Trifonov,Dobri D.V.

d.trifonov@student.fontys.nl

Music Player Project

Individual project for Semester 3, containing music  
 storing capabilities and online search integration

Requirements document

Music Player

Table of Contents

[Version Table 2](#_Toc84625337)

[Product Description 3](#_Toc84625338)

[Product Backlog 4](#_Toc84625339)

[Architecture diagrams 7](#_Toc84625340)

[C1 architecture diagram 7](#_Toc84625341)

[C2 Architecture diagram 8](#_Toc84625342)

[C3 Architecture diagram 9](#_Toc84625343)

[UML diagram 10](#_Toc84625344)

[Report 11](#_Toc84625345)

[Solid Principles Guaranteed 11](#_Toc84625346)

[Single Responsibility 11](#_Toc84625347)

[Open/Closed 11](#_Toc84625348)

[Liskov Substitution 11](#_Toc84625349)

[Interface Segregation 11](#_Toc84625350)

[Dependency Inversion 11](#_Toc84625351)

[Software used 11](#_Toc84625352)

[Backend 11](#_Toc84625353)

[Frontend 11](#_Toc84625354)

[Database Layer 11](#_Toc84625355)

# Version Table

|  |  |  |
| --- | --- | --- |
| Version | Date | Info |
| 1.00 | 17.09.2021 | Initial Documentation |
| 1.10 | 03.10.2021 | Updated and Reformed the User Stories |
| 2.00 | 08.10.2021 | Added Architecture Diagrams, Report, Solid Guarantee, Software used |

# Product Description

This project is dedicated to making a Web Application with the goal of synchronized music storage. The application is to be made simple with the least amount of buttons and configuration needed. It should be able to save music from local storage to a database/cloud storage system. All uploaded content should be synchronized between all logged in devices on a per user basis. Users should be able to source music online from within the app. The app will be connected to other 3rd party services that will provide the internet music sourcing through their publicly available API’s (Youtube, Spotify). The design is to be kept minimalist and all frontend aspects of should be responsive. The app should be usable from any devices’ browser.

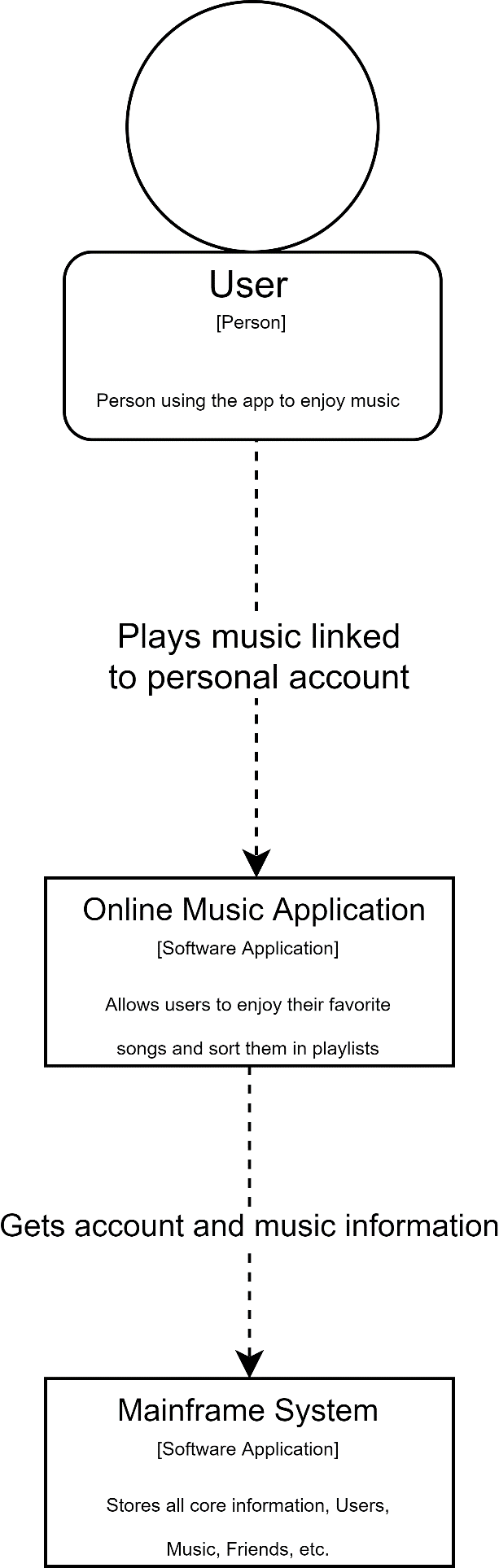
# Product Backlog

User story, Priority (User Story Count - 1), Estimation (Story Points), Acceptance criteria

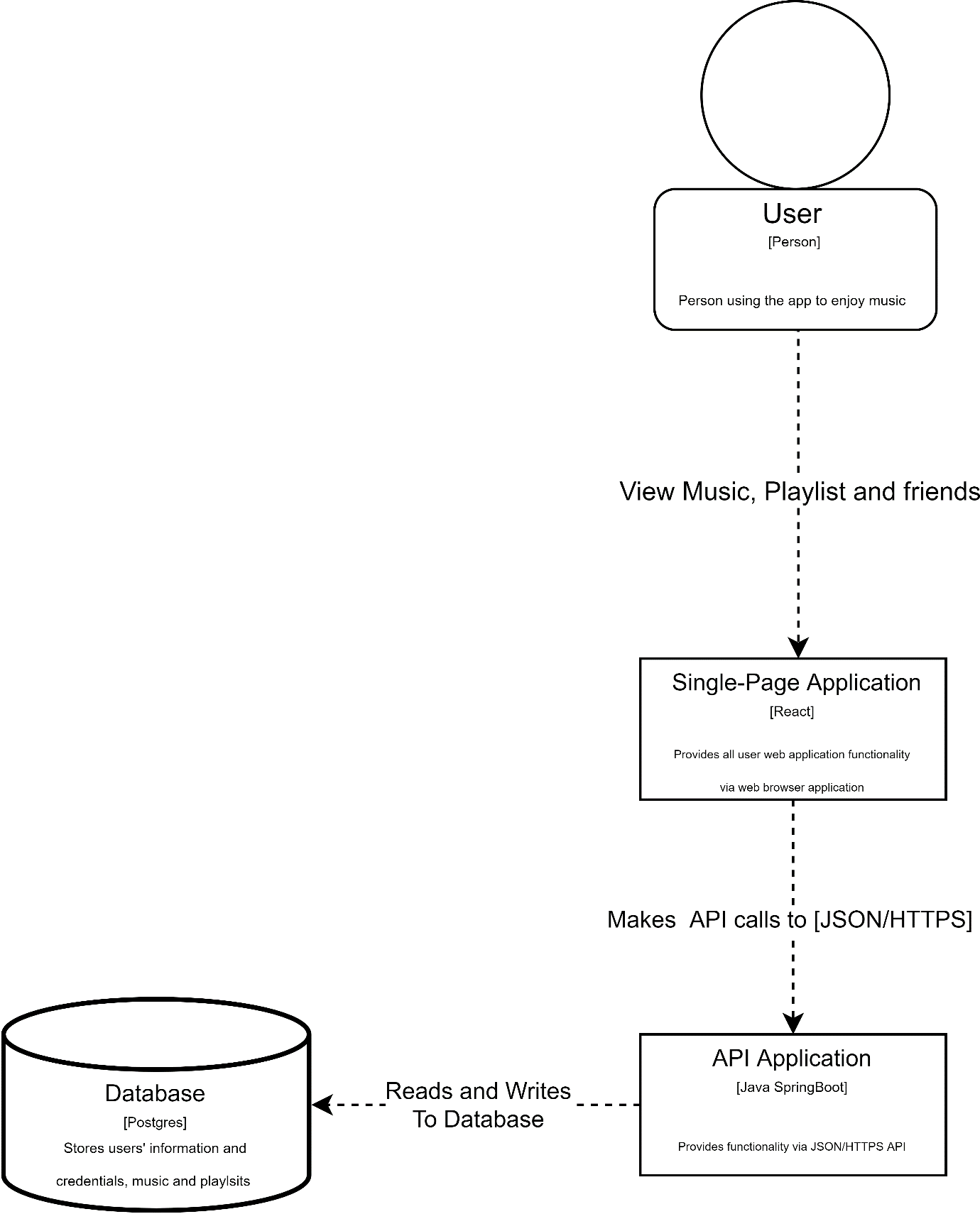
|  |  |  |  |
| --- | --- | --- | --- |
| **Task ID** | **User Story** | **Priority** | **Estimation** |
| 1 | *As a User I want to be able to have an account, so I can use additional features of the app*  **Acceptance Criteria:**   * User provided credentials (Email, Username, Password) * Valid user provided credentials | **5** | **1** |
| 2 | *As a User I want to be able to play music from the app, so I can listen to my favourite songs*  **Acceptance Criteria:**   * The app should be able to play audio files | **1** | **5** |
| 3 | *As a User I want to have media controls, so I can skip, go back or pause certain songs*  **Acceptance Criteria:**   * A fast forward button that skips to the next queued song * A pause and play button that stops the currently playing music or resumes it if stopped * A rewind button that plays the previous song in the queue | **3** | **3** |
| 4 | *As a User I want a simple indicator of the current song, so I can check what is playing*  **Acceptance Criteria:**   * GUI indicator that displays all available metadata about the current playing song | **4** | **1** |
| 5 | *As a User I want to be able to add my downloaded songs, so I can play them through the app*  **Acceptance Criteria:**   * Upload music from local device to application | **2** | **5** |
| 6 | *As a User I want to be able to search through my music collection, so I can find my songs faster*  **Acceptance Criteria:**   * GUI search bar that filters displayed songs | **6** | **1** |
| 7 | *As a User I want to be able to search for music within the app, so I can easily find more songs*  **Acceptance Criteria:**   * GUI search bar that shows result of inputted data * Online connectivity to 3rd party services that offer music | **7** | **13** |
| 8 | *As a User I want to be able to have playlists, so I can sort my music collection*  **Acceptance Criteria:**   * Have multiple folder/directories containing different but not necessarily unique combination of music | **8** | **2** |
| 9 | *As a User I want to be able to synchronize my music between my devices, so I can play my music on any device*  **Acceptance Criteria:**   * Keep user uploaded content store in an online database * Give access to user’s content to any logged in device of the account’s holder | **15** | **5** |
| 10 | *As a User I want to have a recommendation page, so I can find new music tailored to my taste*  **Acceptance Criteria:**   * Specialized page that displays music that is not in your playlist collections yet | **12** | **21** |
| 11 | *As a User I want to be able to find other users on the app, so I can see my friend’s accounts*  **Acceptance Criteria:**   * GUI search bar that displays all users with matching name to the inputted data | **13** | **2** |
| 12 | *As a User I want to be able to see other users playlists, so I can see what other people like listening to*  **Acceptance Criteria:**   * Show playlists and music collections of users on their profile | **14** | **3** |
| 13 | *As an Admin I want to be able to see all Users’ information, so I can navigate through user data easier*  **Acceptance Criteria:**   * Specialized GUI page with user list | **9** | **1** |
| 14 | *As an Admin I want to be able to see all Users’ uploaded content, so I can make sure no inappropriate content is being uploaded*  **Acceptance Criteria:**   * Ability to filter out user uploaded content * Ability to preview user uploaded content | **10** | **2** |
| 15 | *As an Admin I want to be able to have Delete and Update functionality for User content, so I can manage uploaded content easier*  **Acceptance Criteria:**   * Admin GUI option menu that allows for removing user content from the application * Admin GUI option menu that allows for updating user content from the application | **11** | **5** |

# Architecture diagrams

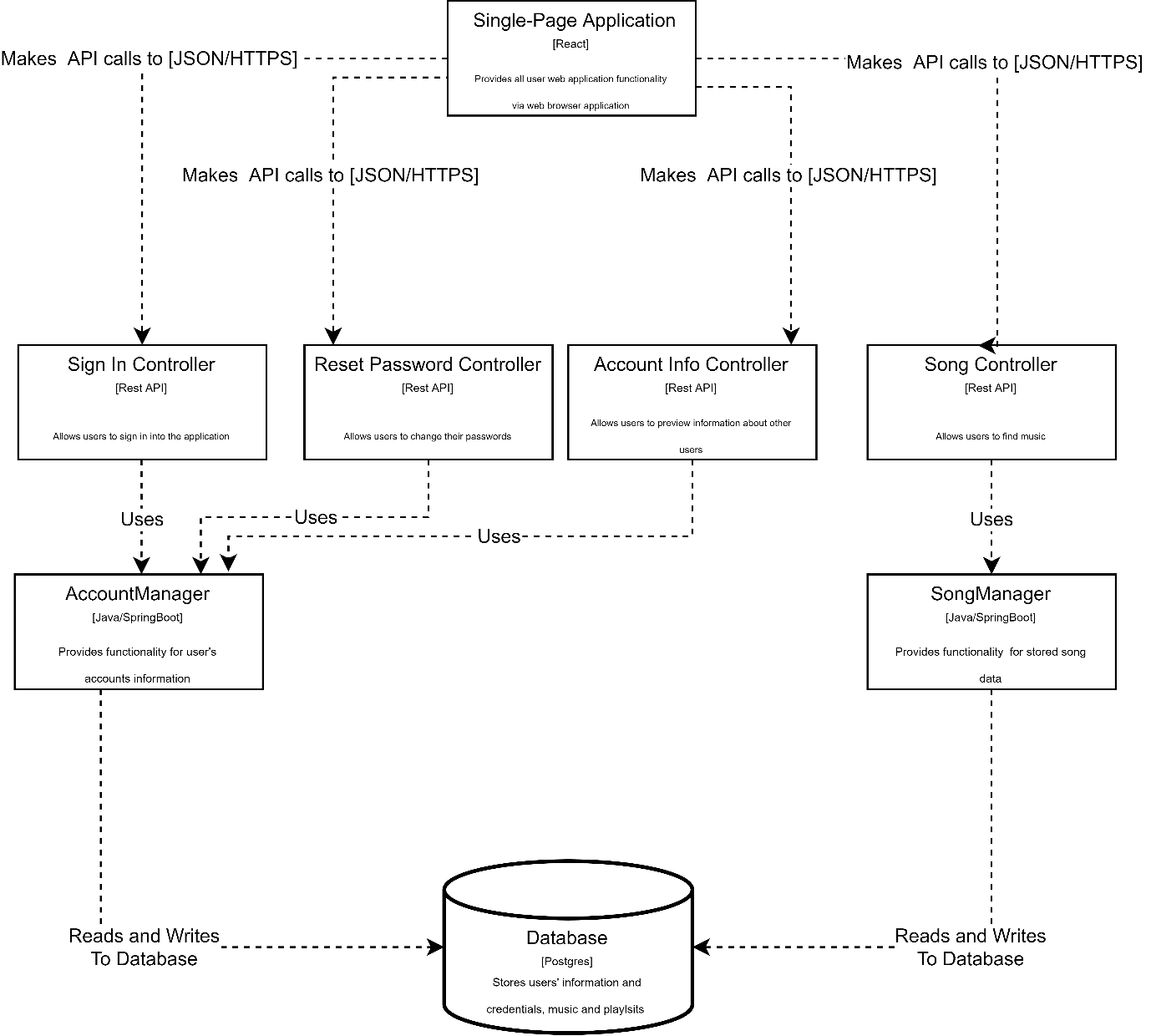
## C1 architecture diagram



## C2 Architecture diagram



## C3 Architecture diagram



## UML diagram

# Report

Duo to not finding the C4 architectural diagram particularly useful in the case of this project I have decided to not make one for now. Instead I will be using a UML diagram in which I will be better able to present the design for the application. The C4 diagram however will be later generated from the written code after the development of the application has ended.

# Solid Principles Guaranteed

## Single Responsibility

Classes are responsible for only one thing. This allows to have fewer unit tests and have less dependencies throughout my project. All classes are single-purpose, small and easy to understand.

## Open/Closed

I will ensure this by adding more to the base application instead of changing the base working code.

## Liskov Substitution

I will ensure this by making every class that implements an interface be able to substitute any reference throughout the code that implements that same interface.

## Interface Segregation

I will ensure this by splitting any interface that becomes too large within the project

## Dependency Inversion

I will ensure this by having implementations that would be easily changeable without altering any complex code.

# Software used

## Backend

For the backend I will use Spring Boot because it lets me create a standalone application that runs on it's own without relying on an external web server such as Tomcat into my app.

## Frontend

For the frontend I will use React because it allows me to create reusable UI components. React is also fast, scalable, and simple.

## Database Layer

The database layer will be a point of research and testing within my project duo to the uncertainty of reading, writing and storing big files, such as audio files, into a database.