Secure Chat App

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.2.1 Spiegazione elementi tabella dei requisiti: 5

2.3 Use case 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 5

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

3.3 Design delle interfacce 6

3.4 Design procedurale 6

4 Implementazione 7

5 Test 7

5.1 Protocollo di test 7

5.2 Risultati test 8

5.3 Mancanze/limitazioni conosciute 8

6 Consuntivo 8

7 Conclusioni 8

7.1 Sviluppi futuri 8

7.2 Considerazioni personali 8

8 Glossario 8

9 Bibliografia 9

9.1 Bibliografia per articoli di riviste: 9

9.2 Bibliografia per libri 9

9.3 Sitografia 9

10 Allegati 9

# Introduzione

## Informazioni sul progetto

In questo capitolo raccogliere le informazioni relative al progetto, ad esempio:

* Allievo: Lukas Moro
* SAMT I3AA
* Sezione Informatica

## Abstract

Lo scopo di questo progetto è di creare un applicazione di messaggistica web con gestione degli utenti, autenticazione con Token JWT e chat privata con un altro utente.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large-Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo del progetto sarà quello di creare l’applicazione web con:

-Tutti i file di Node.js

-Informazioni sull’uso dell’applicazione

-Info sull’uso delle API

-Media usati per le prove

# Analisi

## Analisi del dominio

Questo progetto Node.js consisterà nel creare un applicazione di messaggistica da poter utilizzare durante scuola/lezioni per chattare tranquillamente con i compagni.

Esistono già prodotti simili, un esempio è Whatsapp Web.

Gli utenti saranno tutti quelli che hanno bisogno una rapida applicazione di messaggistica usando semplicemente nome utente e password senza bisogno del telefono e autenticazione a due fattori.

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Registrazione Utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisogna avere un form di registrazione e verificare il formato dell’email, telefono, password ecc.. |
| **Sotto requisiti** | |
| **001** | Compilare tutti i campi obbligatori |
| **002** | Verifica che le informazioni siano scritte correttamente |
| **003** | Verifica che non esista già un account con lo stesso nome |
| **004** | Verifica della sicurezza della password |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Login Utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisogna avere un form di login, compilare campi obbligatori |
| **Sotto requisiti** | |
| **001** | Avere già creato l’account |
| **002** | Verificare la password |
| **003** | Ev. Verifica via email |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Autenticazione dell’utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Avere il DB con tutte le tabelle e avere il sistema per generare il Token |
| **Sotto requisiti** | |
| **001** | Ottenere utente e password |
| **002** | Verificare che l’utente esista |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Chat privata |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Bisogna avere interfaccia delle chat |
| **Sotto requisiti** | |
| **001** | Avere fatto il login e la registrazione |
| **002** | Aggiungere l’utente con cui si vuole parlare e aspettare che accetti |
| **003** | Rifiutare/accettare la conversazione se me lo chiedono |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Inviare messaggi |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Generare chiavi pubbliche private |
| **002** | Criptare messaggi |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Ricevere messaggi |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Avere accettato la conversazione |
| **002** | Decriptare messaggi (anche eventuali sul DB) |
| **003** | Salvare messaggi su DB se sono offline |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Inviare file o immagini |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Avere il sistema per convertire e processare i file ed eventualmente salvare su DB |
| **Sotto requisiti** | |
| **001** | Avere effettuato l’accesso |
| **002** | L’immagine o file non deve superare un certo limite di dimensione |
| **003** | Devo potere inviare al massimo 1 file/immagini alla volta |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Layout |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Colore di sfondo viola scuro con filigrana per le chat.  Mostrare immagine profilo con nome e se è online o no.  Schermata divisa in due colonne, a sinistra gli utenti e a destra la chat. |
| **Sotto requisiti** | |
| **001** | L'interfaccia deve essere responsive |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

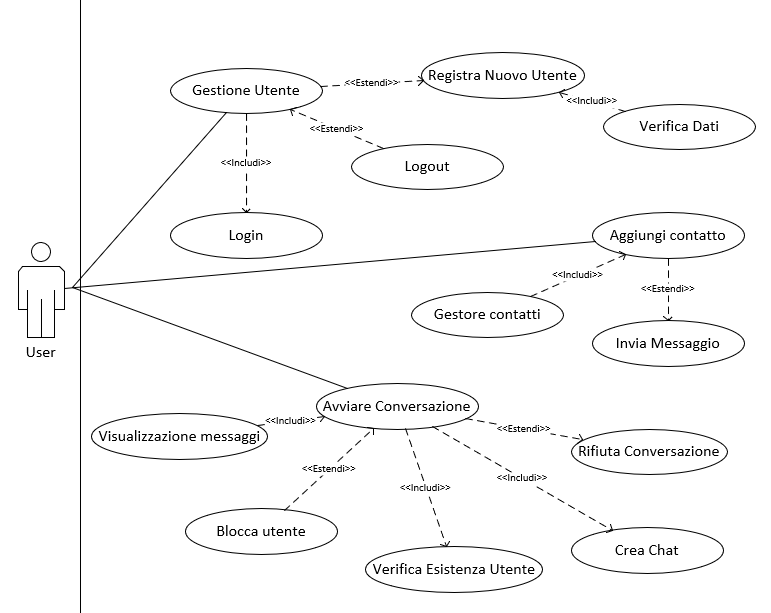
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| Figura 1: Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (p.es. scrum), dovranno apparire in questo capitolo.

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

Come Tool ho utilizzato Node.js insieme a React per fare le varie pagine, invece ho utilizzato Visual Studio Code come software per editare i file.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che HW sarà disponibile durante lo sviluppo?

L’applicazione dovrà essere eseguita su un server o macchina locale con installato Node.js.

Non servirà un hardware particolare dato che Node.js è leggero.

L’importante è che se faccio partire l’applicativo su due macchine differenti cercando di aprire la chat e collegarsi, siano entrambi sulla stessa rete e collegate, o per semplificare tutto utilizzare un server web con Node.js.

Durante il progettò avrò a disposizione una macchina virtuale (Ubuntu) su cui installare Node.js e fare partire l’applicazione e un server web per provarla durante le fasi di testing.

# Progettazione

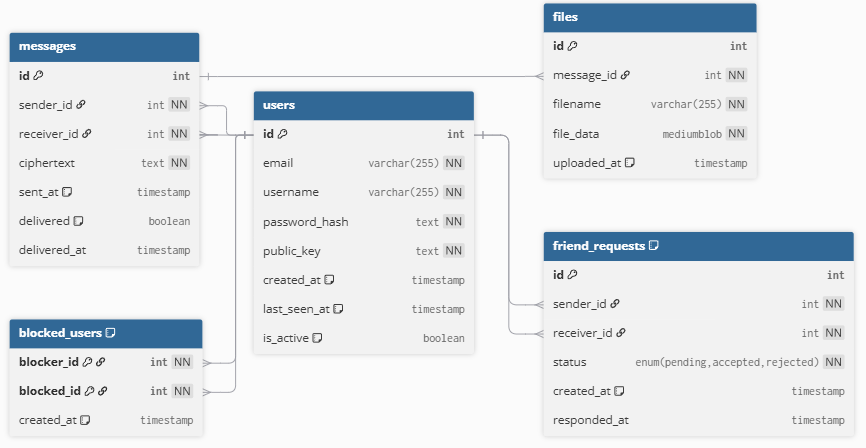
Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database



users(id PK, username UNIQUE, password\_hash, public\_key, created\_at, last\_seen\_at, is\_active)

friend\_requests(id PK, sender\_id FK → users(id), receiver\_id FK → users(id), status ENUM, created\_at, responded\_at, UNIQUE(sender\_id, receiver\_id))

blocked\_users(blocker\_id PK FK → users(id), blocked\_id PK FK → users(id), created\_at)

messages(id PK, sender\_id FK → users(id), receiver\_id FK → users(id), ciphertext, sent\_at, delivered, delivered\_at)

files(id PK, message\_id FK → messages(id), filename, file\_data, uploaded\_at)

Nel DB come tabelle ho creato users, messages, files, friend\_requests e blocked\_users.

Nella tabella Users ci saranno tutte le informazioni sull’utente.

In friend\_requests le richieste di amicizia in sospeso con un ENUM per indicare lo stato della richiesta.

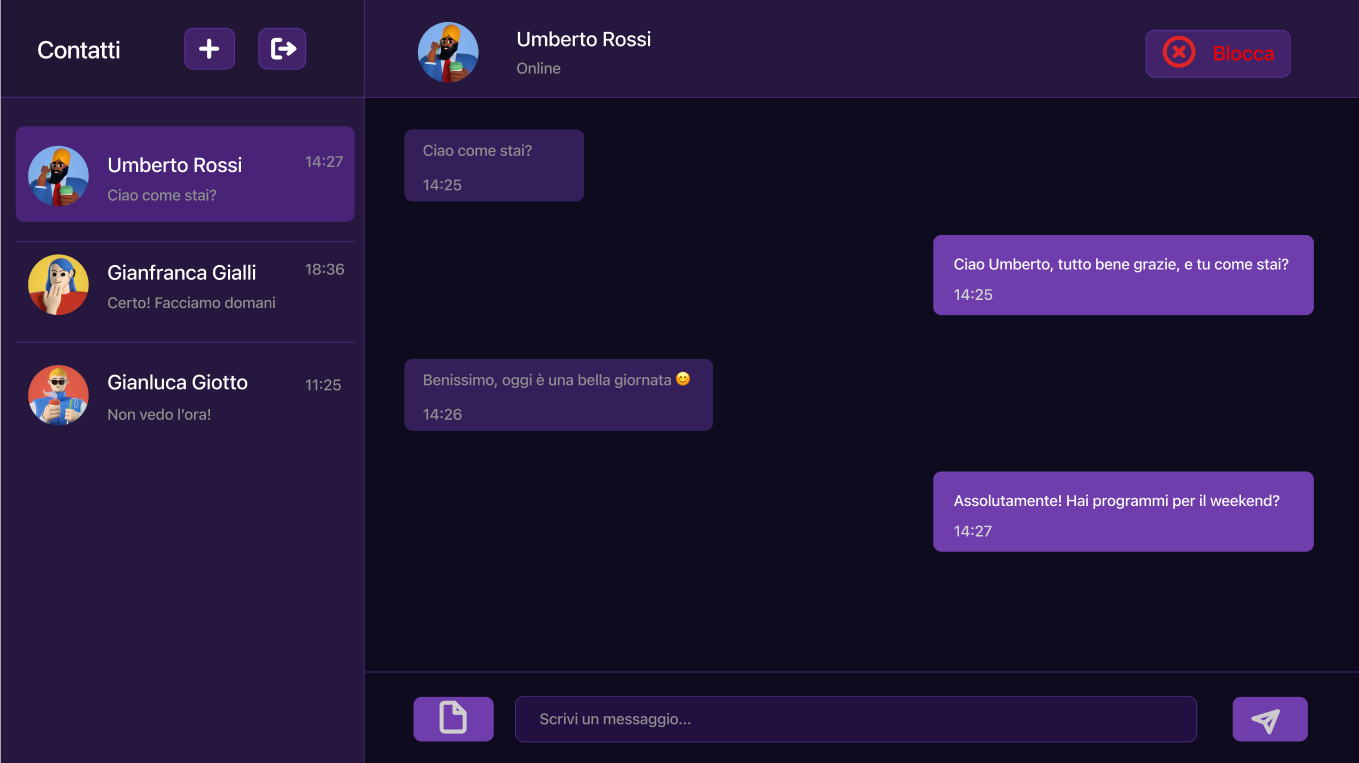
In blocked\_users una lista di utenti che sono stati bloccati da altri.

In messages tutti i messaggi che non sono ancora stati consegnati.

E come ultimo files che conterrà un possibile allegato che si vuole mandare in formato binario, in modo da rendere più facile tenerlo sul database.

## Design delle interfacce

Schermata Home:

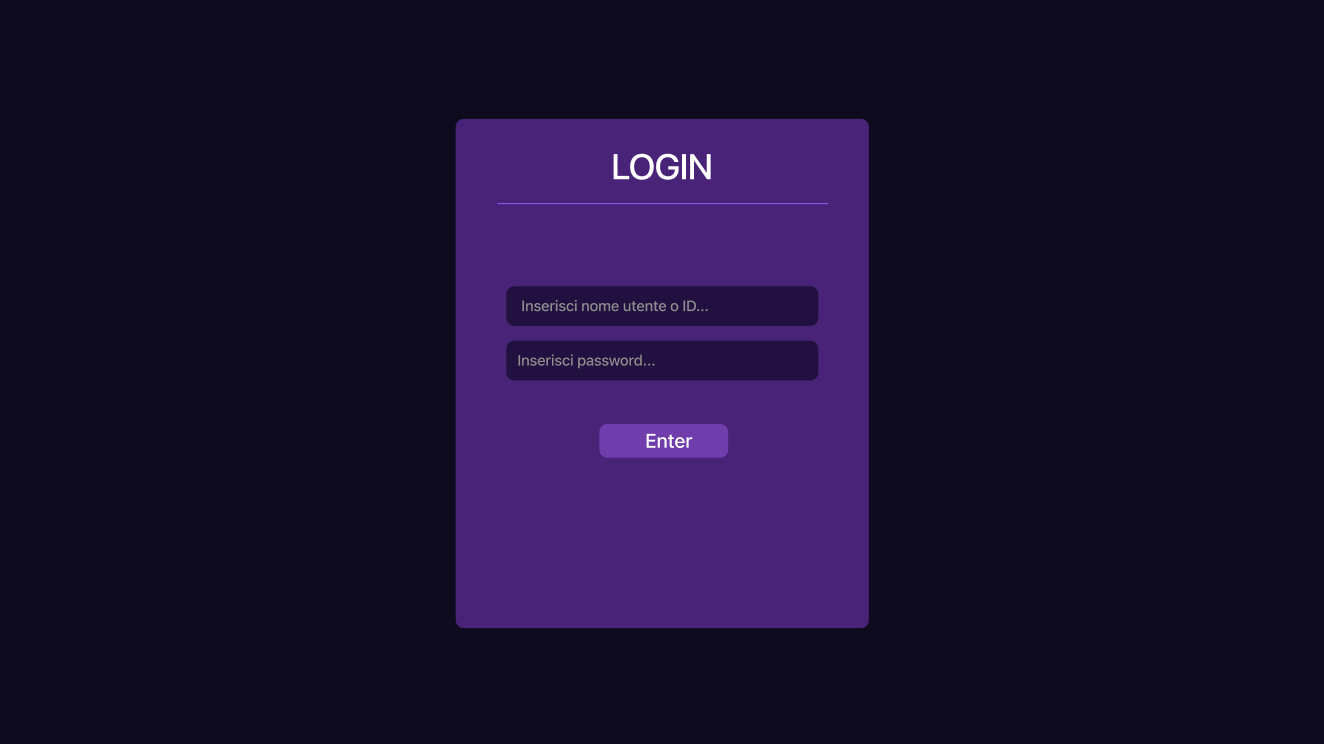


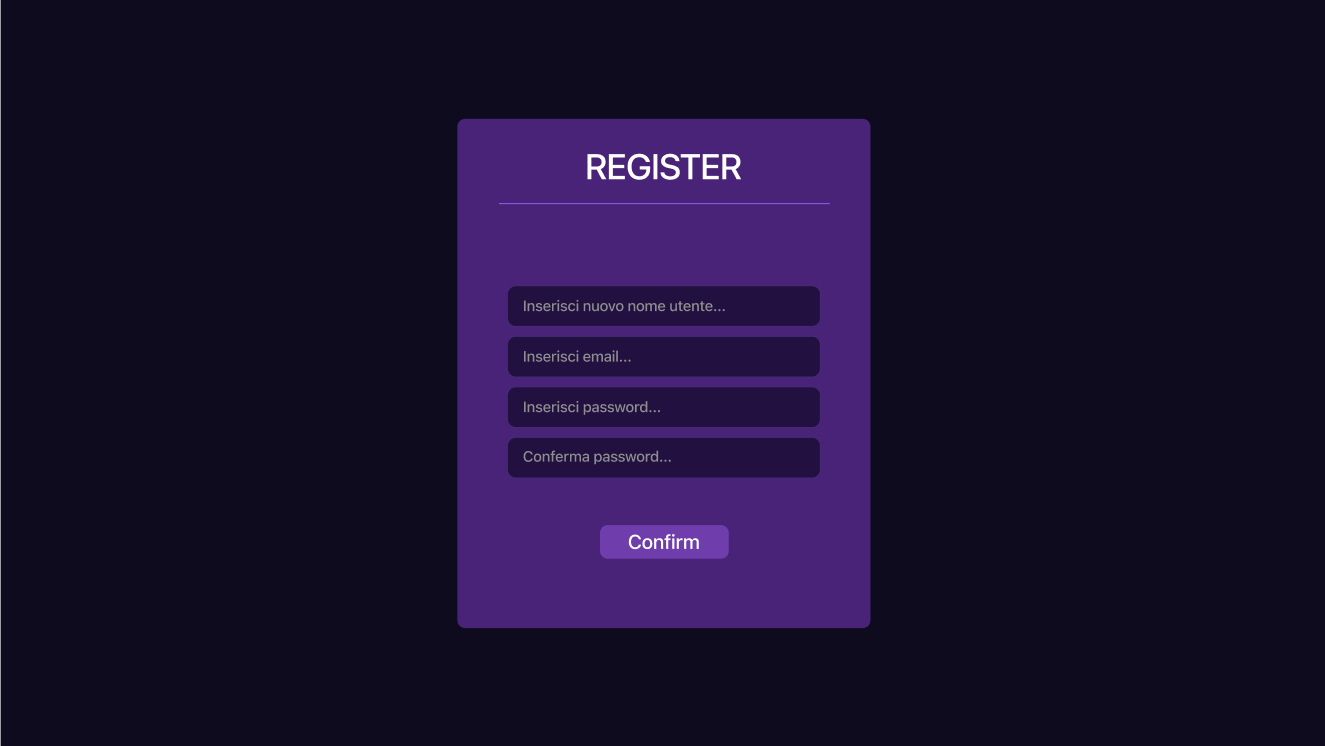
A sinistra ci sono gli utenti con l’ultimo messaggio mandato e l’orario a cui è stato inviato o riceuto.

Sopra la lista delle chat ci saranno i pulsanti per aggiungere un nuovo contatto, fare il logout e un pulsante per modificare le propie informazioni come nome utente e nome visualizzato.

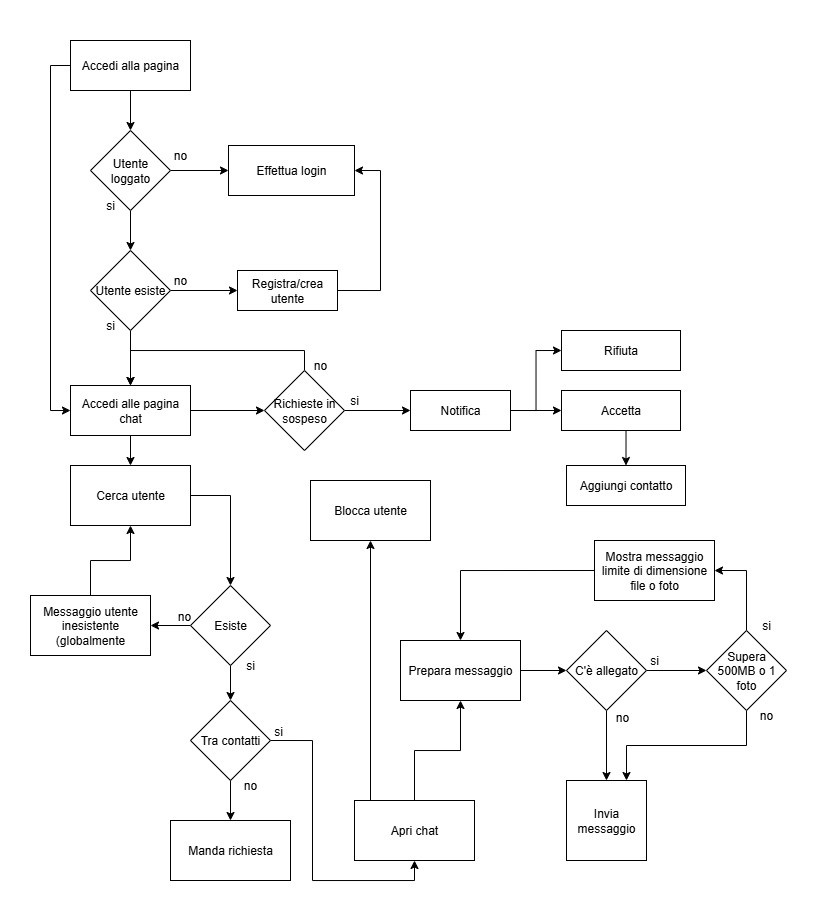
A destra ci sarà la chat vera e propia con gli ultimi messaggi, uno spazio per scrivere il messaggio e due pulsanti per inviare il messaggio e inviare un file.

Appena sopra la chat ci sarà l’immagine profilo della persona con cui sto parlando, il suo nome e il suo stato (online, offline) con a destra un pulsante per bloccare l’utente

Schermata Login:  


Schermata Registrazione:  


## Design procedurale



Il diagramma rappresenta lo schema logico dell’architettura dell’applicazione, mostra i principali componenti e le interazioni.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …