LoopLab

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 7

1.7 Pianificazione 8

1.8 Analisi dei mezzi 9

1.8.1 Software 9

1.8.2 Hardware 9

2 Progettazione 10

2.1 Design dell’architettura del sistema 10

2.2 Design dei dati e database 10

2.3 Design delle interfacce 10

2.4 Design procedurale 10

3 Implementazione 11

4 Test 11

4.1 Protocollo di test 11

4.2 Risultati test 12

4.3 Mancanze/limitazioni conosciute 12

5 Consuntivo 12

6 Conclusioni 12

6.1 Sviluppi futuri 12

6.2 Considerazioni personali 12

7 Bibliografia 12

7.1 Bibliografia per articoli di riviste: 12

7.2 Bibliografia per libri 12

7.3 Sitografia 13

8 Allegati 13

# Introduzione

## Informazioni sul progetto

* Allievo: Eros Marucchi
* Docente Responsabile: Maurizio Di Florio
* Perito 1: X
* Perito 2: X
* Data di inizio: 27.01.2025
* Data di fine: 04.04.2025
* Luogo di sviluppo: Scuola Arti e Mestieri di Trevano
* Sezione scolastica: Informatica 4° anno

## Abstract

The objective of the project is to develop a web application that allows users to create and manage custom soundtracks through an interactive and intuitive interface. The project is to be completed in 168 hours, ensuring all specified requirements are met.

The operational goal of the application is to provide users with a dynamic interface that allows them to visually compose music using a central circle as a reference point. Users can add circles of different sizes, each representing a sound, and connect them through lines that define timing. The size of the circles corresponds to the volume, and their interaction determines the playback of sounds.

The system will include a default sound library, with the ability for users to upload their own sound files and apply effects such as reverb. Users will be able to record, save, and download their compositions in mp3 format.

The application will be built using Node.js for the backend with Handlebars as the template engine, JavaScript for the frontend, and MySQL for data management. A user authentication system will be implemented, differentiating between standard users, who will be able to create soundtracks, and administrators, who will be able to manage CRUD functionalities.

In conclusion, the final product will offer an innovative platform that blends visual and audio interaction, making soundtrack creation accessible even to users with no prior musical experience.

## Scopo

L’obiettivo del progetto è sviluppare un’applicazione web che permetta agli utenti di comporre, modificare e gestire le proprie colonne sonore in modo semplice, interattivo e accessibile a tutti, indipendentemente dall’esperienza musicale.

L’interfaccia dell’applicazione offre la possibilità di aggiungere suoni tramite cerchi di dimensioni variabili. Questi cerchi, che rappresentano i diversi suoni, sono connessi da linee che definiscono il tempo della composizione. La dimensione dei cerchi determina il volume del suono e il movimento degli stessi lungo le linee consente una visualizzazione dinamica della musica che si sviluppa nel tempo. Ogni suono interagisce visivamente con gli altri, creando una composizione musicale fluida e coinvolgente.

La piattaforma include una libreria di suoni predefiniti, con la possibilità per gli utenti di caricare i propri file audio e personalizzare i suoni con effetti aggiuntivi, come il riverbero, per arricchire le composizioni. Una volta completata la composizione, gli utenti possono salvare il progetto nel proprio profilo e scaricare il file audio in formato mp3, rendendo la produzione musicale facile da gestire e condividere.

## Analisi del dominio

L’applicazione è progettata per rendere la creazione musicale accessibile a chiunque, anche senza esperienza pregressa in musica.   
Il contesto di utilizzo include utenti educativi, creativi e appassionati di musica, che cercano un’interfaccia semplice ed interattiva per comporre e gestire colonne sonore personalizzate.   
Esistono prodotti simili nel mercato, come *Soundation* e *GarageBand*, questi spesso richiedono una certa conoscenza musicale, mentre il mio progetto punta su un approccio visuale e intuitivo che rende la composizione sonora facile e coinvolgente.   
Gli utenti possono essere sia hobbisti, studenti o educatori, con competenze musicali di base o assenti, e hanno bisogno di strumenti per creare e personalizzare suoni in modo dinamico.   
L'applicazione si propone come una risorsa educativa e creativa, senza richiedere conoscenze teoriche avanzate in musica.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita un’interfaccia grafica semplice ed intuitiva per ogni tipo di utente. |
| **Sotto requisiti** | |
| **001** | SI necessita una maschera di registrazione |
| **002** | Si necessitano gli input per inserire le informazioni richieste |
| **003** | Si necessita di una mail di conferma |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita un’interfaccia grafica semplice ed intuitiva per ogni tipo di utente. |
| **Sotto requisiti** | |
| **001** | SI necessita una maschera di login |
| **002** | Si necessita della possibilità di rimpostare la password |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione colonne sonore |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita un’interfaccia grafica semplice ed intuitiva per ogni tipo di utente. Queste funzionalità saranno solo per gli utenti di tipo user. |
| **Sotto requisiti** | |
| **001** | SI necessita una schermata con la possibilità di disegnare la propria colonna sonora |
| **002** | Si necessita la possibilità di avviare la propria composizione tramite il tasto play |
| **003** | Si necessita la possibilità di stoppare la propria composizione tramite il tasto pausa |
| **004** | Si necessita la possibilità di registrare la propria composizione tramite il tasto registra |
| **005** | Si necessita la possibilità di salvare le composizioni create |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Multilingua |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | SI necessita una la possibilità di cambiare la lingua dell’applicativo |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | User e Admin |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita il tipo di utente user |
| **002** | Si necessita il tipo di utente admin |

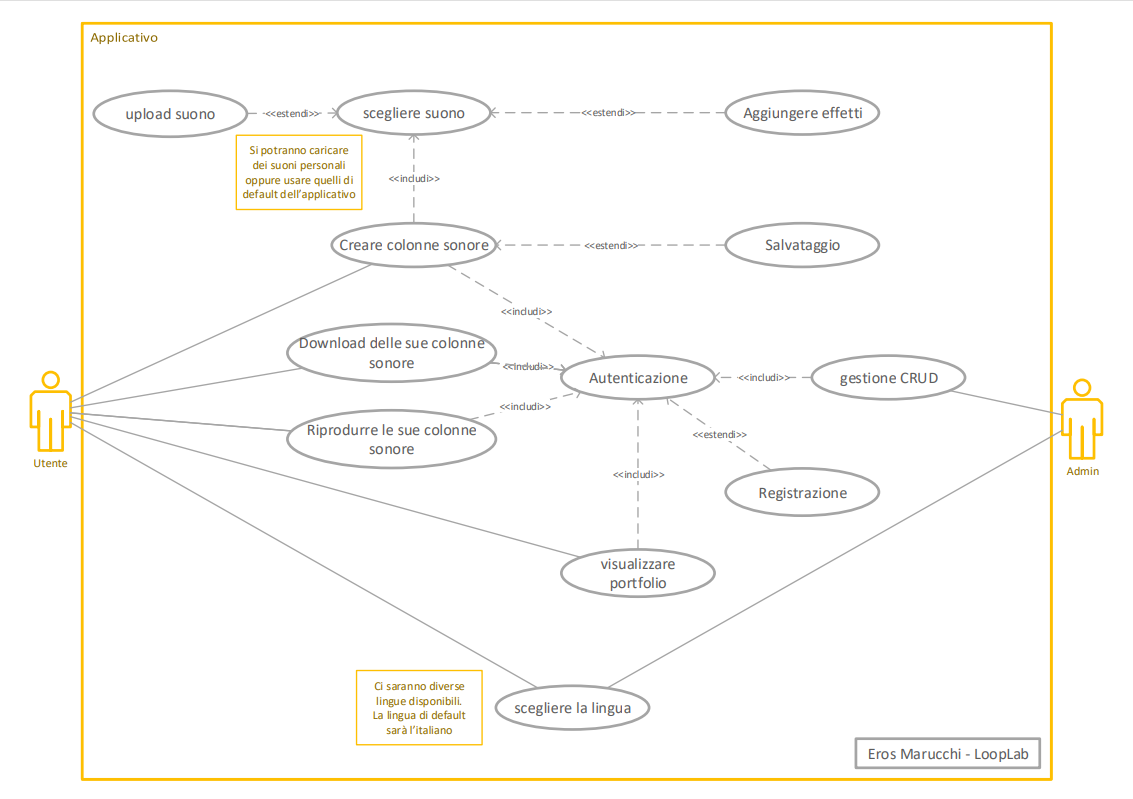
|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Gestione suoni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita un’interfaccia grafica semplice ed intuitiva per ogni tipo di utente. |
| **Sotto requisiti** | |
| **001** | Si necessita una libreria di suoni di default presenti nell’applicativo |
| **002** | Si necessita la possibilità di caricare dei suoni personali |
| **003** | Si necessità la possibilità di aggiungere degli effetti ai suoni |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Portfolio |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita un’interfaccia grafica semplice ed intuitiva per ogni tipo di utente. |
| **Sotto requisiti** | |
| **001** | Si necessita il fatto di avere un portfolio per ogni utente |
| **002** | Si necessità di poter scaricare le proprie colonne sonore |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Gestione CRUD |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Queste funzionalità saranno solo per gli utenti di tipo admin |
| **Sotto requisiti** | |
| **001** | Si necessita il fatto di poter gestire la parte CRUD dell’applicativo. |

## Use case

Descrizione…

Ecco il diagramma di Use Case del progetto:

## 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 2: Esempio di diagramma di Gantt. |

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

## Analisi dei mezzi

Per fare questo progetto mi è stato fornito un PC scolastico.

### Software

* Visual Studio Code 1.78.2, editor di testo per sviluppare l’applicativo web.
* Google Chrome 116.0.5845.188, browser per testare e testare l’applicativo.
* FireFox 110.0, browser per testare l’applicativo.
* Microsoft Edge116.0.1938.81, browser per testare l’applicativo.
* GanttProject, per fare il GANTT.
* Microsoft Visio Professional 2019, per fare lo Use Case

### Hardware

* Nome dispositivo: 427-19
* Nome completo del dispositivo: 427-19.CPT.local
* Processore: Intel Core i7-9700 CPU 3.00 GHz
* RAM installata: 32.0 GB
* Scheda Video: NVIDIA GeForce RTX 2060
* Tipo sistema: Sistema operativo a 64 bit, processore basato su x64
* Versione di Windows: Windows 10 Enterprise, versione 22H2

# 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.

# 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
* …