UE20CS303: SOFTWARE ENGINEERING

SOFTWARE REQUIREMENTS SPECIFICATION

MusicWreck

VERSION 1.0

Prepared By-

PES1UG20CS331 - RENITA KURIAN

PES1UG20CS334 – RICHA SHAHI

PES1UG20CS337 – RIMJHIM SINGH

PES1UG20CS344 – RIYA JHA

INTRODUCTION

1.1 DOCUMENT PURPOSE

The product whose software requirements are specified in this document is MusicWreck.

The purpose of this document is to present a detailed description of the product, MusicWreck. This document is intended to

- Explain the purpose and features of the product, MusicWreck
- The constraints under which the product must operate
- How the product would respond to different users' requests?

The document's primary goal is to help the reader get a better understanding of the project. The document is intended for the developers of the software, the end users of the product who have been identified in the later sections, and to the professors who would review the project.

1.2 Product Scope

The software being developed is Machine Learning based Music Recommendation System. The product would recommend songs to the user based on her preferences of genres and songs she like. The product would –

- Recommend songs to the user based on genres
- User can also select certain songs from the dropdown option to get a personalised playlist
- Would be provided with a Spotify playlist with all these songs

The points specified above would make a competent music recommendation system for the user according to her likelihood of musical tracks.

1.3 Intended Audience and Document Overview

1.3.1 Intended Audience:

This document is primarily intended for the:

- Developers of this software
- Software engineers who would work on further development of the project
- The professors who would review the document and finally,
- Clients that is novice or professional event managers, volunteers.

1.3.2 Document Overview:

The first chapter, that is the Introduction section of the document is intended to introduce the reader to the product, MusicWreck.

The second chapter, Overall Description section of SRS document provides an overview of the overall functionality of the product. It describes the informal requirements.

The third chapter, Specific Requirements section, of SRS document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

The second and the third chapter of the document describe the same software product, but are intended for different audiences and thus use different language.

1.4 Definitions, Acronyms and Abbreviations

1	Employer	Employer is an individual who has contacted the event organiser.
2	Event Manager	Event Manager is an individual who is responsible for the whole event and can view the entirety of the event being planned on the software. He/She is usually the lead event organizer.
3	HTTPS	HTTPS stands for Hypertext Transfer Protocol Secure. This protocol is a widely used communications protocol for secure communication over a computer network, with especially wide deployment on the Internet.
4	Spotify	Organisation that's going to provide all the dataset and playlist interface for the user.
5	SRS	SRS stands for Software Requirement Specification. It is a document that completely describes all of the functions of a proposed system and the constraints under which it must operate.
6	Team Head	Team head is an individual who is responsible for all the actions undergoing under his/her team.
7	UI	UI stands for User Interface. It is defined as the space where interaction between humans and machines occurs.
8	View	View means to display and look at data on screen.

9	ML model	The Machine Learning Model being used for the recommendation system.
10	Dataset	The dataset being used for building the recommendation system taken kaggle.
11	API	Application Programming Interface to use HTTP requests to use Spotify Developers Features.

1.5 Document Conventions

Formatting Conventions:

- The font style for the headings of each section is Arial Bold and the font size is 15.
- The font style for the headings under each section is Arial Bold and the font size used is 12.
- For the remainder of the document, the font style is Arial and the font size is maintained at 11.
- Italics has been used to indicate comments.
- The text is single spaced and margins are maintained at 1'' separation.

1.6 References and Acknowledgments

1.6.1 References:

- https://docs.google.com/document/d/e/2PACX-1vRIW_TuZ6z0ASjAoxgJgmzjGYLCDx019tKvphaTwK_Za7fnMKywUuXI0-s5wr0nQI_gprm6J6y7L9rL/pub
- https://jinja.palletsprojects.com/en/3.0.x/templates/
- www.python.com
- https://developer.spotify.com/documentation/web-api/quick-start/

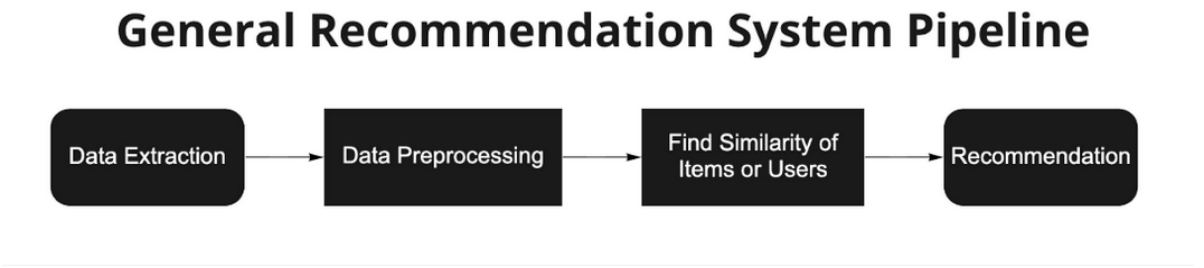
2. Overall Description

2.1 Product Perspective

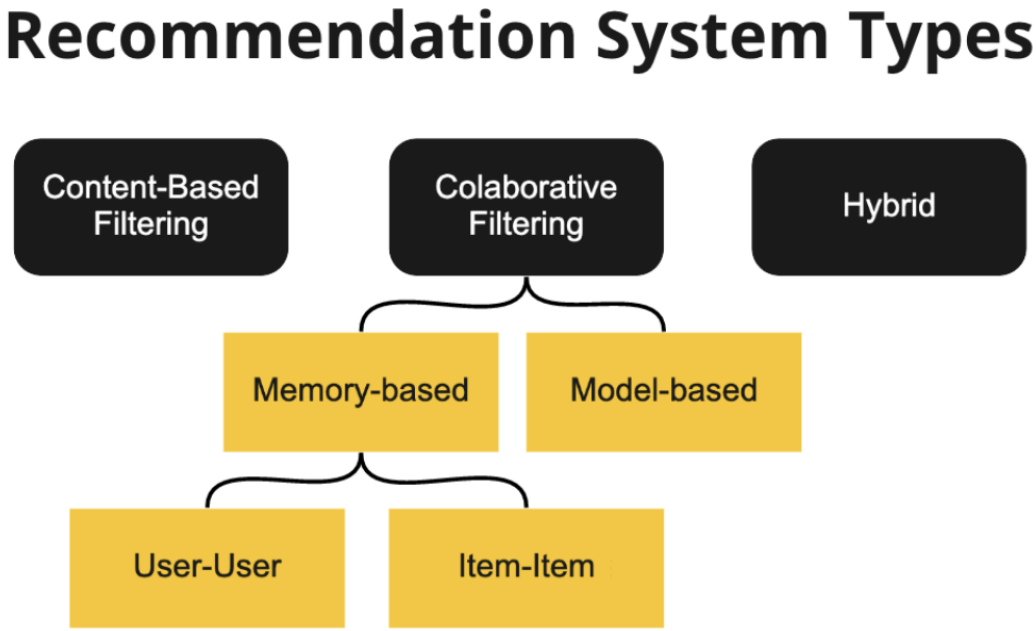
The product MusicWreck is a music recommendation system product. It mainly is trained on a Spotify dataset. The user is suggested with song recommendation after she picks her most liked songs. By using music recommender system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of the music that has been heard previously. In a nutshell, recommendation systems ***recommend things that the***

people might like based on your own watch history or you and friends watch history as a collective.

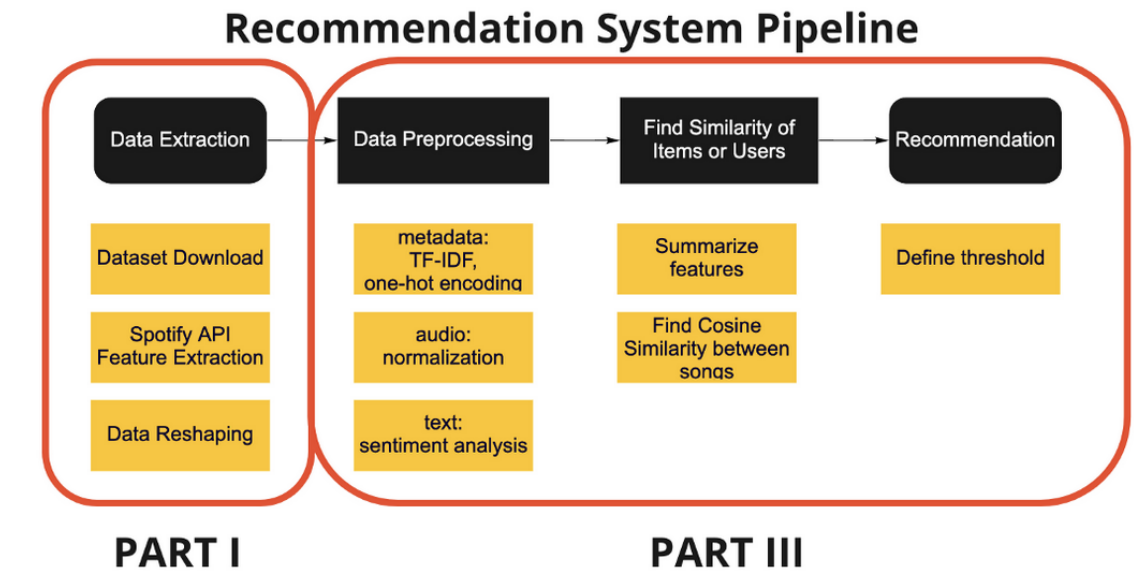
2.2 Product Functionality



General Recommendation System Pipeline. (Image by author)



Different Types of Recommendation System. (Image by author)



Recommendation System Pipeline for this project. (Image by author)

2.3 Users and Characteristics

The system will support four types of user privileges:

- Developer
- User
- Employer

The various users that we expect the software to be used by are:

1.	Developers	Developer is an individual who is responsible for all the actions going on in the project.
2.	User	Has to enter her choices and the playlist is accordingly displayed
3.	Employer	Owner once the software is sold.

2.4 Operating Environment

The software will be designed to work on any version of Windows, Linux (kernel 2.7 and above) and Mac platform. The software is completely web based and runs on popular web browsers namely firefox, chrome, internet explorer (IE8 and above). These web browsers are preferred since they support HTML.

2.5 Design and Implementation Constraints

We have to design different pages for different types of users such as developers and target audience. The implementation part is yet to be done. But, we have a clear picture as to how our pages would look. The communication protocol will be http and smtp. There are a number of tools which can be used for its implementation. The maximum number of users at a time is yet to be decided we will try to deploy it at an online hosting website or make an installable python executable.

2.6 User Documentation

No tutorials have been developed as of now.

2.7 Assumptions and Dependencies

Assumptions

The user is familiar with internet and web based software like social networking sites.
The browsers which the user is using is either Google Chrome 10.0 and above or Mozilla Firefox 4.0 and above.

Dependencies

- Spotify Dataset (public data)
- Spotify Developers API
- Flask
- Other module and packages are going to be decided during implementation

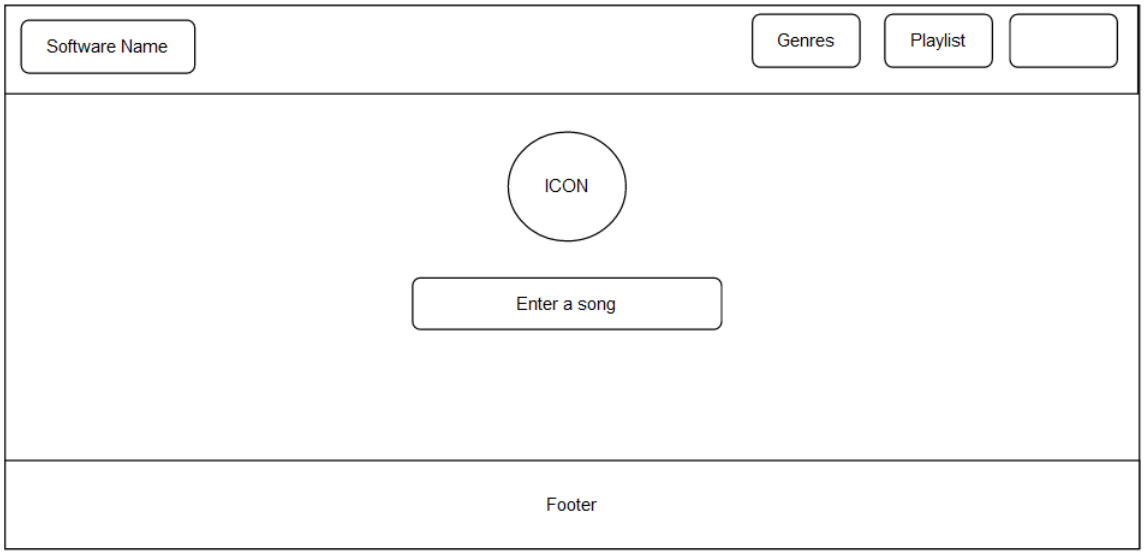
3. Specific Requirements

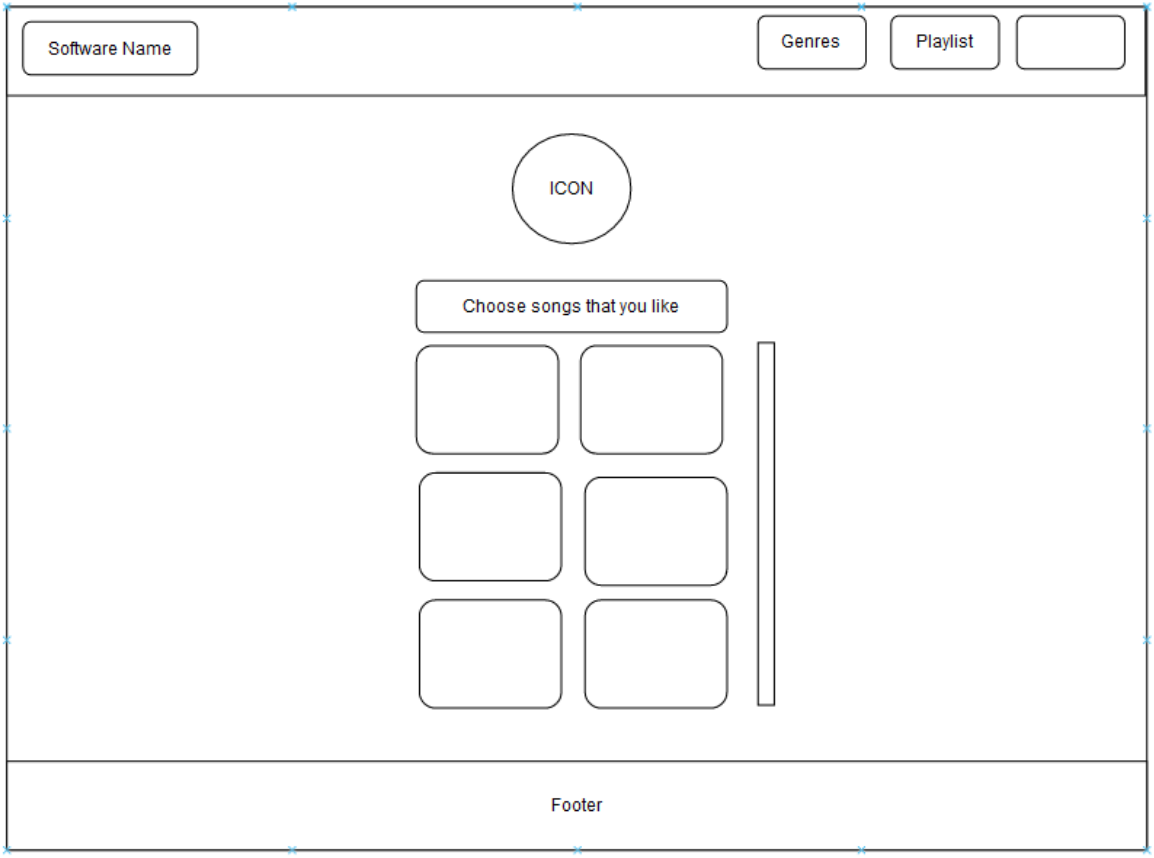
3.1. External Interface Requirements

3.1.1. User Interfaces

The user interface design is simple and clear. Users can simply choose songs from dropdown list or enter songs in the search bar. Personalised playlist based on genres are also available.

USER VIEW:





Developers View:

BASH or CONSOLE view. (Does not support GUI)

3.1.2. Hardware Interfaces

Not applicable.

3.1.3. Software Interfaces

The software is operating system independent. It would run on Linux, Windows and Mac.

3.1.4. Communications Interfaces

A web browser is a basic necessity for the software to be deployed. Playlist is shared via SMTP protocol or User has to enter Spotify credentials to get it.