Chat con Firebase

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 1

1.8 Analisi dei mezzi 1

1.8.1 Software 1

1.8.2 Hardware 1

2 Progettazione 1

2.1 Design dell’architettura del sistema 1

2.2 Design dei dati e database 1

2.3 Design delle interfacce 2

2.4 Design procedurale 2

3 Implementazione 2

4 Test 2

4.1 Protocollo di test 2

4.2 Risultati test 4

4.3 Mancanze/limitazioni conosciute 4

5 Consuntivo 4

6 Conclusioni 4

6.1 Sviluppi futuri 4

6.2 Considerazioni personali 4

7 Bibliografia 4

7.1 Bibliografia per articoli di riviste: 4

7.2 Bibliografia per libri 4

7.3 Sitografia 5

8 Allegati 5

# Introduzione

## Informazioni sul progetto

Sono un’allieva del terzo anno di informatica alla Scuola d’Arti e Mestieri a Trevano. Ho realizzato questo progetto durante il modulo 306 con la supervisione del docente Geo Petrini. Il progetto è iniziato il 16.09.2021 è si è concluso il 23.12.2021.

## Abstract

È una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

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

Il progetto è pensato per utenti che vogliono chattare fra di loro in gruppi

## Analisi

## Analisi del dominio

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

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Installazione server Firebase |
| **Priorità** | alta |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita una maschera di login |
| **002** | Si dovranno poter immettere nuovi allievi |
| **003** | Dovrà essere possibile la ricerca di allievi |

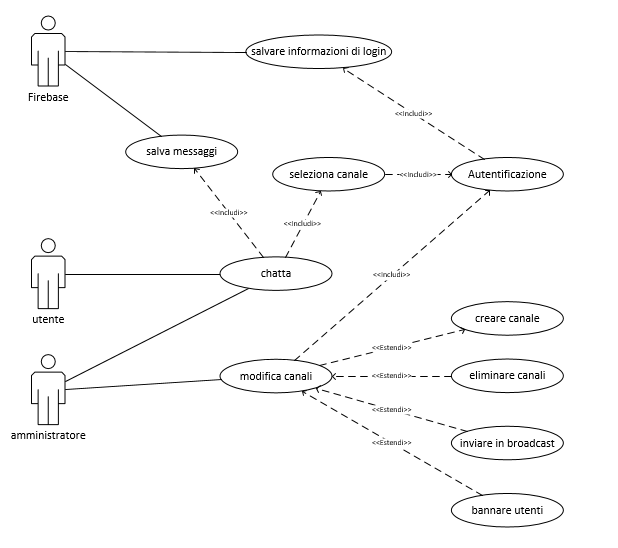
|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Autentificazione degli utenti |
| **Priorità** | 1 |
| **Versione** | alta |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | È necessario salvare nome e password |
| **002** | Le password devono essere salvate in modo sicuro sui database di Firebase |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Gli amministratori devono poter eseguire determinate azioni in più rispetto agli utenti normali |
| **Priorità** | 1 |
| **Versione** | media |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Creare canali |
| **002** | Modificare canali |
| **003** | Eliminare canali |
| **004** | Inviare messaggi in broadcast |
| **005** | Possibilità di bannare utenti |
| **006** | Ban dei messaggi (opzionale) |

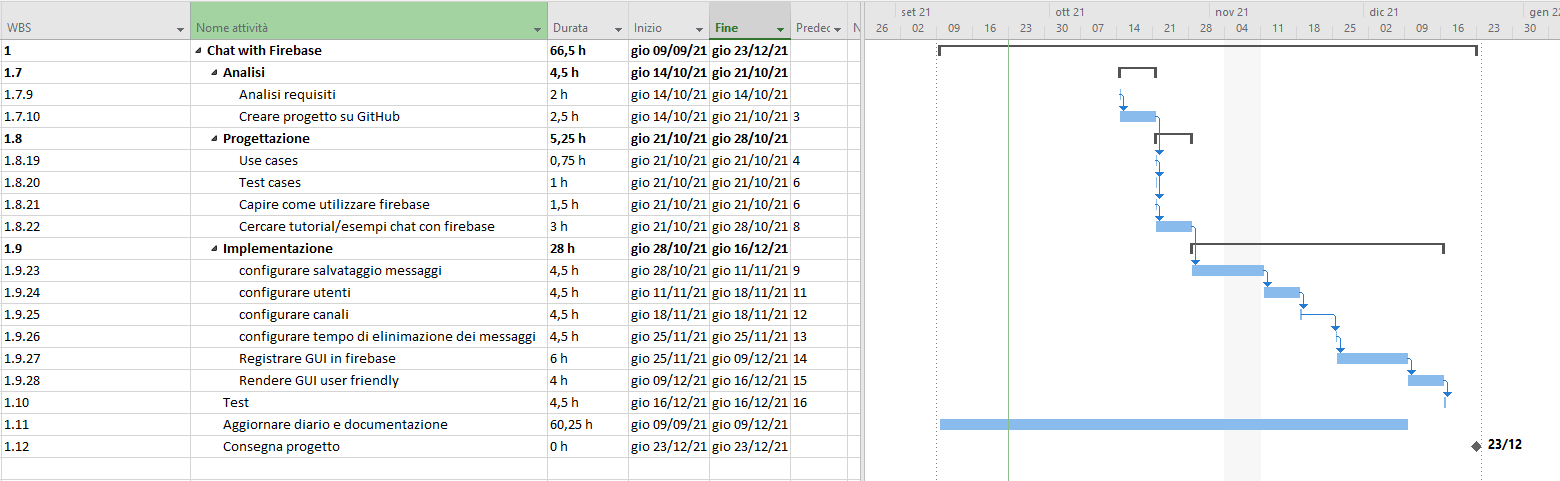
|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Configurazione dei messaggi |
| **Priorità** | bassa |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’applicazione deve gestire i messaggi auto purge |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Deve essere supportata l’intera mappa unicode |
| **Priorità** | bassa |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** |  |

## Use case



## Pianificazione



## 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

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### 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?

# Progettazione

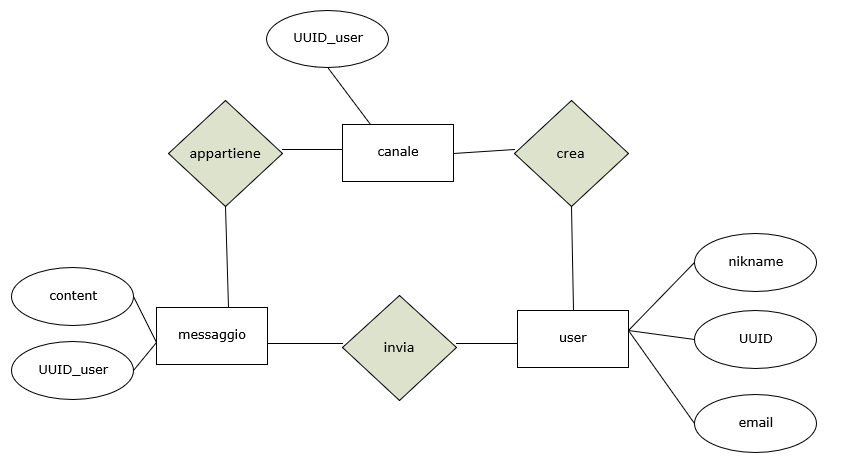
Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## 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



## Design delle interfacce

Wireframe: https://wireframepro.mockflow.com/editor.jsp?editor=on&spaceid=MdlnWCz73h&bgcolor=white&perm=Create&pcompany=C624bea04c2134b5bb452efc4fff21f7a&ptitle=prova&category=bootstrap4&projectid=MUAx2CiyLmb&publicid=4c02aef144d74813bf24e895a7c3d70f#/page/D2fc7f7c9c5fbabeef99897fd0d66f4cc

## 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-01  REQ-01 | **Nome:** | Installazione server Firebase |
| **Descrizione:** | Controllare che il server sia online e che gestisca correttamente le richieste al DB | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | 1. Aprire la console online di firebase 2. All’invio di un messaggio nell’applicazione verificare che nel database appaia il nuovo record | | |
| **Risultati attesi:** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-02  REQ-02 | **Nome:** | Autentificazione degli utenti |
| **Descrizione:** | Controllare che gli utenti di possano autentificarsi nella piattaforma. | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | 1. Aprire la schermata di login 2. Inserire una mail e una password 3. Controllare che sulla console di firebase si veda il nuovo utente | | |
| **Risultati attesi:** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-03  REQ-03 | **Nome:** | Gli amministratori devono avere la possibilità di modificare i canali. |
| **Descrizione:** | Gli amministratori devono poter:   1. Creare canali 2. Eliminare canali 3. Inviare messaggi in broadcast 4. Bannare utenti | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | - | | |
| **Risultati attesi:** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-04  REQ-04 | **Nome:** | Configurazione messaggi |
| **Descrizione:** | L’utente deve poter decidere se i messaggi si devono auto cancellare alla lettura degli altri utenti | | |
| **Prerequisiti:** | Aver eseguito il login nell’applicazione | | |
| **Procedura:** | 1. Autentificarsi con due utenti diversi 2. Inviare un messaggio con l’opzione auto-purge attivata 3. Controllare con l’altro utente che una volta letto il messaggio si cancelli | | |
| **Risultati attesi:** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-05  REQ-05 | **Nome:** | Deve essere supportata l’intera mappa unicode |
| **Descrizione:** | Deve essere supportata l’intera mappa unicode comprendendo anche alcune emoji che ne fanno parte. | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | 1. Accedere all’applicazione con un account 2. Provare ad inviare varie emoji | | |
| **Risultati attesi:** | - | | |

## 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.

# 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
* …