggio di dover leggere tutto il periodo dall'inizio, anche se parte di esso è già stato letto; tuttavia, l'elaborazione del testo è molto veloce e risulta un'operazione quasi trasparente all'utente.
Col tempo, il team di sviluppo è arrivato ad una soluzione che potrebbe permettere di tornare alla divisione del testo per parole, anziché per periodi. Infatti, se la divisione del testo venisse fatta al caricamento del documento, tramite PHP, e quindi via server, vi sarebbe solo l'attesa durante tale fase; pertanto, tutto il lavoro potrebbe essere svolto senza che l'utente abbia bisogno di rimanere collegato al sito.

6.11.5. Il processo di salvataggio del file audio

I file audio generati dal sintetizzatore vengono dapprima memorizzati nel server, poi caricati nel client. La generazione di questo file audio richiede un'accurata analisi, poiché si presentano diverse problematiche che potrebbero rilevarsi di difficile gestione se non studiate al meglio. Tra di esse, elenchiamo le seguenti:

- la generazione di un nuovo file non deve sovrascrivere file già esistenti;
- se un utente legge lo stesso libro deve essere gestita la generazione di un nuovo file con un nome differente o con una propria cartella utente;
- se un utente apre più pagine web dello stesso libro o di un libro differente i file devono essere differenti;
- se un utente sta facendo il login o il logout sarebbe efficiente eseguire una pulizia dei propri file audio, all'interno del server;
- i file audio wav sono molto pesanti, è possibile pensare a un programma che comprima nei formati mp3 o amr; in questo modo la dimensione passerebbe, es esempio: da 100Mb a 10Mb con mp3, ad 1Mb con amr.

7. Implementazione dell'interfaccia

7.1. Contesto ed Obiettivi

Nel presente capitolo vengono analizzate le pagine web realizzate. Si partirà dalla bozza di prima realizzazione, per poi presentare e descrivere la pagina realizzata nel prototipo. Si rimarca il fatto che le interfacce sono state realizzate con coerenza rispetto alle scelte progettuali prese inizialmente.

Pagina di login

Mockup

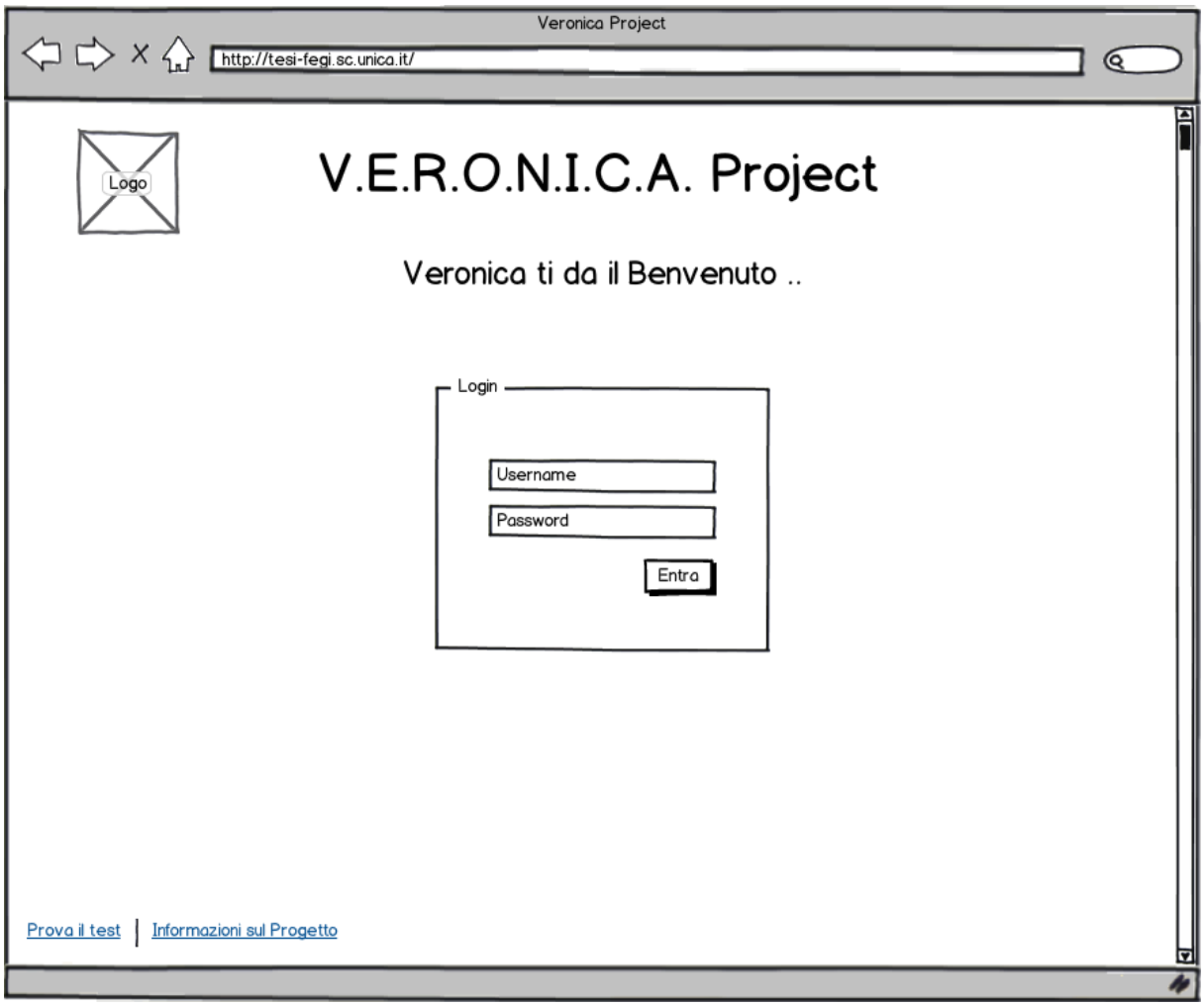

Immagine 7.1

Pagina realizzata

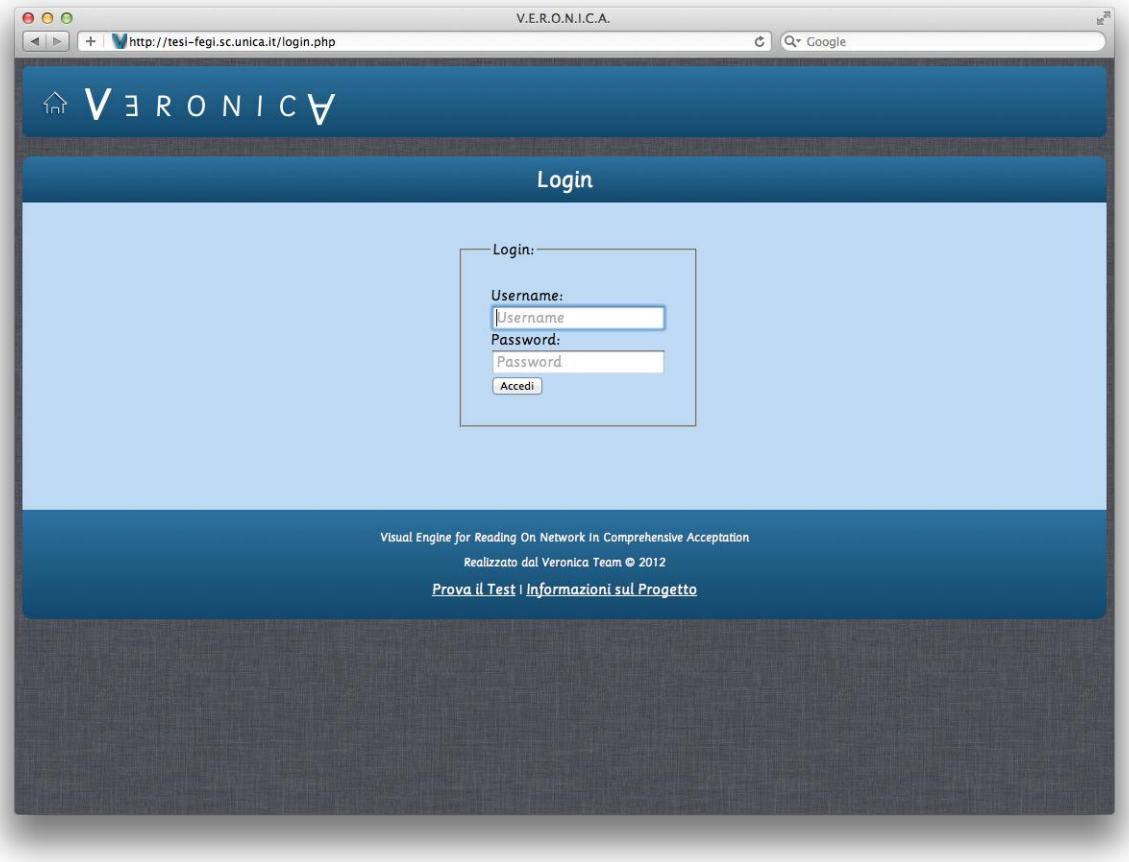

Immagine 7.2

Descrizione della pagina

La pagina di login è la prima che viene mostrata all'utente. Serve per effettuare l'accesso al sistema, in modo che possano accedere ai servizi solo gli utenti registrati da un amministratore. Se si tenta ad accedere ad altre pagine esistenti senza aver effettuato il login, si viene rimandati a questa pagina.
Oltre al login, permette di effettuare un test, senza valenza medica, ad esempio per sapere quante sillabe al secondo si riesce a leggere, e di ottenere informazioni sul sistema.

Home Page

Mockup

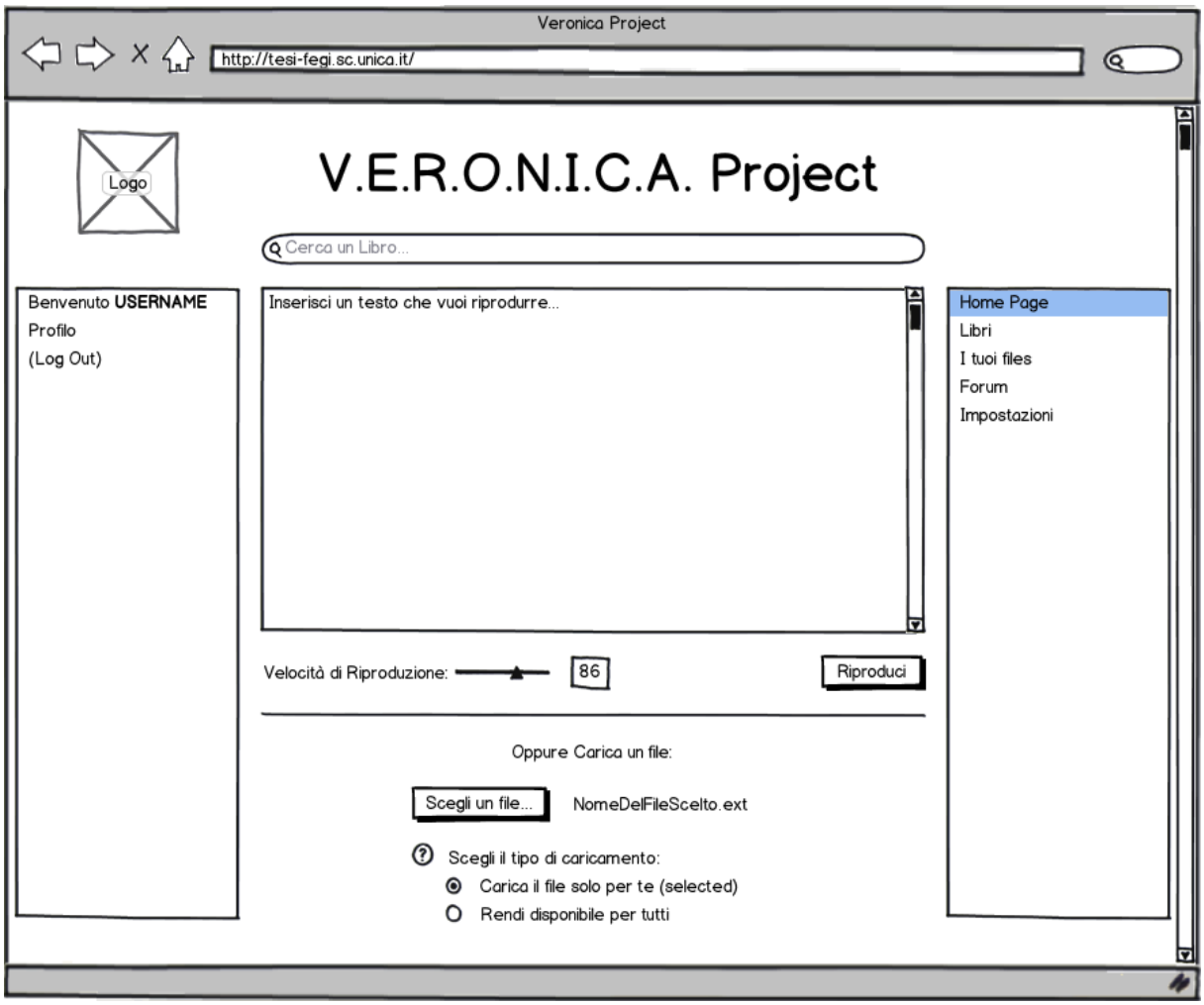

Immagine 7.3

Pagina realizzata

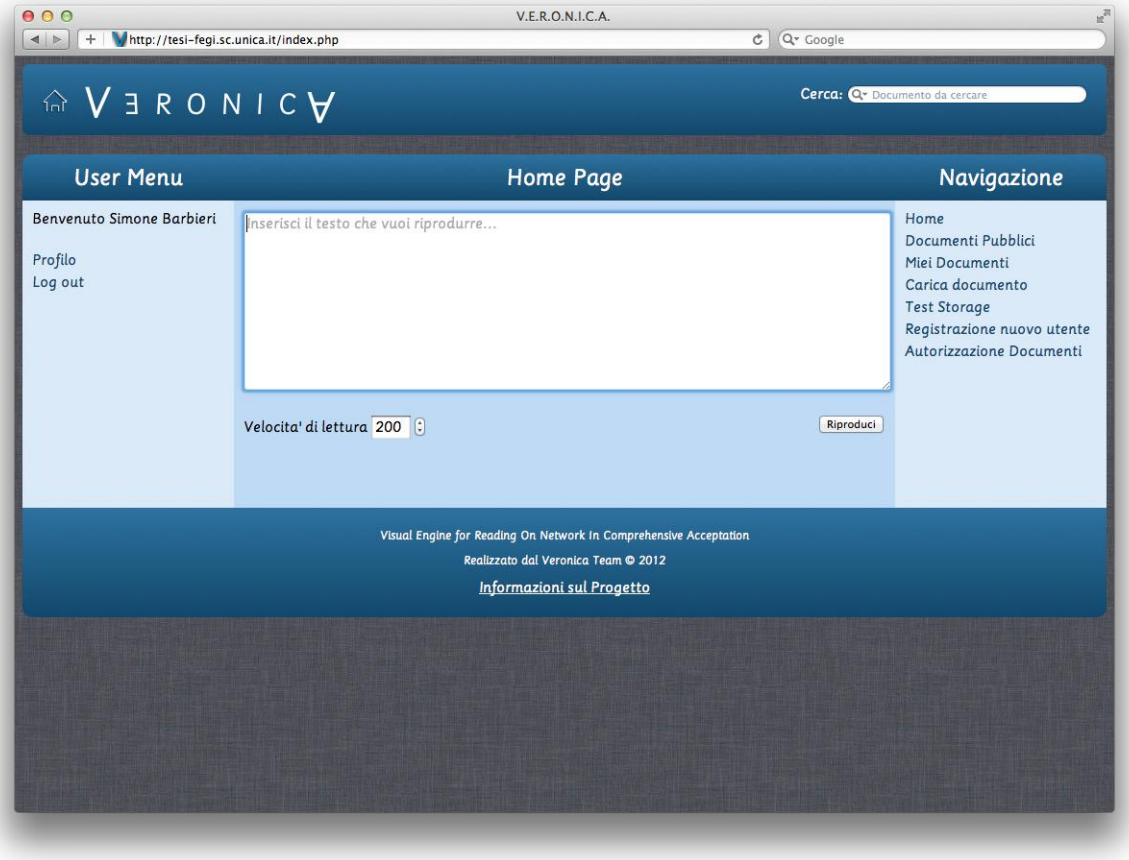

Immagine 7.4

Descrizione della pagina

È la pagina principale del sito. Se non si stava tentando di accedere ad altre pagine, una volta effettuato il login, si viene re-indirizzati a questa pagina. In questa pagina c'è la possibilità di inserire del testo nella text box che occupa il centro della pagina, ed esso verrà letto dal sintetizzatore vocale, mediante il tasto "Riproduci".
Inizialmente era prevista la possibilità di caricare anche i file direttamente dalla home page, come mostrato dal mockup, ma è stato deciso di spostare questa funzione in una pagina dedicata.

Elenco Documenti

Mockup

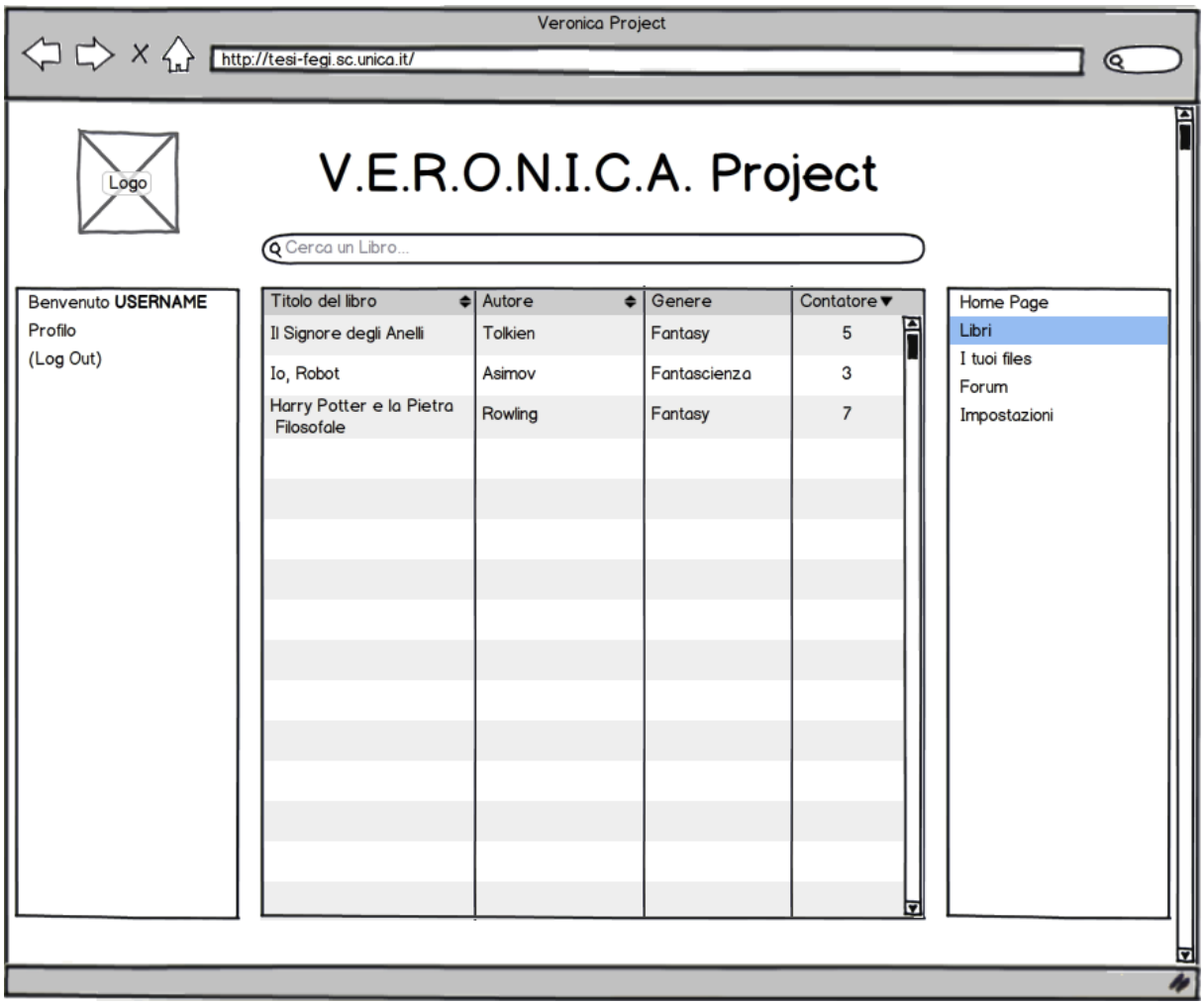

Immagine 7.5

Pagina realizzata

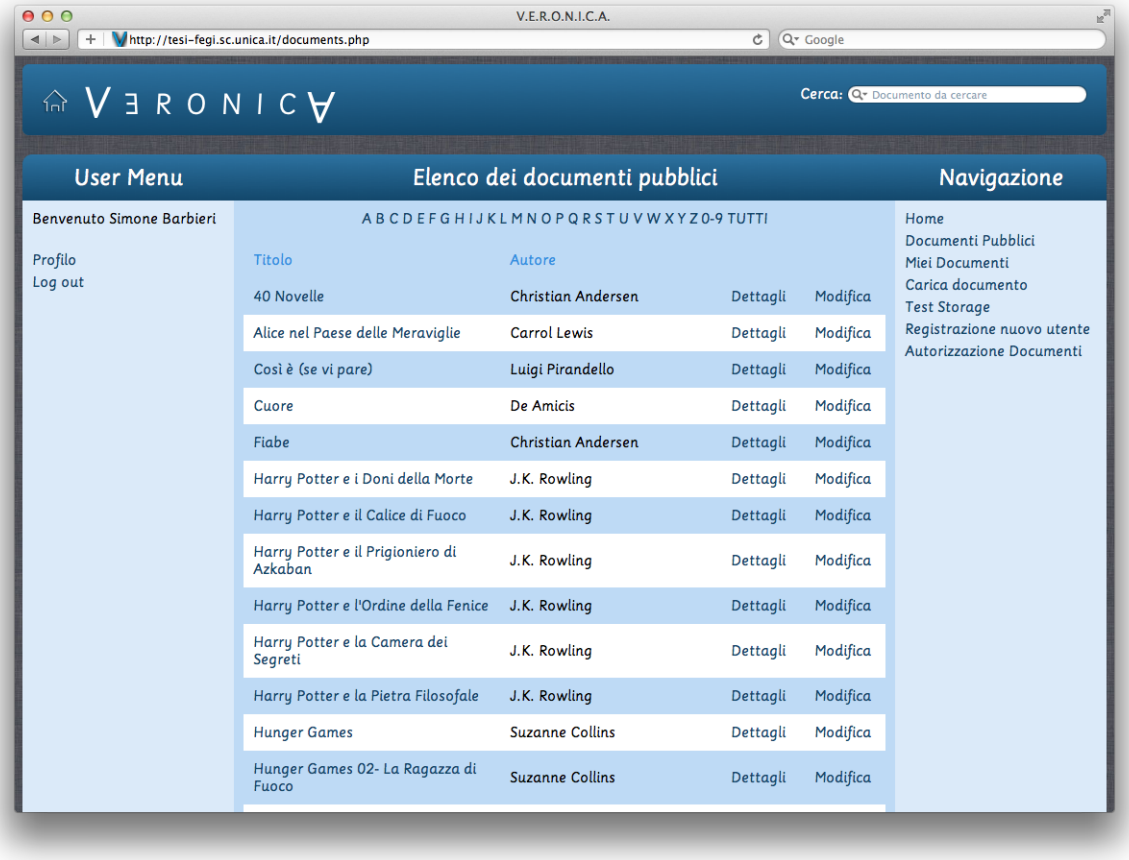

Immagine 7.6

Descrizione della pagina

È la pagina che viene visualizzata quando si vuole visualizzare l'elenco dei documenti pubblici, l'elenco dei propri documenti oppure quando si effettua una ricerca. Da questa pagina è possibile iniziare la lettura di un libro, visualizzarne o modificarne i dettagli, segnalarlo ad un amministratore e anche assegnare un voto al libro (funzione non presente in questo primo prototipo).

Lettura di un documento

Mockup

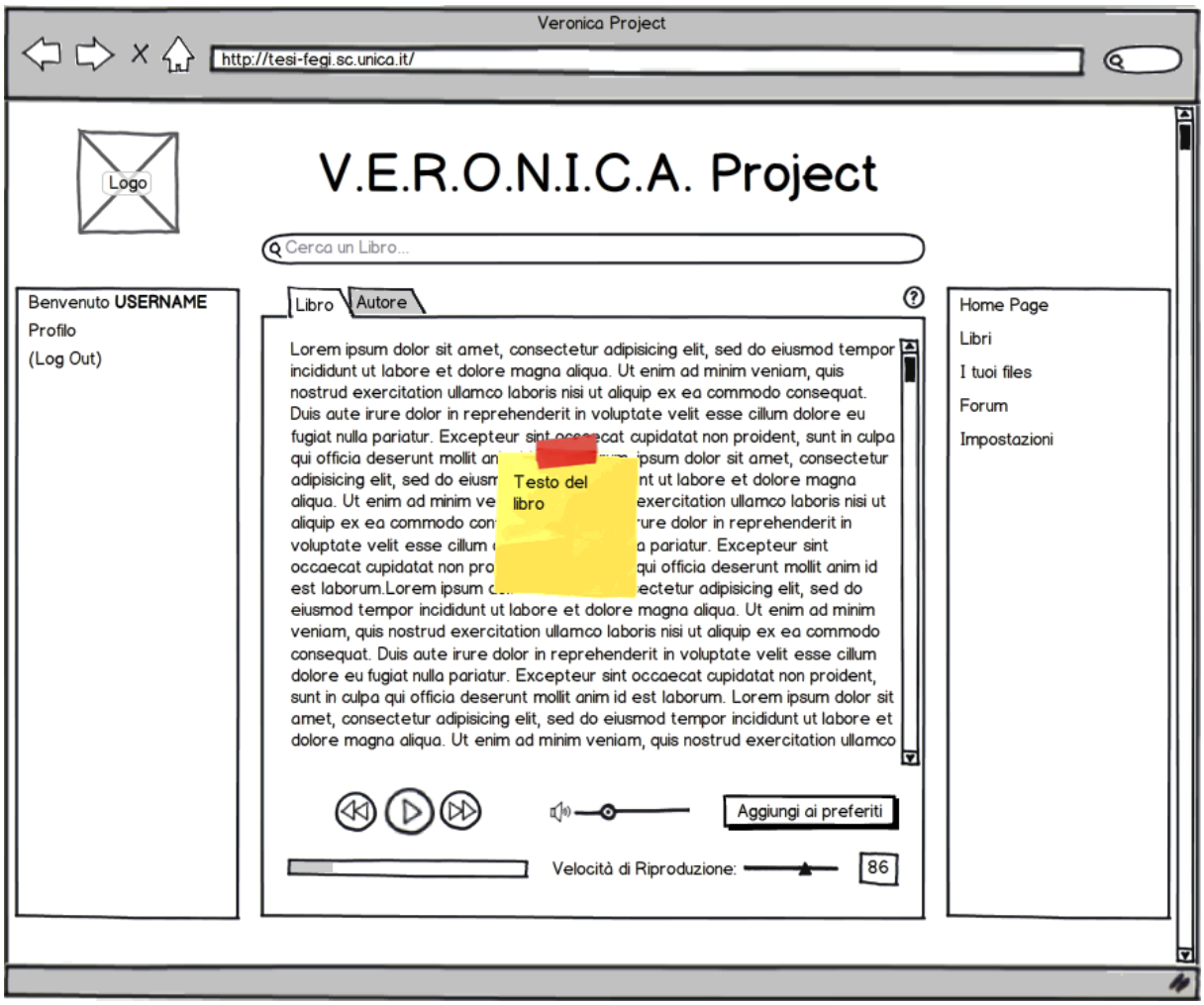

Immagine 7.7

Pagina realizzata

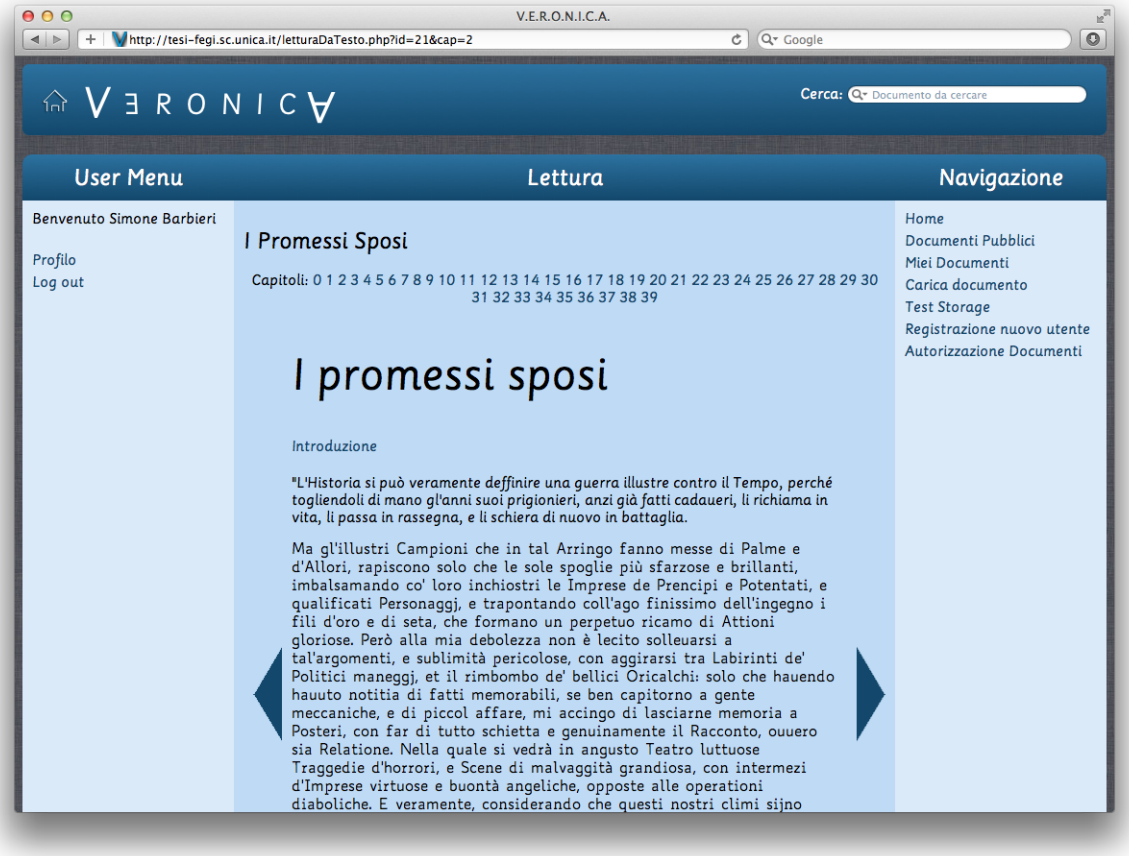

Immagine 7.8

Descrizione della pagina

Questa pagina permette la lettura di un documento. Il testo del documento è visualizzato con il font realizzato per dislessici, biancoenero (vedi paragrafo 6.5); cliccando su di esso, il testo viene inviato al sintetizzatore, che inizia la riproduzione dal termine cliccato del testo in poi.
Il testo viene letto con la velocità di lettura di default, definita nelle impostazioni. È tuttavia possibile modificarla per la corrente sessione, attraverso una casella di testo presente sotto il contenuto del documento.
Il testo viene diviso in capitoli e in pagine, come descritto nel paragrafo 6.11. Per scorrere tra le pagine sono presenti delle frecce ai lati del testo, o alternativamente possono essere usati i tasti freccia della testiera, per velocizzare l'operazione. I capitoli, invece, sono elencati in alto, e permettono di raggiungere determinati punti del testo più rapidamente.

- Contiene il testo del libro.
- Permette di avviare la lettura del testo.
- Permette di selezionare la velocità di lettura.
- Permette di aggiungere il libro ai preferiti.
- Permetterà di valutare il libro, esprimendo un giudizio su 5 stelle.

Carica Documento

Mockup

Non è stato realizzato un mockup per questa pagina, in quanto il progetto iniziale prevedeva di inserire questa funzione nella home page.

Pagina realizzata

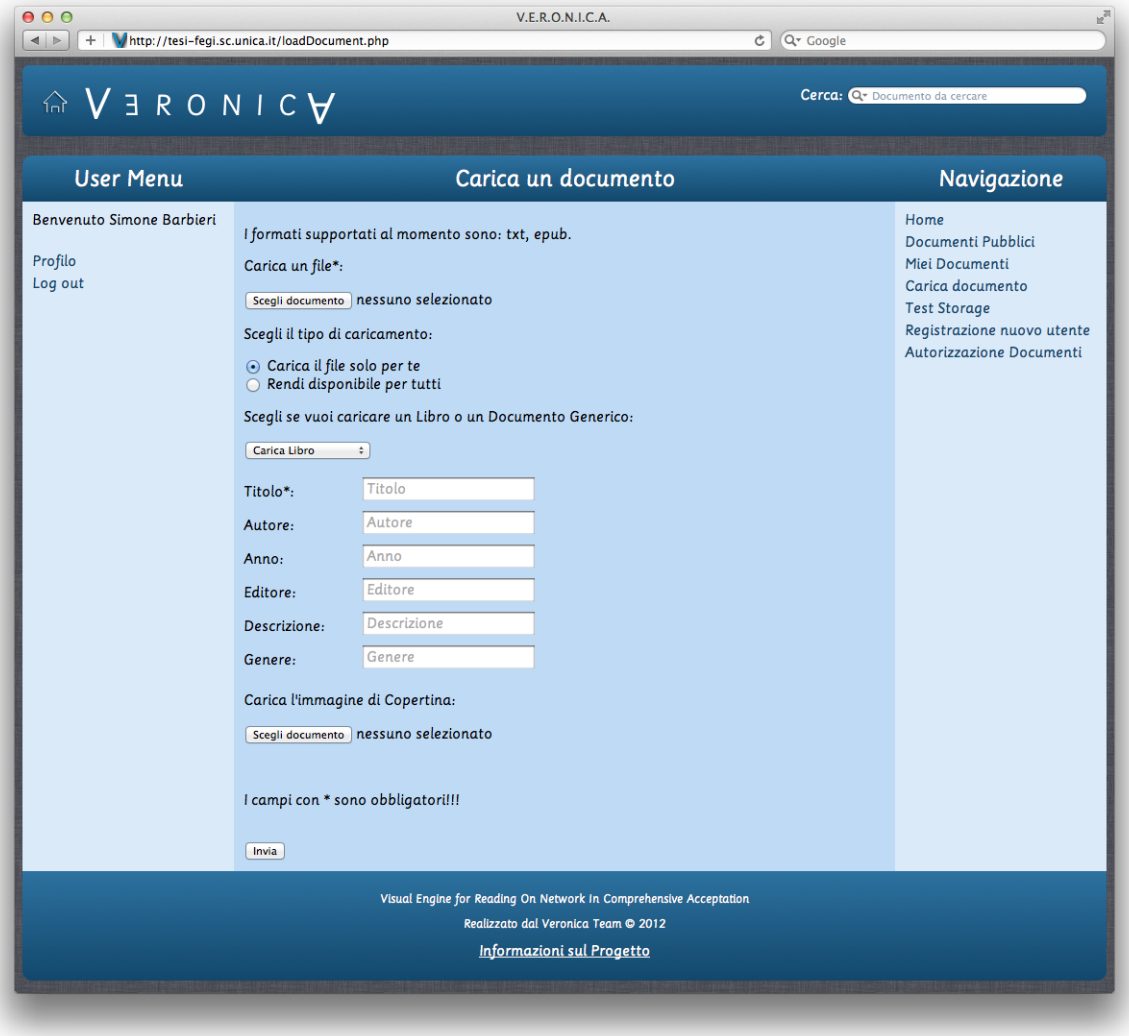

Immagine 7.9

Descrizione della pagina

È la pagina che permette di caricare un documento nel sistema. Si può scegliere di rendere il documento pubblico o privato e anche il tipo di documento (libro oppure documento generico). Inoltre, c'è la possibilità di inserire ulteriori informazioni sul documento, compresa la copertina, se il documento è un libro.

Inserimento Utente

Mockup
Non è stato realizzato un mockup per questa pagina.

Pagina realizzata

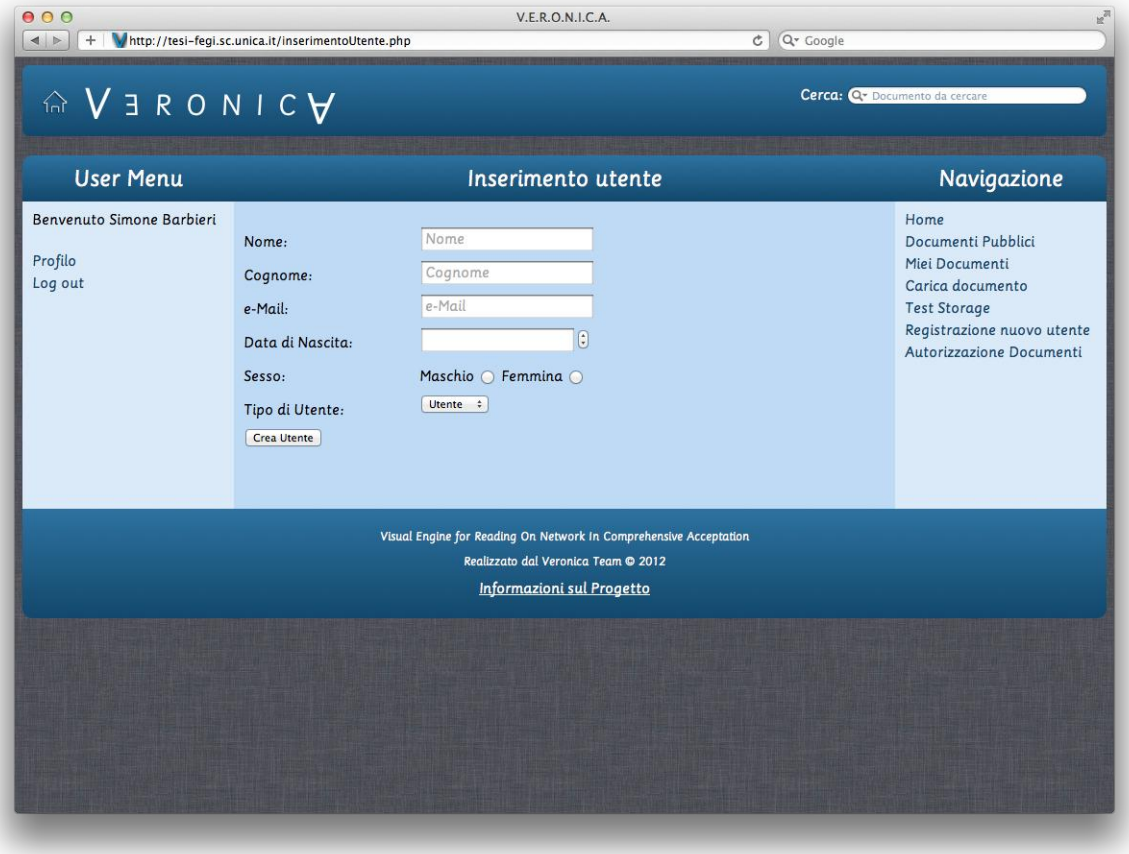

Immagine 7.10

Descrizione della pagina
Questa pagina è visibile solamente dagli amministratori; tramite essa, è possibile inserire un nuovo utente nel sistema, come descritto nel requisito R07.

Profilo e Impostazioni

Mockup
Non è stato realizzato un mockup per questa pagina.

Pagina realizzata

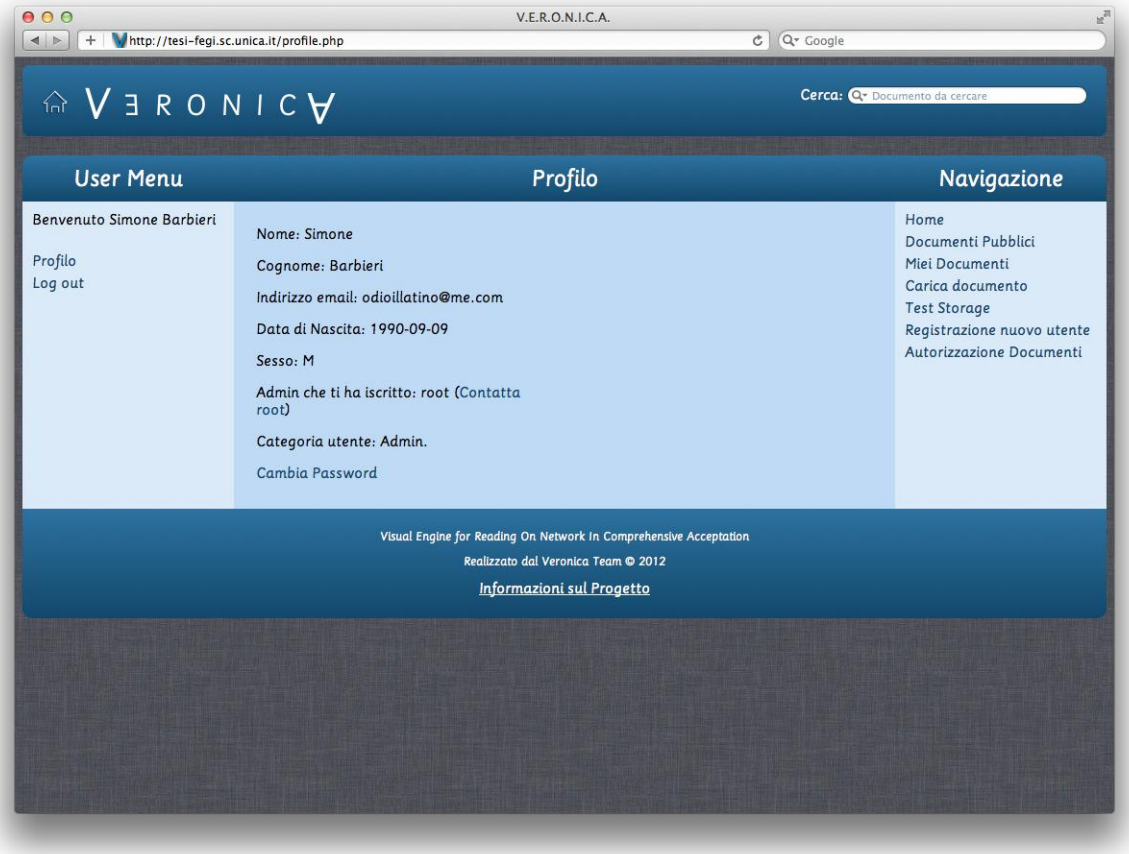

Immagine 7.11

Descrizione della pagina
È la pagina che contiene le informazioni personali dell'utente. È possibile aggiornare informazioni come l'indirizzo email e password.
Da questa pagina sarà anche possibile modificare le seguenti impostazioni:

- scegliere il tema;
- scegliere la velocità di default di lettura;
- scegliere la dimensione di default del testo.

Informazioni

Mockup
Non è stato realizzato un mockup per questa pagina.

Pagina realizzata

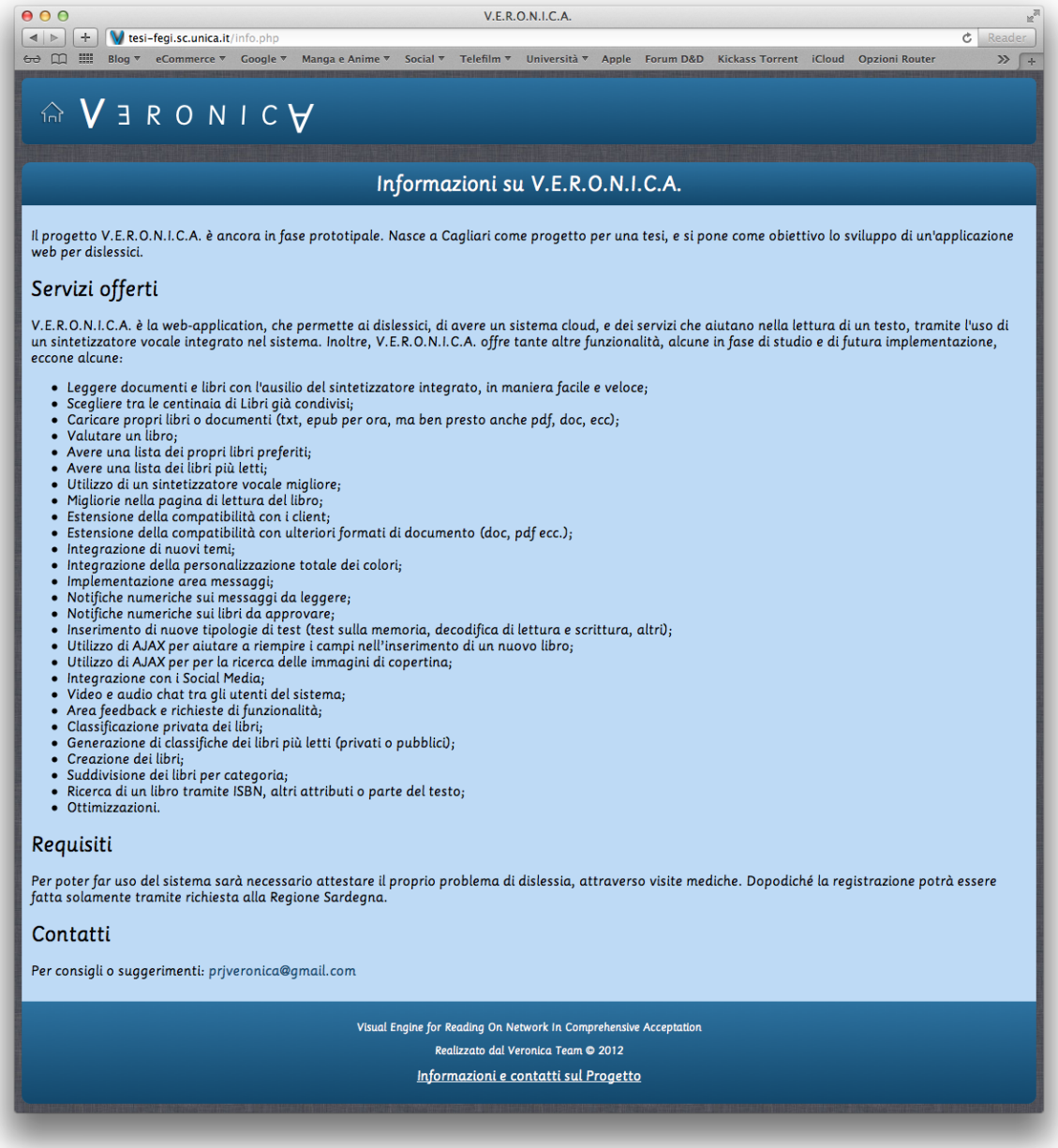

Immagine 7.12

Descrizione della pagina
Questa pagina riporta informazioni sulla dislessia e sul progetto V.E.R.O.N.I.C.A. Questa pagina è visibile anche se non si è utenti registrati, e può essere raggiunta attraverso un link posizionato nell'header.

Test

Mockup

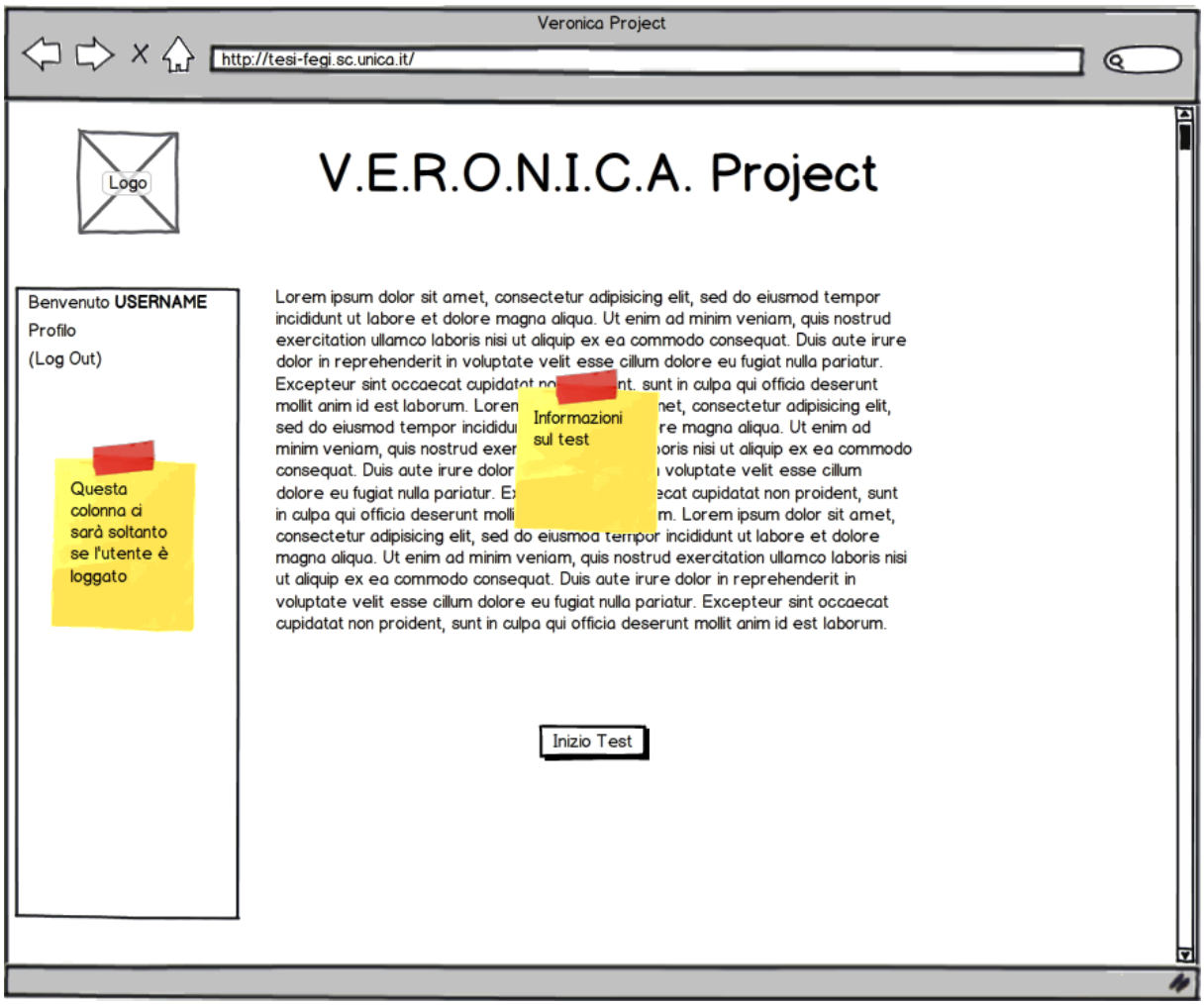

Immagine 7.13

Pagina realizzata

Pagina ancora non realizzata. Il prototipo presenta una descrizione di ciò che potranno essere i Test.

Descrizione della pagina

In questa pagina è possibile effettuare una serie di test senza valenza medica, che permettono di ottenere il grado di dislessia dell'utente in maniera indicativa. Questa pagina (o una molto simile) è visualizzabile sia da utenti registrati, che da visitatori.

8. Descrizione della Base di Dati

8.1. Contesto ed Obiettivi

Il sistema V.E.R.O.N.I.C.A. è pensato, in linea di principio, per fornire un servizio di lettura automatizzato tramite sintetizzatore vocale.
Chiaramente, il servizio di lettura, come precisato al capitolo 3.3, Modello del sistema, non è l'unico di cui si può fruire: in generale, sono stati pensati ulteriori servizi per tutte le tipologie di utenti che utilizzeranno il sistema.
A livello di base di dati, tale aspetto implica le necessità di salvare una moltitudine di dati in maniera precisa e sistematica.
In prima analisi, sono state individuate tre tipologie di utenti al quale garantire l'accesso al sistema; di conseguenza, si è reso necessario il salvataggio dei loro dati, sia di accesso che personali, all'interno database.
Le tre tipologie sono state classificate come generalizzazione dell'entità principale "utente". Vengono indicate di seguito:
il soggetto dislessico, indicato come "Dislessico";
i medici, indicati come "Medici"
gli amministratori del sistema, indicati come "Admin".
L'utente deve possedere gli attributi relativi alla propria anagrafica, nello specifico è richiesta la memorizzazione di nome, cognome, data di nascita, sesso, username e password, da cifrare.
I dati aggiuntivi che contraddistinguono gli utenti dislessici dovranno essere relativi alla malattia, in particolare la sua fascia all'interno di V.E.R.O.N.I.C.A. ed il suo grado di dislessia. L'interfaccia grafica mostrata all'utente dislessico sarà differente a seconda della fascia identificata; egli ha, comunque, la possibilità di cambiare la propria interfaccia.
Per quanto riguarda i medici, sono richiesti i dati che contraddistinguono la sua professione, eventualmente integrabili con nuovi dati.
È stato, sin dal principio, ritenuto opportuno tenere traccia dei dati di riferimento dei vari files caricabili all'interno del sistema.
Nello specifico, si è distinto, inizialmente, tra i documenti, sottoponibili al sintetizzatore per una lettura, ed i test, forniti dal sistema per ottenere una valutazione indicativa sul grado di dislessia dell'utente che lo effettua. Va rimarcato, anche in tale circostanza, che i test non hanno alcuna valenza medica e sono liberamente accessibili anche ad utenti non registrati al sistema.
La possibilità di caricare un test è offerta in linea esclusiva ai soli medici registrati al sistema. Tale scelta implica delle conseguenze importanti nella gestione dei dati.
È, infatti, opportuno tenere traccia di tutti i test caricati da ogni singolo medico: nello specifico, le informazioni relative ai caricamenti effettuati, si pensi alla data e all'ora, in aggiunta ai dati tipici di un particolare test.
Allo stesso modo della condivisione dei documenti pubblici, che verrà analizzata in seguito nel dettaglio, anche l'inserimento di un test implica un controllo rigido sul contenuto: sebbene l'utenza sia regolarmene registrata e verificata, è fondamentale garantire la sicurezza a favore degli utenti. Infatti, ogni test caricato, prima di essere effettivamente disponibile, deve ricevere l'approvazione da parte di un admin, essenzialmente per motivi legati alla tutela dei minori: è importante ricordare che V.E.R.O.N.I.C.A. è un sistema pensato, in primis, per bambini dislessici.
A tali necessità, si affianca l'esigenza di un tracciamento dei login effettuati dagli utenti del sistema, principalmente per verificare le operazioni eseguite dagli utenti e prevenire eventuali

problemi legati al malfunzionamento del sistema. In prima analisi, si è preferirito creare una tabella "Login" che contenesse tutte le informazioni di sorta, a scapito dell'utilizzo di un file di log, essenzialmente per motivi di accessibilità, velocità nella fruizione dei dati e per evitare di dover analizzare continuamente un file di testo.
Durante lo sviluppo della base di dati, l'entità "Libro" è stata più volte analizzata e rimodellata per soddisfare le esigenze di programmazione.
Inizialmente, i libri furono ideati come entità a se stante, dal momento che l'inserimento di appunti ed altre tipologie di documenti è avvenuto procedendo con lo sviluppo.
È stata proprio l'estensione del supporto a tali tipologie che ha permesso una riqualificazione completa delle categorie di files gestite da V.E.R.O.N.I.C.A.: il libro è stato, infatti, classificato come sottocategoria del Documento; quest'ultimo è una generalizzazione di qualsiasi tipologia di documento che potrà essere caricato su V.E.R.O.N.I.C.A.
Al momento in cui si scrive, si tratta di libri e appunti (files di testo semplice, quali i .txt), le cui problematiche di gestione sono affrontate, dettagliatamente, nel capitolo 9.1, Gestione dei Documenti: caricamento e salvataggio.

8.2. Schema Concettuale definitivo

Di seguito è riportato lo schema concettuale E-R definitivo del database.
Per maggior chiarezza, le entità, che, secondo i formalismi, dovrebbero essere descritte nel seguente modo:

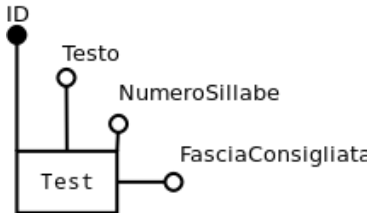

Immagine 8.1

verrano, invece, descritte come di seguito riportato:

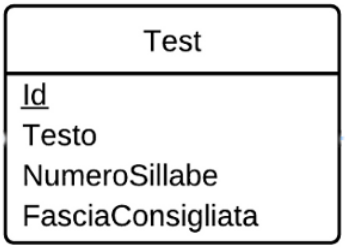

Immagine 8.2

In questo modo si semplifica il diagramma e lo si rende più gradevole e comprensibile.

Lo schema concettuale definitivo è il seguente:

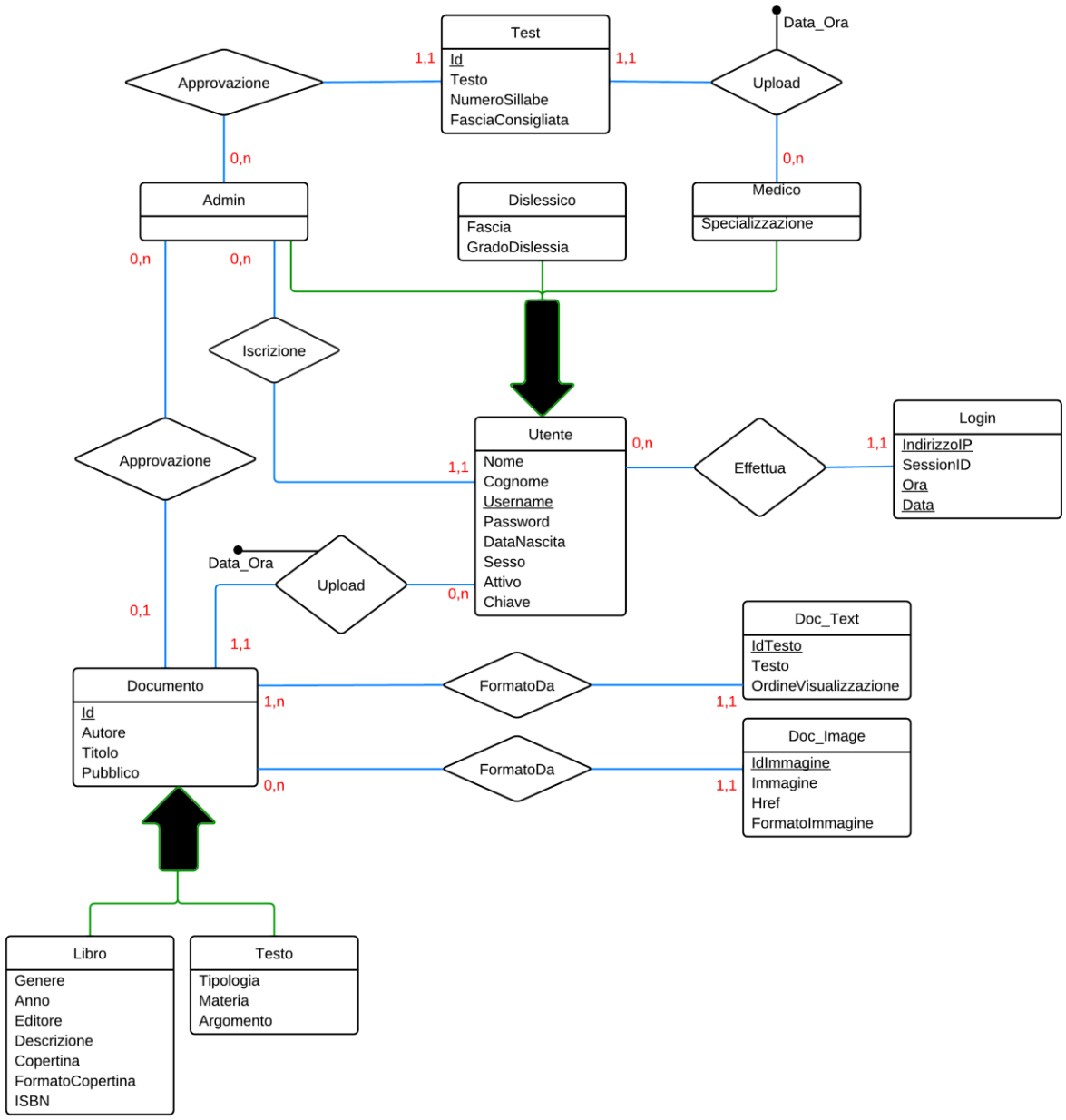

Immagine 8.3: Schema concettuale della Base di Dati

L'entità "Utente" è una generalizzazione delle entità "Admin", "Dislessico" e "Medico". L'entità "Medico" può effettuare l'upload di un "Test" che permette di ricevere un feedback riguardante il grado di dislessia di un determinato utente. Tale test potrà essere effettuato da un qualsiasi utilizzatore (anche non registrato) del sistema. Nello specifico, un medico può caricare 0 o più test, e un test può essere caricato da un solo medico. Per poter essere reso accessibile, un test deve essere sottoposto ad approvazione da parte di un Admin. Un Admin può approvare 0 o più Test, mentre un Test può essere approvato da un solo Admin (infatti, una volta approvato, non è più necessario che un altro admin lo riapprovi).
Un utente generico, per poter accedere al sistema, deve effettuare il login. Per tener traccia di tutti i login che vengono effettuati nel sistema, è stata creata un' apposita entità login, destinata a memorizzare i relativi dati di ogni accesso effettuato dagli utenti. Ogni login viene

identificato da indirizzo IP, data e ora: quindi, un utente può effettuare 0 o n login nel sistema, ma ogni login (composto da indirizzo IP, data e ora) è effettuato da un solo utente.
Un Admin ha anche la possibilità di iscrivere altri utenti (di qualsiasi categoria). Nello specifico, un Admin può iscrivere 0 o più utenti generici al sistema, mentre un qualsiasi utente generico può essere iscritto da un solo Admin. Ogni utente (di qualsiasi categoria) può caricare nel sistema 0 o più Documenti. Viceversa, un documento può essere caricato da un solo utente.
Nel sistema vi possono essere documenti pubblici o privati. Nel caso in cui venga fatta richiesta di caricare un documento pubblico, tale richiesta deve essere approvata da un Amministratore. Quindi, un documento può essere approvato da un Admin (se è fatta richiesta di pubblicazione per gli utenti registrati al sistema), e un Admin può approvare 0 o più documenti (può comunque approvare solamente documenti di cui è stata richiesta la pubblicazione).
Per via della difficoltà di stabilire se più documenti generici sono uguali tra loro, si assume, inizialmente, che per ogni caricamento è identificato un documento diverso da quelli già presenti all'interno di V.E.R.O.N.I.C.A. L'entità documento è, però, una generalizzazione delle entità "Libro" e "Testo" (con quest'ultimo termine si vogliono rappresentare i documenti, di qualsiasi tipologia, che non siano libri). Vi è, inoltre, la possibilità di identificare se un libro è già presente nel sistema attraverso il suo codice ISBN. Nel caso in cui venga fatta richiesta di condivisione di un libro già pubblico nel sistema, ossia con lo stesso ISBN di uno già presente, l'utente che ne ha richiesto la condivisione riceverà una segnalazione da parte del sistema che lo informa dell'esistenza del libro.
Per via di scelte implementative, si è deciso di memorizzare i documenti direttamente sul database piuttosto che nel filesystem del server. Sono entrate a far parte del database due entità: Doc_Text, che rappresenta un singolo contenuto testuale di un documento, e Doc_Image, che rappresenta un'immagine contenuta in un documento. Un documento può essere composto da uno o più contenuti testuali (ma non zero) e ogni contenuto testuale appartiene ad un solo documento. Un documento, inoltre, può essere composto da 0 o più immagini, mentre ogni immagine appartiene ad un solo documento.

8.3. Schema Logico definitivo

Test (**ID**, Testo, NumeroSillabe, FasciaConsigliata, Admin, Medico, DataOra_Upload);
Utente (**Username**, Password, Nome, Cognome, DataNascita, Sesso, AdminIscr, Chiave);
Admin (**Username**);
Dislessico (**Username**, Fascia, GradoDislessia);
Medico (**Username**, Specializzazione);
Login (**Username**, **Data**, **Ora**, **IndirizzoIP**, SessionID);
Documento (**ID**, Utente, Autore, Titolo, Pubblico, AdminApprovazione, DataOra_Upload);
Libro (**Documento**, Anno, Editore, Descrizione, Genere, Copertina, FormatoCopertina, ISBN);
Testo (**Documento**, Materia, Argomento);
Doc_Text (**Documento, IDTesto**, Testo, OrdineVisualizzazione);
Doc_Image (**Documento, IDImmagine**, Immagine, HRef, FormatoImmagine).

8.4. Descrizione e Decisioni Prese

Si è decisa la creazione di una tabella "Utente" per gestire al meglio i dati anagrafici degli utenti ed altri dati, relativi al sistema, a titolo di esempio l'attributo "Chiave", un codice identificativo inviato dall'ente insieme a nome utente e password, necessario, in ambito di

sicurezza, per l'autenticazione iniziale e la negoziazione delle chiavi, come specificato nel capitolo 11, Sicurezza.
Sono state, successivamente, create tre tabelle allo scopo di contenere gli attributi specifici di ogni categoria di utente: "Dislessico", "Medico", "Admin".
Le alternative alla scelta presa sono precisate di seguito:

- creare un'unica tabella "Utenti", contenente anche gli attributi specifici di ogni categoria di utente, e un attributo flag indicante la categoria alla quale appartiene ogni utente. Questa scelta non è stata presa in considerazione perché ci sarebbero stati molti attributi settati a NULL per ogni tupla (a titolo esemplificativo, un Admin non è un Paziente: quindi gli attributi GradoDislessia e Fascia sarebbero settati a NULL perché privi di significato per un Admin);
- creare tre tabelle "Admin", "Paziente", "Medico", contenenti, oltre agli attributi specifici per ogni categoria, anche gli attributi generali (non essendoci una tabella generale "Utente"). Tale alternativa non è stata considerata, poiché avrebbe, potenzialmente, complicato la gestione delle query. È sotto esame la possibilità di una sua futura integrazione.

Il Test è pensato in modo tale che sia formato da una porzione di testo, e dal numero di sillabe dal quale il testo è caratterizzato. Per ogni Test, oltre al testo di cui è composto e il numero di sillabe, vengono memorizzati anche il Medico che l'ha caricato, e l'Admin che ha autorizzato la pubblicazione di tale test.
Successivamente, il concetto di Test sarà esteso a diverse tipologie (non solo un testo con un numero di sillabe da valutare), pertanto è da prevedere una diversa gestione.
La tabella "Login" è necessaria per mantenere registrati i dati di login degli utenti: data e ora di ogni accesso e indirizzo IP.
L'entità Documento è, concettualmente, divisa in due sottocategorie:

- Libro;
- Testo.

Nella prima categoria sono indicati i libri, nella seconda tutti gli altri tipi di documenti elettronici che non sono libri. Partendo dallo schema concettuale, si è deciso di mantenere tutte e tre le tabelle (Documento, Libro, Testo) a causa delle stesse problematiche incontrate nella gestione delle tabelle degli utenti:

- l'inserimento di un'unica tabella contenente i Documenti con un flag indicante la tipologia (Libro o Testo) avrebbe causato troppi attributi settati a NULL;
- la creazione delle sole due tabelle Libro e Documento è un'alternativa che non è stata inizialmente presa in considerazione in quanto avrebbe comportato difficoltà dal punto di vista della realizzazione delle query e delle varie relazioni che legano i documenti. Si pensa comunque di adottarla in futuro.

Quindi, la tabella che descrive il documento generico (sia libri che non) è caratterizzata da un ID (un valore intero per identificare il documento), l'Autore e il Titolo del documento, un Flag per indicare se il documento è pubblico o privato, e lo username dell'Admin che ha approvato il documento (nel caso in cui sia pubblico; viceversa, se tale attributo non è settato significa che ancora nessun Admin ha approvato il documento, e se il documento è privato quest'attributo sarà NULL).
Per quanto riguarda invece le tipologie di documento specifiche, il libro sarà caratterizzato dall'ID, chiamato "Documento", per indicare a quale entità della tabella documento è riferito, l'anno di pubblicazione, editore, genere, una breve descrizione e la copertina (il file vero e proprio). Per la visualizzazione di immagini prelevate dal database, è necessario conoscerne anche il formato specifico (il "mimetype"): quindi vi è nella tabella libro anche un attributo FormatoCopertina, indicante il formato della copertina caricata.

Se il documento non è un libro, sarà ugualmente caratterizzato dall'ID per identificare il documento al quale è associato nella tabella "Documento", con l'aggiunta della materia e dell'argomento del documento stesso.
Vi sono inoltre due tabelle, Doc_Text e Doc_Image, che servono a memorizzare i contenuti veri e propri dei documenti (non vengono salvati sul filesystem del server, ma direttamente sul database).
La tabella Doc_Text, come le tabelle Libro e Testo, è caratterizzata dall'ID del documento al quale si riferisce. Inoltre, è necessario memorizzare anche un IDTesto, in quanto ogni documento può essere diviso in più parti di testo, tipicamente capitoli, il (o la parte di) testo vero e proprio del documento, e l'ordine di visualizzazione delle varie parti di testo che lo compongono.
Anche per quanto riguarda la tabella Doc_Image, si memorizza l'ID del documento, un IDImmagine, in quanto anche in questo caso, per ogni documento, possono esserci più immagini. Viene poi memorizzata l'immagine vera e propria, rappresentato da codice binario, il formato dell'immagine (per l'eventuale visualizzazione su sito web), e l'attributo HRef, che serve per identificare, attraverso i tag <img> del documento, ogni immagine. Per capire nel dettaglio l'utilità di quest'ultimo attributo, vedere il paragrafo 9.1, Caricamento e salvataggio.

8.5. Possibili Miglioramenti

Dal punto di vista progettuale, si prevede l'inserimento di una tabella riguardante le segnalazioni, rilasciabili dagli utenti nel caso in cui ritengano che un documento contenga dei contenuti non consoni. Si prevede, inoltre, la creazione di una tabella contenente i feedback che ogni utente può rilasciare dopo la lettura di un documento. Sarebbe, inoltre, necessario integrare la rimozione di un account utente da parte degli Admin, delegando agli stessi la gestione dei dati che dovranno essere necessariamente rimossi e quelli che invece possono essere mantenuti.
È prevista l'integrazione, all'interno della tabella utente, di un attributo "tema", indicante quale tema si desidera avere come predefinito ad ogni accesso.
Si prevede di memorizzare dati statistici di vario genere, rigorosamente in forma anonima, come il grado di dislessia medio degli utenti del sistema V.E.R.O.N.I.C.A., i risultati medi dei test, ecc.
È possibile integrare, inoltre, un sistema di conteggio di letture per ogni documento pubblico, in modo tale che possa essere creata una sezione contenente i libri più letti.
È pianificata l'integrazione di un sistema di accesso al sistema mediante Tessera Sanitaria; pertanto, sarà necessario integrare i dati delle tessere degli utenti all'interno del sistema V.E.R.O.N.I.C.A. Per un ulteriore livello di sicurezza, è sotto esame la fornitura di un dispositivo, da fornire all'utente, che permetta l'accesso al sistema mediante lettura della stessa Tessera Sanitaria.

8.6. Piano delle Query

Durante il progetto della base di dati, si è tenuto conto degli aspetti legati alla stesura del codice per la realizzazione dell'applicativo: in particolar modo, è stato necessario pianificare le query durante lo sviluppo delle diverse funzionalità. Si può affermare che esse hanno condizionato la gran parte delle scelte effettuate per la realizzazione della base di dati. Di seguito vengono indicate le principali query progettate ed utilizzate per realizzare diverse funzionalità del sistema.

Creazione tabella Utente

```
CREATE TABLE Utente (
Username VARCHAR(20) PRIMARY KEY,
Password VARCHAR(20),
Nome VARCHAR(20),
Cognome VARCHAR(20),
DataNascita DATE,
Sesso CHAR,
AdminIscr VARCHAR(20),
Chiave VARCHAR(20),
FOREIGN KEY AdminIscr REFERENCES Admin(Username)
)
```

Creazione tabella Dislessico

```
CREATE TABLE Utente (
Username VARCHAR(20) PRIMARY KEY,
Fascia INT,
GradoDislessia FLOAT,
FOREIGN KEY Username REFERENCES Utente(Username)
)
```

Creazione tabella Documento

```
CREATE TABLE Documento (
ID INT PRIMARY KEY,
Utente VARCHAR(20),
Autore VARCHAR(20),
Titolo VARCHAR(20),
Pubblico INT,
AdminApprovazione VARCHAR(20),
DataOra_Upload TIMESTAMP,
FOREIGN KEY Utente REFERENCES Utente(Username)
)
```

Creazione tabella Libro

```
CREATE TABLE Libro (
Documento INT PRIMARY KEY,
Anno INT,
Editore VARCHAR(20),
Descrizione VARCHAR(200),
Genere VARCHAR(20),
Copertina MEDIUMBLOB,
FormatoCopertina VARCHAR(20),
ISBN VARCHAR(20),
FOREIGN KEY Documento REFERENCES Documento(ID)
)
```

Approvazione di un documento

```
UPDATE Documento
SET AdminApprovazione =  $usernameAdmin
WHERE  ID = $key
```

(L'ID del Documento è contenuto nella variabile "$key")

È la query da eseguire quando l'amministratore, dopo aver visionato un documento, decide di approvare la richiesta di pubblicazione effettuata dal proprietario del documento. Viene modificato un solo attributo, fino a quel momento settato a NULL, con l'inserimento dello username dell'admin che ha approvato il documento.

Visualizzazione dettagli documento

SELECT Titolo, Autore, Utente
FROM Documento
WHERE ID=$key
(L'ID del Documento è contenuto nella variabile "$key")

In generale, quando vi è necessità di visualizzare i dettagli di un dato documento presente nel sistema (indipendentemente dalla tipologia), viene utilizzata la query descritta. Per venire a conoscenza del documento a cui far riferimento, è sufficiente conoscerne l'Id.

Eliminazione documento

DELETE FROM Documento WHERE ID ='$key
DELETE FROM Libro WHERE ID =$key (se è un libro)
DELETE FROM Testo WHERE ID =$key (se non è un libro)
DELETE FROM Doc_Text WHERE Documento =$key
DELETE FROM Doc_Image WHERE Documento =$key (se possiede immagini)

L'eliminazione di un documento viene solitamente comandata dal proprietario del documento stesso (se è privato), quando questi non desidera più tenerlo nella propria raccolta personale (se il documento è stato approvato come pubblico, la giurisdizione del documento passa agli amministratori, e l'utente che ha caricato il documento non ha più la possibilità di eleminarlo dal sistema). Dal database, sarà necessario eliminare la tupla contenente i dati generali del documento, la tupla della tabella Libro (se il documento che si vuole eliminare è un libro) o quella della tabella Testo (nel caso in cui il documento non è un libro), la tupla o le tuple contenenti il contenuto del documento nella tabella Doc_Text, e le eventuali tuple contenenti le immagini associate al documento nella tabella Doc_Image. Anche in questo caso, il documento da eliminare viene identificato attraverso l'Id.

Visualizzazione dei libri con richiesta di pubblicazione e non ancora approvati

SELECT Titolo, Autore, Utente
FROM Documento
WHERE Pubblico=1 AND AdminApprovazione IS NULL
ORDER BY Titolo

Tale query viene eseguita quando un Amministratore desidera visualizzare l'elenco dettagliato dei documenti con richiesta di pubblicazione pendente. La clausola "WHERE" esegue due controlli:

- Pubblico=1: il documento risulta pubblico (un documento privato possiede l'attributo Pubblico=0);
- AdminApprovazione=NULL: il documento non è stato approvato da un amministratore (questo attributo perde valore nel momento in cui un documento risulta privato).

Per una miglire comprensibilità per l'amministratore, l'elenco dei documenti viene visualizzato in ordine alfabetico in base al Titolo.

Visualizzazione dettagli di un libro

SELECT Titolo, Autore, Pubblico, Anno, Editore, Genere, Descrizione, Copertina, TipoCopertina
FROM Documento, Libro
WHERE Documento.ID=Libro.Documento AND Documento.ID=$id
(L'ID del Documento è contenuto nella variabile "$id")

Questa query è utilizzata quando vi è necessità di visualizzare i dettagli di un dato libro presente nel sistema. Si differenzia dalla visualizzazione del documento in quanto è sviluppata solo ed esclusivamente per i libri, identificati mediante ID.

Visualizzazione dettagli di un testo

SELECT Titolo, Autore, Tipologia, Materia, Argomento
FROM Documento, Testo
WHERE Documento.ID=Testo.Documento AND Documento.ID=$id
(L'ID del Documento è contenuto nella variabile "$id")

La query indicata è simile alla precedente, con la differenza che l'obiettivo della stessa è visualizzare i dettagli di un dato documento che non sia un libro (e quindi presente nella tabella Testo). Anche in questo caso, il "testo" è identificato mediante ID.

Visualizzazione dei documenti pubblici approvati

SELECT ID, Titolo, Autore, Utente
FROM Documento
WHERE Pubblico=1 AND AdminApprovazione IS NOT NULL
ORDER BY Titolo

In questo caso, la query ha come obiettivo la sola visualizzazione dei documenti pubblici già approvati e visualizzabile, quindi, da tutti gli utenti. Rispetto alla query che visualizzava i documenti da approvare, l'attributo AdminApprovazione questa volta deve essere diverso da NULL, ossia il documento deve essere già approvato da un Admin. Per una visualizzazione più semplice, la lista di documenti viene mostrata in ordine alfabetico in base al titolo dei documenti.

Visualizzazione dei libri pubblici approvati che iniziano per una data lettera

SELECT ID, Titolo, Autore, Utente
FROM Documento
WHERE Pubblico=1 and Titolo REGEXP "^$lettera" AND AdminApprovazione IS NOT NULL
ORDER BY Titolo
(La lettera fornita è memorizzata nella variabile $lettera)

Anche tale query è simile alla precedente, ad eccezione di un controllo aggiuntivo nella clausola "WHERE": il titolo dei libri visualizzati deve necessariamente iniziare per una data lettera. Il suddetto controllo è eseguito mediante espressioni regolari, nei quali il simbolo "^" indica che si fa riferimento al primo carattere di una stringa. Questa query è utilizzata per fornire una funzionalità aggiuntiva agli utenti, semplificando una ricerca che sarebbe, altrimenti, troppo pesante (anche se i documenti si dovessero trovare in ordine alfabetico) nel

caso in cui la lista sia composta da molti elementi. L'utente che sta cercando uno specifico documento e ne conosce il titolo, deve selezionare la lettera iniziale del titolo, ed avrà una visualizzazione dei soli documenti che iniziano per la lettera selezionata.

Visualizzazione di un capitolo di un documento

```
SELECT Testo
FROM Doc_Text
WHERE Documento=$id AND Ordine=$cap
```

(La chiave del Documento è contenuta nella variabile $id, il capitolo in $cap)

Questa query viene effettuata quando un utente ha intenzione di leggere (o di far leggere al sintetizzatore) un dato documento, e, nello specifico, un dato capitolo di tale documento. Ogni documento presente all'interno di V.E.R.O.N.I.C.A. è sempre composto da almeno un "capitolo": quindi, per poter leggere un documento, la query dovrà essere eseguita almeno una volta.

Visualizzazione dei documenti appartenenti ad un utente

```
SELECT ID, Titolo, Autore, Utente, Pubblico
FROM Documento
WHERE utente=$username
ORDER BY titolo
```

(Lo username dell'utente è contenuto nella variabile $username)

Quando un utente vuole visualizzare l'elenco dei propri documenti, aprirà una pagina che eseguirà la query indicata in precedenza. È opportuno rimarcare che non viene fatto nessun controllo sull'attributo "Pubblico" della tabella Documento, in quanto non è necessario fare questa distinzione: i documenti che si vogliono visualizzare sono quelli che sono stati caricati da un determinato utente, indipendentemente dal fatto che siano visibili da tutti gli altri utenti o meno.

Visualizzazione dei dati generali di un utente

```
SELECT Nome, Cognome, Mail, DataNascita, Sesso, AdminIniscrizione
FROM Utente
WHERE username=$username
```

(Lo username dell'utente è contenuto nella variabile $username)

Per poter visualizzare i dati di un utente, è necessaria l'esecuzione della query indicata. Vengono visualizzati solamente i dati generali di ogni utente; per visualizzare anche i dati specifici per ogni categoria, sarà sufficiente eseguire un join sull'attributo username, tra la tabella Utente e la tabella rappresentante la categoria specifica dell'utente di cui si vogliono visualizzare i dettagli.

Ricerca di un documento attraverso un pattern

```
SELECT ID, Titolo, Autore, Utente, MATCH ( titolo ) AGAINST ( $pattern )
AS attinenza
FROM Documento
WHERE MATCH ( titolo ) AGAINST ( $pattern ) AND Pubblico=1 AND
AdminApprovazione IS NOT NULL
ORDER BY attinenza DESC
```

(Il pattern da ricercare è memorizzato nella variabile $pattern)

Oltre alla visualizzazione mediante ordine alfabetico, è possibile effettuare la ricerca di un documento pubblico inserendo una stringa nella barra di ricerca e visualizzando successivamente i risultati più consoni rispetto alla stringa inserita. La query sovrastante esegue questa ricerca, a partire da un determinato pattern di ricerca.

Selezione dell'id massimo dei documenti

```
Select MAX(ID)
FROM Documento
```

Questa semplice query è utilizzata solamente per conoscere quale sarà l'Id da assegnare al prossimo documento da inserire nel sistema, ossia (Max(Id))+1.

Caricamento di un nuovo documento

Nel caso di caricamento di un documento, come per l'eliminazione, vi sono 5 differenti query che possono essere eseguite. Vengono descritte di seguito.

```
INSERT INTO Documento
VALUES ($id, $username, $autore, $titolo, $pubblico, NULL, $pubblico)
```

Questa query viene sempre eseguita al caricamento di un documento. Si tratta dell'inserimento dei dati nella tabella di un documento generico, a prescindere della tipologia di documento caricato. Quando il documento viene inserito, non è stato ancora approvato da un amministratore (sempre se deve esserlo), quindi AdminApprovazione è settato a NULL.

```
INSERT INTO Libro
VALUES ($id, $anno, $editore, $descrizione, $genere, $copertina, $formatoCopertina, $ISBN)
```

Questa query viene eseguita solamente nel caso in cui il documento caricato sia un libro. Vengono quindi inseriti, nella tabella dedicata esclusivamente ai libri, tutti i dettagli del documento caricato.

```
INSERT INTO Testo
VALUES ($id, $materia, $argomento)
```

Questa query, invece, viene eseguita solamente nel caso in cui il documento caricato non sia un libro. Anche in questo caso, vengono caricati i dati riguardanti un semplice "testo" caricato da un utente.

```
INSERT INTO Doc_Image
VALUES ($id, $idImmagine, $image, $href, $mimetype)
```

Questa query viene eseguita solamente se il documento caricato possiede almeno un'immagine, e verrà eseguita tante volte quante immagini vi sono all'interno del documento.

```
INSERT INTO Doc_Text
VALUES ($id, $idcap, $testo, $ordine);
```

Questa query viene, invece, eseguita sempre, in quanto ogni documento possiede almeno un "capitolo", e viene anch'essa eseguita un numero di volte pari al numero di capitoli per ogni documento.

Login
SELECT Utente FROM Login WHERE SessionID = (sessionID)

È la query che garantisce che l'utente abbia effettuato il login in precedenza ed abbia una sessione valida, nello specifico non scaduta a causa di inutilizzo, attraverso un doppio controllo. Il primo dato recuperato è l'id di sessione dal relativo cookie di sessione, impostato dal sistema sul computer client dell'utente che sta accedendo a V.E.R.O.N.I.C.A.
La presenza di tale id all'interno della tabella login rappresenta il secondo controllo effettuato, dal momento che i dati all'interno della tabella specificata, saranno inseriti solo dopo che l'utente ha effettivamente effettuato il login e ricevuto conseguente approvazione di accesso da parte del sistema.

8.7. Impostazioni Specifiche

8.7.1. Gestione iniziale del database tramite Drupal

Inizialmente, durante la definizione delle strategie di sviluppo iniziali, è stata valutata l'idea di utilizzare un ambiente di sviluppo, Drupal, che fornisse un modulo per la registrazione degli utenti e un'interfaccia già pronti, con l'aggiunta di altri moduli preinstallati o installabili. Tale ambiente è stato adottato come struttura di sviluppo per un periodo iniziale di alcune settimane. Durante il suo utilizzo, il team si è reso conto che sarebbe stato solo un rallentamento per lo sviluppo della piattaforma, dal momento che le conoscenze del team si sono rilevate sufficienti ma, soprattutto, poiché la tipologia di lavoro da realizzare non obbligava in alcun modo la dipendenza da altre piattaforme. Dopo qualche settimana di studio e utilizzo, con scrittura delle prime bozze di pagine si è quindi deciso di abbandonare Drupal e realizzare l'applicazione senza alcune basi pronte. Tale scelta, unita alla potenza dei linguaggi di programmazione adottati si è rilevata ottima per quanto riguarda tutta la fase di sviluppo della piattaforma, ma non solo: il DBMS MySQL ha, infatti, consentito una ampia libertà per quanto riguarda la gestione, lo sviluppo e la manutenzione della base di dati, al contrario dei moduli preinstallati forniti da Drupal, che hanno rallentato la stesura del progetto a causa della presenza di schemi preesistenti dal quale V.E.R.O.N.I.C.A. avrebbe dovuto dipendere per differenti funzionalità.

8.7.2. Gestione Documenti su Database

Inizialmente, l'intenzione era di creare sul server, per ogni libro caricato, una cartella contenente, oltre il libro, anche la sua copertina e eventuali altri dati ad esso correlati. Successivamente, con l'intenzione di aggiungere il formato epub all'insieme di formati di libri accettati dal sistema, si è deciso di gestire tutto tramite il database sql, questo per permettere una una gestione più rapida (e comoda) degli epub, visto che la struttura fisica del file prevede una suddivisione in capitoli e l'inserimento di eventuali immagini all'interno del testo. Per questi motivi, sono state create due nuove tabelle, direttamente collegate alla tabella Documento: Doc_Text e Doc_Image. Nella prima, per ogni libro, sarà caricato il testo dei file xml che nell'epub compongono ogni capitolo. Nella seconda, invece, verranno immagazzinate, per ogni libro, le eventuali immagini presenti. La gestione delle immagini è più complicata durante la lettura di un documento, in quanto è necessario scorrere tutti i file xml dei capitoli cercando i riferimenti alle immagini, e ogni volta che se ne trova uno, cercare l'immagine relativa nel database.
Vi sono diversi vantaggi e svantaggi nel gestire tutti questi dati sul database. Per quanto riguarda i file di testo, vi sono perlopiù solo vantaggi:

- MySQL è in grado di gestire in modo performante stringhe di testo anche molto lunghe;
- Avere il testo sul database e non sul filesystem gestito dal sistema operativo del server, permette al web server Apache di mantenere in cache un eventuale testo che viene richiesto più volte contemporaneamente, permettendone un caricamento più veloce, a differenza del file su filesystem che richiede almeno un lettura da disco fisso per ogni accesso.

Per quanto riguarda invece la gestione di immagini, gli svantaggi sono soprattutto due:

- Difficoltà nel gestire i file binari sul database tramite linguaggio PHP;
- Le tabelle tendono ad essere pesanti.

Vi sono anche però alcuni vantaggi:

- Nell'eventualità di dover spostare l'intero sistema in un altro server, sarà sufficiente effettuare una copia del database e inserirla sul nuovo server, senza nessun rischio riguardante incompatibilità dovuta a sistema operativo o filesystem diverso;
- Maggior sicurezza: un attaccante o un amministratore sbadato avrebbe il potere di sostituire l'immagine sul filesystem, mettendone semplicemente un'altra con lo stesso nome; questo, con la gestione delle immagini sul database, non è possibile;
- varie protezioni sulla scrittura dei dati offerte dal database: supporto unico, aggiornamento unico, coerenza dei dati, affidabilità, privatezza, efficienza, efficacia.

Con la decisione di mettere nel database tutti i dati, salvare i dati anche su disco fisso diventa "obsoleto". Quando però i dati risiedevano sul disco fisso, venivano inseriti ognuno in una cartella numerata, e uno script bash si occupava di generare il numero della cartella. Quando un documento, e quindi la sua relativa cartella, veniva eliminato, si creava un "buco" nella numerazione, che però era facilmente rimpiazzabile grazie allo script bash, che verificava il primo numero disponibile. Con la gestione dei dati sul database, però, questo non è più possibile: non vi sono funzioni in grado di comprendere quale sia l'indice ideale col quale andare a salvare il documento caricato. L'unico modo è utilizzare un indice incrementale, che ad ogni caricamento si incrementa automaticamente. L'indice incrementale però non risolve il problema dei "buchi": se un documento viene eliminato, quell'indice rimarrà inutilizzato.

9. Gestione Documenti

9.1. Caricamento e salvataggio

Per quanto riguarda la gestione dei documenti, vi sono diverse problematiche che sono state affrontate. Il fatto che ogni utilizzatore abbia la possibilità di poter caricare un proprio documento, implica la necessità di una gestione dei vari formati in cui un documento digitale si può presentare. Questo perché abbiamo la necessità di estrarre il testo (eventualmente formattato) e visualizzarlo con un font speciale adatto alla lettura per gli utenti. Inoltre, il testo va passato al sintetizzatore in modo tale che possa essere riprodotto. È quindi essenziale che si abbia la possibilità di trattare del testo (stringhe).
Quando un utente desidera caricare sul server un documento, potrà scegliere se caricare:

- un libro;
- un documento generico.

Inoltre potrà scegliere se renderlo pubblico (dovrà prima essere approvato da un amministratore) o privato. Tutte le caratteristiche del documento verranno caricate nelle relative tabelle sul database. Premendo il tasto invia, il file viene caricato e viene controllato se il formato è supportato dal sistema. Dopodiché viene caricato su una directory temporanea sul server, e elaborato in modo tale che venga trasferito sul database, in modo tale che sia sempre a disposizione dell'utente che desidera accedervi.

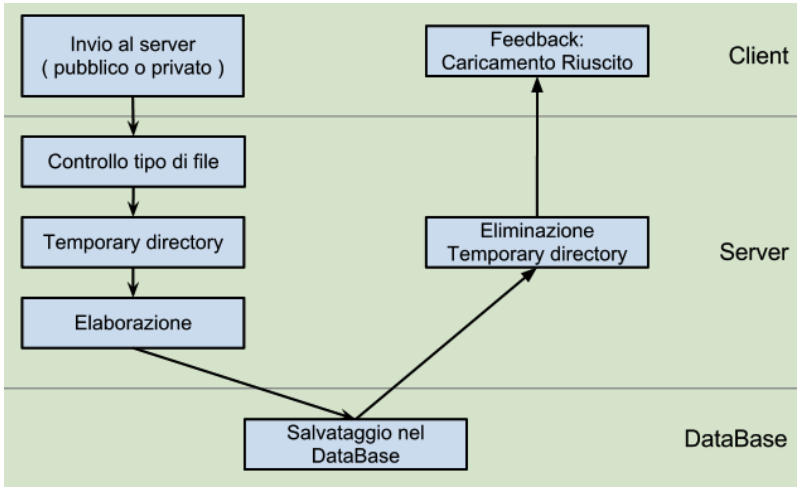

Immagine 9.1

La scelta di salvare il documento sul database offre diversi vantaggi, soprattutto per quanto riguarda l'efficienza del sistema: un documento che viene prelevato dal database, viene gestito dal Web Server Apache, che è in grado di utilizzare diverse ottimizzazioni per quanto riguarda la ricerca e la richiesta di un determinato oggetto nei database. Una di queste è l'utilizzo di una memoria cache che permette quindi il prelievo molto più veloce di un documento che, per esempio, viene richiesto molto spesso. Questo tipo di vantaggio non si potrebbe avere se si gestissero i documenti direttamente sul filesystem del server, in quanto sul database verrebbe salvata solamente la directory della posizione del documento, implicando quindi una successiva lettura da disco, sempre obbligatoria (che verrebbe quindi effettuata anche quando viene richiesto più volte lo stesso documento). D'altro canto, la gestione dei documenti su database complica molto le cose dal punti di vista implementativo, soprattutto per quanto riguarda i documenti con formato epub.
Durante lo sviluppo del sito, si è deciso di supportare inizialmente due diversi formati:

- TXT
- EPUB

Tutti gli altri formati al momento non sono supportati, e il sistema restituirà un messaggio di errore nel caso in cui si tenti di caricare un file con formato diverso dai due citati sopra.
Di seguito, verranno descritte nel dettaglio le gestioni dei formati supportati, le relative problematiche e il salvataggio dei documenti sul database.

9.1.1. TXT

La gestione dei file in formato txt è abbastanza semplice in quanto caratterizzato da un semplice insieme di caratteri, ossia una stringa. Non vi possono quindi essere formattazioni del testo particolari (titoli, paragrafi, grassetto, corsivo, ecc.). È quindi sufficiente, una volta caricato il file sulla directory temporanea, aprire il file di testo, leggerne il suo contenuto e trasferirlo nel database, per poi eliminare la directory temporanea contenente il documento stesso. Trovandoci su una piattaforma linux, la lettura automatizzata dei semplici file di testo viene fata mediante l'istruzione PHP:
$testo=shell_exec("cat file.txt");
L'istruzione shell_exec esegue la stringa passata come parametro nella shell di sistema. Quindi, supponendo che il documento sia chiamato "file.txt", il comando bash cat ne visualizza il contenuto, che viene restituito in output e quindi salvato nella variabile $testo, che quindi, dopo l'esecuzione dell'intera istruzione, conterrà il contenuto del documento.
Come già discusso nel paragrafo che discute del database, ad ogni id di un documento (tabella Doc) sono associate una o più stringhe di testo che lo compongono (Doc_Text). Quindi, quando viene caricato il documento, vengono salvati con opportune query i dati che lo caratterizzano e, attraverso l'istruzione descritta precedentemente, viene caricato il testo del documento in un'unica tupla della tabella Doc_Text. Successivamente si può procedere alla cancellazione del documento dalla directory temporanea.
Per quanto riguarda la lettura, attraverso un'opportuna query, viene richiesto il testo del documento, viene diviso in pagine e visualizzato pagina per pagina.

9.1.2. EPUB

Il formato epub è un formato molto utilizzato esclusivamente per libri digitali (a differenza del pdf, che oltre ai libri, può caratterizzare un'altissima varietà di documenti). Oramai, i principali rivenditori di libri elettronici, oltre al formato pdf (che per quanto riguarda i libri sta cadendo in disuso), permettono il download del libro in formato epub. Per questi motivi è stato scelto di supportare questo formato.
La gestione dei documenti in formato epub è molto più complessa rispetto ad un semplice file di testo. Un generico file in formato epub non è altro che un archivio zip, contenente al suo interno dei file che descrivono la struttura del libro (i file stessi da cui è formato, l'ordine in cui questi file devono essere visualizzati, eventuali immagini, file di stile, ecc.). Nello specifico, in ogni file epub, vi sono i seguenti file:

- mimetype: è un file contenente una sola riga: "application/epub+zip", indica sostanzialmente il tipo del file a cui si sta accedendo;
- container.xml (all'interno della cartella META-INF): è un file scritto in xml, che serve sostanzialmente ad indicare la posizione di un file in formato opf, che analizzeremo più avanti. Un esempio del codice contenuto nel file container.xml è il seguente:

```
<?xml version="1.0"?>
<container version="1.0"
xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
   <rootfiles>
      <rootfile full-path="OEBPS/content.opf"
       media-type="application/oebps-package+xml"/>
   </rootfiles
</container>
```

La struttura generale del codice è sempre la stessa per ogni file epub. Quello che può cambiare (anche se nella gran parte dei casi non cambia) è l'attributo full-path del tag rootfile. Quest'attributo contiene il collegamento al primo file che permetterà di analizzare e comprendere la struttura vera e propria di un epub. Quasi sempre, l'attributo full-path corrisponde a OEBPS/content.opf, ma non essendo questa una scelta obbligata per l'autore dell'epub, è sempre necessario analizzare il file container.xml. Una volta ottenuta la directory del rootfile, possiamo andare ad analizzarlo (per comodità, d'ora in avanti il rootfile lo chiameremo content.opf).

- content.opf: è il file più importante di tutto l'epub. è scritto in codice xml, e serve a descrivere l'intera struttura del documento. è suddiviso in tre parti:
    - Nella prima parte (metadata) vengono definiti i metadati del documento: titolo, autore, editore, numero di pagine, ecc.
    - Nella seconda parte (manifest) vengono elencati tutti i file all'interno del documento. Più precisamente, per ogni file ne è indicata la directory (relativa alla posizione di content.opf), e ad ogni file è associato un id che lo identifica.
    - Nella terza parte (spine) viene elencato l'ordine in cui devono essere visualizzati i file che formano il documento. Più precisamente, è indicato un ordine degli id di ogni file che forma il documento. Quindi, un generico lettore di epub, dovrà aprire il file content.opf, leggere in ordine gli id presenti nella terza parte, estrarre per ogni id la directory (cercando nella seconda parte), e successivamente cercare il file vero e proprio che compone il documento.

N.B. non necessariamente nella terza parte devono essere presenti tutti gli id dei file presenti nella seconda parte del file content.opf. è quindi possibile che l'autore inserisca nel documento dei file che però non verranno visualizzati dal lettore di epub.

- toc.ncx: nonostante l'estensione, è anch'esso un file xml. la sua posizione è indicata nel file content.opf, e il suo scopo è fornire un sommario dei contenuti del documento. Non è detto che l'autore completi in modo soddisfacente questo file (non vi è nessun vincolo).

Oltre questi file standard contenuti in un documento epub, vi sono tutti gli altri file che compongono il documento, che possono essere generalizzati a tre categorie principali:

- pagine xhtml, contenenti i contenuti del documento;
- immagini;
- file di stile (CSS).

Per poter capire meglio come viene gestito un epub, visualizziamo un esempio del file content.opf:

```
<?xml version="1.0" encoding="UTF-8"?>
<package>
    <metadata>
        <dc:title>I promessi sposi</dc:title>
        <dc:subject>Letteratura</dc:subject>
        <dc:creator opf:role="aut" opf:file-as="Manzoni,
Alessandro">Alessandro Manzoni</dc:creator>
        <dc:language>it</dc:language>
        <meta name="cover" content="copertina.png"/>
    </metadata>
    <manifest>
        <item id="ncx" href="toc.ncx" media-type="application/x-
dtbncx+xml"/>
        <item id="copertina.png" href="Images/copertina.png" media-
type="image/png"/>
        <item id="Style0001.css" href="Styles/Style0001.css" media-
type="text/css"/>
        <item id="Section0001.xhtml" href="Text/Section0001.xhtml"
media-type="application/xhtml+xml"/>
        <item id="Section0002.xhtml" href="Text/Section0002.xhtml"
media-type="application/xhtml+xml"/>
    </manifest>
    <spine toc="ncx">
        <itemref idref="Section0001.xhtml"/>
        <itemref idref="Section0002.xhtml"/>
    </spine>
</package>
```

Come si può notare, nello spine, che contiene l'ordine dei file da visualizzare per generare il documento, non vi sono tutti gli elementi presenti nel manifest. Un altro aspetto importante da notare è che i file presenti nello spine sono tutti file con formato xhtml: il documento quindi è una vera e propria pagina web, il cui codice sarà formato dal codice delle pagine presenti nello spine. A sua volta, ogni pagina potrà fare riferimento ad altri file presenti nel manifest.
A questo punto, per poter salvare un intero file epub in un database sarebbe necessario salvare:

- l'elenco dei file presenti nel manifest;
- l'ordine di visualizzazione dei file nello spine.

Come già detto precedentemente però, nel nostro sistema è necessario che il testo dei documenti sia visualizzato con un font che faciliti la lettura agli utenti. Quello che effettivamente serve sul sistema è la visualizzazione del documento, del testo e delle eventuali immagini. Quindi si è deciso di agire nel modo seguente: per ogni file xhtml presente nella sezione spine, si cerca la directory nella sezione manifest, si estrae il testo del file come fatto per i file txt, e si salva sul database. Successivamente, si analizza la sezione manifest e si salvano sul database le immagini (ossia i tag che hanno contengono nell'attributo media-type la sottostringa image). Vediamo più nel dettaglio le varie procedure, analizzando tutte le varie problematiche.
I file che compongono l'intero documento non sono dei txt, ma delle pagine xhtml: sono composte da tag, a differenza dei semplici file di testo. Siccome devono essere visualizzate su un sito web, non è un grandissimo problema memorizzare anche i tag xhtml: nella stampa del testo, quando il documento verrà letto, verrà visualizzato normalmente dal browser. Inoltre, siccome non verrà trovato il file di stile utilizzato dal file epub (abbiamo deciso di non salvarlo sul database), il font con il quale verrà stampato il libro sarà il font del sito. L'unica

problematica da risolvere sta nel fatto che, essendo pagine xhtml pure, contengono anche i tag <body>, e visto che il documento deve essere visualizzato in una pagina web contenente già un tag <body>, si otterrebbe alla fine una pagina di visualizzazione con più <body> annidati, il browser darebbe errore e non verrebbe visualizzato niente. Si è deciso di salvare solo il codice contenuto all'interno del tag <body> di ogni pagina xhtml. Il codice è stato estratto mediante l'xpath.
Come già precisato nel capitolo sul database, ad ogni documento (tabella Doc) è associato un insieme di testi (tabella Doc_Text) e un insieme di immagini (tabella Doc_Image). Per quanto riguarda i file xhtml, quindi, vengono salvati nella tabella Doc_Text sottoforma di stringa (come avviene per i file txt), e per ogni tupla vi è un intero che indica l'ordine di visualizzazione delle pagine xhtml. Per quanto riguarda invece la gestione delle immagini, questa diventa più problematica. Le immagini in un file epub vengono richiamate esattamente come un qualsiasi sito web normale: attraverso un attributo src che ne indica la posizione. Quindi, ogni pagina xhtml contenente immagini, conterrà al suo interno in tag del tipo:

```
<img src="../Images/image.png">
```

Si noti che l'immagine, all'interno del file epub, potrebbe trovarsi dovunque, e la directory all'interno dell'attributo src è relativa alla posizione del file xhtml che la richiama. Questo implica che dobbiamo necessariamente modificare tutti i tag img presenti nelle pagine xhtml del documento se vogliamo conservare anche le immagini dei libri. Più precisamente, ogni tag va sistituito con una query che preleva l'immagine dal database e la visualizza. Il problema che più di tutti ha complicato la gestione del formato epub è quello di identificare correttamente l'immagine che si vuole visualizzare.
Osservando gli attributi di cui sono dotati i tag item di tipo immagine nella sezione manifest del file content.opf, possiamo notare che vi sono un id e un href. Intuitivamente verrebbe da pensare che ogni immagine si potrebbe facilmente identificare con l'attributo id, ma nelle pagine xhtml, ogni immagine si può identificare con l'attributo src. Quindi, sempre intuitivamente, verrebbe da pensare che si potrebbe effettuare l'identificazione mediante l'uguaglianza tra l'attributo src del tag img e l'attributo href che abbiamo sul database, prelevato dal manifest. Questo, però, funzionerebbe solamente se il file content.opf e i vari file xhtml si trovassero sulla stessa directory. Infatti, i riferimenti alle varie immagini sono sempre relativi.
Per risolvere questo problema, è necessario analizzare tutte le possibili casistiche. Abbiamo tre fattori differenti in gioco: la posizione del file xhtml, la posizione dell'immagine e la posizione del file content.opf (si ricorda che in un file epub, tutti e 3 questi file possono trovarsi dovunque).
Ricordiamo che il nostro problema sta nell'identificare una stringa univoca che identifichi un'immagine, avendo come dati in input:

- la posizione del file content.opf;
- la posizione del file xhtml relativamente alla posizione del file content.opf;
- la posizione del file immagine relativamente alla posizione del file content.opf;
- la posizione del file immagine relativamente alla posizione del file xhtml.

Tutti e quattro questi input sono stringhe.

Per semplificare il ragionamento, analizziamo l'esempio più comune di un file epub, ossia la struttura rappresentata in figura:

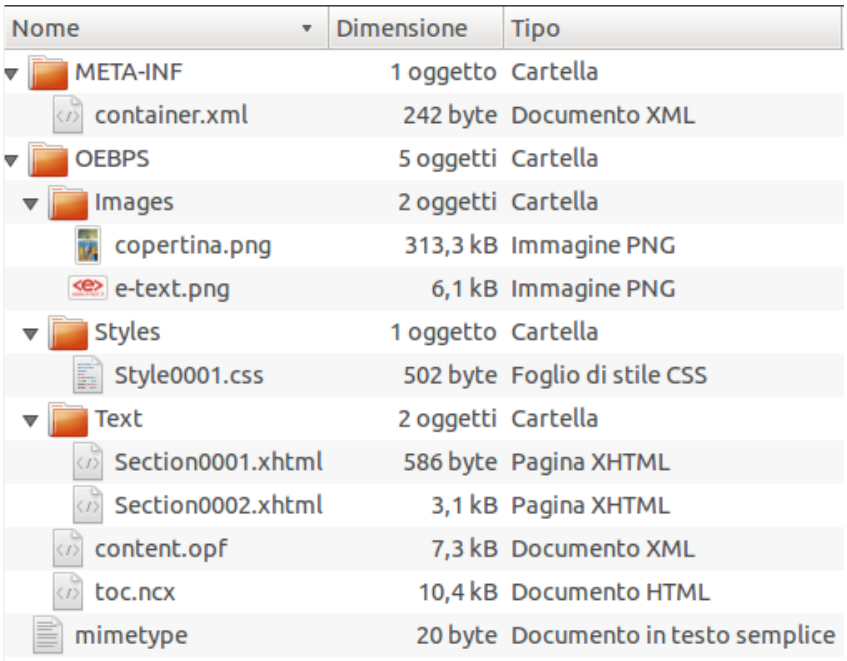

Immagine 9.2: Struttura di un file ePub

Il file content.opf si trova all'interno della cartella OEBPS/, i file xhtml si trovano in OEBPS/Text/, e i file immagine si trovano in OEBPS/Images/.
N.B: Se si avesse avuta sempre la certezza che i file xhtml e il file content.opf si trovassero nella stessa directory, il problema non si sarebbe posto, a prescindere dalla posizione delle immagini: il file content.opf deve per forza contenere la directory relativa di tutte le immagini, ed essendo il file xhtml nella stessa directory, l'attributo src sarebbe stato uguale all'attributo href. Essendo l'attributo href del file content.opf univoco (due immagini non possono occupare la stessa posizione con lo stesso nome), sarebbe stato sufficiente identificare l'immagine con quest'ultimo attributo.
Nel caso in figura, invece, la situazione è più complicata: nel file content.opf, l'attributo href di un'immagine sarebbe del tipo: Images/copertina.png, mentre all'interno di un file xhtml, l'attributo src sarebbe: ../Images/copertina.png (si parte dalla directory Text). In generale, se si vuole identificare un elemento attraverso la propria posizione, è necessario che venga analizzata la directory globale, ossia il percorso per arrivare al file a partire da una directory radice. In questo caso, la directory radice sarà ovviamente la directory contenente l'archivio epub estratto. Dato il percorso globale di un generico file, e il percorso relativo di un altro file, a partire dal percorso del primo, è possibile calcolare il percorso globale del secondo file. Ad esempio, sempre utilizzando il caso precendente, il percorso globale del file content.opf (che possiamo trovare direttamente nel file container.xml), è: OEBPS/content.opf; il percorso relativo a partire dalla directory di content.opf per arrivare all'immagine è: Images/copertina.png. La directory globale dell'immagine diventerà quindi: OEBPS/Images/copertina.png. Immaginando ora che l'immagine venga richiamata dal file Section0001.xhtml, con directory assoluta OEBPS/Text/Section0001.xhtml (calcolabile nello stesso modo descritto precedentemente con il file content.opf), l'attributo src sarà: ../Images/copertina.png. Si otterrà: OEBPS/Text/../Images/copertina.png. Costruendo un opportuno algoritmo che, all'incontro del pattern ".." cancella la directory precedente, si

ottiene la la directory globale: OEBPS/Images/copertina.png. In questo modo, la directory globale diventa una stringa identificativa per ogni immagine.
Quindi, al caricamento del documento, vengono analizzate e salvate sul database tutte le immagini elencate nel manifest di content.opf, e per ogni immagine viene calcolata la relativa directory globale. Successivamente, per ogni pagina xhtml presente nello spine, vengono cercati tutti i tag img (mediante xpath), e il loro attributo src vengono modificati, inserendo un link ad una pagina PHP che si occupa della visualizzazione delle immagini presenti sul database, passandole come attributi la directory globale dell'immagine alla quale si riferiscono e l'id del documento.

10. Modalità di accesso

10.1. Contesto ed Obiettivi

Come indicato finora, è importante fornire una distinzione chiara e precisa a proposito delle tipologie di utenza a cui garantire l'accesso, ma soprattutto, distinguere in maniera rimarcata la gestione delle diverse funzionalità attribuite ai profili utente di V.E.R.O.N.I.C.A.
Di seguito vengono indicate le strategie adottate in relazione alla tipologia di sistema sul quale ci si è basati e alle metodologie, principalmente programmative, utilizzate per lo sviluppo.

10.2. Architettura Server

Come indicato nella sezione Architettura del sistema, il sistema V.E.R.O.N.I.C.A. è pensato sulla base di una architettura a tre livelli, che permette una accurata suddivisione modulare dei servizi da offrire all'utente e dei servizi trasparenti all'utente, ma funzionali al sistema stesso.
A livello progettuale, tale suddivisione è fondamentale per garantire una gestione distribuita del sistema, sia per quanto concerne l'aspetto di programmazione, che di controllo e gestione dei vari moduli.
Ad ogni modo, per la parte implementativa sono state effettuate alcune variazioni rispetto al modello architetturale previsto, che non hanno comportato nessun cambiamento in merito al funzionamento del sistema stesso.
In primo luogo, vanno precisate le caratteristiche della macchina server adottata e le esigenze al momento della prima realizzazione del sistema V.E.R.O.N.I.C.A..
Il server fornito in dotazione possiede le seguenti caratteristiche:

- Processore Intel Pentium III;
- Ram: 2GB;
- Hard Disk: 100GB.

Inoltre, la quantità e la tipologia di dati di cui è richiesta la memorizzazione comporta un carico poco elevato sul sistema. Tali motivazioni giustificano, in conclusione, la scelta di un architettura fisica a due livelli, vale a dire l'architettura tipica client-server, sebbene l'architettura logica rimanga a tre livelli, in modo che l'architettura fisica si possa rimodellare su di essa, nell'eventualità in cui si renda necessario un cambiamento.

10.3. Macchina server

Come accennato nel paragrafo precedente, il server utilizzato è stato dotato di entrambe le componenti che contraddistinguono, nell'architettura logica, l'application server ed il data access server.
Nello specifico, il server è stato dotato del sistema operativo "Linux Xubuntu", descritto nel paragrafo seguente, di cui è stato appositamente dotato dei componenti forniti dalla piattaforma LAMP:

- Web server "Apache HTTP Server";
- DBMS "MySQL";
- PHP.

Come noto, V.E.R.O.N.I.C.A. è pensato come un servizio gratuito a favore delle persone che soffrono di dislessia: non avere dei costi sull'utilizzo dei moduli utilizzati per lo sviluppo e la gestione riduce i costi generali, permettendo al progetto di rimanere gratuito. Per tali

motivazioni la scelta delle componenti è ricaduta, per quanto è stato possibile, su piattaforme di sviluppo e software liberi.

10.3.1. Linux Xubuntu

Sono elencate, in precedenza, le caratteristiche hardware della macchina server: come si è potuto notare, sebbene tale macchina sia stata più che sufficiente per lo sviluppo del progetto, è stato essenziale dotarla di un sistema operativo che si adattasse in maniera egregia alle proprie caratteristiche. La scelta del team è caduta sul sistema operativo "Xubuntu 12.04" per diverse motivazioni che tengono alto il rapporto qualità/prezzo.
Innanzitutto, l'interfaccia grafica del sistema svolge un ruolo fondamentale nel bilancio delle sue prestazioni: in questo caso, Xfce, l'interfaccia grafica fornita in dotazione su Xubuntu, offre prestazioni funzionali pur mantenendo una richiesta di risorse piuttosto limitata, garantendo così un funzionamento ottimale della macchina.

10.3.2. LAMP: Apache, MySQL, PHP

Si tratta della piattaforma che offre moduli open source per lo sviluppo di applicazioni web.
Nell'ordine, Apache è il modulo che permette alla macchina fornita in dotazione di essere considerata un vero e proprio server. Sono delegati ad Apache, infatti, le funzionalità di comunicazione e trasmissione dei dati verso l'esterno, in particolare attraverso l'utilizzo del protocollo HTTP che, come noto, permette il trasferimento di ipertesti tra client e server; si ricorda che Apache è stato impostato per la comunicazione, tramite il protocollo TCP, sulla porta 80 del server di cui si è disposto.
Allo stesso modo, il team, attraverso i rispettivi files di configurazione ha impostato l'utilizzo dei protocolli SSH e SFTP, essenziali per una gestione completa della macchina server e per effettuare i test del sistema direttamente su di esso. In particolare il protocollo SFTP è stato adottato per garantire un trasferimento di dati sicuro ai soli utenti connessi mediante login: è stata opportunamente disabilitata la possibilità di accesso anonimo.
Una volta effettuato tale tipo di suddivisione, è stato sufficiente impostare il DBMS fornito, MySQL, per le esigenze di sviluppo. È opportuno ricordare, anche in questa sede, che l'architettura del progetto nasce distinta su tre livelli, ma le esigenze di sviluppo e le possibilità ridotte hanno portato all'utilizzo di un architettura effettiva a due livelli, come descritto approfonditamente nel paragrafo 3.2) Architettura del sistema.
Infine, come visto nei restanti moduli, il PHP è stato il linguaggio adottato per lo sviluppo delle pagine web lato server.

10.4. Gestione delle sessioni utente

Il sistema V.E.R.O.N.I.C.A. permette, come detto, l'accesso esclusivo agli utenti registrati. Tale aspetto costituisce un fondamento teorico importante per gestire al meglio gli utilizzi del sistema e fornire una navigazione piacevole e sicura, attraverso il salvataggio di informazioni legate a preferenze personali e dettate dalle funzionalità del sistema.
Il sistema di memorizzazione scelto è basato sull'utilizzo delle sessioni, gestito attraverso le funzionalità predefinite offerte dal linguaggio PHP.
Innanzitutto, è opportuno precisare che, dal punto di vista funzionale, non si ha un carico di dati elevato di cui tenere traccia per offrire l'accessibilità; in questa versione di V.E.R.O.N.I.C.A., si vuole garantire il riconoscimento dell'utente ad ogni suo accesso, memorizzare i suoi accessi, garantire che l'utente connesso sia effettivamente il proprietario di tale account.
Per tali motivi, ogni utente che effettua il login a V.E.R.O.N.I.C.A. attiva una sessione, che consente la memorizzazione di dati utili alla fruizione del servizio o alla personalizzazione

della navigazione utente; a titolo di esempio, si prenda il messaggio di benvenuto all'utente, comprensivo di nome e cognome, quando effettua l'accesso.
Più tecnicamente, all'atto della prima connessione tra client e server, il gestore delle sessioni, presente nel linguaggio PHP, si occupa di generare un identificativo univoco e di creare un file sul quale si andrà a memorizzare le cosiddette variabili di sessione, allo scopo di tener traccia di dati informativi sulla navigazione dell'utente, indicati in precedenza. Il salvataggio di tale file avviene sulla macchina server: ciò non è sufficiente per effettuare la comunicazione, verso il client, dei dati salvati. A tal proposito, PHP offre due strategie differenti, descritte di seguto:

- utilizzo di un cookie di sessione: il server si occupa dell'invio al client di un particolare cookie, sul quale saranno memorizzate le informazioni presenti all'interno del file di sessione salvato, in precedenza, sulla macchina server;
- utilizzo dell'id di sessione negli URL delle pagine: volendo evitare l'utilizzo del cookie di sessione, il linguaggio PHP offre la possibilità di inserire, sugli URL delle pagine web fornite, l'id di sessione, recuperabile mediante funzioni predefinite.

Si è optato per la prima scelta, sebbene sia stato l'unico caso di utilizzo di cookie all'interno del sistema, principalmente per motivi legati all'insicurezza offerta dal primo metodo, che mostra, in maniera evidente, l'id della sessione utente nell'URL delle pagine web aperte dal client; ad ogni modo, nonostante la memorizzazione del cookie non garantisca l'inaccessibilità di tale informazione, si è preferito offrire una metodologia più sicura da affiancare ai controlli eseguiti sul server e sul database, specificati nel dettaglio nel capitolo 8) Descrizione della Base di Dati. Nello specifico, il principale controllo, legato all'autenticazione dell'utente, avviene confrontando l'id di sessione, fornito dal cookie, con l'id di sessione memorizzato nella base di dati, nell'opportuna tabella login, in modo da verificare se il client che possiede tale cookie sia effettivamente lo stesso dell'utente che ha eseguito il login. Il controllo descritto richiede un carico computazionale maggiore, in quanto si deve eseguire una query sulla base di dati, ma le esigenze di sicurezza lo giustificano appieno.
La sessione ha termine una volta effettuato il logout, oppure dopo un periodo di inutilizzo del sistema pari ad un ora.
In linea di principio, sarebbe notevolmente utile fare uso dei cookies, dal momento che essi vengono salvati permanentemente sul computer client: l'aspetto negativo, che è sufficiente ad evitare il loro utilizzo all'interno di V.E.R.O.N.I.C.A., è che essi portano in dote problematiche legate alla sicurezza, essenzialmente perché i dati all'interno di essi vengono salvati in chiaro, inoltre è possibile modificare i valori all'interno dei cookies stessi. Per queste ragioni, l'unico cookie utilizzato è riferito alla sessione dell'utente e viene opportunamente eliminato al termine della sessione.

10.5. Permessi

La gestione dei permessi di accesso, garantiti agli utenti, nelle differenti pagine web che compongono il sito non è semplice. Ricordiamo le tipologie di utenti che hanno accesso al servizio:

- Utente dislessico;
- Medico;
- Amministratore.

Ogni pagina web del sistema V.E.R.O.N.I.C.A. garantisce una funzionalità indirizzata ad un determinato profilo utente. La gestione di tali pagine comporta un rigido controllo sulla loro accessibilità da parte degli utenti. Nello specifico, è fondamentale garantire l'accesso delle

varie tipologie di utenti ai servizi effettivamente progettati per essi, ed impedire loro l'accesso se non ne possiedono i permessi.
Al momento in cui si scrive, per gestire i vari permessi, sono stati necessari i seguenti controlli:

- l'utente che accede è, o meno, un amministratore;
- l'utente che accede è o non è medico;
- l'utente che accede è o non è un dislessico;
- il documento a cui si accede è privato o pubblico;
- il documento a cui si accede è stato approvato da un amministratore (chiaramente, questo permesso ha senso solamente se il documento è pubblico);
- il documento a cui si accede appartiene all'utente che accede.

Ovviamente, in base alla funzionalità a cui si vuole accedere, si sceglierà il sottoinsieme di controlli necessario affinché solamente chi ha diritto acceda al servizio.
Un ulteriore controllo, da effettuare su differenti pagine, è il valore delle variabile passate mediante metodo GET. Nonostante la bassa sicurezza offerta da tale metodo di invio dei dati, fornito dal PHP, essa viene, comunque, ampiamente utilizzata in gran parte dei siti web (V.E.R.O.N.I.C.A. incluso) per differenti motivi:

- permette di inviare dati ad altre pagine PHP, come fossero dei semplici collegamenti ipertestuali, senza dover utilizzare un form per ogni scelta, con bottone di invio submit, come impone il metodo POST;
- inserendo gli opportuni controlli sulle variabili ricevute nella pagina PHP, non ci sono pericoli di accesso a zone riservate del sito.

Più nel dettaglio, le variabili inviate mediante metodo GET vengono passate alla pagina PHP accedendo, a titolo di esempio, al seguente link come un normale collegamento ipertestuale: pagina.PHP?variabile=valore.
Nell'esempio sopra citato, a "pagina.PHP" viene inviata "variabile", a cui è assegnato "valore". Dal momento che questo indirizzo compare in chiaro nella barra degli indirizzi di ogni browser, un attaccante potrebbe modificare a piacimento sia il nome della variabile passata, sia il valore ad essa associato. Il problema, come già accennato in precedenza, si risolve inserendo gli opportuni controlli, nello specifico, si deve verificare che l'utente abbia i permessi necessari per accedere alla pagina che riceve i determinati valori inviati mediante metodo GET.

11. Sicurezza

Il sistema V.E.R.O.N.I.C.A. dedica particolare attenzione alla sicurezza.
Il termine sicurezza indica l'insieme di provvedimenti adottati per proteggere utenti, informazioni e servizi inerenti il sistema V.E.R.O.N.I.C.A.
Vengono di seguito trattate le motivazioni che hanno spinto verso lo sviluppo di tale caratteristica, descrivendo successivamente le decisioni adottate ed il lavoro svolto per aumentare la sicurezza del sistema nel suo complesso.

11.1. Contesto ed Obiettivi

La sicurezza è una materia estremamente vasta, comprende conoscenze relative anche a tematiche che tradizionalmente non appartengono all'informatica. Per semplificarne l'analisi, viene ora tralasciata la suddivisione per tematiche e si osserva la materia categorizzando le azioni orientate alla sicurezza in tecniche appartenenti a: prevenzione, rilevazione, azione.
La maggior parte degli sforzi dedicati alla sicurezza nel progetto V.E.R.O.N.I.C.A. riguardano la crittografia, che si inserisce nell'ambito della prevenzione.
Tuttavia, come illustrato in seguito, sono state intraprese numerose azioni mirate alla rilevazione delle minacce verso il sistema e conseguenti azioni per la risoluzione delle problematiche.
La sicurezza è una tematica spesso trascurata, oggi e in passato.
Il processo di astrazione, la cui importanza è indiscutibile, ha causato parallelamente alla diffusione dell'informatica all'interno della nostra quotidianità, una mentalità da parte dell'utilizzatore finale rivolta ad un forte interesse per il reale utilizzo finale del sistema, astraendo (e quindi trascurando), spesso e volentieri, tematiche interne quali la sicurezza dei sistemi informatici e in generale del calcolatore.
Se da parte dell'utente finale non vi è una forte richiesta in termini di sicurezza, certo non la si può riconoscere nelle azioni intraprese da parte delle industrie del settore.
Il mercato dell'informatica è in crescita continua da decenni, e le industrie del settore, in forte competizione tra loro, sono lanciate in un processo di continua produzione orientato alla soddisfazione delle esigenze finali, generando quindi un vortice che porta a una corsa continua orientata a migliorare le prestazioni dei sistemi, quasi mai la sicurezza se non per particolari settori in cui è indispensabile.
Per la maggior parte dei settori ingegneristici, la sicurezza è sempre in primo luogo, se si scopre un problema di sicurezza si compie un passo indietro e si rivede l'intero progetto.
Tuttavia, quando i computer hanno cominciato a dare problemi sulla sicurezza, la corsa delle industrie orientata alle prestazioni è continuata e nessun passo indietro è avvenuto, creando una tragedia sempre più sproporzionata nel campo della sicurezza.
Il comportamento adottato è dovuto al fatto che, nella realizzazione di un progetto software, capita sovente di avere obiettivi in contrasto fra loro e la sicurezza è il caso più evidente di tale fenomeno: utilizzare algoritmi mirati a migliorare la sicurezza del sistema incide sulle prestazioni in termini di tempi di risposta ed efficienza. La perdita di prestazioni a favore di una maggiore sicurezza avrebbe significato per molte aziende una conseguente perdita di competitività.
Il sistema V.E.R.O.N.I.C.A. cerca di raggiungere il giusto equilibrio tra sicurezza ed efficienza, in quanto il sistema necessita sia di buoni tempi di risposta sia di una protezione ottimale. L'attività di sicurezza sviluppata per il sistema V.E.R.O.N.I.C.A. ha quindi come obiettivi primari:

1. tutela delle informazioni trattate nel sistema;

2. tutela degli utenti iscritti al sistema;
3. tutela dei servizi forniti dal sistema.

11.1.1. Informazioni trattate nel sistema

Il sistema V.E.R.O.N.I.C.A. gestisce una modesta quantità di dati, comprese informazioni personali e sensibili, quali nome utente e password, dati anagrafici, dati relativi ai test (con e senza valenza medica) effettuati sulla dislessia ed altre informazioni riservate.
Obiettivo primario della sicurezza è tutelare la segretezza di tutte le informazioni del sistema, assicurando che gli utenti iscritti possano accedere solo alle proprie informazioni e nessun utente non iscritto possa accedere ai dati degli utenti registrati al sistema.
In particolar modo, la gestione della possibilità da parte delle diverse categorie di utenti di accedere ai dati è stata sviluppata in accordo con le normative vigenti, per maggiori informazioni è possibile consultare il capitolo Aspetti Legali.

11.1.2. Utenti iscritti al sistema

Il sistema V.E.R.O.N.I.C.A. individua tre differenti categorie di utenti:

- amministratori del sistema;
- personale medico;
- utenti dislessici.

Ciascuna categoria viene tutelata nello svolgimento del proprio lavoro.
Una prima misura di tutela consiste nell'evitare un sovraccarico del sistema dovuto ad eccessive richieste da parte di utenti curiosi di conoscere il sistema ma non registrati, quindi sui quali non si ha nessuna garanzia del regolare e corretto utilizzo del sistema.
Il problema diventa particolarmente evidente in quanto la dislessia è molto diffusa e la ricerca di sistemi virtuali che possano offrire un supporto alle persone dislessiche è in costante aumento. Come misura preventiva alla problematica descritta, la sicurezza si propone di garantire l'accesso al sistema V.E.R.O.N.I.C.A. solo da parte di utenti regolarmente registrati, previa autorizzazione da parte dell'ente gestore del servizio e verifica di test medici da parte degli organi competenti.
La sicurezza mira ad evitare l'intromissione di attaccanti intenzionati (per motivazioni di qualsiasi natura) ad introdursi nei moduli riservati di V.E.R.O.N.I.C.A.
Per quanto l'ipotesi possa essere giudicata remota, (il sistema non tratta dati bancari o altre possibili fonti di interesse da parte dei comuni attaccanti), questa non è una valida motivazione per ignorare le problematiche di sicurezza. Inoltre il sistema potrebbe ampliare le proprie prospettive e categorie di informazioni (o utenti e servizi) utilizzate in un periodo successivo: è necessario pertanto garantire un buon livello di sicurezza già in fase di progettazione.
V.E.R.O.N.I.C.A. è pensato principalmente (ma non solo) per categorie di utenti in una fascia di età tra i sette e i diciotto anni, appunto persone in fase di sviluppo.
La sicurezza intende proteggere gli utenti, garantire che i contenuti ai quali accedono siano rigorosamente controllati dagli amministratori e dal personale medico di V.E.R.O.N.I.C.A. e ogni utente con il quale si interagisce sia registrato e rintracciabile.

11.1.3. Servizi forniti dal sistema

Gli utenti possono interagire col sistema usufruendo di diversi servizi, i principali sono l'accesso alla propria area riservata, la possibilità di effettuare un test di prova (senza alcuna valenza medica) sulla dislessia, inserire del testo da sottoporre al sintetizzatore vocale, caricare libri nel sistema e pubblicarli a favore di tutti gli utenti registrati.

La sicurezza intende sorvegliare i diversi servizi, impedendo malfunzionamenti dovuti ad azioni (dolose o colpose) di altri utenti, al fine di garantire il corretto funzionamento del sistema.

11.2. Possibilità di Attacco

Un sistema informatico può avere svariati problemi che possono causare la momentanea o la permanente interruzione del servizio all'utente. In un sistema web-based come V.E.R.O.N.I.C.A. si può ad esempio pensare ad una temporanea congestione della rete, un problema hardware improvviso legato alla macchina server, o altro ancora.
L'attaccante di un sistema informatico tuttavia rimane l'aspetto più importante e più complesso da considerare quando si parla di sicurezza. A differenza dei fenomeni precedentemente descritti, (che possono comunque creare problemi all'interno del sistema), l'attaccante possiede due importanti caratteristiche che lo contraddistinguono:

1. Continuare nell'intento ostile finché non si persegue lo scopo;
2. Ricercare nuove metodologie di attacco, senza mantenere un comportamento prevedibile.

Per analogia, se un fenomeno naturale come la pioggia avesse tale caratteristica, sarebbe una pioggia che intenzionalmente continua a cadere finché non raggiunge lo scopo di far straripare il fiume e cadrebbe sempre in maniera diversa senza nessuna possibilità di esser studiata e prevista da parte degli esperti meteorologi.
Il sistema V.E.R.O.N.I.C.A., come detto in precedenza, non dovrebbe attirare grande attenzione da parte di un attaccante comune: non ha scopo di lucro, non gestisce soldi in alcuna maniera, non porta avanti alcun tipo di ideologia che possa provocare singoli o gruppi di persone.
Per una migliore riuscita, il lavoro sviluppato sulla sicurezza, ignora quali possano essere le cause di attacco o i profili psicologici e tecnici dell'attaccante, e focalizza l'attenzione sulle possibili modalità di attacco, chiunque sia ad attaccare il sistema.
Di seguito vengono illustrate, raggruppate per tipologia le possibili modalità di attacco:

- azioni offensive legate all'intromissione nel canale di comunicazione;
- azioni offensive legate al sistema crittografico utilizzato;
- azioni offensive legate a debolezze nei restanti moduli del sistema V.E.R.O.N.I.C.A.

1.1	Capire il significato dei messaggi intercettati
1.2	Cancellare i messaggi senza che possano mai arrivare al destinatario
1.3	Conservare i messaggi e ritrasmetterli sulla linea alterati
1.4	Conservare i messaggi e ritrasmetterli al destinatario in ritardo
1.5	Conservare i messaggi e ritrasmetterli al destinatario in ordine diverso dall'originale
1.6	Inviare dei messaggi al destinatario fingendosi il reale mittente

Tabella 11.1: Azioni offensive legate all'intromissione nel canale di comunicazione

Nel coordinamento della sicurezza di V.E.R.O.N.I.C.A. si provvede contro ciascuna delle possibilità ora presentate, principalmente attraverso la realizzazione di un canale di comunicazione sicuro che, si basa sull'utilizzo di un sistema crittografico. Conseguentemente, vengono quindi elencate le possibili linee d'attacco verso un sistema crittografico.

2.1	Ciphertext-Only	L'attaccante cerca di rompere la codifica conoscendo esclusivamente il testo cifrato.
2.2	Known Plaintext	L'attaccante cerca di rompere la codifica conoscendo anche il testo in chiaro, ed effettuando dei confronti per ricavarne la chiave utilizzata.
2.3	Chosen Plaintext	L'attaccante sceglie un particolare testo in chiaro da proporre al sistema per la codifica, e procede come al punto precedente.
2.4	Chosen Ciphertext (and Plaintext)	L'attaccante è in grado di ottenere anche un testo in chiaro a partire dal corrispondente cifrato e cerca di ottenere la chiave come al punto precedente.
2.5	Distinguishing Attacks	L'attaccante non ha più come obiettivo ricercare la chiave utilizzata ma capire il significato di un messaggio o parte di esso.
2.6	Birthday	L'attaccante utilizza il paradosso del compleanno per studiare possibili collisioni tra i messaggi scambiati.
2.7	Meet in the Middle	L'attaccante lavora come al punto precedente, ma utilizzando una tabella inizializzata con ulteriori valori da poter confrontare

Tabella 11.2: Azioni offensive legate al sistema crittografico utilizzato

Gli algoritmi di cifratura e autenticazione utilizzati nella realizzazione del canale sicuro, sono in grado di resistere agli attacchi elencati. Poiché è estremamente difficile attaccare il sistema crittografico utilizzato, l'attaccante medio si concentra sulla ricerca di falle all'interno dei restanti moduli del sistema, attraverso le tipologie di attacco di seguito descritte.

3.1	Remote file inclusion
3.2	SQL Injection
3.3	Debolezze dei linguaggi utilizzati
3.4	Debolezze dei moduli integrati nel sistema

Tabella 11.3: Azioni offensive legate a debolezze nei restanti moduli del sistema V.E.R.O.N.I.C.A

Le azioni di difesa mirano a contrastare i problemi ora descritti.

11.3. Modalità di azione

Sulla base degli obiettivi definiti al paragrafo Contesto ed Obiettivi, e delle modalità di attacco descritte al paragrafo Attaccante e Modalità di Attacco, vengono di seguito illustrate le modalità di azione adottate in supporto alla sicurezza del sistema V.E.R.O.N.I.C.A.

Applicazione di scelte progettuali orientate alla sicurezza	Fase di Progettazione
Realizzazione di un canale di comunicazione sicuro	Fase di Implementazione
Meccanismi di Difesa	Fase di Implementazione
Attacco al sistema attivo	Fase di Manutenzione

Tabella 11.4: Piano di azione per la sicurezza nel progetto V.E.R.O.N.I.C.A.

Le Scelte di Progettazione consistono in una serie di principi adottati, in fase di progettazione, in relazione agli obiettivi stabiliti, ovvero la salvaguardia di informazioni, utenti e servizi del sistema V.E.R.O.N.I.C.A.
L'aspetto più importante per la sicurezza è costituito dalla realizzazione, in fase di implementazione del sistema, di un Canale di Comunicazione Sicuro tra client e server, tramite un sistema di cifratura e autenticazione applicato ai dati.
Tale misura non è sufficiente a garantire un buon livello di sicurezza, in quanto è necessario garantire la sicurezza dell'intero sistema nei minimi aspetti.
Infatti, come dimostrato dalla proprietà dell'anello debole, "il massimo livello di sicurezza di un sistema è pari alla sicurezza del suo componente più debole". Tra i vari moduli, la crittografia (caratterizzata da una complessa matematica), è raramente l'anello debole, anzi è quasi sempre la parte più forte. Ne consegue che dotare V.E.R.O.N.I.C.A. di un modulo crittografico è un lavoro insufficiente se non si dedica una certa attenzione alle caratteristiche delle altre componenti, intesi come moduli creati ex-novo e moduli pronti integrati in fase di sviluppo.
Per tali motivazioni sono state apportate, in fase di implementazione, numerose azioni orientate a rafforzare la Difesa del Sistema.

11.4. Scelte Progettuali Orientate alla Sicurezza

La sicurezza è un'attività estremamente complessa, comprende conoscenze che variano su una infinità di aree tematiche e possono variare radicalmente nel tempo.
Un sistema infatti è progettato per durare anche decenni, quindi un attaccante che tenta di introdursi in un sistema più vecchio può avere a disposizione una rilevante quantità di informazioni, conoscenze e tecnologie non disponibili a chi ha progettato il sistema.
Con tutte le difficoltà citate risulta essere estremamente difficile incrementare il livello di sicurezza di un sistema esistente: per questo motivo la maggior parte degli sforzi per rendere più sicuro il sistema V.E.R.O.N.I.C.A. si concentra su una fase di ricerca iniziale che procede la progettazione del sistema, cercando di realizzare un sistema il più sicuro possibile da subito.
Le scelte progettuali sono di seguito presentate, suddivise in base al principale obiettivo che soddisfano tra i principi definiti nel paragrafo contesto ed obiettivi.

1. Scelte progettuali orientate alla tutela delle informazioni trattate nel sistema:
    - realizzazione di un canale di comunicazione sicuro tra client e server tramite un meccanismo di cifratura di tutti i dati che transitano nel canale;
    - realizzazione di un canale di comunicazione sicuro tra client e server tramite un meccanismo di autenticazione di tutti i dati che transitano nel canale;
    - requisiti minimi legati alla gestione della sessione dell'utente.
2. Scelte progettuali orientate alla tutela degli utenti iscritti al sistema:

- possibilità di registrazione al sistema negata all'utente e delegata esclusivamente all'ente gestore del sistema;
- creazione di un sistema di permessi per regolare l'accesso dei medici ai dati dei dislessici nel rispetto delle normative vigenti;
- requisiti minimi legati alle password utilizzate dall'utente al fine di utilizzare password con un buon livello di sicurezza.

3. Scelte progettuali orientate alla tutela dei servizi forniti dal sistema:
   - ancora una volta intervengono i meccanismi di cifratura e autenticazione del canale sicuro, con l'obiettivo di garantire che i file audio scaricati e ascoltati siano effettivamente messaggi provenienti dal server e non messaggi inseriti da terzi;
   - processo di approvazione dei dati resi pubblici dagli utenti da parte dell'amministratore di sistema.

11.5. Realizzazione di un Canale di Comunicazione Sicuro

La maggior parte degli sforzi nel campo della sicurezza per il progetto V.E.R.O.N.I.C.A. si concentrano sulla realizzazione di un canale di comunicazione sicuro tra client e server, che permetta lo scambio di messaggi in forma cifrata.
L'obiettivo è impedire che chiunque si metta in ascolto sul canale possa ostacolare la comunicazione tra le due parti in questione, attraverso una qualsiasi delle possibili azioni definite al paragrafo Attaccante e possibilità di attacco.
Il lavoro dell'attaccante si può riassumere in due principali possibilità:

1. spiare la comunicazione e quindi capirne il significato;
2. fingersi una delle due parti della comunicazione per poter inviare false informazioni.

Il primo problema viene risolto attraverso un processo di cifratura dei dati, e viene descritto nel paragrafo Cifratura. Il secondo problema, viene invece contrastato attraverso un meccanismo di autenticazione di ogni messaggio scambiato, che permette al destinatario di capire con certezza chi sia il reale mittente del messaggio: il processo è descritto nel paragrafo Autenticazione.
Ogni volta che un messaggio deve essere spedito, prima dell'invio viene sottoposto sia alla funzione di autenticazione che alla funzione di cifratura.
Solo una volta compiute tali operazioni il messaggio può viaggiare attraverso la rete, finché non giunge al destinatario che provvede a decifrare l'informazione e verificarne il mittente tramite il meccanismo di autenticazione: se l'autenticazione fallisce il messaggio viene scartato.
Vengono di seguito espressi i dettagli del canale realizzato: partendo da un'accurata analisi dei principi sulla crittografia, si discute successivamente delle modalità di cifratura, autenticazione e gestione delle chiavi, terminando con le personali conclusioni sul sistema realizzato.

11.5.1. Principi generali

La solidità di un sistema crittografico si basa principalmente sulla correttezza degli algoritmi utilizzati e sulla robustezza delle chiavi applicate a tali algoritmi.
Il primo istinto nella crittografia, potrebbe essere creare un algoritmo nuovo e mantenerlo segreto, con l'obiettivo di rendere nulla la potenzialità offensiva dell'attaccante anche se quest'ultimo venisse in possesso della chiave, in quanto non saprebbe come poterla utilizzare.
L'approccio è assolutamente sbagliato, e porta al risultato opposto a quanto sperato, infatti come enunciato dal principio di Kerckhoffs: "La sicurezza del sistema crittografico deve

dipendere esclusivamente dalla segretezza delle chiavi e non dalla segretezza dell'algoritmo utilizzato".
Le motivazioni sono molto importanti.

- In primo luogo è estremamente difficile creare un sistema di codifica (corretto). Se il sistema creato viene reso pubblico, eventuali problemi possono essere trovati dall'intera comunità scientifica e risolti; diversamente, al sistema nascosto può lavorarci esclusivamente l'attaccante, che ricerca sull'algoritmo una falla da sfruttare.
- È molto più complesso sostituire il sistema di codifica utilizzato piuttosto che modificarne esclusivamente la chiave. Spesso gli algoritmi vengono realizzati a livello hardware, il che rende il processo di sostituzione ancora più complesso.

Gli algoritmi di cifratura hanno in ogni caso una larga diffusione quindi, per il modello della paranoia, sono insicuri: l'attaccante potrebbe conoscere l'algoritmo, bisogna concentrare i propri sforza sulla segretezza della chiave.
Vengono di seguito descritti gli algoritmi utilizzati, sia per la cifratura che per l'autenticazione, in seguito viene presentato il modello di gestione delle chiavi sviluppato.
Nelle Considerazioni Inerenti il Canale di comunicazione si tratta dettagliatamente la tematica della sicurezza degli algoritmi scelti, delle chiavi utilizzate e dell'intero canale di comunicazione.

11.5.2. Cifratura

Definizione
Viene ora descritto il primo componente del canale sicuro: il cifrario a blocchi.
Un cifrario a blocchi consiste in una funzione per la codifica di blocchi di dati, la cui dimensione massima trattabile è fissata per l'algoritmo di cifratura utilizzato.
La funzione si occupa di codificare il blocco di testo in chiaro in un blocco cifrato, dove anche il blocco cifrato è caratterizzato da una dimensione fissa, definita come dimensione del cifrario.
Il processo di cifratura è una operazione invertibile, ovvero esiste la funzione inversa che a partire dal blocco di dati codificato effettua la decodifica rendendo il blocco di dati originale.
Ciò permette a chi riceve il messaggio cifrato di ricavarne il significato iniziale.

Fondamenti Teorici
Anche se il cifrario a blocchi determina in maniera univoca per ogni singolo input quale debba essere il corrispondente output cifrato, il cifrario a blocchi ideale deve dare l'impressione di una mappatura del tutto casuale dei dati, in modo che l'attaccante non possa trarne informazioni e capire come, da un blocco cifrato, si possa risalire al corrispondente blocco in chiaro.
Il tutto segue il principio di Shannon di confusione e diffusione: dove la confusione indica la complessità nel relazionare l'output alla chiave, mentre la diffusione è la proprietà che garantisce una distribuzione dei simboli nel blocco di output tale da evitare correlazioni statistiche.
Il principio di Shannon influenza fortemente la struttura del cifrario a blocchi: vi sono due principali tipologie di strutture per cifrari che si sono affermate nell'ultimo secolo:

- rete di Feistel;
- rete a sostituzione e permutazione;

Per quanto le reti di Feistel garantiscano una forte similitudine tra l'algoritmo di codifica e l'algoritmo di decodifica (semplificando fortemente la realizzazione hardware del sistema), entrambe le tipologie citate rispettano il principio di Shannon.

Un cifrario a blocchi può essere definito sicuro se non esiste un metodo non banale che possa permettere all'attaccante di decodificare i dati, (dove il metodo banale consiste nel provare tutte le possibili chiavi finché non si trova la chiave corretta).
Come già definito in precedenza però, è estremamente difficile creare un cifrario a blocchi sicuro. In accordo con il principio di Kerckhoffs, il cifrario utilizzato deve essere pubblico, così avviene anche per il progetto V.E.R.O.N.I.C.A.

Cifrario adottato: A.E.S.
L'algoritmo utilizzato è A.E.S. (Advanced Encryption Standard), il nuovo standard per la cifratura adottato dagli Stati Uniti in seguito a concorso per cifrari terminato nel novembre 2001.
Esistono svariati cifrari ritenuti abbastanza sicuri dalla comunità scientifica odierna, ad esempio altri concorrenti per A.E.S. quali Serpent o Twofish danno dei buoni risultati.
La scelta di A.E.S. è dovuta alla considerazione del buon compromesso dell'algoritmo tra il livello di sicurezza garantito e le buone prestazioni.

Caratteristiche tecniche
A.E.S. utilizza una dimensione massima del blocco di testo in input pari alla dimensione del blocco in output, ovvero 128 bit.
Riguardo la dimensione della chiave, l'algoritmo lascia una libera scelta tra: 128, 192, e 256 bit.
La scelta progettuale per il sistema V.E.R.O.N.I.C.A., giustificata in Funzionamento e Considerazioni Inerenti il Canale di Comunicazione, è mirata ai 256 bit.
A partire dalla chiave descritta, A.E.S. genera delle chiavi di sessione utilizzate nella computazione.

Funzionamento
La maggioranza dei cifrari a blocchi sono caratterizzati da una struttura ed operazioni simili: ciò che li distingue è il modo in cui tali operazioni sono applicate.
Similitudini tra gli algoritmi sono senza dubbio l'iteratività del processo di codifica (e decodifica), la suddivisione in più frammenti del blocco di informazioni ricevuto in input e alcune operazioni applicate ai frammenti di blocco considerati:

- funzione XOR: Operazione XOR tra la chiave di sessione e il frammento di blocco;
- funzione Sub: Sostituzione di un frammento con un altro specifico;
- funzione Mix: Mescolamento dei frammenti di dati ricevuti in input.

Terminata la panoramica su alcuni aspetti ricorrenti nella struttura dei cifrari a blocchi, viene presentato di seguito il funzionamento di A.E.S.
La struttura è basata su una rete a sostituzione e permutazione, di tipo iterativo.
Il numero di iterazioni è dipendente dalla dimensione della chiave: in caso di chiave a 128 bit si eseguono 10 iterazioni, 12 se la chiave ha dimensione 192 bit, 14 per una chiave a 256 bit.
Viene svolta un'operazione preliminare:

1. suddivisione del blocco di 128 bit in input in frammenti di dimensione fissa;

All'interno di ciascuna iterazione vengono eseguite le operazioni descritte:

1. funzione di XOR di ogni frammento con la chiave di sessione (calcolata in base alla chiave iniziale e fornita da un apposito gestore delle chiavi);
2. applicazione della funzione Sub, per la sostituzione del frammento ricevuto con un nuovo specifico frammento che possa eliminare eventuali correlazioni statistiche;
3. applicazione delle funzioni Mix, che ricevono in ingresso frammenti di blocchi e rendono in uscita li stessi frammenti ma in ordine differente.

Terminato il terzo punto, l'iterazione ricomincia con i nuovi frammenti generati.

Terminata l'ultima iterazione, i frammenti vengono resi in uscita in un unico blocco da 128 bit.

11.5.3. Modalità di cifratura

Definizione
Una modalità di cifratura a blocchi, è un algoritmo che affianca il lavoro del cifrario utilizzato, al fine di risolvere problemi legati alla gestione del flusso dei blocchi trattati.

Fondamenti Teorici
Il cifrario è una funzione che, come descritto in precedenza, associa ad ogni possibile blocco in chiaro di dimensione massima fissata, un blocco cifrato di dimensione statica predefinita.
Vi sono due problemi fondamentali ai quali il cifrario non risponde:

1. modalità di cifratura di un messaggio con dimensione maggiore rispetto all'input massimo accettato dalla funzione di cifratura;
2. gestione di un flusso di blocchi codificati.

Il primo problema è che, per quanto definito fino a questo momento, se si vuole cifrare un messaggio con dimensione maggiore rispetto all'input di A.E.S., bisogna creare un modulo che si occupi della suddivisione in più blocchi del messaggio stesso e ricomporre il messaggio cifrato.
Il destinatario deve possedere inoltre un modulo che scomponga il messaggio cifrato prima di sottoporre i singoli blocchi alla funzione di decodifica e recuperi il messaggio in chiaro ottenuto.
Il secondo punto indicato, costituisce un vero problema per la sicurezza.
Poiché A.E.S. è una funzione, ad ogni input corrisponde (a meno di variazione di chiave) il medesimo output: molte comunicazioni possiedono uno stesso inizio, o parti di messaggio ricorrenti, quindi verrebbero codificate sempre nella medesima maniera.
L'attaccante, può osservare il flusso dei blocchi sul canale di comunicazione e se, per motivi di qualsiasi natura, è a conoscenza di quali siano le parti in chiaro nella comunicazione che si ripetono, è in grado di studiare il funzionamento del sistema creato effettuando attacchi di tipo known plaintext: il flusso di blocchi non appare più come una sequenza di informazioni casuali.
Una buona modalità di cifratura a blocchi associata all'algoritmo di cifratura scelto, garantisce che, pur avendo stessi blocchi da codificare all'interno di uno o più messaggi scambiati, non vi siano in uscita blocchi cifrati identici a blocchi precedentemente computati.

Modalità di cifratura adottata: Counter
La modalità di cifratura utilizzata nel sistema V.E.R.O.N.I.C.A. è Counter, conosciuta anche come CTR. Basa il suo funzionamento su un contatore il cui requisito essenziale è che vari continuamente, senza che alcun numero venga riciclato.

Caratteristiche tecniche
L'algoritmo è in grado di gestire messaggi in input di qualsiasi dimensione, sia durante il processo di codifica che decodifica.

Funzionamento
CTR mode utilizza un valore chiamato NONCE, ovvero Number used ONCE.
Come indica il nome, si tratta di un numero utilizzato esclusivamente per un singolo messaggio, al fine di evitare che blocchi uguali appartenenti a messaggi differenti producano uno stesso blocco cifrato.
L'algoritmo si compone di due passi fondamentali:

1. Ki = E(K, Nonce || i) for i=1,...,k
2. Ci = Pi XOR Ki

In primo luogo si calcola la chiave i-esima da utilizzare nella computazione.
La chiave è calcolata codificando con l'algoritmo utilizzato (A.E.S. per il progetto V.E.R.O.N.I.C.A.), il "Nonce" concatenato al numero "i" che indica il numero del blocco nel messaggio attuale.
La codifica descritta avviene con l'utilizzo della chiave da 256 bit scelta per A.E.S.
Il secondo passo consiste nell'effettuare un'operazione di XOR tra la chiave i-esima appena calcolata e il blocco di dati che si intende codificare.

11.5.4. Funzione Hash

Definizione
La funzione Hash è uno strumento crittografico alla base del processo di autenticazione.
Consiste in una funzione simile al processo di cifratura: da un blocco di dati in input rende un blocco di dati non comprensibile a chi intercetta il messaggio.
La sostanziale differenza con la cifratura è che una funzione hash, pur restituendo un messaggio di dimensione fissata, può analizzare in input un messaggio di grandezza arbitraria, il che implica che si tratta di un processo non invertibile.

Fondamenti Teorici
Il processo di non invertibilità di una funzione hash trova fondamenti matematici nella legge del buco della piccionaia, in quanto i possibili input della funzione sono strettamente maggiori dei possibili messaggi in output: più possibili messaggi differenti tra loro possono essere codificati nel medesimo messaggio e, a partire dal messaggio non è possibile determinare a quale degli input corrisponde.
Le funzioni hash devono rispettare due importanti caratteristiche, precisamente, dati un messaggio "m" codificato attraverso la funzione hash "h(m)" nel corrispondente codificato "x":

1. a partire da x non deve essere possibile ricavare h(m) oppure m;
2. si deve evitare il processo di collisione, (nonostante l'ampio raggio di input).

Come nella cifratura, anche nelle funzioni hash l'output deve dare la sensazione di una sequenza di simboli casuali, dalla quale non è possibile ricavare alcuna informazione.
Per maggiori informazioni è possibile osservare quanto definito nei Fondamenti Teorici relativi al processo di cifratura dei dati.

Funzione Hash adottata: SHA 256
Nell'ambito del sistema V.E.R.O.N.I.C.A. si è scelto di utilizzare una funzione hash appartenente alla nota famiglia SHA. Tale famiglia di funzioni ha avuto come membro originale SHA1, funzione caratterizzata da un output a 160 bit.
Per le potenze di calcolo degli elaboratori attuali, 160 bit di chiave non garantiscono un buon livello di sicurezza, SHA1 inoltre possiede aspetti strutturali non convincenti, il che suscita dei dubbi anche sull'utilizzo delle funzioni SHA2 (in quanto caratterizzate dalla medesima struttura).
Lo standard SHA3 è ancora in fase di definizione, quindi la scelta è ricaduta su una funzione appartenente all'insieme SHA2, precisamente la funzione SHA256.
Per quanto come detto in precedenza SHA2 possieda una struttura secondo gli esperti non più adatta alle funzioni moderne, (SHA3 dovrà basarsi su una struttura diversa), SHA 256 offre un output di 256 bit, in perfetto accordo con i 256 bit utilizzati come chiave per A.E.S. nella realizzazione del sistema di cifratura.

Caratteristiche tecniche
Come tutte le funzioni hash l'input può avere una dimensione arbitraria, l'output è caratterizzato da una dimensione di 256 bit.

Funzionamento
Il funzionamento di SHA256 è di tipo iterativo, come la maggior parte delle funzioni hash attuali.
L'input viene suddiviso in parti di dimensione costante che, una volta elaborate, vengono ricomposte nell'output della funzione, definito digest.

11.5.5. Autenticazione

Definizione
L'autenticazione è il processo attraverso il quale il destinatario identifica se il mittente del messaggio appena ricevuto è realmente il mittente che ci si aspettava.
Gli algoritmi per il processo di autenticazione in crittografia vengono definiti MAC (Message Authentication Code).
MAC è una funzione che prende in ingresso una coppia di valori, ovvero una chiave segreta ed un messaggio da autenticare. In uscita viene reso un messaggio codificato da allegare a una coppia del messaggio in chiaro nel processo di invio dei dati.

Fondamenti Teorici
I MAC vengono realizzati mediante altre primitive crittografiche esposte nei paragrafi precedenti, quali algoritmi di cifratura oppure funzioni hash.
Devono essere sempre rispettati i requisiti di sicurezza definiti per le precedenti primitive di crittografia, (si vedano i Fondamenti Teorici per il processo di cifratura).
Tuttavia MAC deve possedere un nuovo particolare requisito, precisamente:

1. l'attaccante non deve mai potersi fingere il reale mittente del messaggio.

Se infatti lo scopo di utilizzare un algoritmo di cifratura è impedire all'attaccante di comprendere le informazioni scambiate, l'utilizzo di un MAC ha l'obiettivo di garantire al destinatario di non poter dubitare sull'identità del mittente dei messaggi ricevuti.

MAC adottato: HMAC
Per il progetto V.E.R.O.N.I.C.A. si è scelto di utilizzare la funzione di autenticazione HMAC.
HMAC si basa su l'utilizzo di una funzione hash.
La sua particolarità tuttavia è l'indipendenza da una specifica funzione: se per qualsiasi motivo la funzione hash utilizzata con HMAC dovesse risultare debole, la si sostituisce con una differente funzione, continuando a mantenere l'impianto HMAC.

Caratteristiche tecniche
Le caratteristiche tecniche sono legate alla funzione di hash utilizzata in HAMC, nel caso del progetto V.E.R.O.N.I.C.A. si tratta di SHA256.

Funzionamento
Viene descritto ora come lavora HMAC.
Dati:

- "K": Chiave utilizzata per la funzione hash;
- "m": Messaggio da autenticare;
- "h": Funzione hash scelta (SHA256);
- "a": Costante specifica;
- "b": Costante specifica;

- "f": Risultato della computazione.

Si calcola: f = h((K XOR a) || h((K Xor b) || m)).
Il risultato della computazione "f", deriva da una funzione hash che, come detto in precedenza, non è invertibile: il destinatario deve possedere altre informazioni per poter completare il processo di verifica dell'autenticazione. Il meccanismo è piuttosto semplice: il mittente calcola "f" e lo invia al destinatario assieme al messaggio originale "m". Il destinatario, a partire da "m", ricalcola (mediante la medesima funzione MAC con la stessa stessa chiave del mittente) l'autenticazione, quindi la confronta con quella ricevuta "f". Se il risultato ottenuto è lo stesso, allora il messaggio è stato realmente inviato dal mittente che ci si attendeva, diversamente il messaggio viene scartato perché qualcuno ha tentato di impersonare il mittente atteso.

11.5.6. Protocollo di Negoziazione delle Chiavi

Definizione
Il protocollo di negoziazione delle chiavi consente di calcolare le chiavi da utilizzare per gli algoritmi di cifratura e autenticazione presentati sino a questo momento.

Fondamenti Teorici
In prima approssimazione, si potrebbe pensare di fornire all'utente una coppia di chiavi (chiave di cifratura e chiave di autenticazione), assieme ai dati necessari al login, utilizzando un canale di comunicazione tradizionale considerato maggiormente sicuro.
Ad esempio, all'atto della registrazione, il gestore del sistema potrebbe inviare i dati citati tramite il sistema postale e l'utente potrebbe inserire i dati attraverso una sezione apposita del sito mostrata al primo ingresso, in modo tale che a ogni futuro accesso siano sempre presenti le chiavi iniziali e si possano riutilizzare.
Il metodo però non è corretto, perché le chiavi crittografiche non devono mai essere statiche, (fisse per tutta la durata del sistema): devono cambiare di frequente. Maggiore è il tempo per il quale si utilizza una chiave, maggiore è la probabilità che venga scoperta dall'attaccante: una chiave crittografica viene ritenuta sicura se l'intervallo di tempo con il quale viene sostituita è minore del tempo necessario all'attaccante per ottenerla.
Per questi motivi, si instaura un protocollo di negoziazione delle chiavi, che produca nuove chiavi ogni volta che il client richieda una nuova connessione al server. La particolarità del protocollo è che la negoziazione avviene tramite uno scambio (tra client e server) di parametri in chiaro utilizzando un canale non sicuro.
Nonostante il meccanismo sia visibile all'attaccante in ascolto sul canale, il processo utilizzato permette alle parti di ottenere una chiave sicura in un tempo molto minore del tempo necessario all'attaccante per calcolare la chiave tramite i dati intercettati.
La ragione che giustifica ciò è l'utilizzo di funzioni matematiche difficilmente invertibili: le parti nella comunicazione possono generare i parametri da scambiare sul canale insicuro in maniera molto rapida ma l'attaccante, per calcolare i valori che hanno originato i dati scambiati (ovvero applicare la funzione inversa), ha bisogno di tempi troppo elevati.

Protocollo Negoziazione Chiavi adottato: Evoluzione del protocollo Diffie-Hellman
Il protocollo di negoziazione sviluppato nel progetto V.E.R.O.N.I.C.A. è una variante del protocollo Diffie-Hellman, caratterizzato dallo scambio di parametri ottenuti tramite elevamento a potenza che costringono l'attaccante ad affrontare il problema matematico del logaritmo discreto. La chiave viene generata attraverso differenti passaggi, svolti alternativamente dal client e dal server tramite lavorazione degli stessi dati comuni.

Caratteristiche tecniche
Il protocollo è un semplice algoritmo che genera una terna di valori, meglio definiti in seguito, e utilizza i valori descritti per generare una chiave a 256 bit. Altre caratteristiche tecniche interessanti riguardano la dimensione dei valori della terna, tuttavia non sono indicabili a priori per il singolo protocollo in quanto vengono negoziate tra le due parti e sono quindi variabili.

Funzionamento
Il funzionamento del protocollo è rappresentato nella seguente Figura 11.1.
A e B sono le due parti coinvolte nella comunicazione, dal punto di vista del progetto si può affermare che A sia il client, mentre B rappresenti il server. Come accennato in precedenza, la negoziazione avviene tramite lo scambio di alcuni parametri in chiaro, utilizzati per formare successivamente la chiave.
I parametri di partenza sono:

- P;
- Q;
- G.

Si tratta di tre numeri (spesso in questo ambito sono utilizzati numeri primi): costituiscono la base per le successive operazioni aritmetiche quali l'elevamento a potenza dei valori che porteranno alla generazione della chiave. La robustezza del protocollo è legata alla dimensione dei numeri: più sono grandi, maggiore è il carico di lavoro per l'attaccante, (più lo schema è sicuro).
Per questo motivo, la prima azione del protocollo consiste in una proposta da parte del client sulla dimensione minima da utilizzare per il numero "P".
Tale valore, rappresentato da "S", viene inviato al server.
Il server sulla base dell'informazione in input sceglie la terna P, Q, G.
Attraverso i parametri scelti, si genera un numero "x" compreso tra 1 e Q-1. "x" è un parametro privato dal server, il client non ha modo di conoscerlo (e quindi allo stesso modo neppure l'attaccante). Il valore "x" è tuttavia parte del processo di generazione della chiave: a partire da esso il server ricava "X" tramite l'elevamento a potenza "G^x".
"X", accompagnato dalla terna di valori scelti e dall'autenticazione del server stesso, viene inviato al client.
Il client immagazzina i dati ricevuti.
Allo stesso modo del server, procede alla generazione di un valore privato compreso tra 1 e Q-1, il valore verrà indicato come "y". Ancora sulla stessa idea delle azioni compiute dal server, il client calcola "G^y", il cui risultato è immagazzinato in "Y" e inviato al server assieme alla propria autenticazione.
A questo punto la generazione della chiave è pronta: il client calcola la chiave "k" come risultato di "X^y", mentre il server ottiene lo stesso risultato calcolando "Y^x".
I due valori ottenuti sono esattamente gli stessi, in quanto:

- chiave client $= X \wedge y = (G \wedge x) \wedge y = k$;
- chiave server $= Y \wedge x = (G \wedge y) \wedge x = k$.

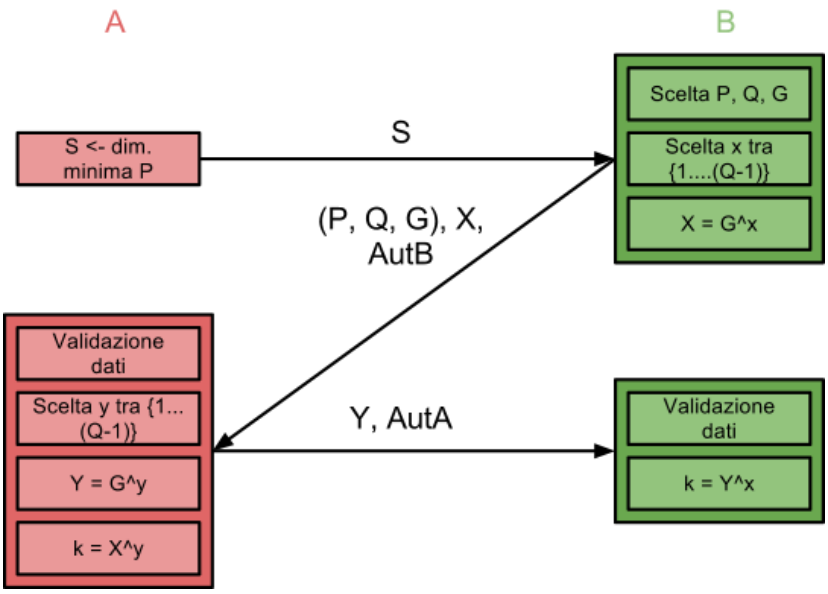

Figura 11.1: Protocollo di Negoziazione delle Chiavi

L'attaccante ha a disposizione in chiaro (nel canale) i valori di "X" ed "Y", ma non i valori privati "x" ed "y". Sulla base del principio di Kerckhoffs, può anche essere a conoscenza di come sia l'elevamento a potenza con G a creare i valori scambiati, quindi potrebbe pensare di applicare l'operazione inversa all'elevamento a potenza per ricavare "x" ed "y" a partire dai valori spiati e quindi calcolare la chiave.
Da un punto di vista computazionale però, il problema è troppo costoso, (ed il costo aumenta con la dimensione dei numeri): prima che l'attaccante riesca a scoprire la chiave, le due parti hanno concordato una chiave sicura e chiuso la comunicazione.
Le autenticazioni citate nel processo di negoziazione si basano su una chiave fornita dal gestore del sistema assieme ai dati di login e inserita al primo ingresso da parte dell'utilizzatore finale. Il procedimento sin qui illustrato viene adottato in V.E.R.O.N.I.C.A. per dare origine alla chiave del protocollo di cifratura. Successivamente si scambiano dei messaggi cifrati per ottenere una chiave per il protocollo di autenticazione. Client e Server hanno ottenuto le chiavi di cui avevano bisogno senza dover utilizzare vie traverse per la comunicazione e potendo cambiare le chiavi ad ogni nuova sessione di comunicazione in maniera trasparente all'utente.
A questo punto il canale cifrato è completo e pronto all'utilizzo.

11.5.7. Funzionamento e Considerazioni Inerenti il Canale di Comunicazione

Analizzate singolarmente tutte le componenti del canale di comunicazione sicuro, è possibile definire ora il funzionamento complessivo del sistema realizzato, in relazione alle possibilità dell'attaccante trattate nel paragrafo Attaccante e Modalità di Attacco.

11.5.7.1. Difesa dalle azioni offensive legate all'intromissione nel canale di comunicazione

Nella tabella 11.1 sono state presentate le diverse possibilità di un attaccante che si intromette nel canale di comunicazione spiando i messaggi scambiati tra le due parti.
Come detto in precedenza, il canale di comunicazione sicuro ha lo scopo di limitare tali attacchi.
Alcune azioni offensive sono gestite direttamente dalla rete di comunicazione:

- cancellare i messaggi senza che possano mai arrivare al destinatario;

- conservare i messaggi e ritrasmetterli al destinatario in ritardo;
- conservare i messaggi e ritrasmetterli al destinatario in ordine diverso dall'originale.

Infatti, al livello, rete vengono gestite la numerazione dei pacchetti, il tempo di vita, e la completa ricezione di messaggi inviati, fattori che ostacolano le azioni descritte. Mentre:

- conservare i messaggi e ritrasmetterli sulla linea alterati

potrebbe non essere di alcuna importanza, se il destinatario è in grado di riconoscere che il mittente del messaggio non è effettivamente chi ci si aspettava.
Si concentrano dunque gli sforzi sulle due azioni più pericolose che l'attaccante può effettuare:

- capire il significato dei messaggi intercettati;
- inviare dei messaggi al destinatario fingendosi il reale mittente.

La cifratura risolve il primo aspetto, l'autenticazione il secondo.
Ogni messaggio scambiato tra client e server viene sottoposto ad entrambi gli algoritmi prima dell'invio sulla rete (giudicata non affidabile in quanto non controllata dal sistema realizzato). Un'importante decisione da prendere è quale dei due obiettivi abbia la priorità e in conseguenza diretta quale delle due operazioni debba essere svolta per prima sui messaggi: cifrare il messaggio e autenticarlo oppure autenticare il messaggio e cifrarne il risultato. A parere dei maggiori esperti in sicurezza entrambe le modalità portano vantaggi e svantaggi ma comunque sono giudicate soluzioni sicure. Generalmente, è preferibile che il destinatario sia sempre in grado di identificare la fonte dei messaggi ricevuti, con una priorità maggiore rispetto alla riservatezza del messaggio stesso. Per questo motivo, all'interno del sistema V.E.R.O.N.I.C.A., il canale in primo luogo autentica i messaggi, successivamente protegge il processo di autenticazione attraverso la cifratura.
Di seguito è rappresentato lo schema di comunicazione per ogni singolo messaggio scambiato tra client e server, Figura 11.2.

1. Il procedimento inizia con la costruzione di un messaggio in chiaro "plaintext", che si vuole inviare, generato attraverso le operazioni svolte dal mittente (client o server).
2. Il plaintext, viene quindi autenticato. Come detto in precedenza, l'autenticazione viene svolta mediante un particolare MAC definito HMAC che, allo stato attuale del progetto, sfrutta la funzione hash SHA256, rendendo in uscita la stringa "a".
3. L'intera autenticazione viene protetta mediante cifratura. Poiché si vogliono poter gestire messaggi di dimensione variabile e si vuol mantenere il controllo dei flussi dei blocchi, si utilizza la modalità di cifratura CTR applicata all'algoritmo di cifratura scelto, ovvero AES. Il messaggio in ingresso è una stringa costituita dalla concatenazione del "plaintext" con il plaintext autenticato "a". In uscita dal procedimento si ha il "ciphertext".
4. Il "ciphertext" viaggia per la rete di comunicazione sino al destinatario.
5. Il destinatario riceve il "ciphertext". Come prima operazione effettua la decodifica sfruttando nuovamente la modalità di cifratura CTR con l'algoritmo di cifratura AES. Il risultato di tale operazione è nuovamente il "plaintext" concatenato ad "a", ovvero l'autenticazione del plaintext stesso.
6. Prima di poter fruire il plaintext bisogna accertarsi che il mittente sia effettivamente il mittente atteso e non un attaccante che invia falsi messaggi; per questo motivo, il destinatario applica al "plaintext" il processo di autenticazione HMAC basato su funzione hash SHA256 e ottiene una stringa definita " a' ".
7. Avviene quindi il confronto di "a" con " a' ": se le due stringhe sono uguali, l'autenticazione è riuscita, diversamente il messaggio viene scartato.

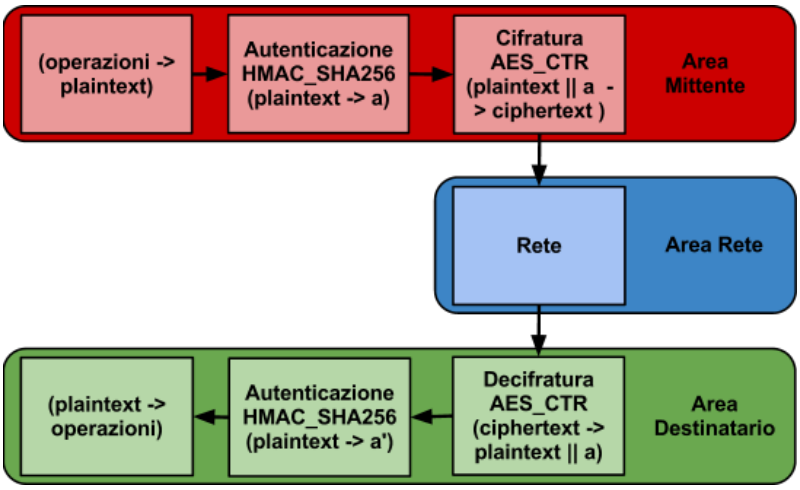

Immagine 11.2: Schema canale di comunicazione

11.5.7.2. Difesa dalle azioni offensive legate al sistema crittografico utilizzato

Come già definito, una volta adottato un sistema crittografico per rendere nulle le azioni di attacco descritte alla tabella 11.1, l'attaccante può procedere contro il modulo crittografico con gli attacchi descritti alla tabella 11.2. La sicurezza del modulo, è data dalla sicurezza (e quindi correttezza) degli algoritmi scelti e dalla sicurezza e gestione delle chiavi crittografiche.

Gli algoritmi scelti sono pubblici, in accordo con il principio di Kerckhoffs.

Attualmente non sono noti gravi problemi di sicurezza riguardo agli algoritmi scelti: per quanto si stiano rivoluzionando gli standard legati alle funzioni hash mediante la realizzazione del nuovo SHA3, (SHA2 lascia qualche perplessità) e gli attacchi basati su formule matematiche siano un campo in continuo sviluppo (specialmente verso lo studio dei cifrari), all'interno del progetto V.E.R.O.N.I.C.A. sono stati utilizzati ottimi algoritmi.

Di seguito, viene riportata una trattazione dei tipi di attacco mirati alle funzioni cifratura:

- Ciphertext-only;
- Known Plaintext;
- Chosen Plaintext;
- Chosen ciphertext and Plaintext.

In un attacco di tipo ciphertext-only, l'unica informazione a disposizione dell'attaccante è il testo cifrato, disponibile ad esempio nella rete di comunicazione tra mittente e destinatario.

L'attaccante tenta di ottenere la chiave di cifratura o il testo in chiaro, a partire esclusivamente da un'analisi del testo cifrato.

In cifrari classici quali il Cifrario di Cesare, è possibile senza troppa difficoltà realizzare questo tipo di attacco, basandosi su strumentazioni di vario genere, ad esempio l'analisi della frequenza dei caratteri, legata alla lingua utilizzata (escludendo ora la ricerca per tentativi della chiave).

Grazie alle caratteristiche di confusione e diffusione espresse dal teorema di Shannon, AES è in grado di resistere all'attacco descritto.

Known Plaintext è un attacco realizzato quando si conosce non solo il testo cifrato, ma anche il corrispondente testo in chiaro. La possibilità di realizzare un attacco di questo genere è molto elevata, in quanto esistono svariati contesti nei quali l'attaccante è a conoscenza di entrambi.

Ad esempio, si sa che in determinati messaggi vi sono delle parole note sempre in punti determinati, o ancora pezzi di testo ripetitivi.

A partire dal confronto del testo in chiaro con il rispettivo cifrato, si cerca di determinare al chiave crittografica utilizzata o altre informazioni importanti.
AES è in grado di resistere all'attacco.
Chosen Plaintext mantiene le caratteristiche descritte per gli attacchi precedenti, con in più la possibilità di scegliere quale sia il testo in chiaro da sottoporre all'algoritmo per ottenerne il corrispondente cifrato e studiare il meccanismo di cifratura.
Il tipo di attacco può essere realizzato inviando al sistema dei messaggi che in seguito verranno cifrati e possono essere intercettati: è stata una tecnica estremamente utilizzata dagli inglesi contro la macchina Enigma durante la seconda guerra mondiale.
AES non mostra problemi con gli attacchi Chosen Plaintext.
Chosen ciphertext and Plaintext è un attacco ancora più forte, dove l'attaccante può scegliere non solo il testo in chiaro da proporre al sistema ma anche un qualsiasi testo cifrato da cui poi ricava il corrispondente in chiaro.
Senza dubbio è l'attacco più forte analizzato sino a questo momento, ma non è comunque sufficiente a rompere il sistema di codifica AES.
Sino a questo momento sono stai analizzati attacchi il cui scopo principale è cercare di ricavare la chiave di cifratura utilizzata. Non essendo possibile, vista la robustezza degli algoritmi scelti, si cercano attacchi non mirati alla ricerca della chiave:

- Distinguishing Attacks

L'attacco intende cercare di ottenere alcune parti di messaggio in chiaro o il suo significato, indipendentemente dalla ricerca della chiave. Tuttavia, il sistema V.E.R.O.N.I.C.A. abbraccia il principio di Horton, ovvero "Autentica ciò che intendi, non solo ciò che dici".
Nel processo di autenticazione e cifratura, non vengono lasciate scoperte informazioni aggiuntive sul significato dei dati inviati, o altri campi contenenti informazioni su come vadano utilizzati i dati scambiati.
Possedendo (sia gli algoritmi che il procedimento con il quale si trattano i dati) un buon livello di sicurezza, ne deriva che la sicurezza del modulo crittografico è demandata alla sicurezza delle chiavi, determinata dalla segretezza della chiave e in particolar modo dalla sua dimensione.
Gli algoritmi utilizzati infatti richiedono l'utilizzo di particolari chiavi di cifratura, come descritto nel paragrafo Protocollo di Negoziazione delle Chiavi. Nell'ambito del progetto V.E.R.O.N.I.C.A. è stato scelto di utilizzare chiavi con una dimensione di 256 bit per ogni algoritmo coinvolto.
La scelta della dimensione della chiave mira a raggiungere un livello di sicurezza di 128 bit, in relazione alle limitazioni poste dai due ultimi tipi di attacco descritti in tabella 11.2:

- Birthday;
- Meet in the Middle.

Gli attacchi menzionati sono basati sul paradosso del compleanno e sfruttano le collisioni. Secondo il paradosso del compleanno, se una variabile può avere "n" valori diversi, dopo aver pescato casualmente "radice di n" variabili dello stesso tipo, vi è una probabilità pari a un mezzo di ottenere una coppia di valori uguali.
Il principio matematico prende il nome di paradosso del compleanno in quanto, essendo 365 i giorni in un anno, se vi sono "radice di 365" persone (circa 20), vi è una probabilità di un mezzo che almeno due persone abbiano il compleanno lo stesso giorno.
L'attacco birthday quindi, sfrutta il paradosso del compleanno per attendere che si verifichi una collisione, ovvero si abbia una coppia di valori identici da poter sfruttare sul sistema.
Poiché le funzioni hash mappano in uno stesso modo input diversi, sfruttando le proprietà statistiche descritte, si cerca un valore "x1" il cui hash sia identico al valore hash di "x2", dove "x2" è il messaggio sconosciuto. Verificata la collisione "x1" può essere sostituito ad

“x2” senza che il processo di autenticazione fallisca. L’attaccante ha inviato un falso messaggio al destinatario.
Il sistema è spesso adottato nell’ambito degli attacchi alle transazioni bancarie.
L’attacco Meet in the Middle, è basato sullo stesso principio ma, anzi che attendere l’arrivo di una collisione tra i vecchi valori utilizzati, l’attaccante prepara una tabella con dei valori computati in precedenza per accelerare il processo di confronto e la comparsa della collisione.
Gli attacchi birthday e meet in the middle quindi comportano che, se una chiave ha dimensione di “n” bit e quindi ha “2 elevato n” possibili valori, allora dopo “radice di 2 elevato n” volte vi è una probabilità pari a un mezzo di avere una collisione, ovvero dopo “2 elevato n-1” valori.Per questo motivo, una chiave di “n” bit, ha un reale livello di sicurezza di “2 elevato n-1” bit.
Nel caso delle chiavi utilizzate per il progetto V.E.R.O.N.I.C.A., si hanno chiavi a lunghezza di 256 bit che garantiscono un livello di sicurezza di 128 bit.

11.6. Meccanismi di Difesa

Nel presente capitolo l’obiettivo è prendere dei provvedimenti mirati che possano rafforzare la difesa del sistema realizzato sino a questo momento.
Nei paragrafi precedenti è stato sviluppato un canale di comunicazione sicuro che possa permettere lo scambio di messaggi cifrati tra client e server. Il canale di comunicazione permette principalmente di evitare all’attaccante di capire i messaggi scambiati o fingersi una delle due parti nella comunicazione inviando dei falsi messaggi.
L’attaccante a questo punto potrebbe attaccare il canale, ad esempio tentando di recuperare le chiavi dei meccanismi di cifratura, o in generale effettuando dei particolari attacchi come descritto nella Tabella 11.2. Gli algoritmi adottati nella realizzazione del canale sono tuttavia in grado di offrire un ottimo livello di sicurezza contro le eventualità citate.
L’unica soluzione rimasta all’attaccante, consiste nel ricercare delle altre debolezze presenti nel sistema, anche se non hanno necessariamente a che fare con la crittografia.
Le principali tipologie di attacco verso i sistemi odierni infatti, non sono mirate alla componente crittografica, in quanto si tratta di attacchi troppo complessi e di difficile riuscita: l’attaccante medio si concentra sul ricercare bug nell’implementazione del sistema che consentano di inserire comandi e false informazioni in modo da infiltrarsi all’interno e prendere il controllo.
La Tabella 11.3 descrive alcune tra le principali possibilità adottate in questo settore, e di seguito verranno analizzate.
L’obiettivo del lavoro è capire quanto gli attacchi presentati possano essere attuabili verso il sistema V.E.R.O.N.I.C.A. e quindi rafforzare il sistema di difesa.

11.6.1. Remote File Inclusion

Il Remote File Inclusion, è una tecnica attraverso la quale l'attaccante inserisce delle proprie pagine di codice all'interno dell'applicazione web che vuole attaccare. Nell'applicazione, si possono effettuare i danni più svariati, in una maniera difficilmente prevedibile a priori.
Le variabili GET e POST del PHP, possono essere utilizzate da parte di chi ha creato il sistema per l’inclusione di pagine web esterne all’applicazione stessa. L’attacco si genera nel momento in cui da parte dello sviluppatore non vi è stato alcun controllo sui possibili dati in input. L’attaccante infatti, modifica semplicemente il codice della pagina, variando il file da includere, in quanto è libero di inserire qualsiasi file senza che vi sia alcun controllo lato server che respinga il codice inviato.

Per difendersi dall'attacco descritto, è necessario semplicemente effettuare un controllo lato server sui file inclusi, ad esempio, se il file non appartiene ad un determinato dominio di file noti a chi ha creato l'applicazione allora viene respinto.
All'interno del progetto V.E.R.O.N.I.C.A. avviene un'attenta suddivisione del codice in diversi file e cartelle. Si evita la ripetizione di codice identico, inserendo in appositi file il codice usato più comunemente e richiamando tali file solo nelle dovute pagine.
Un esempio sono le porzioni di codice utilizzate per la gestione del layout, l'interfacciamento verso la base di dati o ancora la gestione delle sessioni.
L'inclusione dei file citati però, non avviene mediante GET e POST, ma attraverso l'istruzione "include", in quanto nel progetto non avviene mai l'inclusione di file esterni.
In questo modo, i file da includere sono dei parametri statici non modificabili e non legati ad eventuali dati in input, l'attaccante non può modificarli e non può attaccare il sistema nella maniera descritta. Si evita l'inclusione di codice esterno anche perhé si vuole rendere il progetto indipendente da piattaforme o siti esterni, il che ha creato grosse difficoltà di implementazione che saranno ampiamente ripagate in termini di sicurezza e affidabilità del sistema.

11.6.2. SQL Injection

SQL Injection è una tecnica che, come il Remote File Inclusion, precedentemente descritto, permette all'attaccante di inserire del codice maligno all'interno del sistema.
L'obiettivo questa volta è molto più chiaro, in quanto la tecnica è generalmente utilizzata in un form di login per autenticarsi ed entrare nell'account di un utente, pur non avendo le sue credenziali. E' naturale pensare che l'attaccante voglia acquisire i privilegi di amministratore in modo da maturare un pieno controllo sul sistema.
Il funzionamento dell'attacco è abbastanza semplice.
Si pensi ad esempio ad un sistema munito di autenticazione basato su due principali pagine: una che mostri all'utente il form con i campi "nome utente" e "password" ed una seconda pagina che riceva i dati del form e si occupi di verificare l'autenticazione.
L'attaccante inserisce nel form il nome dell'utente con il quale si vuole autenticare (anche se non conoscesse il nome potrà provare "admin", "root" o simili), nel campo password inserisce una stringa che rappresenti una disgiunzione inclusiva, quale "OR user = admin".
Ipotizzando che realmente esista l'utente "admin", quando i dati arrivano alla pagina lato server per la verifica, la disgiunzione (in assenza di controlli) farà si che il risultato dell'interrogazione sia sempre vero, e l'autenticazione venga superata con successo.
L'attacco è stato reso possibile da una totale assenza di controllo lato server sulla natura dei dati ricevuti dal form prima di sottoporre l'interrogazione alla base di dati.
Lo stesso problema si può verificare per qualsiasi form i cui risultati debbano essere utilizzati come parametri per interagire con il database, se non vi è un rigido controllo su quanto ricevuto dal client l'attaccante può cercare di influire sull'esito delle interrogazioni sottoposte al DBMS. La tecnica è inizialmente nata come attacco contro i sistemi basati sull'utilizzo di SQL, tuttavia è generalizzabile verso sistemi che utilizzino linguaggi diversi.
V.E.R.O.N.I.C.A. è munito di alcuni form per interagire con l'utente, alcuni esempi:

- form iniziale per il login;
- form per l'inserimento dati di un file che si vuol caricare;
- form a diposizione degli amministratori per operazioni di varia natura.

Tutti i form prevedono un rigido controllo sui dati.
Si ha un controllo iniziale lato client che consenta di ridurre il carico di lavoro per il server e allo stesso tempo aiutare l'utente nella compilazione del form segnalando prontamente gli errori effettuati (come campo vuoto, password troppo corta, etc.).

In secondo luogo si effettuano dei controlli lato server (che non è possibile delegare al client per motivi di sicurezza) che precedono l'interrogazione della base di dati, o più genericamente precedono l'utilizzo dei dati ricevuti, qualsiasi esso sia.

11.6.3. Panoramica sulla Sicurezza delle Tecnologie Coinvolte nel Sistema

Qualsiasi sistema virtuale è realizzato mediante precise tecnologie e ne utilizza svariate per mantenere operativa la propria attività.
All'interno del presente paragrafo, si cerca di effettuare un'analisi sui possibili problemi di sicurezza delle tecnologie alle quali il sistema V.E.R.O.N.I.C.A. è legato, in quanto i problemi di sicurezza citati si riflettono direttamente sulla sicurezza del sistema.
Nel capitolo Tecnologie utilizzate, è possibile osservare un elenco delle tecnologie coinvolte nella realizzazione del sistema, suddiviso per tecnologie lato client e lato server.
Dal punto di vista della sicurezza, è possibile categorizzare gli strumenti utilizzati secondo la seguente suddivisione:

- linguaggi (HTML, PHP, JavaScritp);
- browser;
- sistemi operativi.

I linguaggi coinvolti, possono avere problemi di sicurezza, legati ad esempio a particolari istruzioni o funzioni del linguaggio non sicure. Tra i linguaggi utilizzati nel progetto, il PHP offre un lungo elenco di istruzioni pericolose e, dopo accurate ricerche, è stato possibile individuare un insieme delle principali funzioni dannose e intervenire.
Le istruzioni e le funzioni considerate, forniscono dei risultati ottenibili tramite un numero infinito di diverse sequenze di istruzioni considerate non pericolose.
Il lavoro all'interno del progetto da questo punto di vista, è stato ricercare le eventuali istruzioni pericolose presenti e sostituirle con una sequenza di istruzioni non dannose.
E' essenziale capire che la fase di manutenzione del sistema svolge un ruolo chiave nella sicurezza. Se un domani vengono individuate dalla comunità scientifica nuove funzioni o istruzioni dannose, si deve procedere nuovamente all'aggiornamento del codice, finché il progetto V.E.R.O.N.I.C.A. rimane attivo.
I browser utilizzati dall'utente, costituiscono un modulo difficilmente controllabile da parte dei progettisti. Per assicurare una buona protezione è consigliabile utilizzare versioni aggiornate dei propri browser, ma attualmente non sono state imposte nel sistema delle limitazioni che blocchino l'utente qualora decida di utilizzare delle versioni del proprio browser non apprezzabili dal punto di vista della sicurezza agli occhi del team di sviluppo.
Il sistema operativo è un discorso simile ai browser web: anche qui non sono imposti dei controlli sulla tipologia o la versione utilizzata, con i dovuti rischi che ne derivano. Le possibilità di azione dei progettisti nei confronti del sistema operativo sono ancora più limitate, inoltre il progetto è un'applicazione web che vuole astrarre dai dettagli del sistema operativo (o del browser web) utilizzato, nell'ottica di un'accettazione globale che entra in contrasto con l'obiettivo di mantenere i massimi livelli di sicurezza.

11.6.4. Funzionamento e Considerazioni Inerenti i Meccanismi di Difesa

La maggior parte degli attacchi odierni, sfrutta uno scarso controllo dei dati in input.
Remote File Inclusion e SQL Injection sono esempi basati proprio su questo tipo di lacuna.
Come già detto, il sistema V.E.R.O.N.I.C.A. accompagna all'utilizzo del canale cifrato un forte controllo sulla natura dei dati, la Figura 11.3 mostra l'intero processo di scambio dati tra il client e il server, costantemente tutelato dagli interventi in ambito di sicurezza.

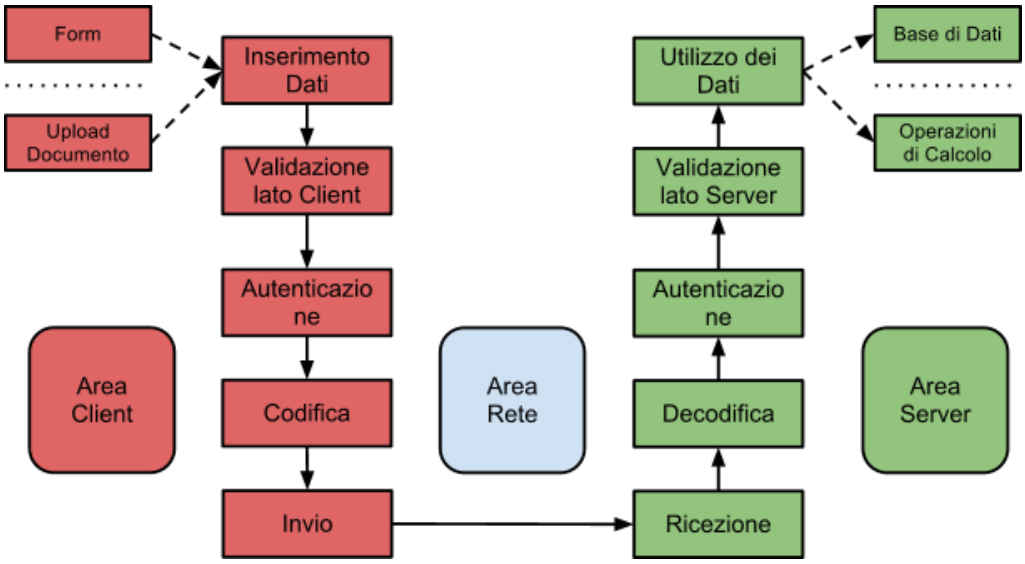

Immagine 11.3: Interventi della sicurezza nella comunicazione tra client e server

Dalla figura si evince che, qualsiasi sia la tipologia di dato inserito in input, attraversa un processo di validazione lato client e, una volta giunto al server, il dato viene sottoposto a maggiori controlli che possano evitare problemi al sistema. Solo dopo tali controlli il dato può essere utilizzato per interagire con la base di dati o effettuare altre operazioni lato server.
In generale, gli attacchi basati sullo scarso controllo dei dati in input rientrano nella tipologia dei Cross-Site Scripting e in ogni progetto web-based dovrebbero essere considerati e studiati con attenzione, in quanto si tratta di tipologie di attacco estremamente semplici da mettere in atto ma estremamente dannose per il sistema.
Nel progetto V.E.R.O.N.I.C.A., per ogni nuova funzionalità realizzata, viene analizzato ogni possibile punto di inserimento dati da parte del client e si procede a realizzare una struttura che guidi i dati sulla base dello schema di Figura 11.3.
Il lavoro descritto, costituisce il nucleo degli interventi nell'ambito della difesa del sistema, e deve essere rispettato per ogni nuova funzionalità futura implementata.
Al supporto della difesa è anche importante scegliere con attenzione i moduli utilizzati, ma il lavoro possibile su questo campo è ben poco da parte del team.
Il team ha ricercato dei moduli più sicuri possibile e, per ogni linguaggio, sistema operativo o browser, si è scelto di utilizzare la versione più aggiornata disponibile sul mercato. Il lavoro di aggiornamento non deve tuttavia fermarsi, costituisce una continua prevenzione da operare durante la fase di manutenzione del sistema V.E.R.O.N.I.C.A.

12. Aspetti Legali

12.1. Contesto ed Obiettivi

Il progetto V.E.R.O.N.I.C.A. affronta tematiche delicati dal punto di vista legale, le principali sono il diritto d'autore nella gestione dei libri caricati e la privacy per via del trattamento di dati sensibili degli utenti registrati al sistema.
Per fornire una visione completa degli aspetti legali e capire come vengono garantiti in ogni momento i diritti degli utenti, come mostrato nella Figura 12.1, viene presentato uno studio delle tematiche legali, a partire dal momento in cui l'utente decide di far parte del sistema sino a quando decide di porre termine al suo utilizzo.

12.2. Iter Burocratico per la registrazione a V.E.R.O.N.I.C.A.

La registrazione dell'utente dislessico è preceduta da una sua formale richiesta scritta di iscrizione, accompagnata dalla documentazione medica che ne attesti il grado di dislessia.
Il consenso al trattamento dei dati personali deve essere espresso dal singolo utente dislessico e non dal medico. Il consenso viene espresso a favore del proprio medico (o pediatra) di base.
Allo stesso modo si procede per la registrazione degli utenti medici ed amministratori: essi dovranno, di conseguenza, fornire il consenso al trattamento dei dati personali relativi alla loro mansione.
Si ricorda che i test resi disponibili (anche ai visitatori) sulla piattaforma, non hanno valenza medica, ma, semplicemente, vogliono dare un'idea all'utente sul problema.
Le precise modalità per la domanda di iscrizione, nonché l'insieme dei documenti da allegare, saranno precisati dall'ente gestore della piattaforma.

12.3. Registrazione al Sistema

Una volta portato a termine l'iter burocratico, i dati forniti dall'utente unitamente alla richiesta di iscrizione sono sottoposti agli amministratori di V.E.R.O.N.I.C.A.
E' loro onere, infatti, inserire tali dati nella fase di registrazione utente: sarà poi un modulo automatizzato del sistema ad inviare una mail all'utente con i dati di riepilogo ed i dati da inserire all'atto del login.
Va specificato che, al primo accesso effettuato dall'utente, verranno sottoposte alcune informative sul trattamento dei dati all'interno del sistema e sulle modalità di utilizzo dei servizi, al fine di garantire la massima efficienza del sistema.
Per quanto riguarda i dati relativi alla salute dell'individuo, è importante specificare che potranno essere trattati unicamente previa specifica informativa e acquisizione del consenso dell'interessato.
Dal momento che la finalità del trattamento non è di carattere sanitario o legata alla cura e terapia medica del paziente, ma bensì legata alla sua identificazione all'interno del sistema ed a semplici rilevazioni statistiche, il gestore del sistema V.E.R.O.N.I.C.A. dovrà indicare nell'informativa, in modo dettagliato, quali saranno le finalità del trattamento dei dati e le tipologie di dati che potranno essere trattati, nonché richiedere l'autorizzazione preventiva del Garante per la privacy, secondo quanto espresso dagli artt. 20 e 26 del Codice privacy.
L'informativa che deve essere fornita all'interessato per ottenerne il consenso al trattamento dei dati deve indicare le finalità del trattamento, i titolari del trattamento, le tipologie di dati trattati ed i soggetti che possono accedere ai dati.

L'informativa fornita deve essere corredata dalla possibilità di offrire il consenso al trattamento per evitare di generare confusione ai soggetti interessati sui due diversi adempimenti. Tale consenso deve essere acquisito specificatamente per le tipologie di dati preventivamente specificate e vale soltanto per determinate tipologie di soggetti abilitati all'accesso, nel caso del sistema V.E.R.O.N.I.C.A. unicamente per i medici che potranno visualizzare soltanto i dati dei loro pazienti, previo loro consenso.
All'interno dell'informativa, deve essere precisato che il consenso è sempre revocabile.

12.4. Utilizzo dei Servizi

L'utilizzo del sistema comporta la possibilità di pubblicare materiale, i cui diritti possono appartenere all'utente stesso, qualora condivida una propria nota, o a terzi, nel caso in cui condivida un libro scritto da un altro generico autore.
Generalmente, è ragionevole avvenga la condivisione di opere letterarie, appartenenti quindi alla seconda categoria descritta: dal punto di vista legale è importante discutere del diritto d'autore, al fine di tutelare gli autori delle opere pubblicate nel sito.

12.4.1. Tipologie di Opere Fruibili nel Sistema

Si distinguono, per le necessità del progetto V.E.R.O.N.I.C.A., tre diverse categorie di opere, a cui rispondono diverse modalità di azione:

- opere cadute in pubblico dominio;
- opere cadute in pubblico dominio ma contenenti note a margine di autori che possiedono ancora diritti nei confronti dell'opera;
- opere non cadute in pubblico dominio.

12.4.2. Possibilità di Acquisizione delle Opere da Inserire nel Sistema

In primo luogo è importante capire come il sistema possa entrare in possesso delle opere da mettere a disposizione degli utenti: non vi sono problemi per le opere cadute in pubblico dominio ma il diritto non obbliga in alcun modo gli autori delle opere ancora protette a rilasciare copie delle proprie opere a società o organizzazioni umanitarie.
Il modo più semplice e veloce per superare il problema è richiedere alle case editrici l'autorizzazione all'utilizzo dell'opera all'interno del sistema, spiegando come essa sia utilizzata senza fini di lucro e per un'opera a fin di bene.
La decisione finale, inoppugnabile, è demandata quindi alla casa editrice stessa.

12.4.3. Possibilità di Fruizione delle Opere Inserite nel Sistema

Nonostante vi sia una libera riproduzione per le opere cadute in pubblico dominio, la stessa possibilità non è concessa relativamente ad opere comprendenti note a margine e le cd. edizioni critiche di un'opera, protette da un diritto connesso di durata ventennale.
Vi sono dei problemi anche con le opere ancora protette da diritti d'autore o diritti connessi, sarebbe necessario ottenere il consenso del titolare del diritto.
In prima analisi, ne deriva, quindi, che si possano utilizzare esclusivamente opere cadute in pubblico dominio prive di note a margine protette; tuttavia la legislatura, tramite l'articolo 71-bis l.a., offre maggiori possibilità nei confronti di portatori di handicap.
Nel caso di fruizione da parte dei portatori di handicap infatti, l'utilizzazione delle opere è gratuita (nelle condizioni indicate dalla legge, come illustrato in seguito) in maniera indipendente dalla tipologia di opera indicata in precedenza.
La dislessia, indipendentemente dalla forma e dal grado, è riconosciuta nel diritto attuale come forma di handicap, con la conseguente fruibilità libera delle opere descritte.

E’ necessario, pertanto, capire se le condizioni di legge per rendere il servizio gratuito e libero nei confronti degli utenti dislessici del sistema V.E.R.O.N.I.C.A. siano verificate.
Per il godimento dell’esenzione è sufficiente che l’utilizzo dell’opera sia riservata al portatore di handicap; in seguito sono richieste le seguenti condizioni:

- l’utilizzo del materiale deve essere collegato all’handicap;
- l’utilizzo del materiale non deve avere fini di lucro;
- l’utilizzo del materiale deve limitarsi a quanto richiesto dall’handicap.

La prima condizione è certamente rispettata, essendo i libri destinati alla lettura digitale da parte degli utenti dislessici. Anche la seconda condizione è verificata: come già detto in precedenza, infatti, il progetto V.E.R.O.N.I.CA. non ha fini di lucro, ma si configura come un servizio libero dedicato gratuitamente agli utenti regolarmente registrati (la registrazione è gratis).
Non sono previsti altri utilizzi dei libri, al di fuori della lettura automatizzata descritta in precedenza: ne consegue che anche la terza condizione sia rispettata.
Vi sono quindi i presupposti perché gli utenti del sistema V.E.R.O.N.I.C.A. possano utilizzare liberamente le opere interne al sistema.
Si ricorda, infine, che l’utilizzo personale delle opere è garantito dalla creazione di appositi account per ogni singolo utente precedentemente registrato e di cui si ha prova certa della dislessia grazie ai documenti forniti all’atto di registrazione.
L’integrità degli account e dell’intero sistema è garantita dalle forti misure di sicurezza intraprese nella realizzazione del progetto, per maggiori informazioni in tema è possibile consultare il capitolo Sicurezza.

12.5. Cessazione dell’utilizzo di V.E.R.O.N.I.C.A.

Il sistema V.E.R.O.N.I.C.A. offre la massima trasparenza verso l’utente che decide di terminare il suo utilizzo o disabilitare temporaneamente il suo profilo.
Come stabilito dall’informativa presentata al primo accesso dell’utente, indipendentemente dal fatto che essa riceva il consenso al trattamento o meno, il sistema garantisce la rimozione totale dei dati medico-sanitari dell’utente dislessico.
Si riserva, per utilizzi futuri, la possibilità di inserire un ulteriore informativa, da sottoporre a consenso dell’utente, nella quale si esplicita la possibilità di mantenere i suoi dati in forma anonima all’interno del sistema, per un trattamento statistico.
Ad ogni modo, il sistema garantisce la rimozione totale dei dati medici anche all’utente che si disabilita temporaneamente da V.E.R.O.N.I.C.A., salvaguardando, in ogni caso, i suoi dati personali per una, eventuale, riabilitazione futura del proprio profilo.

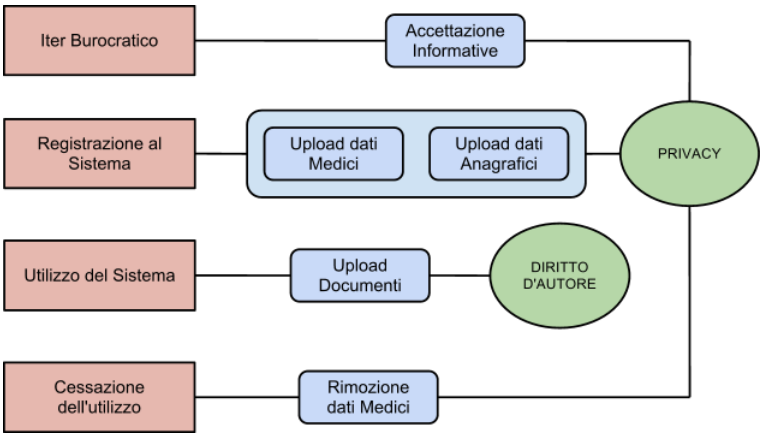

Immagine 12.1: Fasi della fruizione da parte dell’utente del sistema V.E.R.O.N.I.C.A.

13. Conclusioni e Sviluppi Futuri

13.1. Interfaccia e Interazione

Modellare, colorare, ideare l'interfaccia e offrire all'utente un sistema estremamente utilizzabile e piacevole è un obiettivo che richiede un grande spirito autocritico, tanti feedback (tanti pareri esterni), e tanta creatività. È possibile studiare un processo di personalizzazione delle interfacce utente, sebbene sia difficile che tutti gli utilizzatori la ritengano soddisfacente per le proprie esigenze; come in ogni realtà che ci circonda, ci sarà sempre qualcuno che avrebbe fatto diversamente o che avrebbe adottato differenti strategie. Il discorso della personalizzazione, con temi e colori diversi, riesce a colmare le lacune portate da questo problema. L'interazione è, invece, meno restrittiva. La scelta di implementare un metodo anziché un altro è, tipicamente, semplificata dal fatto che bisogna dare all'utente quello che cerca con meno fatica possibile. A prescindere dai gusti di una persona, solitamente è più semplice discutere su quale sia la strada migliore da offrire. Le scelte di implementazione sono molteplici, ma, ad ogni modo, il cerchio si restringe rapidamente per via del modo in cui l'utente dovrà percepire e manipolare le informazioni per portare a termine un compito.
Scelte progettuali che, a volte, costringono l'utente a fare ulteriori controlli o a dare molta attenzione alle operazioni che compie. Questo tipo di stereotipo porta ad avere un sistema di difficile utilizzo che, a tratti, stanca l'utilizzatore medio. Il lavoro che il team ha svolto si concentra tantissimo su queste semplificazioni, assolutamente non banali e da non sottovalutare. Per il design delle interfacce, è molto importante chiedersi cosa l'utente vorrebbe trovare in una determinata posizione, o quando (e come) il sistema debba comunicare un eventuale messaggio, senza turbare le sue aspettative. Affinché il sistema venga adottato, è bene che chi lo utilizza si senta appagato. Le funzionalità da sole non bastano se non vengono presentate nel modo corretto, oppure se è difficile reperire o capire dove trovare ciò che si cerca. Con V.E.R.O.N.I.C.A. verrà creata un apposita area che consentirà la richiesta di eventuali modifiche all'interfaccia o l'aggiunta di funzionalità, in quanto l'utente non deve adattarsi per forza al sistema, ma deve avvenire anche, e soprattutto, il viceversa.

13.2. Base di Dati

Adattività. E' questa la parola chiave con la quale si può descrivere lo sviluppo della base di dati. Adattività alle esigenze maturate con l'avanzamento della realizzazione del sistema, ma, soprattutto, adattività agli sviluppi dei moduli che compongono il sistema.
La fase iniziale è stata caratterizzata da un attenta fase di analisi dei requisiti, in linea con i servizi che il sistema avrebbe dovuto offrire, seguita dalla realizzazione di opportune versioni, studiate appositamente per garantire la massima efficienza nella memorizzazione e fornitura dei dati.
A livello di obiettivi, possiamo citare i seguenti:

- scelta di un DBMS adatto al sistema da realizzare;
- strutturazione dello schema in modo che siano garantite le massime performance;
- memorizzazione dei dati essenziali per la fruizione del servizio;
- scelta di opportune strategie di suddivisione e gerarchizzazione dei dati.

Come già precisato, nel capitolo Descrizione della Base di Dati, lo sviluppo della base di dati è stato realizzato incrementalmente: si è iniziato, infatti, con la definizione delle entità e delle associazioni principali e dei relativi attributi. Proseguendo con lo sviluppo dell'intero sistema

è stato importante curare gli aspetti che consentono il corretto funzionamento dei restanti moduli: ciò ha comportato un'attenta analisi delle fasi di sviluppo realizzate dal team allo scopo di rielaborare le funzionalità messe a punto e di integrarne di nuove o rimuovere o modificare quelle esistenti. Allo stesso modo, la base di dati è stata realizzata con un occhio di riguardo verso gli sviluppi futuri del sistema: lo schema concettuale, al momento in cui si scrive, si presta perfettamente all'integrazione di nuove entità e relazioni, rendendo semplice la traduzione dello stesso nello schema logico. A tal proposito, si ha un esempio rilevante di una modifica apportata in fase di realizzazione: si tratta dell'inserimento e conseguente integrazione delle tabelle riguardanti la memorizzazione fisica dei documenti. Tali tabelle, inizialmente non erano previste, in quanto l'idea iniziale si basava sulla memorizzazione dei dati all'interno del filesystem del server. La scelta di memorizzarli all'interno del database ha comportato una riduzione notevole del carico di dati sul server e ha garantito una gestione ottimizzata dei dati grazie alla cache fornita dal DMBS, permettendo, a livello architetturale, una netta separazione tra i compiti svolti dal server e dal database: infatti, sebbene il server dipenda strettamente dal database per via delle query che è necessario effettuare, non è vero il contrario, in quanto nel database non vi è nessun riferimento a dati presenti sul server. Ciò permette anche di effettuare un eventuale trasferimento futuro del sistema senza pericoli di incompatibilità o sicurezza. In conclusione, si può affermare che gli obiettivi prefissati inizialmente, e durante lo sviluppo, siano stati raggiunti in piena conformità con i requisiti stabiliti nelle diverse fasi di sviluppo realizzate. E' importante rimarcare, infine, la possibilità di integrare o estendere nuove funzionalità del sistema, senza dover praticare modifiche complesse o senza dover prescindere dalle funzionalità realizzate finora.

13.3. Networking

Il modulo citato è stato affrontato seguendo una opportuna divisione degli aspetti da sviluppare: inizialmente, si è impostato l'ambiente di lavoro, con relativi protocolli e software, che avrebbero gestito la comunicazione client-server ed i dati memorizzati internamente al database. In seguito, sono state affrontate le tematiche, prettamente software, legate alla gestione degli accessi al sistema dei vari profili utente e, di conseguenza, dei permessi di accesso alle diverse funzionalità offerte da V.E.R.O.N.I.C.A., come descritto nel capitolo 10, Modalità di accesso.

Un primo obiettivo è stato adattare l'architettura logica, pensata inizialmente a tre livelli, in un architettura a due livelli. In realtà, come visto, tale operazione è stata consentita dalla configurazione dei moduli forniti dalla piattaforma LAMP, essenziale nella gestione ed utilizzo del server.

Riguardo alla gestione degli accessi degli utenti al sistema, è importante dire che, a livello progettuale, era stato previsto un accesso ai soli utenti registrati ed una divisione tra le funzionalità accessibili dai diversi profili utente. Tali obiettivi sono stati pienamente raggiunti grazie al lavoro svolto mediante le funzionalità offerte dal linguaggio PHP, che ha permesso la creazione di sessioni utente, fondamentali per una gestione ottimale dei processi di riconoscimento ed accesso dell'utente al sistema. Si precisa che tale gestione è stata realizzata parallelamente allo sviluppo della base di dati, in quanto tali tipologie di controlli dipendono strettamente dai dati memorizzati.

In conclusione, si può affermare che gli obiettivi stabiliti inizialmente sono stati pienamente raggiunti, in rispetto dei requisiti determinati in fase progettuale.

Per quanto riguarda gli aspetti futuri, il team vorrebbe concentrare l'attenzione sui moduli offerti dalle API di Storage offerte dal linguaggio HTML5, svincolando l'attuale gestione delle sessioni utente e dei login dal linguaggio PHP e, conseguentemente, dall'utilizzo di cookies, dal momento che si ritiene essenziale adattare il sistema V.E.R.O.N.I.C.A. alle nuove

tecnologie sviluppate ed evitare l'utilizzo dei cookie, per i motivi espressi nel paragrafo 10.4, Gestione delle sessioni utente.

13.4. Sicurezza

Il termine sicurezza nel contesto del progetto indica l'insieme di provvedimenti adottati per proteggere utenti, informazioni e servizi inerenti il progetto V.E.R.O.N.I.C.A.

13.4.1. Il ruolo chiave della ricerca

Per questo motivo, precedentemente alla creazione e l'implementazione del sistema, vi è stato un lungo periodo caratterizzato da studio e ricerca sul campo della sicurezza.
In tutti i mesi successivi, la ricerca ha avuto uno spazio minore in favore delle fasi restanti ma non si è mai fermata. Attualmente, mentre il team si occupa di redigere la tesi sul prototipo realizzato ed il lavoro svolto, lo studio nel campo della sicurezza continua, al fine di realizzare gli sviluppi futuri sino a questo momento pianificati.
Come dimostra la Figura 13.1, si può affermare senza dubbio che la ricerca, nonostante sia un'attività intangibile (al contrario di progettazione, implementazione e test), sia stata l'attività più intensa e costosa nell'ambito della sicurezza. E' importante capire che la Figura 13.1 non entra in contrasto con quanto definito nella Figura 4.1, in quanto nella prima si discute di organizzazione delle attività sulla sicurezza che è diversa dal modello del processo adottato per l'intero progetto V.E.R.O.N.I.C.A.

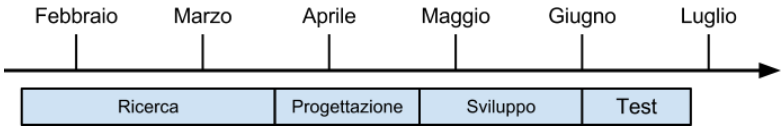

Immagine 13.1: Distribuzione nel tempo delle attività inerenti la sicurezza

La ricerca ha portato diversi riscontri positivi:

- Acquisizione di nuove conoscenze;
- Sviluppo di una mentalità orientata alle problematiche di sicurezza;
- Realizzazione di un progetto migliore (meno propenso ad errori);
- Riduzione dei tempi di sviluppo e test;
- Semplificazione del processo di manutenzione.

Per i motivi descritti, la ricerca nel campo della sicurezza ha costituito una base solida per il progetto e un'occasione di crescita culturale molto importante. Per l'estensione delle tematiche inerenti la sicurezza del progetto, prevista nei mesi successivi (e descritta al paragrafo 13.6, Sviluppi Futuri), il lavoro inizierà con l'avvio di una nuova attività di ricerca.

13.4.2. Obiettivi Iniziali ed Obiettivi Raggiunti

Il lavoro in ambito sicurezza è iniziato con l'obiettivo centrale di rendere il sistema V.E.R.O.N.I.C.A. sicuro, garantendo la massima protezione per:

- informazioni trattate;
- utenti registrati;
- servizi forniti.

Inizia quindi una forte attività di ricerca ed analisi su svariate tematiche della sicurezza, dalle possibilità di attacco ai meccanismi di crittografia e le debolezze tipiche dei sistemi.
Dallo studio nasce un piano di azione, che prevede i seguenti obiettivi (Tabella 11.4):

- applicazione di scelte progettuali orientate alla sicurezza;
- realizzazione di un canale di comunicazione sicuro;
- meccanismi di difesa;

- attacco al sistema attivo.

Gli obiettivi iniziali descritti, vengono in larga parte raggiunti con successo.
L'applicazione di numerose scelte progettuali (vedi 11.4) è stata essenziale per rendere il sistema sicuro prima ancora della sua realizzazione: non è facile incrementare la sicurezza di un sistema già operativo; se sono state applicate delle scelte progettuali poco sicure è più semplice costruire il sistema da capo che correggerne i difetti.
Nella progettazione del sistema V.E.R.O.N.I.C.A. sono state apportate diverse scelte progettuali importanti che si basano essenzialmente sull'esperienza di utilizzo di numerosi altri sistemi, col fine di evitare gli errori tipici osservati.
Non è semplice stabilire se l'obiettivo sia stato raggiunto, in quanto il sistema è progettato per durare decenni e solo il tempo può effettivamente decidere se le scelte progettuali compiute siano corrette o meno: sino al momento in cui si scrive, il team è altamente soddisfatto e non ha dovuto modificare nessuna delle scelte elencate.
La realizzazione di un canale di comunicazione sicuro (vedi 11.5) è il cuore del lavoro svolto sulla sicurezza nell'ambito del progetto V.E.R.O.N.I.CA: è stato il lavoro più impegnativo sia nella fase di ricerca, sia dal punto di vista della realizzazione.
L'importanza del canale è stata più volte evidenziata, con l'obiettivo di impedire all'attaccante di spiare le comunicazioni e fingere di essere una delle due parti autorizzate a comunicare. Il canale realizzato utilizza i migliori algoritmi disponibili attualmente (si pensi ad AES che è lo standard per il governo degli Stati Uniti), supportati da chiavi a 256 bit che comunemente vengono adottate per proteggere documenti classificati top secret.
In questo caso si può affermare con certezza che l'obiettivo è stato raggiunto: il canale è pienamente funzionante e lo scambio delle informazioni tra client e server è cifrato.
Sono stati adottati numerosi meccanismi di difesa (vedi 11.6) nei confronti del sistema: come si è discusso è stato fatto uno studio sulla sicurezza delle tecnologie coinvolte e gli attacchi più frequenti: la sicurezza del sistema è pari alla sicurezza della sua componente più debole quindi lo studio delle componenti utilizzate è davvero fondamentale.
L'attività di aggiornamento, come è stato evidenziato, svolge in questa sede un importante supporto: è importante sia l'aggiornamento delle tecnologie utilizzate, sia l'aggiornamento delle conoscenze sulla sicurezza.
Il lavoro di aggiornamento non è stato sufficiente.
La sicurezza è una caratteristica che per essere raggiunta ha bisogno di grandi sacrifici, in quanto entra fortemente in contrasto con altre caratteristiche comunemente richieste all'interno delle applicazioni, si pensi ad esempio all'efficenza. La realtà è che le tecnologie spesso non sono pensate per la sicurezza: l'esempio più evidente sono i sistemi operativi, creati in un ambito di cooperazione e suddivisione delle risorse tra processi, in opposizione alla filosofia di reciproca diffidenza e autonomia necessarie nella sicurezza.
Se le tecnologie offrissero maggiore supporto alla sicurezza il lavoro sarebbe più semplice.
Il conclusione si può affermare che il lavoro nei confronti della difesa ha pienamente raggiunto gli obiettivi prefissati, ma comunque deve proseguire nel corso del tempo.
Infine si era previsto di testare il livello di sicurezza di V.E.R.O.N.I.C.A. tramite numerose azioni di attacco al sistema, con l'obiettivo di ricercare, in fase di manutenzione, falle nei meccanismi di sicurezza e risolvere le problematiche trovate.
Il lavoro sulla sicurezza prevede numerose nuove idee, presentate nel paragrafo 13.6, sviluppi futuri.
In conclusione si può, sicuramente, affermare che il lavoro svolto rispecchia gli obiettivi prefissati e le aspettative nei confronti della sicurezza del progetto V.E.R.O.N.I.C.A.

13.5. Aspetti legali

Il lavoro svolto riguardo gli aspetti legali del progetto, ha come obiettivo principale il rispetto della legge all'interno del sistema V.E.R.E.O.N.I.C.A. Ne deriva la necessità di proteggere informazioni, utenti e servizi coinvolti per tutta la durata della fruizione del servizio; tale lavoro è svolto per garantire il rispetto dei seguenti obiettivi stabiliti:

- assicurare il trattamento privato dei dati anagrafici e medici dell' utente (privacy);
- assicurare il rispetto del diritto d'autore nelle opere dell'area pubblica;

Il trattamento dei dati utente rientra all'interno delle problematiche di privacy.
Ha costituito la tematica legale più ricorrente, in quanto è presente in tutte le fasi di interazione con il sistema: iter burocratico, registrazione, utilizzo e cessazione dell'utilizzo.
La tematica è stata affrontata pianificando delle informative da sottoporre per ricevere dall'utente l'autorizzazione al trattamento dei dati personali. Bisogna evidenziare l'importanza delle informative come strumento atto a risolvere le problematiche legali, sono infatti utilizzate in svariate parti del sistema: ad esempio al primo ingresso dell'utente su V.E.R.O.N.I.C.A. per indicare le corrette modalità di utilizzo della piattaforma e dei servizi erogati.
Gran parte del lavoro riguardante la privacy tuttavia deve essere ancora svolto, in quanto riguarda l'iter burocratico precedente alla registrazione al sistema e quindi deve essere precisato dall'ente gestore, non può essere gestito attualmente dal team.
Il trattamento delle opere pubbliche riguarda le problematiche inerenti il diritto d'autore.
Come specificato dettagliatamente nel capitolo 12) Aspetti Legali, il problema principale è legato alle opere sulle quali vige il diritto d'autore: nonostante per gli utenti dislessici le norme di legge attuali consentano l'utilizzo libero dell'opera, il gestore del sistema non ha alcun diritto di acquisire le opere senza specifica autorizzazione dall'autore.
L'unico modo possibile per trattare le categorie di opere citate sarà quindi contattare la casa editrice e chiedere di poter integrare l'opera all'interno della piattaforma.
Il lavoro svolto riguardo la privacy e la tutela dei dati gestiti dal sistema ha soddisfatto pienamente le aspettative del team raggiungendo l'obiettivo prefissato.
Per quanto riguarda il diritto d'autore, il team non può ritenersi pienamente soddisfatto delle normative attuali, in quanto non riflettono le esigenze del progetto.
Infatti in principio, si è stabilita la necessità di permettere la libera fruizione delle opere a tutti gli utenti del sistema, in particolar modo gli amministratori, in quanto è fondamentale permettere un controllo delle opere nel momento in cui si richiede la condivisione alla comunità di V.E.R.O.N.I.C.A. Tale necessità non si può soddisfare completamente, eppure, come noto, è importante evitare la pubblicazione di opere che non soddisfino i requisiti richiesti (specialmente vista la bassa età degli utenti utilizzatori).
In conclusione il gestore del sistema deve garantire che le opere condivise siano a norma di legge, ma non può effettuare alcun controllo non avendo il diritto di fruire delle opere stesse, (diritto riservato ai soli dislessici).
Il team è pienamente soddisfatto del lavoro svolto anche se avrebbe gradito un maggiore supporto dalle normative di legge per semplificare la gestione del sistema.

13.6. Sviluppi futuri

Gli sviluppi futuri prevedono un ampia scelta di funzionalità e la compatibilità con più dispositivi e browser. Questo è uno dei principali punti di forza del sistema V.E.R.O.N.I.C.A. che è stato progettato pensando il più possibile all'aggiunta di funzionalità, seppur cercando di mantenere le basi. Durante la realizzazione di questo primo prototipo, infatti, ci si è concentrati molto su quali aspetti potessero essere integrati in futuro e di come potesse essere

creata una base solida ma allo stesso tempo funzionale; per questo V.E.R.O.N.I.C.A. non manca di funzionalità e garantisce l'integrazione di numerosi servizi senza la necessità di reimplementare l'intero sistema daccapo.
Attualmente il sistema è stato testato su un solo tablet, per la precisione un iPad, e il team ha constatato che le funzionalità di questo primo prototipo si sono mantenute tali e quali a quelle fornite nel testing su PC. Si consideri il fatto che il sistema è utilizzabile su ulteriori tablet, eventualmente basati su differenti piattaforme, senza particolari differenze rispetto alla versione testata su iPad. Le eventuali modifiche da realizzare per ottenere la massima compatibilità sarebbero in numero ridotto, grazie alla portabilità del sistema garantita dalle tecnologie utilizzate in fase di sviluppo.
Il futuro di V.E.R.O.N.I.C.A. mira a raggiungere più sistemi possibile, per svincolare l'utente dall'acquisto di una nuova piattaforma. Ciò implica, quindi, che il team, una volta che il prodotto avrà terminato la sua fase prototipale, prenderà questo come uno dei primi presupposti. Ma non sarà sicuramente l'unico: infatti, una breve lista delle "novità" che potranno essere integrate, alla quale il team di sviluppo ha già riservato la dovuta attenzione, anche in fase di progettazione, sono le seguenti:

- utilizzo di un sintetizzatore vocale migliore;
- migliorie nella pagina di lettura del libro;
- estensione della compatibilità con i client;
- estensione della compatibilità con ulteriori formati di documento (doc, pdf ecc.);
- integrazione di nuovi temi;
- integrazione della personalizzazione totale dei colori;
- implementazione area messaggi;
- notifiche numeriche sui messaggi da leggere;
- notifiche numeriche sui libri da approvare;
- inserimento di nuove tipologie di test (test sulla memoria, decodifica di lettura e scrittura, altri);
- utilizzo di AJAX per aiutare a riempire i campi nell'inserimento di un nuovo libro;
- utilizzo di AJAX per per la ricerca delle immagini di copertina;
- integrazione con i Social Media;
- area feedback e richieste di funzionalità;
- classificazione privata dei libri;
- generazione di classifiche dei libri più letti (privati o pubblici);
- creazione dei libri;
- suddivisione dei libri per categoria;
- ricerca di un libro tramite ISBN, altri attributi o parte del testo;
- attacchi al sistema per testarne la sicurezza.

14. Glossario

A.E.S.
Advanced Encryption Standard è l'algoritmo di cifratura a blocchi attualmente adottato come standard dal governo degli Stati Uniti d'America. Deriva dall'algoritmo Rijndael, il vincitore del concorso tra algoritmi di cifratura conclusosi nel 2001 per definire quale sarebbe stato il nuovo standard, chiamato appunto A.E.S.

Apparato critico
Si tratta di una sezione dell'edizione critica di un documento, destinata a definire la traduzione del testo, per riportare il contenuto originale che l'autore avrebbe voluto esprimere. Nello specifico, è usato per definire le opere modificate da terzi, che si differenziano dal documento originale.

Architettura Fisica
Termine utilizzato per riferirsi all'architettura implementata in fase di sviluppo, a causa della macchina hardware di cui si è disposto. Si tratta di un architettura a due livelli.

Architettura Logica
Termine utilizzato per indicare l'architettura sviluppata in fase di progetto, sulla quale avrebbe dovuto realizzarsi il sistema. Si tratta di un architettura a tre livelli.

Aventi causa
Si tratta dei soggetti che riceveranno una situazione di diritto da un altro soggetto. In particolare, si tratta il caso di diritti d'autore tramandati dall'autore a parenti, o terzi.

Cifrari Classici
Rappresenta una categoria di cifrari, indicata spesso come cifrari storici: si distingue dai cifrari moderni. La categoria dei cifrari classici racchiude tantissimi cifrari, tra i quali il cifrario di Cesare indicato nel documento.

Cifrario a blocchi
Un cifrario a blocchi è un algoritmo che lavora su un blocco di dati di dimensione massima fissata. Viene definito cifrario a blocchi perché genera l'output in blocco, in opposizione agli algoritmi di cifratura a flusso che elaborano un elemento alla volta.
Il cifrario è composto da due funzioni, la codifica e la decodifica.
La prima prende in ingresso un blocco di testo in chiaro di dimensione massima fissata e rende in uscita un blocco di testo codificato. La seconda effettua l'operazione contraria.

Cifrario a blocchi ideale
Un cifrario a blocchi ideale possiede l'importante caratteristica di creare blocchi cifrati paragonabili a blocchi di dati casuali: l'attaccante non deve poter trarre alcuna informazione dal blocco cifrato spiato, non deve avere modo di risalire al blocco originale.

Cifrario di Cesare
Antico cifrario utilizzato da Giulio Cesare. L'algoritmo prevede la traslazione di ogni lettera dell'alfabeto utilizzato con una lettera di "k" posizioni precedenti o successive. Giulio Cesare utilizzava una traslazione in avanti di tre posizioni: quindi la chiave è tre.

Ciphertext
Il termine inglese indica il testo cifrato. Attraverso la funzione di decifratura applicata al testo cifrato è possibile ricavarne il corrispondente testo in chiaro.

Collisione
Per quanto definito dalla legge del buco della piccionaia, è ovvio che due diversi input di una funzione hash possano produrre il medesimo output.
Tale situazione, viene definita nella crittografia come collisione e viene sfruttata dall'attaccante del sistema in svariate modalità dipendenti dal contesto in cui si opera.

Counter
Indicata anche come CTR, è una modalità di cifratura a blocchi: è la modalità utilizzata per il progetto V.E.R.O.N.I.C.A.

CTR
Vedi Counter.

Digest
Il termine indica il generico output di una funzione hash.

Documento Privato
Si tratta di documenti caricati dagli utenti registrati al sistema all'interno della propria area personale. Sono consultabili, pertanto, esclusivamente da chi li ha caricati.

Documento Pubblico
Con il termine "documento pubblico" si intende l'insieme dei documenti condivisi all'interno del sistema V.E.R.O.N.I.C.A. e, pertanto, da intendersi pubblici per gli utenti registrati al servizio.

DSA
Si intendono i disturbi specifici nell'apprendimento di alcune abilità specifiche che non permettono una completa autosufficienza nell'apprendimento poiché le difficoltà si sviluppano sulle attività che servono per la trasmissione della cultura, come, ad esempio, la lettura, la scrittura e/o il far di conto.

Funzione Hash
Costituisce una primitiva crittografica, si distingue dal cifrario in quanto è una funzione unidirezionale, mentre il cifrario prevede due funzioni: cifratura e decifratura.
Anche se non esiste la funzione inversa per il meccanismo di Hash, è ugualmente estremamente utile. Ad esempio è utilizzata per memorizzare informazioni codificate nella base di dati, in modo che chi si intromette nel sistema e vede le informazioni, non può capirle ne ottenere di nuovo i dati che le hanno originate.

HMAC
Uno dei possibili algoritmi per implementare l'autenticazione: lavora sfruttando altre primitive crittografiche, in particolar modo funzioni hash. La caratteristica più importante di HMAC è l'indipendenza dalla funzione associata, se si dovessero scoprire delle debolezze inerenti la funzione scelta la si sostituisce con una più adeguata.

Legge del buco della piccionaia
Noto principio matematico conosciuto anche come principio dei cassetti, afferma che se (n + x) oggetti con "n" ed "x" maggiori di zero sono messi all'interno di "n" cassetti, allora

almeno un cassetto dovrà contenere più di un oggetto. Dal punto di vista della crittografia il principio illustra che avendo gli insiemi finiti A (insieme dei testi in chiaro) e B (insieme dei risultati delle funzioni hash), dove B ha cardinalità strettamente minore di A, allora non esistono funzioni iniettive da A a B.

Livello di sicurezza
Vorrebbe indicare quanto un sistema è sicuro.
Di difficile quantificazione; solitamente viene pesato mediante la quantità di lavoro che deve fare l'attaccante per riuscire a forzare il sistema: la quantità di lavoro è rappresentata tramite il numero di passi elementari da compiere, dove un passo può corrispondere a diverse azioni a seconda del contesto, ad esempio confronti in tabella oppure operazioni aritmetiche basilari.

MAC
Vedi Message Authentication Code.

Macchina Enigma
Si tratta di un'apparecchiatura per la cifratura utilizzata ampiamente dai tedeschi durante la seconda guerra mondiale. Gli inglesi, tramite Alan Turing, per contrastare l'azione tedesca, crearono una macchina per rompere il codice nemico: è la Bomba di Turing.

Message Authentication Code
Message Authentication Code è un meccanismo utilizzato per gestire il processo di autenticazione. Per autenticare un blocco di dati, viene associato al suo rispettivo output in hash. Il destinatario legge la parte di dati in chiaro e ricalcola l'hash: se il risultato ottenuto è identico all'hash inviato, allora l'autenticazione è superata, diversamente un'attaccante ha tentato di modificare il messaggio e quindi viene scartato.

Modalità di cifratura a blocchi
È un algoritmo che viene associato al cifrario a blocchi: permette di cifrare (e decifrare) blocchi di dimensione variabile. Il cifrario a blocchi infatti lavora esclusivamente su blocchi di dimensione massima prefissata, la modalità di cifratura permette di suddividere l'input di dimensione variabile in blocchi di dimensione fissa da sottoporre al cifrario.

Modello della paranoia
Descrive l'atteggiamento da assumere nella progettazione della sicurezza. Si deve sempre diffidare dei moduli esterni coinvolti nel sistema, garantendo che la sicurezza sia una proprietà locale e partendo dal presupposto che ogni entità esterna coinvolta possa rappresentare una minaccia. Il discorso non vale solamente per i moduli, ma deve essere necessariamente esteso anche a persone, società e ogni altra possibile entità coinvolta.

Paradosso del Compleanno
Il principio matematico descrive che: se vengono presi "n" elementi casuali, che hanno "m" possibili valori, allora dopo un "n" pari alla radice di "m" vi è una probabilità elevata di ottenere due elementi identici. Il principio prende il nome di paradosso del compleanno, in quanto se si considerano "n" persone, la cui data di compleanno è in un range di "m = 365" valori, allora dopo aver preso radice di "m" ovvero "n = 19" persone, si ha un elevata probabilità che due persone compiano gli anni lo stesso giorno. La percentuale apparentemente è incredibile se si considera l'ampiezza del range.

Plaintext
Il termine inglese indica il testo in chiaro. Il testo può essere dato in input a una funzione di cifratura per ricavarne il ciphertext, ovvero il testo cifrato.

Principio di Horton
Il principio di Horton suggerisce di codificare non solo "ciò che viene detto" ma "ciò che viene inteso": indica l'importanza di non lasciare alcuna informazione utile all'attaccante.
Un esempio pratico può essere la codifica di un determinato testo, lasciando in chiaro nel messaggio di invio però il significato del contesto al quale il messaggio si riferisce. L'unica cosa che un attaccante deve poter capire di un messaggio è la sua lunghezza e il tempo di invio, il resto deve apparire come una sequenza di dati casuali incomprensibile.

Principio di Kerckhoffs
Un sistema crittografico è basato essenzialmente su un insieme di algoritmi ed un insieme di chiavi. Secondo Kerckhoffs la sicurezza del sistema deve dipendere unicamente dalla segretezza delle chiavi, non dell'algoritmo.

Principio di Shannon di confusione e diffusione
Il principio è generalmente riferito ad un algoritmo di cifratura o autenticazione.
Suggerisce che un algoritmo debba produrre un blocco in uscita con le importanti caratteristiche di confusione, cioè difficoltà a ritrovare la chiave a partire dal blocco spiato, e diffusione ovvero eliminazione delle correlazioni statistiche.

Proprietà dell'anello debole
In tema di sicurezza rappresenta un principio fondamentale.
Stabilisce che nel valutare il livello di sicurezza di un sistema bisogna prenderne in considerazione le debolezze, non i punti di forza. "*Il massimo livello di sicurezza di un sistema è pari alla sicurezza del suo componente più debole*".

Rete a sostituzione e permutazione
Rappresenta un modello di architettura per un algoritmo di cifratura alternativo alla rete di Feistel: Twofish utilizza una rete di Feistel, mentre A.E.S. si basa su una rete a sostituzione e permutazione.

Rete di Feistel
Consiste in un particolare cifrario con la proprietà di rendere molto simili le operazioni di cifratura e decifratura, semplificando le implementazioni hardware. La rete di Feistel è un meccanismo integrato in tanti cifrari moderni, ma non ad esempio A.E.S.

SHA
Identifica una famiglia di funzioni HASH tra le più utilizzate. All'interno del progetto V.E.R.O.N.I.C.A. si è scelto di utilizzare l'algoritmo SHA256, appartenente alla famiglia SHA2. Attualmente, è in corso di definizione il nuovo standard SHA3.

15. Bibliografia

Niels Ferguson, Bruce Schneier.
Pratical Cryptography.
Wiley Publishing, Inc., 2003.
ISBN 978-0-471-22357-3

Mark Wandschneider.
PHP e MySQL.
Apogeo, 2006.
ISBN 978-88-503-2541-2

Jim Arlow, Ila Neustadt.
UML e Unified Process.
McGraw-Hill, 2003.
ISBN 978-88-386-6144-8

Andrew S. Tanenbaum, David J. Wetherall.
Computer Networks.
Pearson, 2010.
ISBN 978-0-13-255317-9

Gabriele Gigliotti.
HTML5 e CSS3.
Apogeo, 2011.
ISBN 978-88-503-3011-9

16. Sitografia

La consultazione di tutti i siti presentati è aggiornata alla data 12/07/2012.

Dislessia
Sito AIRIPA, Associazione Italiana per la Ricerca e l'Intervento nella Psicopatologia: http://www.airipa.it/
Associazione Italiana Dislessia: http://www.aiditalia.org
AIDNetwork: http://www.aid.it/chi.htm
Allegato CC DSA: http://www.snlg-iss.it/cms/files/Allegato_CC_DSA.pdf
Disturbi Specifici dell'Apprendimento Consensus Conference: http://ww2.istruzioneer.it/wp-content/uploads/2011/07/Consensus+conference+ist+sanita.pdf

Demografia e Statistiche
Dati statistici sulla quantità di studenti in Italia: http://datablog.ahref.eu/i-numeri-della-scuola-2-2013-la-popolazione-scolastica-italiana
Numero degli studenti in Sardegna: http://www.regione.sardegna.it/j/v/491?s=97031&v=2&c=1489&t=1

Sintetizzatori
I migliori sintetizzatori: http://www.robertosconocchini.it/dsa-dislessia/408-i-migliori-programmi-text-to-speech-nel-web.html

Font per la Lettura
Sito Font: http://www.biancoeneroedizioni.com

Apache
Configurazioni Apache: http://leotardi.no-ip.com/download/manuali/Apache%20-%2030.pdf

EPUB
Guida alla struttura degli EPUB: http://www.guidaebook.com/guida-epub/

HTML
Tutorial Introduttivo: http://www.html5today.it/tutorial/websockets-html5-tutorial-introduttivo
Test delle pagine: http://html5test.com

JavaScript
AES: http://www.movable-type.co.uk/scripts/aes.html
Crypto-js: https://code.google.com/p/crypto-js/
Guida invio dati: http://www.compago.it/manuali/33-programmazione/316-inviare-dati-in-javascript-e-riceverli-correttamente-in-php.html
Gestione delle stringhe: http://www.morpheusweb.it/html/manuali/javascript/javascript_stringhe.asp

PHP
Sito ufficiale PHP: http://www.php.net/
PHP Wikipedia: http://it.wikipedia.org/wiki/PHP#Sicurezza
Guida Base PHP: http://php.html.it/guide/leggi/99/guida-php-di-base/
Sessioni: http://www.siforge.org/articles/2003/11/10-guida_sessioni_php.html
Stringhe: http://php.html.it/guide/lezione/2658/le-funzioni-in-php-gestire-le-stringhe/

Sicurezza: http://www.icosaedro.it/articoli/php-security.html
Injection: http://antirez.com/post/33
AES: http://www.phpaes.com/
AES CTR: http://www.movable-type.co.uk/scripts/aes-php.htmlripts/aes-php.html
HMAC: http://php.net/manual/en/function.hash-hmac.php

Drupal
Guida alla creazione di un tema per Drupal: http://www.tecnomeme.it/articoli_web/creare-tema-drupal
Tema di partenza per creare nuovi temi: http://drupal.org/project/zen

Ingegneria del software
Guida per la realizzazione di un progetto:
https://docs.google.com/viewer?a=v&q=cache:U0Ui0vh4pSsJ:venus.unive.it/matdid.php%3Futente%3Dcortesi%26base%3DComputer%2BScience%253A%2BIngegneria_del_Software%252FEsempio_PianoProgetto_1.pdf%26cmd%3Dfile+&hl=it&gl=it&pid=bl&srcid=ADGEESgOmFbnc9cDgtvInqxxtevJ66VYCngxT41RRjBrBnq8LojbMej54PZKgmhIA3tHl7MX-UmkmY4FXi_qH3CAvJ3PdO7vqtW2R4bZn7I3Rbe_IsW4eufRKq7cZG-OGQojABvZpYUG&sig=AHIEtbRfc-qcV5ob5Q0e5DuNle-W7vKhtg

Sicurezza
MAC: http://nazarenolatella.myblog.it/archive/2009/07/14/mac-message-authentication-code.html

Altro
Disney: http://www.disney.it

Ringraziamenti

Si ringraziano il Prof. Fenu e la Dott.ssa Loi per l'idea del progetto.
Si ringrazia il Prof. Fenu per aver creduto nel team di sviluppo e avergli affidato il progetto.
Si ringraziano il Prof. Bartoletti, la Prof.ssa Corso, il Prof. Fenu, la Prof.ssa Pes e il Prof. Scateni, nostri relatori, per l'impegno e la passione con cui ci hanno aiutato a sviluppare il progetto.
Si ringraziano la Dott.ssa Fadda, la Dott.ssa Zuddas e il Dott. Iacolina per l'aiuto e per i preziosi suggerimenti fornitici; il Dott. Nitti per la pazienza con la quale ci ha sopportato durante i mesi di sviluppo del progetto.
Ringraziamo tutte le persone che ci sono state vicine durante il nostro percorso universitario, le nostre famiglie, i nostri amici e colleghi.
Infine, si ringraziano gli studenti Barbieri, Loddo, Mameli, Muntoni e Pompianu, perché senza di loro il progetto non sarebbe stato sviluppato (un ringraziamento da parte di ognuno agli altri membri del team).