Esempio di documentazione

Sommario

[1 Introduzione 3](#_Toc208687889)

[1.1 Informazioni sul progetto 3](#_Toc208687890)

[1.2 Abstract 3](#_Toc208687891)

[1.3 Scopo 3](#_Toc208687892)

[2 Analisi 4](#_Toc208687893)

[2.1 Analisi del dominio 4](#_Toc208687894)

[2.2 Analisi e specifica dei requisiti 4](#_Toc208687895)

[2.2.1 Spiegazione elementi tabella dei requisiti: 8](#_Toc208687896)

[2.3 Use case 8](#_Toc208687897)

[2.4 Pianificazione 8](#_Toc208687898)

[2.5 Analisi dei mezzi 9](#_Toc208687899)

[2.5.1 Software 9](#_Toc208687900)

[2.5.2 Hardware 9](#_Toc208687901)

[3 Progettazione 9](#_Toc208687902)

[3.1 Design dell’architettura del sistema 9](#_Toc208687903)

[3.2 Design dei dati e database 9](#_Toc208687904)

[3.3 Design delle interfacce 9](#_Toc208687905)

[3.4 Design procedurale 9](#_Toc208687906)

[4 Implementazione 11](#_Toc208687907)

[5 Test 11](#_Toc208687908)

[5.1 Protocollo di test 11](#_Toc208687909)

[5.2 Risultati test 12](#_Toc208687910)

[5.3 Mancanze/limitazioni conosciute 12](#_Toc208687911)

[6 Consuntivo 12](#_Toc208687912)

[7 Conclusioni 12](#_Toc208687913)

[7.1 Sviluppi futuri 12](#_Toc208687914)

[7.2 Considerazioni personali 12](#_Toc208687915)

[8 Glossario 12](#_Toc208687916)

[9 Bibliografia 13](#_Toc208687917)

[9.1 Bibliografia per articoli di riviste: 13](#_Toc208687918)

[9.2 Bibliografia per libri 13](#_Toc208687919)

[9.3 Sitografia 13](#_Toc208687920)

[10 Allegati 13](#_Toc208687921)

# Introduzione

## Informazioni sul progetto

* Allievo: Huynh Anh Nguyen
* Docente: Mariano Fasano
* SAMT Progetti

## Abstract

## Scopo

Lo scopo del progetto (scopi didattici/scopi operativi). Dovrebbe descrivere il mandato, ma non vanno ricopiate le informazioni del quaderno dei compiti (che va invece allegato).

Lo scopo del progetto è quello di realizzare un’applicazione che premetta agli utenti di ascoltare la musica online: In questa maniera può ascoltare la musica personalizzata e creare playlist.

Un ulteriore scopo è quello di imparare ad utilizzare il nuovo framework: React js, Tailwinds, css, node js.

Questo progetto mi premetterà di acquistare nuove conoscenze di lavorare con un Framework dinamico, capire come mettere in funzione le API e approfondire sul lato di server (backend).

# Analisi

## Analisi del dominio

Nel momento ci sono tante applicazioni di questo tipo sul mercato, però ci sono pochi che hanno bella interfaccia. Questo progetto cerca di implementare di creare un UI intuitivo e moderno usando i nuovi frameworks come React js. Questo framework premette di manipolare DOM in maniera semplificata e di rendere una pagina di sito più responsive e dinamico.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisogna di un protocollo di encription/ Creazione di un account |
| **Sotto requisiti** | |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve login nell’applicazione usando un’interfaccia. |
| **Sotto requisiti** | |
| **002** | L’utente deve ottenere già requisiti di ID REQ-01 |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Ascolto musica online (Streaming) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si richiede il streaming di musica, usando API ID:REQ02 |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | L’interfaccia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Crea una interfaccia visuale per gli utenti, dove si può visualizzare brani e playlist |
| **Sotto requisiti** | |
| **001** | Riferisce ID REQ-03 |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Gestione playlist |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Creazione, cancellazione di playlist personalizzate. |
| **002** | Si richiede il streaming di musica, usando API ID:REQ03 |
| **003** | Modifica al REQ-04 |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Salvataggio dei brani |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’utente può visualmente salvare i brani tramite un buttone |
| **002** | Modifica al REQ-05 |
| **003** | Mostra una lista per salvare i brani |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Fermare/avviare la canzone |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’utente deve avere abilità di fermare or partire una canzone |
| **002** | Riferisce ID REQ-03 |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Scegliere la canzone precedente/succesivo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’utente può scegliere di andare prossima canzone o indietro usando due freccette. |
| **002** | Riferisce ID REQ-03 |

|  |  |
| --- | --- |
| **ID: REQ-09** | |
| **Nome** | Ricerca |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’utente può immettere il nome del brano nella sbarra per ricercare la canzone |
| **002** | Riferisce ID REQ-03 |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Cronologia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | L’utente può visualizzare i brani ricercati recentemente |
| **002** | Riferisce ID REQ-03 |
| **003** | Riferisce REQ-09 |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Usabilità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Interfaccia intuitiva, adatta anche a utenti non esperti. |
| **002** | Influenza al REQ-04 |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Accessibilità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Specifica i colori per i daltonici |

### 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

Utente: Interagisce con l’app per scegliere, ascoltare la musica, scoprire nuovi brani, e salvare le preferenze.

Server musicale esterno: fornisce i brani musicali e i dati tramite API.

Applicazione: interfaccia con cui l’utente interagisce.

## 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:

|  |
| --- |
| A  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

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

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

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
* …