Documentazione YourMusic

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: Mirco Gava
* SAM Trevano I3BB – Labo progetti
* Inizio: 12.09.2025 – Fine: 19:12.2025

## Abstract

I decided to do this project because I wanted to get better at programming with objects. I wanted an interface to see, and so I decided that .NET MAUI was my best option.  
The idea for this project was born simply because I’m a single person and I had limited time to finish it. I wanted to work on something that I like, and this is why I’m doing something with music.  
I’m trying to solve a problem that I have with Spotify: I can see my music stats only once a year, so I decided to solve this problem by creating an application that allows you to see your stats whenever you want. I want to make a better version of Apple Music that solves the problem of stats by letting you see them whenever you want, but their interface is not very simple and practical to use.

## Scopo

Lo scopo del progetto è creare un applicativo dove poter ascoltare musica, creare playlist e vedere le proprie statistiche. È una buona occasione per imparare a programmare usando i database di supporto.

Ho scelto di fare qualcosa che come tema centrale avesse la musica in modo che il progetto mi potesse interessare da fare e mi spingesse a lavorare.

Questa è anche una buona occasione per prendere la mano ad usare php.

# Analisi

## Analisi del dominio

Il mio progetto non è nulla di innovativo: esistono già diverse applicazioni per creare playlist o per vedere le tue statistiche (ore di musica ascoltata, brani/artisti preferiti, …), che posso sfruttare per prendere spunto e vedere cosa voglio cambiare.

Potrebbe essere utilizzato da diverse persone e fasce di età: principalmente però il range dell’età penso sia tra i 12-30 anni. Le persone che la userebbero sono principalmente gli utenti di Spotify o YouTube music: possono creare le playlist ma non hanno modo di vedere all’intero della loro app le statistiche quando vogliono: così sono obbligati a usare tools esterni per vederle.

Il tool dovrà funzionare su pc e telefono, che sono i due modi principali dove la gente ascolta musica

## Analisi e specifica dei requisiti

I bisogni del committente sono di avere un applicativo facile da capire e da usare, in modo da rendere l’esperienza gradevole e non far passare gli utenti ad un altro tool. Il tool dovrà essere in grado di poter dire all’utente le sue statistiche, in modo semplice e chiaro.

L’utente dovrà essere in grando di aggiungere i brani che vuole alla sua playlist e di poterla denominare come vuole.

La sicurezza del prodotto può essere bassa, anche se ci fosse una fuga di dati non ci saranno dati sensibili ma solo musica all’interno dell’applicativo.

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Log in |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Gli utenti devono avere un account per usare la musica |
| **002** | Per entrare devono mettere username e password (oppure email e password) |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Creazione/utilizzo playlist |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessità di un’interfaccia pulita e facile da capire |
| **002** | Si potrà inserire un titolo alla playlist |
| **003** | Si potrà inserire una bio alla playlist |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Implementazione database |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Devo poter registrare canzoni |
| **002** | Si dovranno salvare anche artisti, album e genere |
| **003** | A dipendenza di come vorrò farlo verranno salvate anche le playlist |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Aggiunta brani alla playlist |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | aggiungere brani in base al titolo |
| **002** | Quando si aggiunge un brano bisogna anche inserire: album, genere e artista |
| **003** | In caso ci sia si possono mettere le feature |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Salvataggio brani, artisti, album e genere |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Salvare tutti i dati sul database |
| **002** | Non permettere id salvare due volte la stessa cosa |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Ascolto brani |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si dovranno poter ascoltare i brani che sono stati aggiunti |
| **002** | Si potranno anche mettere in pausa |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Statistiche user |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessità di un’interfaccia pulita e facile da capire |
| **002** | Si dovranno vedere album, artisti e generi preferiti |
| **003** | Ci saranno dei grafici per vedere meglio le statstiche |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Ricerca brani |
| **Priorità** | 4 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Barra di ricerca dove inserisci titolo e ti trova il brano |
| **002** | Si potranno cercare anche artisti, album e genere |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: YourMusic

**Priorità**: I requisiti marcati sono le mie priorità assolute: la cosa più importante è avere un’interfaccia pulita e funzionante: quindi la prima cosa che farò è creare il sistema delle playlist e la possibilità di poter aggiungere brani e poterli ascoltare (livello 1): poi pulirò l’interfaccia e aggiungerò le statistiche (livello 2) e poi penserò alla ricerca dei brani e alla pulizia finale di tutta l’interfaccia (livello 3). Quando avrò finito con questo potrò pensare agli extra

**Versione**: Questa è la prima versione dei requisiti, potrebbero cambiare più in avanti

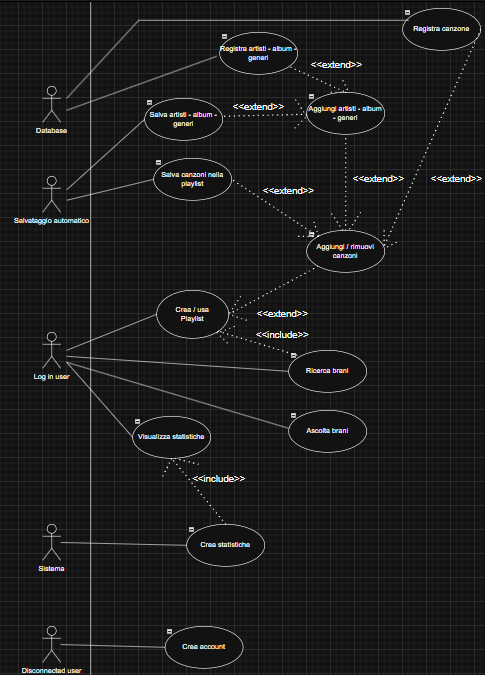
**Note**: Per ora non ci sono note

**Sotto requisiti**: Requisiti più importanti che avranno la priorità nello sviluppo dell’applicativo

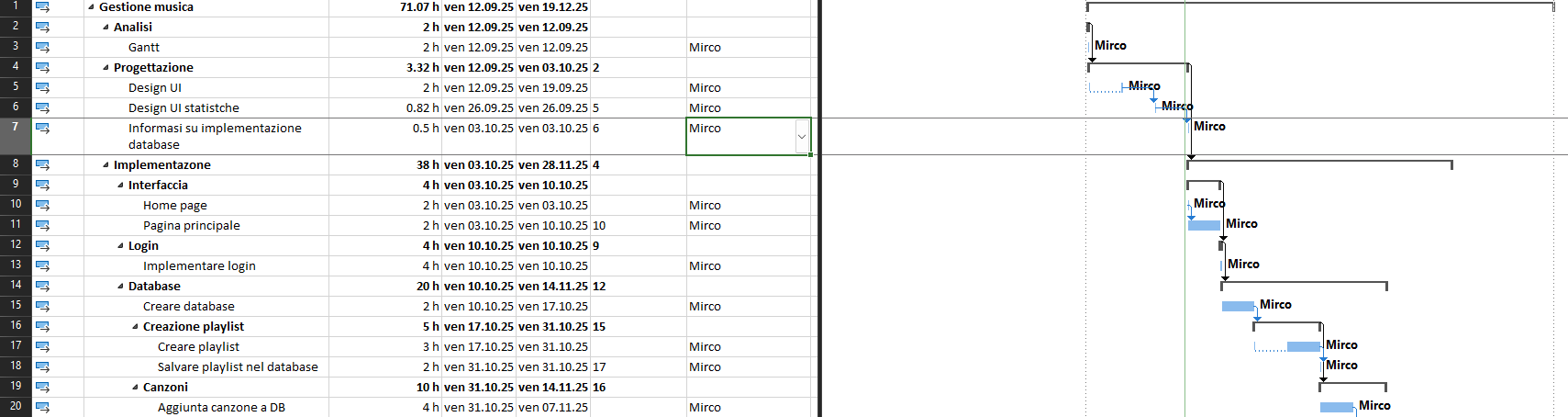
## Use case

C’è user, che potrà guardare le statistiche, aggiungere canzoni, creare playlist e ascoltare musica.

Il sistema si occuperà di creare le statistiche mentre il database e il salvaggio automatico di memorizzare le canzoni



## Pianificazione





## Analisi dei mezzi

Ho a disposizione vecchi progetti ed internet

### Software

HTML – visual studio 2022 – Mysql – Javascript – CSS - php

### Hardware

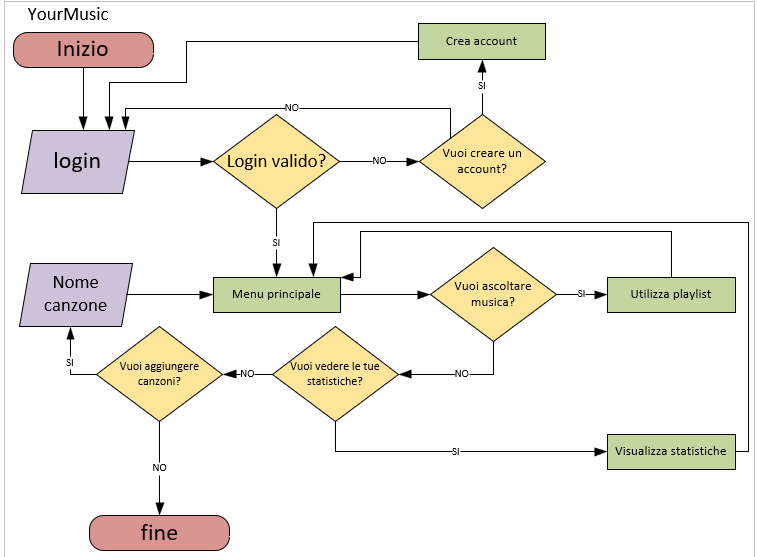
Pc dell:

* Scheda grafica: 4GB
* RAM: 32GB
* Processore: 12th Gen Intel(R) Core(TM) i7-12700

# Progettazione

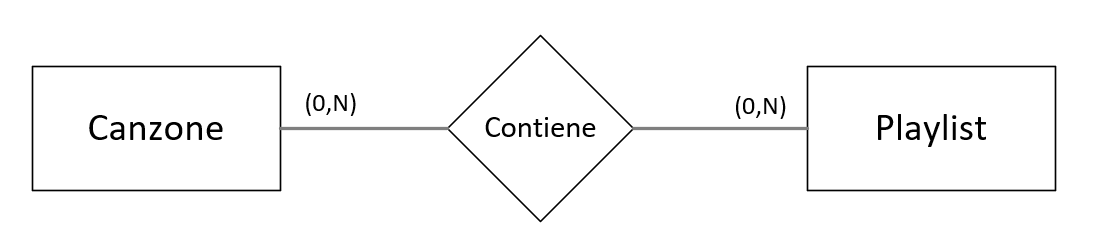
Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

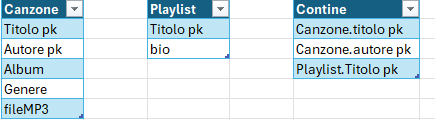
## Design dell’architettura del sistema



## Design dei dati e database

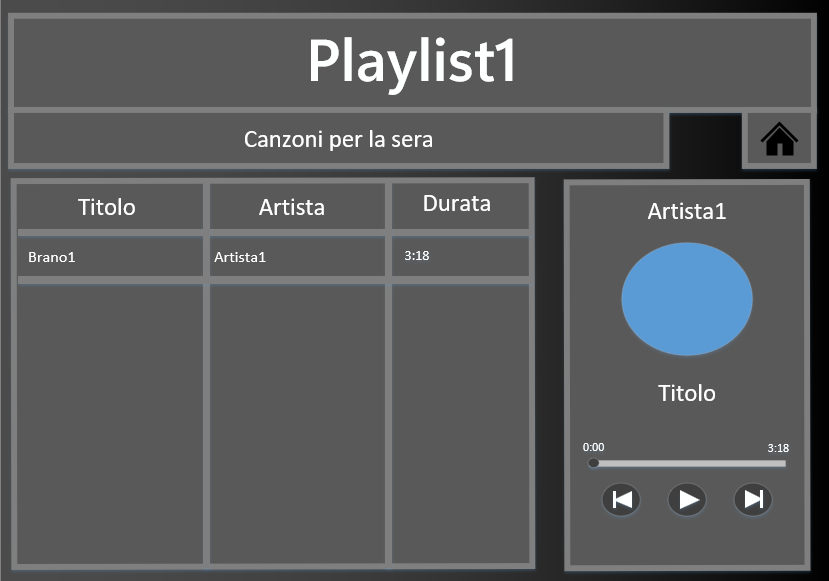
Nel mio progetto farò uso dei database per memorizzare le canzoni:

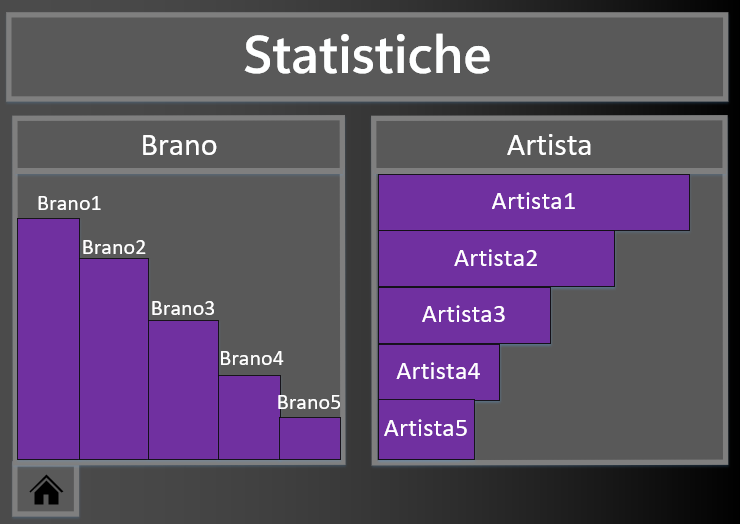




## Design delle interfacce







## Design procedurale

# 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
* …