University of British Columbia, Department of Computer Science

CPSC 304

Summer 2018

Project Part 3

Group Members:

Name	Student Number	Unix ID	Tutorial Section	Email Address
Bijan Karrobi	44307156	z8j0b	T1C	bijankarrobi@gmail.com
Shaun Yuson	20953162	t3a1b	T1E	shaunyuson@gmail.com
Terence Chen	42602136	w2b9	T1F	tcbc@alumni.ubc.ca
Jonathan Fleming	35659135	o7d9	T1F	jfleming@alumni.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Formal Specification (Apollo 18):

Platform:

We plan to use JDBC along with Oracle's database tools.

Functionality:

In terms of functionality, our users will be split into two types: Listeners and Artists. Listeners will be able to rate songs and create playlists (add/delete song in a playlist). Artists will be able to produce music and create albums (add/delete song in an album). Both will be able to listen to songs and update the database through any of these methods. If a listener decides to delete their account, all their playlists will also be deleted. We will also have triggers that will update the database accordingly, such as when a user listens to a song and that song changes or updating an audio or album when its respective artist is removed.

Queries we will implement include finding songs based on ratings, year of release, artist, genre, etc. We will allow the ability to query all users' playlist to find playlists with specific songs or artists, as well as the listener who created the playlist and other playlists they have created. Users can also query about artists' record labels and group artists by record labels. Other types of queries would look at the intersection of Audio and Artist based on genre, the cross product of Artist and Album, and the projection of Audio based on attributes (show audio made after a certain year).

Data:

Since our project is based on a music domain, we will supply our data based on Spotify's database of songs, artists, albums etc. We expect that our data will mostly be strings and integers.

Division of Labour:

Given that we do not know all the roles we will need, backend just specifies any roles that might need to be filled once we begin the implementation stage. We also expect every team member to have equal participation in the project, taking leads based on personal comfort level and will help each other out when needed.

Bijan: Database Population, GUI, Backend
Shaun: Database Structure, Backend
Terence: Database Population, GUI, Backend
Jonathan: Database Structure, Backend

Feedback:

What other types of queries would be interesting based on our database?
Is there some basic functionality we are missing that can be added based on our entities and relationships?
How big should we make our database to best suit the project demo?