Song Recommendation System

Software Requirements Specification

Version 1.0

Tejas Patil
Nitin Narayana Prabhu
Pratik Lala
Ashwath Narayanan
Abhishek Kanchan

Prepared for
CS 441 – Distributed Object Programming using Middleware
Instructor: Mark Grechanik
Fall 2012

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide complete requirements specification of a Song Recommendation System. The document provides a complete description about the objective and goals of the application, the different functionalities supported by the system, the interfaces requirements of the system, the non-functional requirements that help to evaluate the system and the constraints involved in developing this application. The document serves as a useful reference for developers as well as the stakeholders of this system.

1.2 Scope

This application is a Song Recommendation System that is aimed towards the general public. The application allows users to obtain a list of songs, artists and albums that the application feels the user will prefer. To improve user experience, the application also recommends the highest rated songs, based on the user's preferences. The application also provides functionality to users to create personal profiles, rate song attributes, like/dislike song attributes and make simulated shares and purchases of songs and albums. The application has a distributed architecture that allows multiple users to access song data concurrently. It also interacts with external systems such as relational and non-relational databases. Different middleware are used for interfacing the application with external systems.

The application does not provide any audio/video playback capability for any of the songs in its database. Users of the application are not allowed to modify (add/remove/edit) the song list in the database.

From a developer's perspective, the application provides a learning experience in implementing a distributed rule-based architecture, the use of large data and its storage in relational and non-relational databases and concurrency and performance issues when accessing large volumes of data in a distributed system.

1.3 Reference

830-1998 - IEEE Recommended Practice for Software Requirements Specifications

1.3.1 Webpages Refered

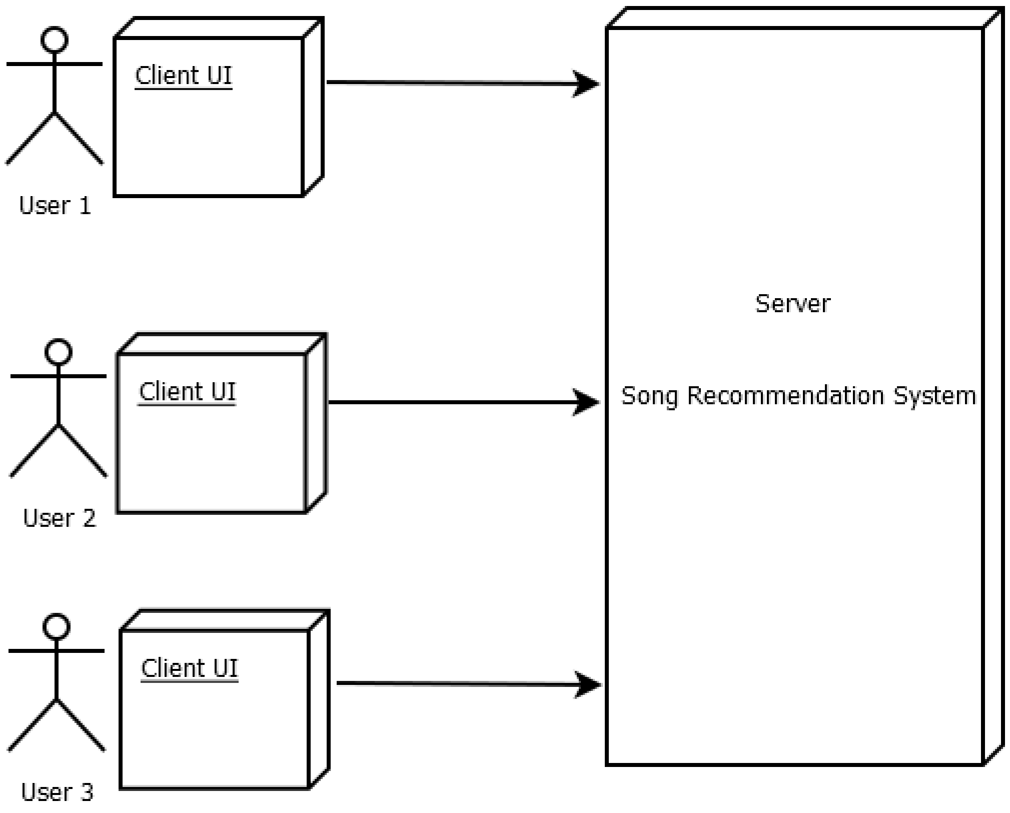
1. www.cse.msu.edu/~chengb/RE-491/.../SRSExample-webapp.doc
2. www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc

1.4 Overview

The next section provides an overall description of the application. It provides a more in-depth explanation of the application.

The third section details the requirement specifications of the system.

2. General Description



System Description

2.1 System Description

Users of the system serve as actors. They interact with the recommendation system through a web interface that allows the user to login with personal credentials, edit their preferences in their profiles, obtain a list of songs, albums and artists based on preferences and perform simulate operations on the data returned by the system.

2.2 Product Perspective

With the advent of high-speed Internet, there are numerous web-based applications that provide streaming content to large audiences. Examples of such applications are Youtube, Spotify and Grooveshark. Internet radio services, such as Pandora and Last.fm are also in vogue. With a very large customer base, such services aim to improve customer experience by providing a more personalized service.

Our application aims to provide recommendations on the user’s preferences, hence providing personalization. However, it is a stand-alone application and does not derive any recommendation logic from these services.

2.3 Product Functions

The product primarily supports the following functions

1. Allows users to provide personal preferences.
2. Allows users to provide ratings, like/dislike songs. Users are also allowed to make a simulated purchase of songs or albums.
3. Provides recommendations to the users based on user preference and searched songs.

2.4 User Characteristics

The application is targeted towards the general public, especially those with a string affinity to music. Users have the ability to interact with the system and retrieve information that is relevant to them.

2.5 General Constraints

Neither the list of songs in the song database being used in the application nor the users who have rated the songs in the application are universal. They are constrained by the data sets we have used.

Custom rules have been used in place of sophisticated data mining and machine learning techniques to provide recommendations to the users. Hence the accuracy of the recommendations may not be as high as those achieved with statistical models.

The application requires interfacing with both relational and non-relational database that serve as data stores for all song-related data.

2.6 Assumptions and Dependencies

The requirements stated below are dependent on the data set we are using in our application. The application is assumed to be operable on Windows and Linux operating systems. It is dependent on different middleware that provides interfacing capability to different services.

3. Requirements Specification

3.1 Functional Requirements

3.1.1 Use Case: Recommendation Functionality

Brief Description: The system provides song recommendations to the user based on user preferences.

Initial step by step description:

1. The user can explicitly ask for recommendations on songs based on personal preferences that the user can enter at the Client UI.
2. The system will implicitly use rules to generate a list of recommended songs for the user.

3.1.2 Use Case: Profile Creation

Brief Description: The user can create a profile on the system.

Initial step by step description:

1. The user creates a username and password for his/her profile.
2. The user has the option to enter his/her personal preferences that will be used by the system for recommendations.
3. The user logs into his/her profile and is automatically recommended songs based on the user profile.

3.1.3 Use Case: Rating a song/album/artist

Brief Description: The user has the options to rate a song/album/artist on a numeric scale.

Initial step by step description:

1. The user will receive a list of recommended songs/albums /artists.
2. The user will select a particular result to get the detail page for the item.
3. The user enters a rating on the item detail page.

3.1.4 Use Case: Likes for a song/artist

Brief Description: The user can like/dislike a particular song, album or artist.

Initial step by step description:

1. The user will receive a list of recommended songs/albums /artists.
2. The user will select a particular result to get the detail page for the item.
3. The user can like/dislike the selected item.

3.1.5 Use Case: Song/ Album Share

Brief Description: The user can simulate sharing of a song or album of his/her choice. This will not be an actual share but recording the song/album share in the database. This feature can be an indicator of the item’s popularity and allow the recommendation system to make improved recommendations.

Initial step by step description:

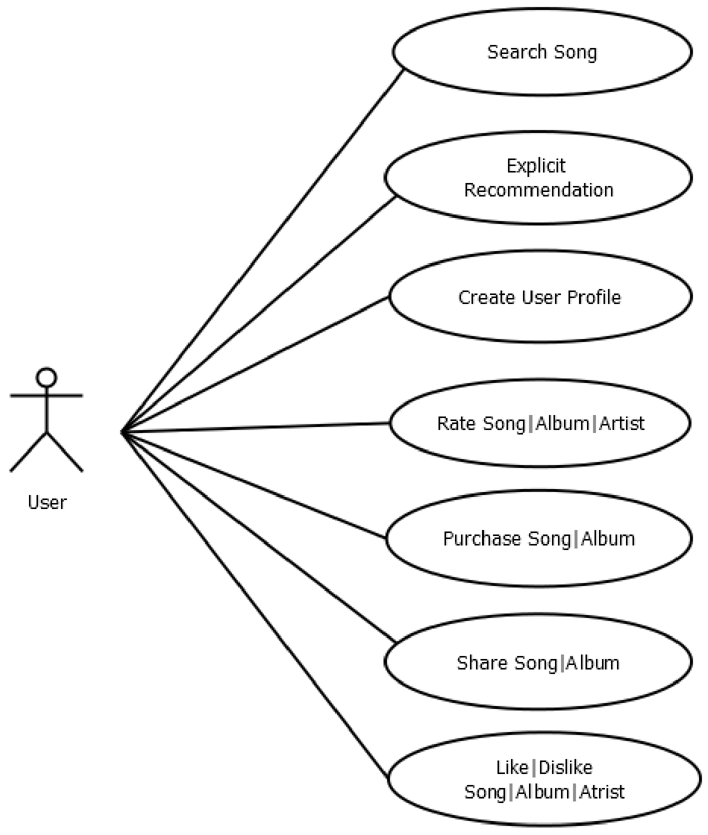
1. The user will receive a list of recommended songs/albums.
2. The user will select a particular result to get the detail page for the item.
3. The user has the option to share the item.

3.1.6 Use Case: Purchase song/album

Brief Description: The user can simulate purchasing a song or album of his/her choice. This will not be an actual purchase but recording the song/album purchase in the database. This feature can be an indicator of the item’s popularity and allow the recommendation system to make improved recommendations.

Initial step by step description:

1. The user will receive a list of recommended songs/albums.
2. The user will select a particular result to get the description page for the item.
3. The user has the option to purchase the item.



Use Case Diagram

3.2 Non-Functional Requirements

3.2.1 Performance

The response time of the application aimed not to exceed 10 seconds for each user interaction.

3.2.2 Reliability

For any client transaction, the server shall respond and should not result in failure. In the case of any failure, the client shall reattempt connection to the server and pass the client query again.

3.2.3 Availability

The server shall be available for the client for any transaction that the client tries to make with the server. If the server is unavailable, the client shall reattempt to connect to the server at most 3 times before ending in failure.

3.2.4 Security

User profile data is secured and will not be visible to any other users. Communication between a client and the server will be invisible to other clients.

3.2.5 Maintainability

The components for the client and server will be independent of each other. Any addition /modification/ deletion of modules can be done without affecting other modules.

3.2.6 Portability

The application will be independent of the underlying OS or hardware. The application only requires the presence of any incorporated middleware and the databases used on the machine on which the application is being ported.