SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT FOR
MUSIC RECOMENDER SYSTEM

By

ABHIMANYU OLLA

Contents

1. INTRODUCTION

1.1 Problem Definition

Websites and applications that offer their users or customers an item or a social element, have been trying to recommend them relevant content in order to their items/elements which they are interested in. Increasing the time that user spends on the website and increasing the interest of the user to the items in the website are the main reasons why the recommender systems are being used. Accuracy and time-efficiency are the most common problems of recommender systems. We will be trying to design an accurate and fast algorithm which will solve these problems.

People often complain about irrelevant recommendations of the websites they are using. User sometimes even complains about websites like Amazon or Netflix even though they are considered as having the best recommender systems. Since most of the global websites that offer their users an item, the problem can be considered as a worldwide problem.

Some of the global websites that are using a recommender system have been using the insourcing method to get the system. However, outsourcing is the most common way among most websites. There are many software companies and university labs that are working on recommender systems. Websites like Amazon, Netflix, Google, Facebook have their own recommender systems. Also, there are researchers in universities like Stanford who are working on recommender systems. Even competitions have been organised to get the most accurate and fast recommender system worldwide. Netflix Prize is the best known competition. It began on October 2, 2006 and on 21 September 2009, the grand prize of US$1,000,000 was given to the BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06%.

For people who are totally unaware of how a recommendation system works, we can explain evaluation of the effectiveness of the recommender systems which is one of the most important issues to understand the topic. As it is explained in the website of Creighton University, recall, precision and DCG (Discounted Cumulative Gain) are the common metrics to assess the quality of the recommendation method. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage too. DCG measures the usefulness, or gain, of a document based on its position in the result list.

The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks. These are the basic concepts related to calculating recommendations.

1.2 Purpose

The purpose of this document is to give a detailed description of the requirements for "Music Recommender".

It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the customer company for their approval and a reference for development of the system.

1.3 Scope

Music Recommender is a music recommendation system, which provides users songs which they may like, based on the songs that they previously listened. Every logged in user should have access to the recommender system. The system will go through the songs that user previously listened and rated, then according to those information it should provide songs to the user. The project's main aim is to provide accurate music recommendations to the user. This project is beneficial for the user as well as for certain companies. For users, they may find songs that they may like without consuming time and even they can encounter new songs which they like from the recommendations. For the company, they make the website more attractive, so they draw more users to the website and the system makes the users of the website spend more time online.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition
User	Recommender system users.
Music Recommender	An information filtering system that seeks to predict the preference that user would give to a song.
JSON	JavaScript Object Notation.
Std	Standard

1.5 References

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- http://tinman.cs.gsu.edu/~raj/8711/sp13/djondb/Report.pdf
- Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35
- Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. (2000). "Application of Dimensionality Reduction in Recommender System A Case Study",
- Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, David M. Pennock (2002). "Methods and Metrics for Cold-Start Recommendations". Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002). New York City, New York: ACM. pp. 253–260. ISBN 1-58113- 561-0. Retrieved 2008-02-02.

1.6 Overview

The remainder of this document includes six chapters and appendixes. The second chapter of the SRS describes the general factors that affect the product and its requirements. It includes product perspective, product functions, constraints, assumptions and dependencies.

The third chapter contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. In the fourth chapter information domain of the software is explained. The fifth chapter describes the behaviour of the software. In the sixth chapter the planning of the software development team is explained. The team structure, an estimation of a schedule of development processes, and process models are shown.

2. OVERALL DESCRIPTION

2.1 Product Perspective

2.1.1 System Interfaces

Recommender system is a sub-component of a larger music system. It uses the larger system's data and makes recommendations to users. Therefore, the recommender system needs to be compatible with larger system. The data used by the recommender system, which includes user profiles and music types, has to be the same type with the one collected in the website. Nosql database will be used to work on the database.

2.1.2 Hardware Interfaces

Since the web portal does not have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system on the web server.

2.1.4 Software Interfaces

The web component communicates with the database in order to get the user rating logs of the larger system. The communication between the database and the web component consists of operation concerning both reading and modifying the data, while the communication between Music Recommender and the website consists of sending a JSON object from the recommender to the website to be displayed on the webpage.

2.1.5 Communication Interfaces

The communication between the different parts of the system is important, since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal.

2.2 Product functions

With Music Recommender, the user will be able to see music recommendations made especially for him/her. The recommendations will be based on the user's previous actions and the actions of the other users who have a similar taste of music as the user, who will get the recommendations. Since the recommendations are made based on the user, they are likely to be unique.
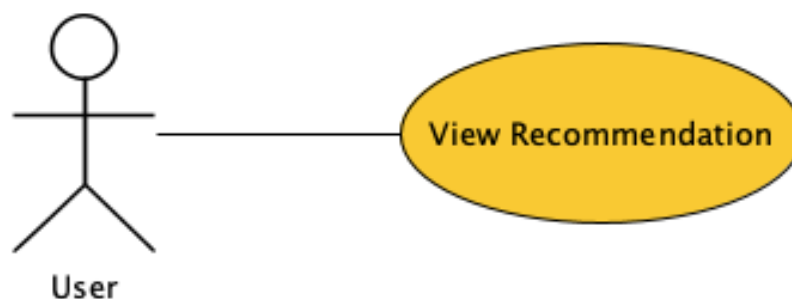

FIGURE 1. USE CASE RECOMMENDATION

2.3 Constraints

The Internet connection is a constraint for the application. Since the application fetches data from the server over the Internet, it is crucial that there is an Internet connection for the application to function.

The web portal will be constrained by the capacity of the database. Since the database is shared with the larger system, it may be forced to queue incoming requests and as a result, increase the time it takes to fetch data.

The computers must be equipped with web browsers such as Internet explorer. Execution time for the algorithm should take no longer than one second. All Java code shall conform to the Java Code Convention standards. Music Recommender will be a sub-component of Music website. Users shall be required to log in to the website to get recommendations. Music Recommender shall be available to users 99.9% of the time when the Music website is available. The system must be operational for each user. It also needs to give unique recommendations for each user.

2.4 Assumptions and dependencies

It is assumed that at least daily one million logs will be provided for the system's use and the past 60 days' logs will be provided. Another assumption is that the user has a web browser and a capable hardware in order to launch the website.

It is also assumed that larger system is able to provide necessary requirements for the recommender system to run.

3. SPECIFIC REQUIREMENTS

3.1 Interface Requirements

The user needs to click the link to the website. Then he/she needs to register to the system by providing a password and an email, otherwise he/she won't be able to use the Recommender System. Then, to benefit from the Recommender System he/she needs to be active on the website by listening and downloading music, adding his/her favourites to the list or sharing them on Facebook/Twitter.

3.2 Functional Requirements

This section outlines the use cases for each user registered to the website(larger system).

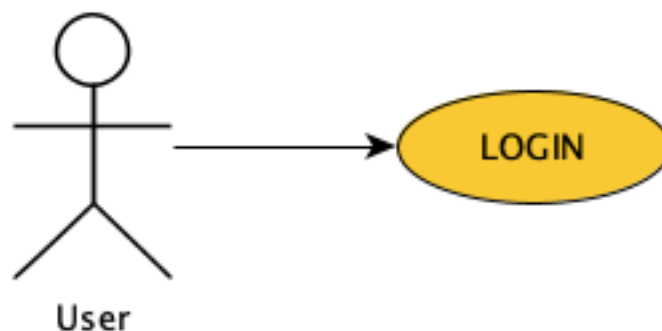### 3.2.1 Login Use Case

Use case: Login

Diagram:

FIGURE 2. LOGIN USE CASE

Brief Description: The user logins to the system to see what recommendations he/she has. Initial Step-By-Step Description

1. The user needs to open the website.
2. The user needs to type his/her username and password to login.

3.2.2 View History Use Case
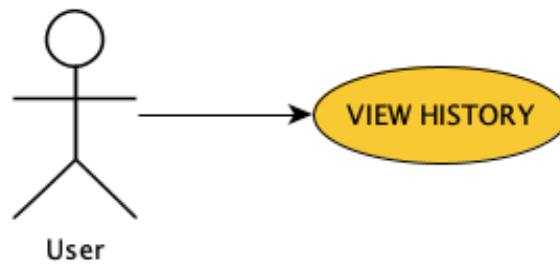
Use case: View History

Diagram:

FIGURE 3. VIEW HISTORY CASE

Brief Description: The user can listen to music on the website online. Initial Step-By-Step

Description

1. The user needs to open the website.

2. The user needs to login to the system.

3. After being logged in, the user can see his/her past actions.

3.2.3 View Recommendations

Use Case Use case: View recommendations
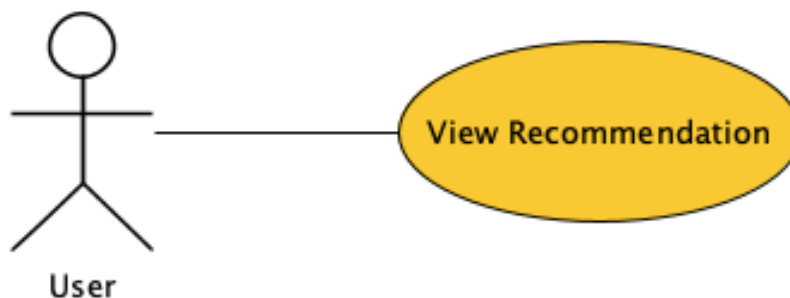
Diagram:

FIGURE 4. VIEW RECOMMENDATION USE CASE

Brief Description: The user will be able to view recommendations next to music lists on the

webpage. Initial Step-By-Step Description

1. The user needs to open the website.

2. The user needs to login to the system.

3. After listening music, the user will be able to view music recommendations selected using

    by their past actions and preferences.

3.3 Non-functional Requirements

3.3.1 Performance requirements

As a larger system, the website has a monthly traffic of over 4 million users. Since recommendation system is planned to be designed for the use of every user in the larger system, it is easy to say that this system will have a monthly traffic of over 4 million users. Every day, over 3 million songs are listened on the larger system. However, not all 3 million songs are logged by the website or not all logged information will be given for the use of the system. By the company, daily 1 million logs will be given to be handled. User information of the past 60 days will be the reference point of the system, that's why 60 million logs need to be handled in order to implement the whole recommender system.

3.3.2 Design constraints

In the implementation process of this system, Java Programming Language will be the main development language. Since Java is selected to be the main development language, Java Programming Language Code Convention published by Oracle is chosen as a standard for the development process of the system. In the process of the documentation of the system, IEEE standards will be used and UML standard will be used while designing the diagrams. Since this system will be a part of much larger system, it must be portable to this larger system. That's why portability is one of the most important attributes of this system. Since the larger system is a website that has the potential of increasing its number of users, user traffic and number of songs, this system needs to be scale up with the website in the correct order. Therefore, scalability must be the number one attribute that system will have.

4. BEHAVIOURAL MODEL AND DESCRIPTION

4.1 Description for software behaviour

This subsection describes the major events and states of our software. When the user open the webpage at first, user will see a login screen. In this screen user will required to provide 18 username and password. If the information is valid the user will be able to move on to the main page. On this page user will be able to open music lists and listen music. Every song that the user listened will be added to the user history. By checking the user history, every user will be able to view recommendations. Users will be able to logout of the system.
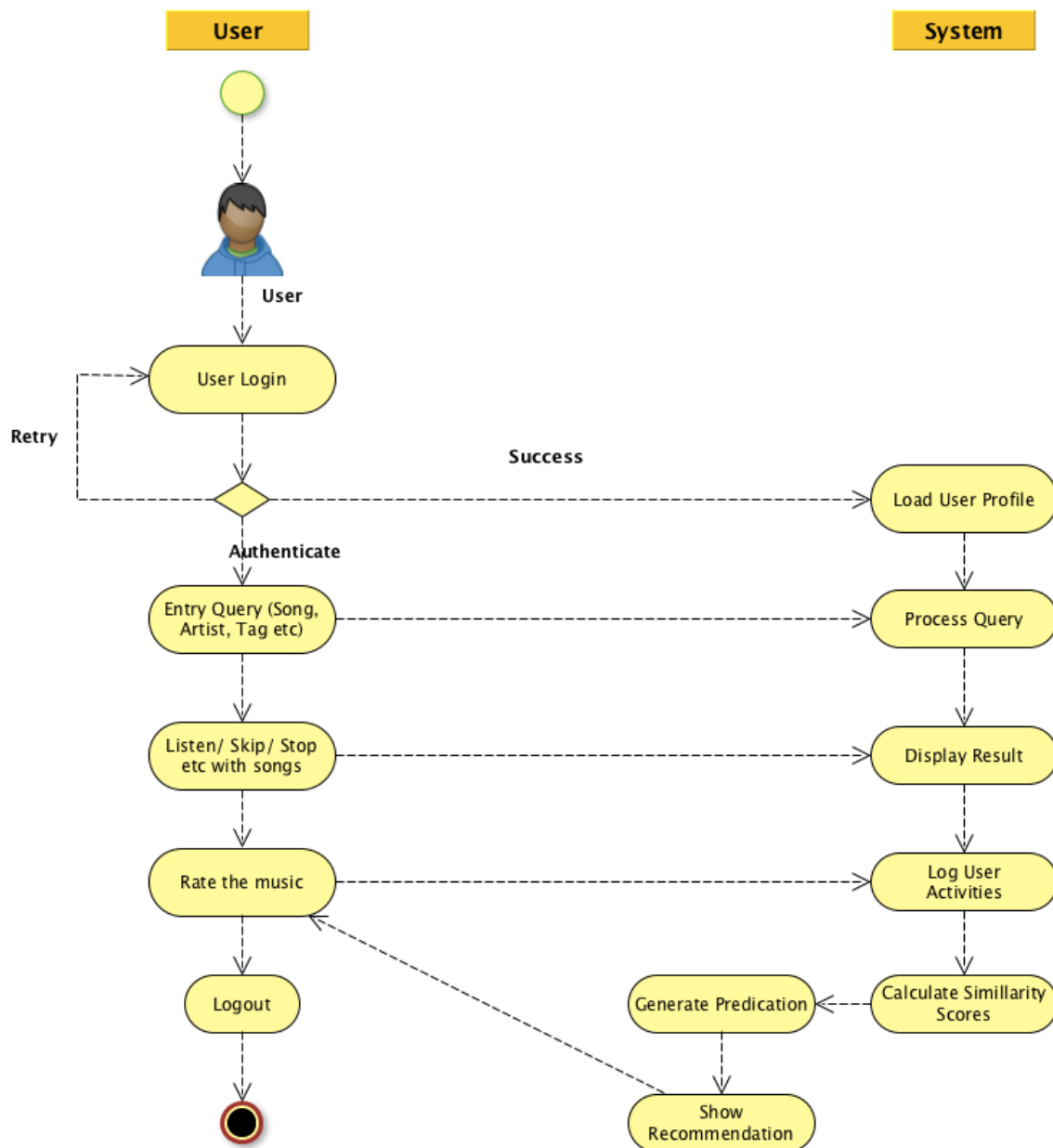
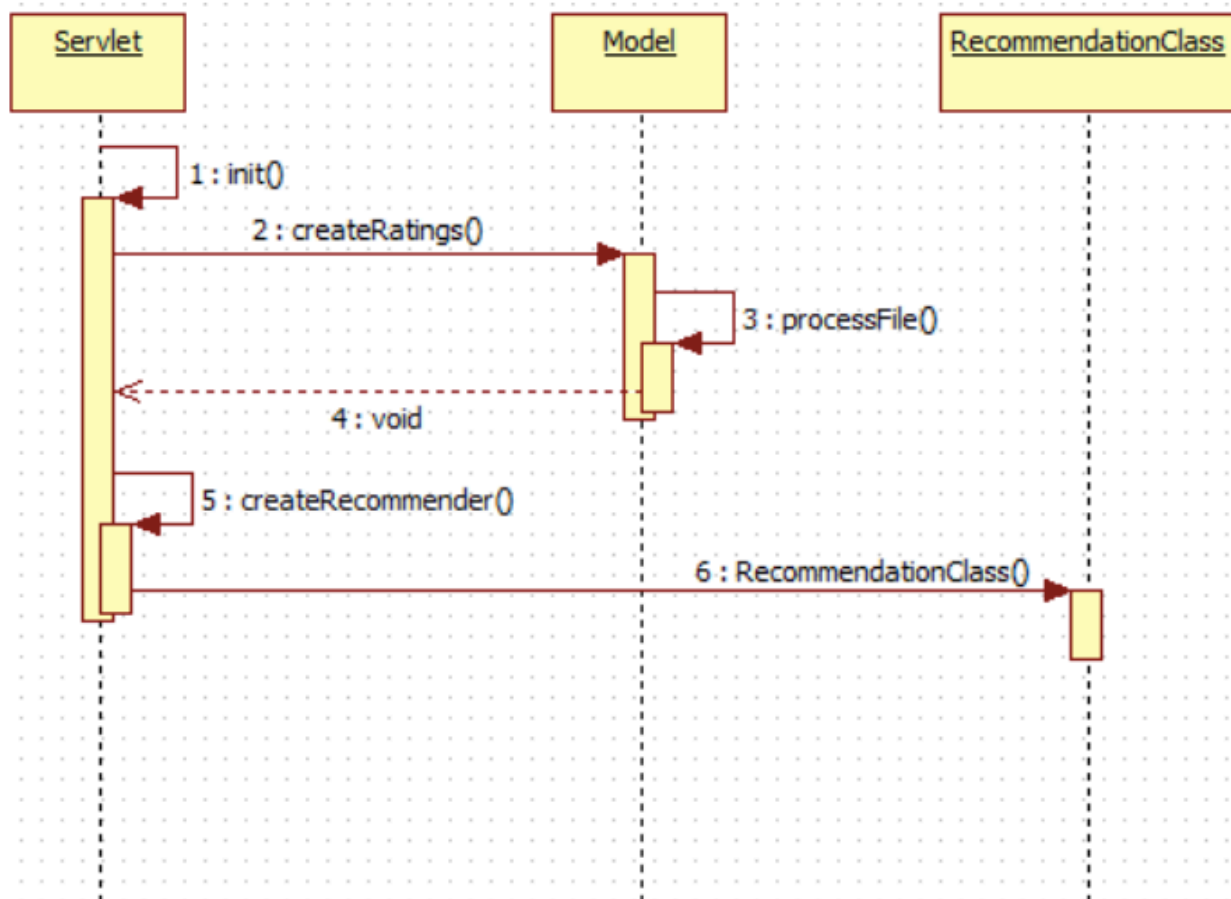## 4.2 State Transition Diagrams

FIGURE 5. STATE TRANSITION DIAGRAM
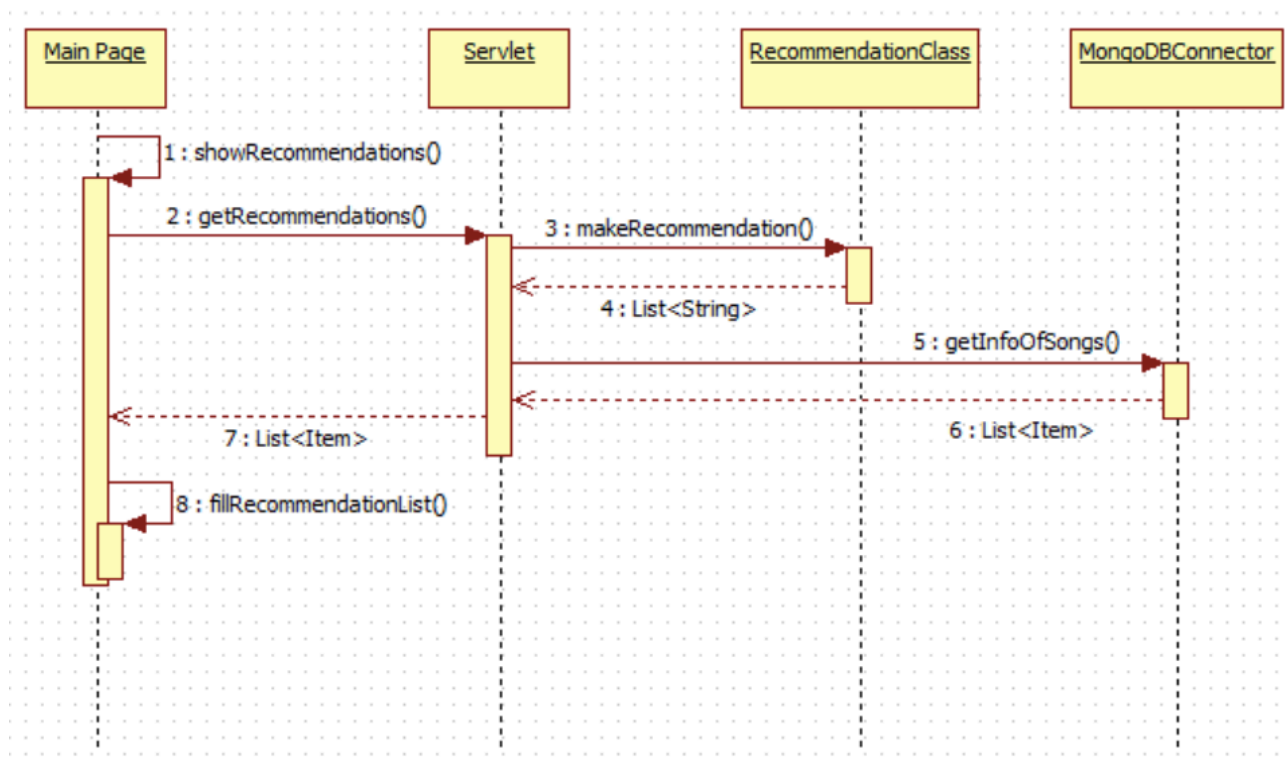
11

FIGURE 6. GENERATE RECOMMENDATION SEQUENCE DIAGRAM

FIGURE 7. VIEW RECOMMENDATION SEQUENCE DIAGRAM

5. PLANNING

5.1 Estimation

| Task	Start Date	End Date	Duration
Gather Requirements	1-Aug-19	10-Aug-19	10
UI Design	11-Aug-19	25-Aug-19	15
Coding	26-Aug-19	14-Sep-19	20
Unit Tests	15-Sep-19	20-Sep-19	6
System Test	21-Sep-19	23-Sep-19	3
Bug Fixes	24-Sep-19	30-Sep-19	7
Improvements	1-Oct-19	10-Oct-19	10
Final Testing	11-Oct-19	15-Oct-19	5
Documentation	16-Oct-19	25-Oct-19	9

Music Recommendation System
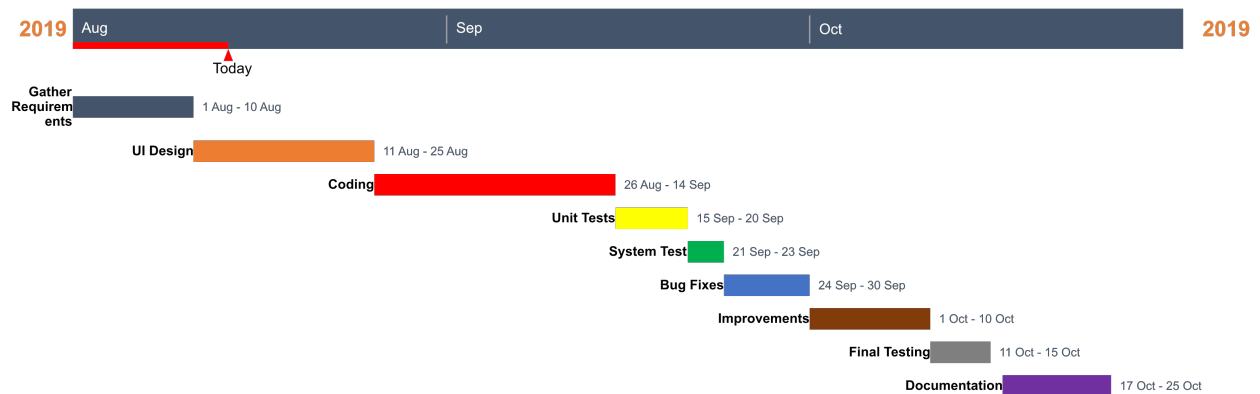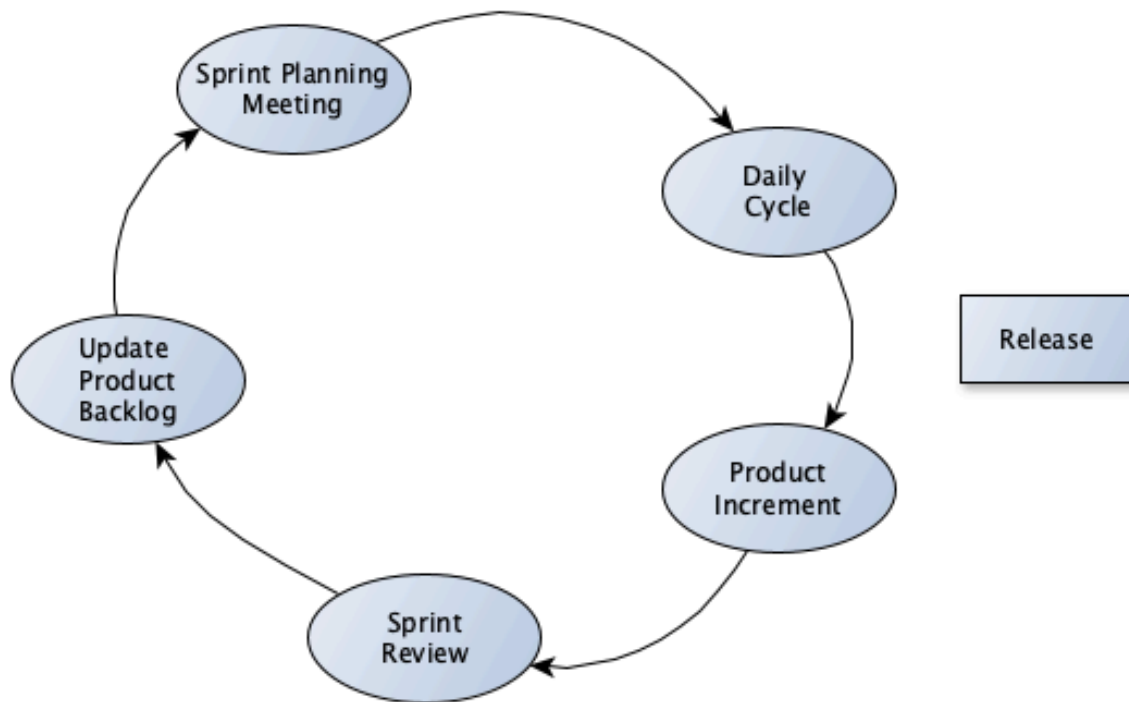
5.2 Pert Chart
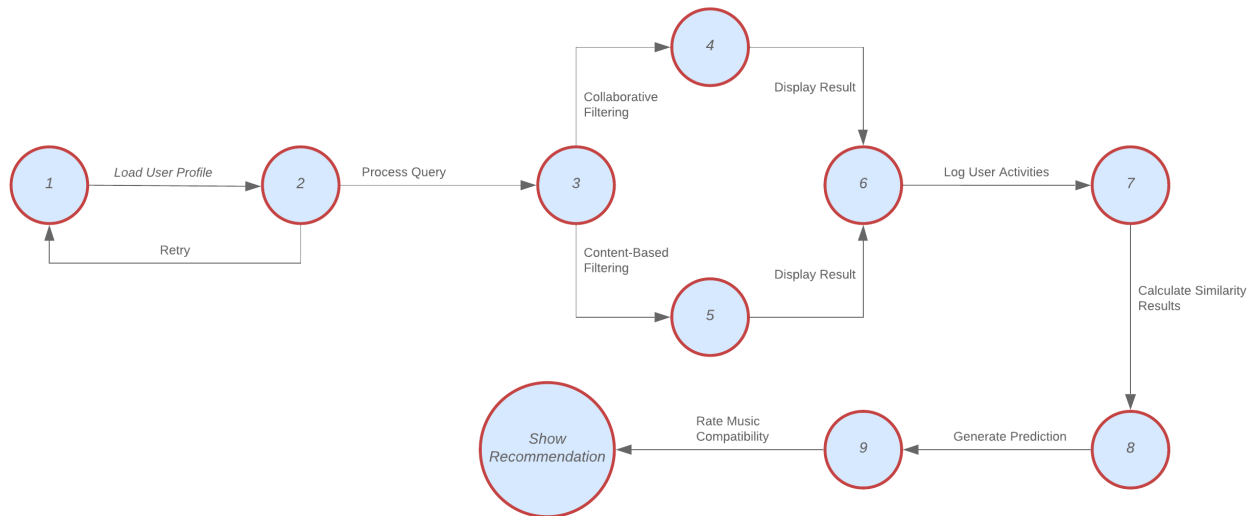
5.3 Work breakdown Structure
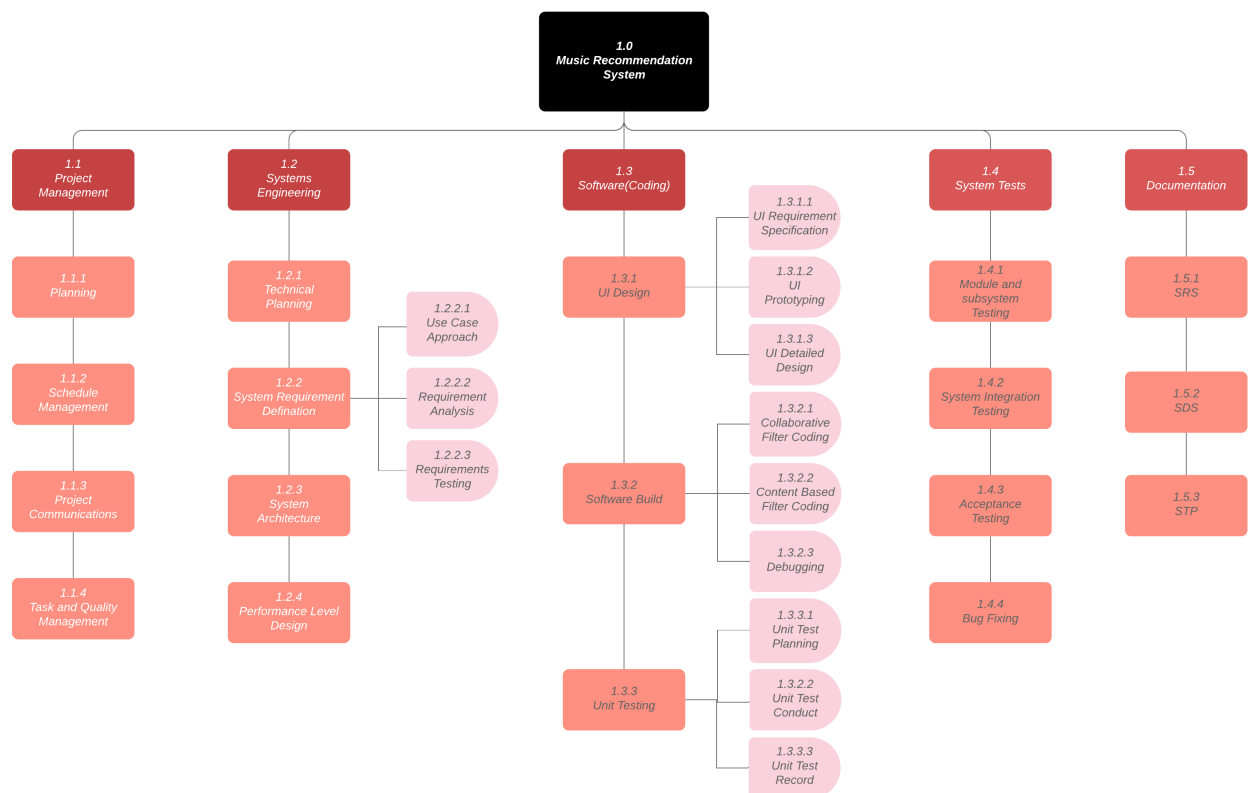
5.4 Process Model

As a process model, it is planned to be used Scrum. It is much easier to demonstrate this process model. Here is the visual demonstration of the chosen process model.

7. CONCLUSION

This Software Requirement Specification document is prepared to give requirement details of the project, "Music Recommender System". First, definition of the problem and the general description of the system are given. Then, all the functional, non-functional and interface requirements, data and behavioural models are stated in a detailed manner. Finally, structure of the development team, basic planning and estimation of the development process and the model of the development process are explained. This document will be helpful at constituting a basis for design and development of the system to be developed. Design details of the project will be explained in the Software Design Description document.