CSCE 3513

Nick Mize, Jenifer Sandoval, Dillon VanBuskirk, Osvaldo Bobadilla

Software Engineering Requirements Specification Group #2

A brief description of software:

We will be designing a music referral site that doubles as social media. It will be web based that allows users to create accounts using a username and password that will be used to favorite certain musical groups. From there, users will be able to view artists similar to the ones liked and other artists which are favorited by other users who share the same likes. Users will be able to filter certain groups by disliking them, or even disliking a genre entirely. There will be a search function that will search by artists, genre, or even by songs. Upon finding a musical group that a user likes, they will be able to open the artist page and find miscellaneous information about them as well as external links. All of the user account information will be stored as well as the entire musical group database. Users will be able to also create playlists of music using Spotify API available on their profiles. Their profiles will be able to be moderately customized by creating a playlist and decorating with favorited artists. Users will also be able to create a friends list and be able to search for users and view their profiles. A mobile app will be attempted to be designed as a feature of our software.

Part 1: User Requirements

Standard User:

1. (1) Web interface - As a user, I want to be able to use an interface so I can create an account, log in to an existing account, as well as perform all other functions relating to liking artists. This consists of the graphical user interface for the web. This interface will take a large amount of person-hours, estimated 8.
2. (1) Database - The database will be used behind the scene for the user because it will be used to store all of their information. There will be two databases: one will be for storage of user account information, the other will be for storage of the artist information. These two will be able to communicate with each other, in that the information on the artist database could be written to the account information database. This will take a large amount of person-hours, estimated 8.
3. (1) User accounts - As a user, I need to be able to enter my information in predesigned forms that will allow me to create a profile or sign into an existing profile. The user will use this profile to modify information about themselves and their favorites. This will consist of user login info as well as additional miscellaneous information about each artist. This will take a small amount of person-hours, estimated 3.
4. (2) Friend list - As a user, I want to be able to ‘follow’ friends that will display on my sidebar news feed. This will allow users to find current friends and view their profiles as well as the ability to find new friends with same music taste, leading to new music discoveries. This will take an average amount of person-hours, estimated 4.
5. (3) Social Media Integration - As a user, I want to be able to send notifications to my other social media sites that display artists I like or playlists I’m listening to. This will allow users to share artists that they’ve favorited on social media sites using their respective API. This will take a small amount of person-hours, estimated 2.
6. (3) Music Playback and Playlist - As a user, I want to be able to add several songs or artist to a playlist that allows me to save and reopen a group of songs. This will allow users to create

playlists of their favorite artists' songs and allow for playback of said songs. This will take an average amount of person-hours, estimated 4 as we have to learn and use Spotify API.

7. (4) Mobile app – As a user, I want to be able to access the site through my mobile device. The users will download the app and either login in or create a profile under a portable device. This will allow access from anywhere through a smart phone. This will take a large amount of person-hours, estimated 8 since we need to learn mobile programming.

Part 2: System Requirements

For Standard User:

1. Web Interface -
    a. Design homepage of website that welcomes users
    b. Design a GUI that allows users to use different functions (sign up, sign in, friends list, music search, logout, artist pages, profile pages)
    c. Add cookies and/or sessions to site
2. Database -
    a. Need a database that stores user login information
    b. Need a database that stores user account description
    c. Need a database that stores artist information
3. User Accounts -
    a. Design user sign up and login page
    b. Design form for sign up and login, including javascript to check for appropriate passwords and usernames. This will give the user feedback based on what they input
    c. Send form information to server and add to or check against database storing user information
    d. Design a homepage for user profiles when signed in
    e. Design a log out button and script that signs a user out when pressed
    f. Design a function that allows user to reset or change their password that properly talks to database
4. Friends List -
    a. Create a page on the user profile page that allows for users to 'follow' friends as well as the function to handle following. This will add the user to the user's news feed that will be in the side bar of a user profile
    b. Allow for users to see who is following them as well as a function that allows users to stop others from following them
5. Social Media Integration -
    a. Work with Facebook API and other social media sites to allow users to share their favorites on their social media sites. This integration will be visible on user's profiles as well as each artist page
6. Music Playback and Playlist -
    a. Work with Spotify API or Grooveshark API to add the ability to have music playback when searching for artists and their songs
    b. Use the Spotify API or Grooveshark API to add playlists through each user account. Design a portion of the user profile that allows for the playlists to be visible
7. Mobile App -

   a. Design a mobile ported version of the site, available to iOS and Android
   b. Design mobile UI that simplifies the site and still allows all functions with an easy-to-use interface

Part 3: Non-functional Requirements

1. (Performance) User login takes no more than 120ms by only error checking once
2. (Reliability) When user edits account information, it is properly saved in database and is appropriately updated on the webpage
3. (Reliability) When users are not signed in, account information is not lost or changed.
4. (Security) A user account password shall not be able to be compromised (a user must provide exact password to sign in)
5. (Portability) Webpage will work as intended on multiple systems and browsers
6. (Stability) If the server hosting the webpage were to go offline, data contained within the database should not be altered
7. (Usability) Interface will be appropriate for the target user community
8. (Security) A user should not be able to add another user without mutual agreement
9. (Reliability) The user homepage should not be altered without user request and should maintain its state despite user login activity
10. (Maintainability and Extensibility) The website should be easily maintained and updated by admins and developers when necessary. The website will be designed in a way that updates will be easily able to add new features