Genesis

By Joshua Bullard,
Alex Brelsford,
Trevor Bali,Serge
Dominique

CSC434
2/21/2019

We acknowledge that this document uses material from the Volere Requirements Specification Template, copyright © 1995 – 2007 the Atlantic Systems Guild Limited.

Contents

Project Drivers

Project Constraints

Functional Requirements

Nonfunctional Requirements

Project Issues

1. The Purpose of the Project

1a. The User Business or Background of the Project Effort

The basis for this project, is for it to be used as a graded assignment within our Software Engineering class. The work to be done within the project will consist of the creation of a program which will give users a new outlook on the music which they listen to. Our main goal for this project is to procure a decent grade for the final product, as well as for the finished product to be used as an example within our individual resumes.

1b. Goals of the Project

The goals of this project is to give people a new taste on music. Helping them learn new types of music that they never heard of. Giving them new songs choice that might be similar to their music upcoming music list. The motivation of this project is to helps the user get out of his or her comfort zone. Also, help them listen to new type of music during their daily life. Giving them a new reason to wake to something new. We would want the user to say this everyday, "Wow, I never heard of this song before.", everytime the user logged into.

2. The Client

The client for this project is Dr. Kimberly Cornell at The College of Saint Rose. The project will be developed during the spring semester of 2019.

3. Users of the Product

Regular Users

- User name/category: Music enthusiasts
- User role: The user must fill out the query questions to generate the music upcoming music list. Once the upcoming music list is created, the user will be able to listen, edit, and share with others.
- Subject matter experience: Novice, the user will only need to know how to type and access the Internet.
- Technological experience: Novice, this project is not technical.
- Other user characteristics: Users of this project must have a spotify account (free or premium). Users also must have access to the Internet due to this project being a web based application using Spotify's API. This project is for all age groups and genders.

Administrator Users

- User name/category: Developers
- User role: Can do everything regular users can do as well as setting parameters for the upcoming music list.
- Subject matter experience: maser
- Technological experience: master

4. Mandated Constraints

4a. Solution Constraints

Description: The application must connect to a web server using Amazon AWS

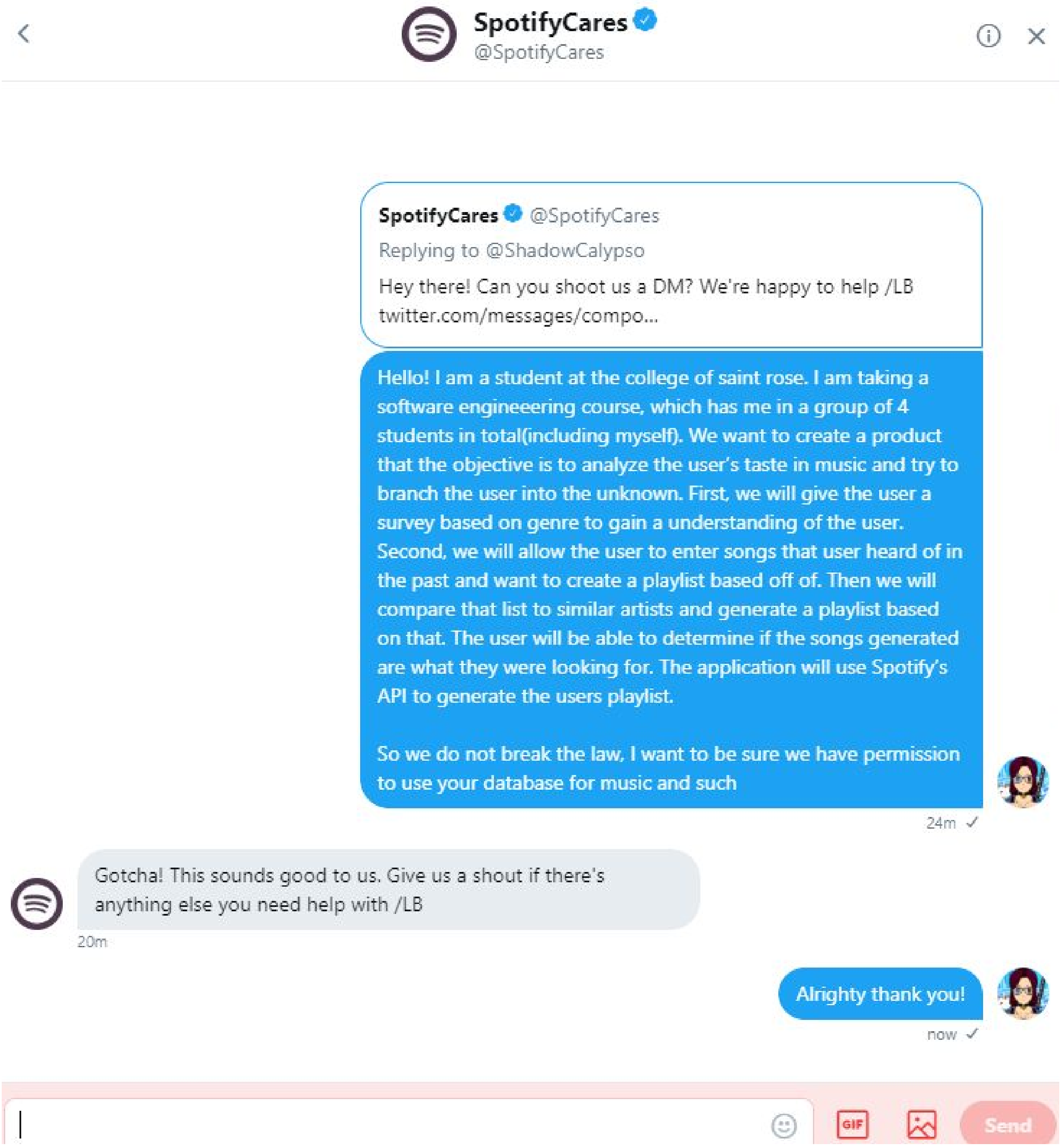
Rationale: With AWS, the application will be accessible from non-local servers.

Fit Criterion: The application will use AWS to run the application on the Internet

Description: The application must be able to connect to Spotify to receive tracks and artists

Rationale: With spotify, the music selection will allow for a better database for material and track suggestions.

Fit Criterion: The application will use Spotify API to link to Spotify's database. Spotify API gives developer an access token which can be used to access Spotify's database.



SpotifyCares
@SpotifyCares

SpotifyCares @SpotifyCares
Replying to @ShadowCalypso
Hey there! Can you shoot us a DM? We're happy to help /LB
twitter.com/messages/compo...

Hello! I am a student at the college of saint rose. I am taking a software engineeering course, which has me in a group of 4 students in total(including myself). We want to create a product that the objective is to analyze the user's taste in music and try to branch the user into the unknown. First, we will give the user a survey based on genre to gain a understanding of the user. Second, we will allow the user to enter songs that user heard of in the past and want to create a playlist based off of. Then we will compare that list to similar artists and generate a playlist based on that. The user will be able to determine if the songs generated are what they were looking for. The application will use Spotify's API to generate the users playlist.

So we do not break the law, I want to be sure we have permission to use your database for music and such

24m ✓

Gotcha! This sounds good to us. Give us a shout if there's anything else you need help with /LB

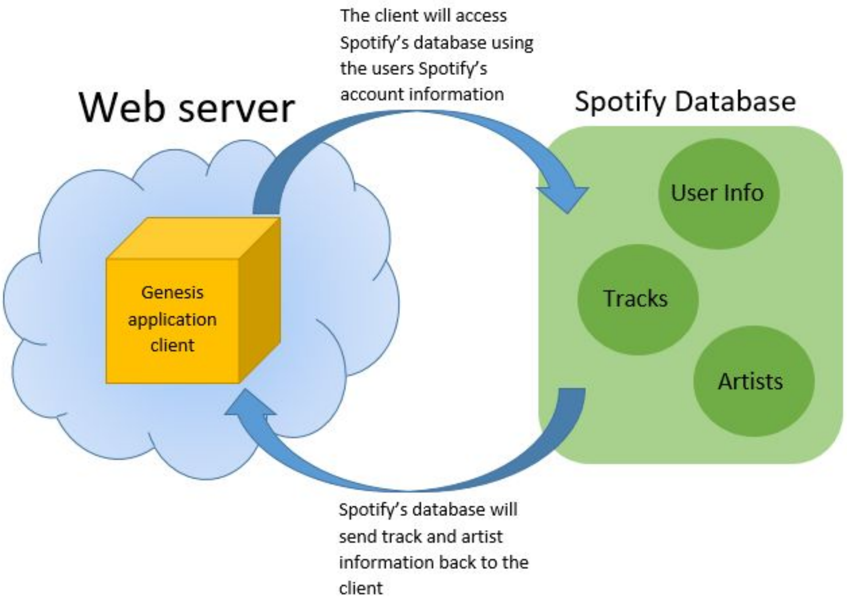
20m

Alrighty thank you!

now ✓

4b. Partner or Collaborative Applications

This application is going to have interactivity with Spotify Music.


The client will access Spotify's database using the users Spotify's account information
Web server
Spotify Database
Genesis application client
User Info
Tracks
Artists
Spotify's database will send track and artist information back to the client

4c. Schedule Constraints

The project deadline is the last day of the spring semester, which is May 9, 2019.

5. Definitions

Database - A structured of data held on the computer to be easily accessed, managed, and updated

MySQL - An open source relational database management system.

Eclipse - An integrated development environment (IDE) used in computer programming

Excel - A spreadsheet program included in the Microsoft Office suite of applications.

Github - A web-based hosting service for version control using Git

Public chat - The communication over the Internet that offers a transmission of messages between sender and receiver.

Private chat - The communication over a private form accessible by the users participating in the message.

Like Button - An option which allows the user to be able like his/her own songs.

Dislike Button - An option which allows the user to be able to dislike his/her own songs.

Web API - An Interface on the user's browser of choice

Javascript - A programming language that is commonly used for creating interactive effects within web browser.

Spotify - A digital music service that allows you to stream music.

6. Assumptions

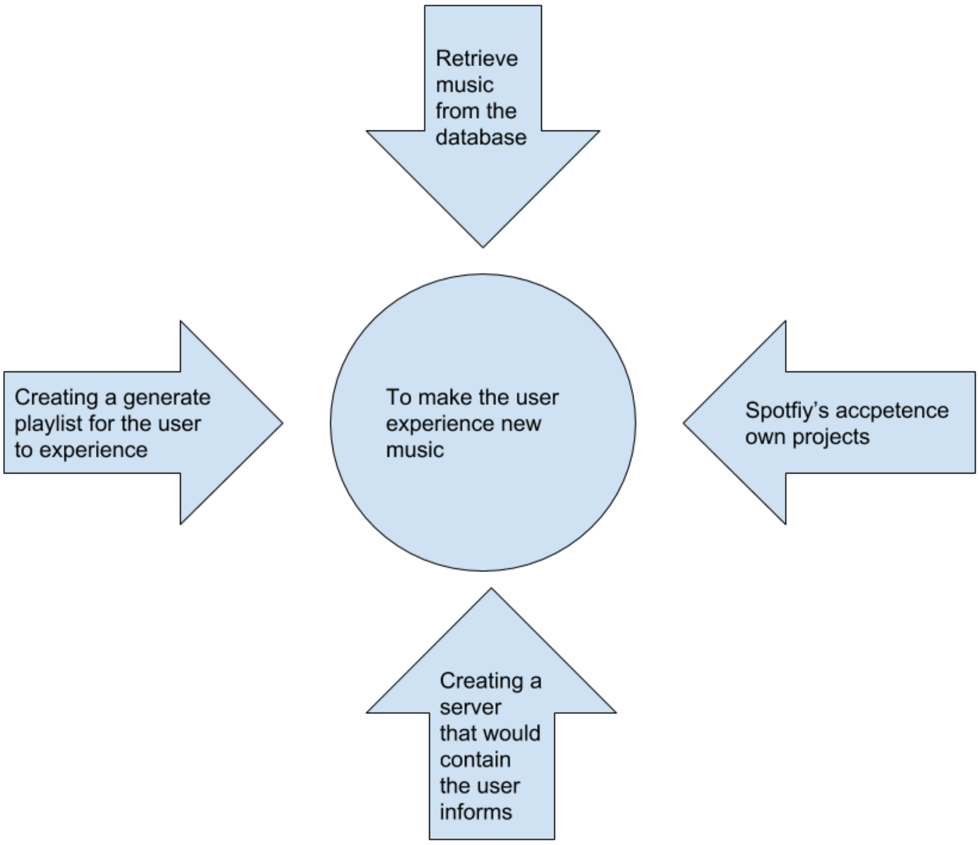
- All users will be able to connect to the Internet
- All users will be able to type and read
- All users will be able to navigate a basic website
- All user will have a Spotify account (free or premium)
- The software will be compatible with multiple web browsers and operating systems

7. The Scope of the Work

7a. The Current Situation

Currently the Computer Science students will have GitHub available to work to help them work as a group on the project. The students will also have Spotify WebAPI available to them. The Spotify API is a great tool to use. It allows people who want to build an addition onto Spotify, or if they just need some tips on how to code certain things. Adding this project to the existing Spotify application will allow the user to, as stated before, find new Artists based upon the idea of, if they like the Artist ____ then maybe this Artist who isn't that well known with the same type of music is someone they would like to listen to. With the API provided to the students by Spotify, it can allow the students then can create a Like and Dislike button so it can tell the program that the User either really liked/disliked the music/Artist they just listened to and adjust the music accordingly.

7b. The Context of the Work


Retrieve music from the database
Creating a generate playlist for the user to experience
To make the user experience new music
Spotfiy's accpetence own projects
Creating a server that would contain the user informs

7c. Work Partitioning

1. Event name: User logs into Spotify account

   Input: User enters Spotify account username and password

   Output: The application uses information from users Spotify account

   Summary: The user will be asked to log into a Spotify account when using Genesis software. When the user attempts to create a upcoming music list the application will notify the user to enter their Spotify credentials.

2. Event name: User enters an artist name to create upcoming music list

   Input: User enters a desired artist in a text field that they would like to create a upcoming music list from.

   Output: The application will generate a upcoming music list based on related artists and tracks

   Summary: The user will be able to add multiple artists to generate a random upcoming music list of related artists and tracks.

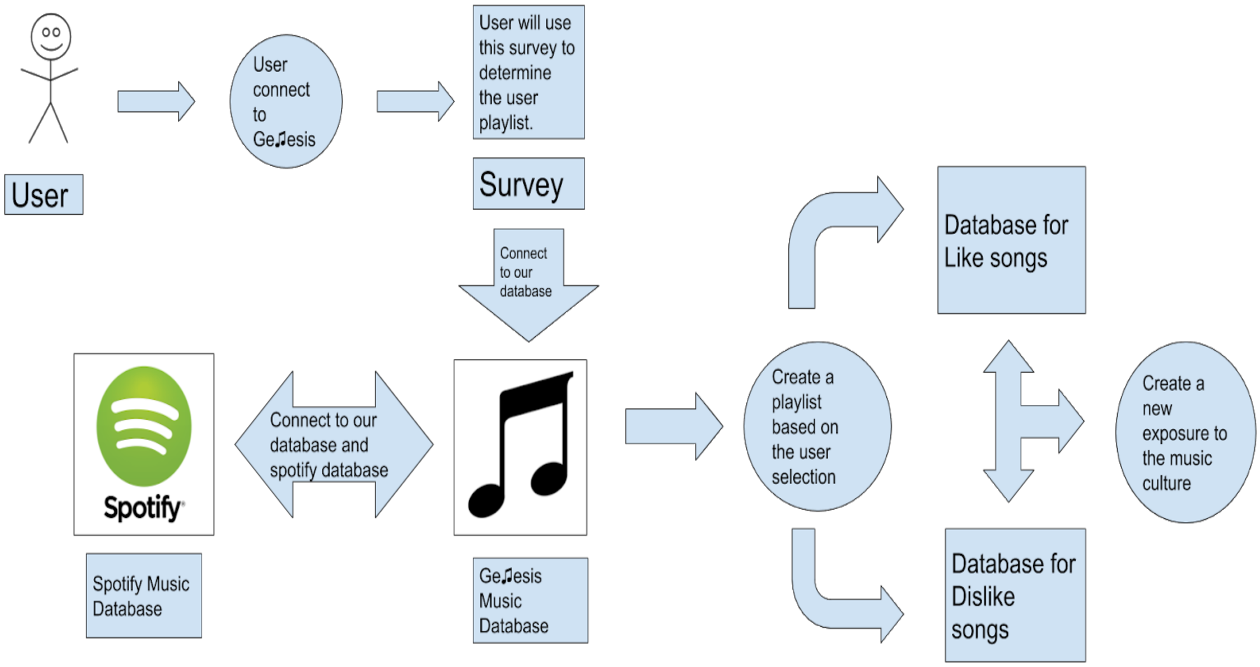
3. Event name: User saves custom upcoming music list

   Input: User clicks save button on application

   Output: The application will save the upcoming music list to the users Spotify account

   Summary: The user will be able to save the custom upcoming music list to their personal upcoming music lists on their Spotify account if they desire to.

8. The Scope of the Product


User
User connect to Ge♫esis
User will use this survey to determine the user playlist.
Survey
Connect to our database
Spotify
Spotify Music Database
Connect to our database and spotify database
Ge♫esis Music Database
Create a playlist based on the user selection
Database for Like songs
Database for Dislike songs
Create a new exposure to the music culture

9. Functional and Data Requirements

9a. Functional Requirements

Content

Requirement #1

Use Cases: Standard User: Account Linking

Description: The product shall allow users to register their Spotify accounts.

Rationale: Allowing the linking of the criteria provided by users with music within the Spotify music database that matches the given criteria.

Fit Criterion: Connection between list of liked songs within main Spotify program, and songs within the Genesis extension program.

Priority: 1

Conflicts: None

Author: Trevor Bali

Date: 2/19/2019

Requirement #2

Use Cases: Standard Users: Liking and Disliking of Songs

Description: The user has the ability to like or dislike a song.

Rationale: If a user likes or dislikes a song, it will tell the program whether more music within this genre, or by the same musician should be recommended to the user or not.

Fit Criterion: Whether a song by the same artist will be within the recommended list given to the user.

Priority: 1

Conflicts: None

Author: Trevor Bali

Date: 2/19/2019

Requirement #3

Use Cases: Standard Users: List of Disliked/Liked Songs

Description: Users require access to a list of songs which they have liked or disliked in the past.

Rationale: People's taste in music changes over time, thus what they may have liked or disliked in the past, they may dislike or like now, respectively.

Fit Criterion: Access to a list of songs previously liked or disliked, with songs switching between the two as needed.

Priority: 1

Conflicts: None

Author: Trevor Bali

2/19/2019

Requirement #4

Use Cases: Standard Users: User Input

Description: Users will be able to enter a desired artist's name, genre, or song name.

Rationale: If a user has a preferred artist or genre that they listen to, then they should be allowed to be recommended songs which are similar to their current tastes.

Fit Criterion: User inputs an artist's name and the program returns a recommendation to a song by a different artist within the same genre.

Priority: 1

Conflicts: None

Author: Trevor Bali

2/19/2019

Requirement #5

Use Cases: Standard Users: User Input

Description: The user will be given a survey when they first begin to use the program.

Rationale: The program will take the results of the survey and create a basic upcoming music list of songs that the user may like, before expanding upon by it by using the users feedback to the given songs.

Fit Criterion: Only open first access of the Genesis program should the survey be given, and will only be given multiple times if no response has been received by the server.

Priority: 2

Conflicts: None

Author: Trevor Bali

Date: 2/19/2019

Requirement #6

Use Cases: Standard User: Upcoming Songs

Description: The user is given a list of upcoming songs which will be played and replaced as they are listened to.

Rationale: An ever-changing list of songs rather than a set playlist that is updated after a set amount of songs are played.

Fit Criterion: Once a song has been played, it will not be re-added to the bottom of the given list, and a new song will take its place.

Priority: 1

Conflicts: None

Author: Trevor Bali

Date: 2/21/2019

Requirement #7

Use Cases: Standard Users: Share Feature

Description: Allowing the user to share their new music with a friend.

Rationale: Can be used to promote new music that people have never heard of and creating a buzz for the song.

Fit Criterion: The song is shared with a user's friend, which the friend has the choice to either listen to the song or decline the song.

Priority: 3

Conflicts: None

Author: Serge Dominique

Date: 2/21/2019

Requirement #8

Use Cases: Standard Users: Chat Feature

Description: An in application chat to be used between users.

Rationale: Can be used by users to discuss music which they have found without the need to share a single song at a time, either with a friend or a group of friends, allowing other users within the chat to search for the given song(s) and find it for themselves.

Fit Criterion: A working chat program for communication between users.

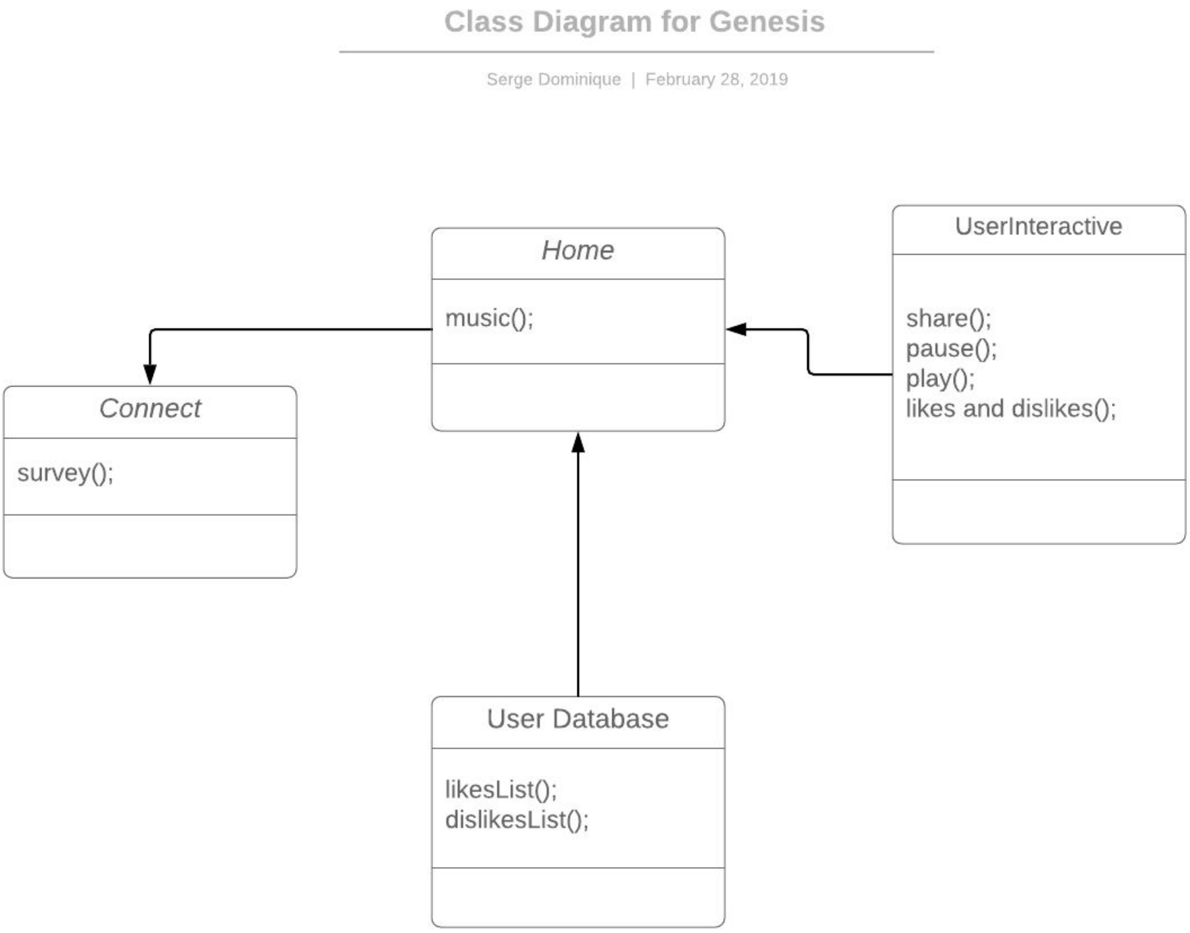
Priority: 0

Conflicts: None

Author: Trevor Bali

Date: 2/21/2019

9b. Data Requirements



Class Diagram for Genesis

Serge Dominique | February 28, 2019

10. Look and Feel Requirements

10a. Appearance Requirements

The visual appearance of the project should take parts of Spotify and GitHub. The project should be a yellow, green, and white.

10b. Style Requirements

This project should have some similar features like Spotify to make it more user friendly. Users should be able to have no problem to learn this website.

Fit Criterion

Users should feel happy to logged in everyday, having the ability to listen to a new song everyday.

Users will be able to keep track of their songs choice after finding a new song.

11. Usability and Humanity Requirements

11a. Ease of Use Requirements

- This project should be easy to use for all users
- This product should also look and feel very user friendly and not hard to use
- If a tutorial is needed on how to use it, the user can find that easily as well
- The process of logging on should feel easy to use

Fit Criterion:

- The Majority of the users should be able to use this product within minutes of being logged in and if they need it, watching the tutorial.

11b. Learning Requirements

- Ge♫esis is an application that easily and efficiently allows the user to listen to music of their tastes that they may not have listened to before
- The application should be easy to use and access

Fit Criterion:

- The user should be able to create the upcoming music list within minutes of watching the tutorial and being logged on.

11c. Understandability and Politeness Requirements

The project will contain simple steps to guide the user to create the playlist to their liking.

11d. Accessibility Requirements

The product should be up to date on all the latest screen reader technology, as well as being accessible to those who are color blind.

12. Performance Requirements

12a. Speed and Latency Requirements

- Every time a user like a song, the database should be update every 5 seconds.
- Each time the user logged off, user information should be save and updated.
- User should be able to logged without no problem

12b. Reliability and Availability Requirements

The project shall be available 24 hours a day, 7 days a week except for the scheduled maintenance on Monday and Friday mornings between 9 and 10 AM.

12c. Capacity Requirements

A maximum of 3,000 online users shall be accommodated simultaneously at any time of the day, except during Monday's and Friday's maintenance hours.
The database shall hold up to a maximum of four terabytes of user and admin data.
The user upcoming music list is set to 20. Only the admin is allow to change the limit.

13. Maintainability Requirements

The application will be offline Monday and Friday mornings between 9 and 10am for updates. Other than that the application will be up 24 hours a day.

14. Security Requirements

14a. Access Requirements

- Administrators will have access to all parts to the system except for the personal spotify login for the user
- All people with an internet connection will have access to this.
- The user and the Administrator have access to their individual private song likes and dislikes, but the Administrator can access anyone's private list
- Only the Administrators can change the cap on the number of songs and how the product saves a song
- Except for the administrators, all users who login for the first time will take a survey on their music taste
- All changes will be announced after an update

14b. Privacy Requirements

- The program will keep all login information private
- The product will adhere to the Internet laws on copyright and policy in the United States

15. Open Issues

- How will this application be hosted on the network?
- Will Spotify give the application access to users data to retrieve user account information such as liked tracks?
- Will users be able to understand how to fully use the application?

16. Risks

- Lack of features: A potential risk with this project will be that our group does not have enough time to develop a bug free application. With only the school semester to finish the project, we may not have enough time to implement all the features we want in the application.
- Not optimized: With the time constraint of the school semester the application code may not be fully optimized which may cause slow speeds when running the application.
- Server: If the application gets disconnected by the server it will unuseable by all users.
- Maintenance: Once users start to use the application bugs in the software may be found. A potential risk is that no one will maintain the application.

17. User Documentation

A document will be included with the application to provide the user with information on how to use the program and its features. Along with a written 'manual' document, the application will have some on screen instructions such as 'enter an artist' and 'create playlist' to guide the user into using the application to its fullest.

On the development side of the application, comments will be added to the source code to describe in detail what each function does and its effect on other functions. Comments will also be added to the source code to allow developers to maintain or modify the application in the future.

18. Waiting Room

Requirements that will not be part of the next release that may be implemented in future releases include:

- Sharing mode - After a playlist is created, the user would be able to send the playlist to another account.
- Purchase music - A button will be added to the application that will redirect the user to a music store such as Google play store or Apple store (depending on what OS the user is using) that will allow the user to purchase the songs in the playlist.
- Music guessing game - A mini-game in the application that will allow the user to guess what song is playing. The application will remove the name of the song that is playing and give the user multiple choice options on potential songs where only one is correct.