Documentazione progetto chat-locale

Samuele Ambrosetti

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: Samuele Ambrosetti
* Docente: Guido Montalbetti
* Scuola: Scuola d’Arti e Mestieri di Trevano
* Sezione: Informatica
* Inizio: 04.09.2024
* Fine: 18.12.2024

## Abstract

*Nowadays anyone uses messaging or social media applications to communicate, in fact as a project*

*it was chosen to program an application for messaging from the terminal. Knowing the username of*

*someone you can send a request to communicate.*

## Scopo

Lo scopo didattico del progetto è riuscire a gestire nel modo più ottimale possibile un progetto IT con le risorse datoci dalla scuola. Questo servirà successivamente a prepararmi all’esame pratico di fine scuola. Lo scopo operativo invece riguarda il creare un sito con Javascript e un server Python con il quale è possibile inviare messaggi conoscendo il nome utente del destinatario. Le comunicazioni e le chat verranno salvate e le più vecchie verranno eliminate.

# Analisi

## Analisi del dominio

Oggi molte persone utilizzano quotidianamente applicazioni di messaggistica. Questo nuovo servizio sarà accessibile principalmente da desktop, mentre l’uso su dispositivi mobili sarà limitato, almeno inizialmente. Sebbene esistano già diversi social e app per comunicare, la maggior parte di essi è fruibile principalmente tramite smartphone. Questa applicazione è pensata per utenti privati che desiderano comunicare senza conservare le chat, offrendo così uno spazio per scambi brevi e immediati. È probabile che venga implementata una crittografia per garantire la sicurezza delle comunicazioni. In generale, si tratta di una piattaforma molto semplice e intuitiva.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Accesso |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Chiunque può creare un account |
| **Sotto requisiti** | |
| **001** | L’accesso al sito necessita un log in |
| **002** | L’account può essere creato tramite una email, nome utente e password |
| **003** | Vanno eseguite due verifiche, se la email esista e se il nome utente sia disponibile |
| **004** | Deve essere possibile eseguire un log out |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Funzionalità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ricercare utenti o conversazioni |
| **002** | Visualizzare informazioni riguardo ai messaggi (volume e data) |
| **003** | I messaggi più vecchi nascosti |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Chat locale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| **Sotto requisiti** | |
| **001** | Si necessita una maschera di login |
| **002** | Si dovranno poter immettere nuovi allievi |
| **003** | Dovrà essere possibile la ricerca di allievi |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Chat locale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| **Sotto requisiti** | |
| **001** | Si necessita una maschera di login |
| **002** | Si dovranno poter immettere nuovi allievi |
| **003** | Dovrà essere possibile la ricerca di allievi |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto (vedi capitolo allegati).

## Pianificazione

La fine del progetto è prevista per il 4.12.2024, due settimane prima della consegna effettiva, per una durata totale prevista di circa 90 ore. Questa programmazione permette di avere una margine di due settimane e un lavoro di circa 12 ore.

1. **Analisi:** Dentro questa fase ho incluso l’inizio della documentazione e l’organizzazione dello spazio di lavoro con GitHub, la discussione col cliente delle richieste e la scrittura del diagramma di use case.
2. **Progettazione:** In questa fase ho incluso la creazione del diagramma di Gantt, la progettazione delle interfacce e una per informarsi e capire come cominciare il lavoro.
3. **Implementazione:** Ho suddiviso questa fase in due:
   1. Creazione del sito in Javascript
   2. Creazione del server in Python (viene incluso il server MySQL)
4. **Documentazione:** La documentazione viene continuamente aggiornata ogni giorno di lavoro.

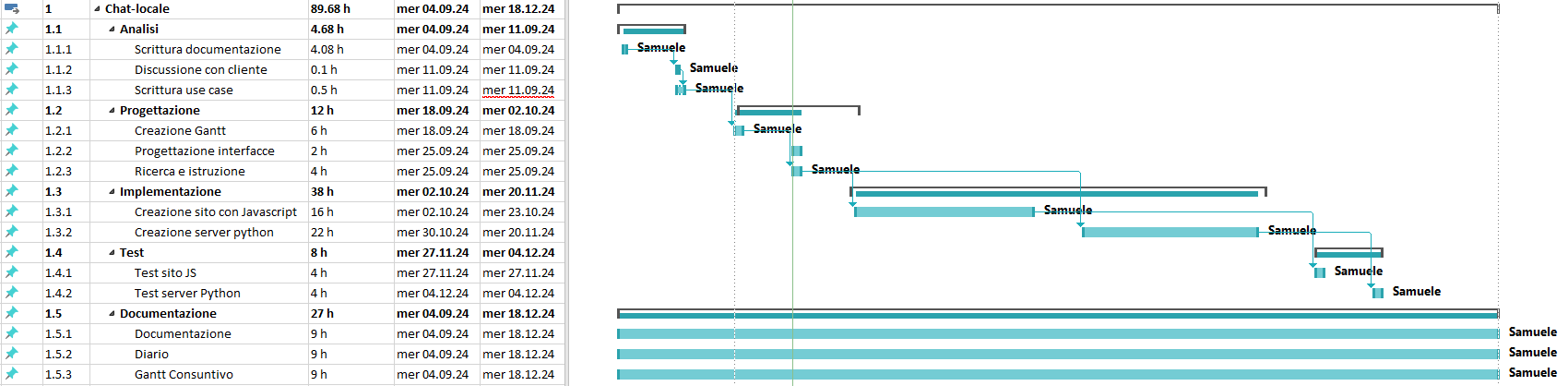
## Analisi dei mezzi

### Software

* Visual Studio Code
* Python
* Javascript
* Vue.js
* MySQL
* React
* Firebase

### Hardware

Un computer è sufficiente finché il traffico è basso.



# 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

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# 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

