**CST-341 Music Manager Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Milestone 3 Music Management Application. | |
| **Date:** | 10/25/2020 | |
| **Revision:** | 1.6 | |
| **Team:** | 1. Vien Nguyen | |
| 1. Roland Steinebrunner | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | Small tweaks to make sure all JDBC is working | Roland | 4 | 0 | | Update Diagrams | Roland | 2 | 0 | | Implement JDBC | Vin | 5 | 0 | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | *https://github.com/viennp84/CST341MileStone.git* | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this report and we agree to the approach we are all taking. |

**Planning Documentation**

**Agile Scrum Product Backlog:**

*N/A*

**Agile Scrum Sprint Backlog:**

*N/A*

**Agile Scrum Burn Down Chart:**

*N/A*

**Agile Retrospective Results:**

|  |
| --- |
| **What Went Well** |
| Project finished on time and done to the satisfaction of both parties |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
|  |  |  |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

*N/A*

**General Technical Approach:**

The milestone project is developing a music web app that provides user the music information, album, singers, and authors. User can create their own favorite song lists and modify their music data.

The music management app will be implemented utilizing the Spring Framework, Bootstrap, and a PostgreSQL. It is designed and built based on Model-View-Controller (MVC) design pattern. The presentation player will show the users the websites and controls so that users can interact with the webpages. The controller will direct the processes to the business logic in the model layer. Model player is responsible for data processing and return the results that are shown in the presentation layer.

The music management application will contain an integrated API service that will allow third parties to access our lists.

The music app is designed for standard users, members, and administrators. Interaction with the app differs per user type.

To implement this project, the team will use GitHub Version Control System to manage the source code and database scripts for, reports, etc.

Team members meeting takes place at least once a week before or after class to discuss problem solving, plan checks, etc. Additionally, the team will communicate over Slack, Discord, or email during the project development process to solve issues. Team members will send a weekly report to professor and asking for technical help if needed.

Team will use GitHub Project board to manage the project for developing and testing processes.

**Key Technical Design Decisions:**

The project uses Java Spring Framework combine with MySQL, Bootstrap to build up pages, controller, and model, and is going to be deployed on Apache Tomcat Server locally. Team members need a computer with Eclipse, MySQL, Apache Tomcat, a text editor, and a web browser installed. The team will obey the planed schedule every month to evaluate the project processes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SEPTEMBER 2020** | | | | | | |
| **Sunday** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** |
| 30 | **31** | **1** | **2** | **3** | **4** | 5 |
| 6 | **7**  Labor Day | **8**  First day of Class | **9** | **10**  Team build up,  Project Idea,  Plan | **11** | 12 |
| 13  Report submission | **14** | **15**  Assign tasks to members.  Implement design layout, database, architecture | **16** | **17**  Combine, evaluate, modify the project | **18** | 19 |
| 20 | **21** | **22**  Initialize project, create database, data collection. | **23** | **24**  Combine, evaluate, modify the project | **25** | 26 |
| 27 | **28** | **29**  Create pages,  Model,  Business,  Playout presentation testing | **30** | **1** | **2** | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **OCTOBER 2020** | | | | | | |
| **Sunday** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** |
| 27 | **28** | **29** | **30** | **1**  Spring MVC, Function testing,  Modify code, design, data if needed | **2** | 3 |
| 4  Milestone 2 submission | **5** | **6**  Using Spring bean in the project, build up new features | **7** | **8** | **9** | 10 |
| 11 | **12**  Columbus Day | **13** | **14** | **15** | **16** | 17 |
| 18 | **19** | **20** | **21** | **22** | **23** | 24 |
| 25  Milestone 3 submission | **26** | **27** | **28** | **29** | **30** | 31  Halloween |

**Known Issues:**

Team member are unfamiliar with the Spring Framework at this time.

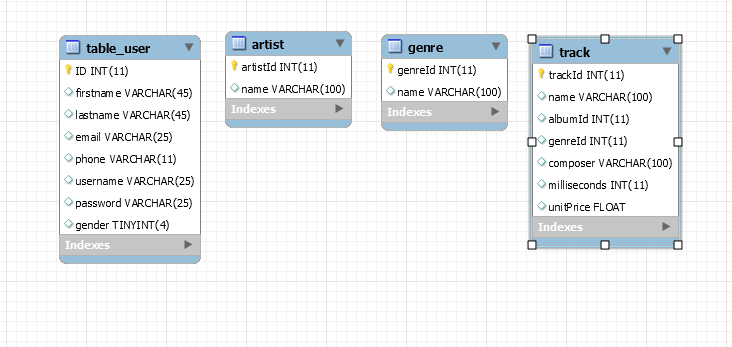
**Risks:**

This is the first building and deploying an application on Apache Tomcat Server, which could be different from other servers and may cause issues in the future.

Collecting data for the database could be difficult as the raw data for music may not be readily available

The application is being built during the semester which time is extremely tight with other classes. Lack of communication between team members and technical issues could arise.

**ER Diagram:**



**DDL Scripts:**

*-- MySQL Workbench Forward Engineering*

*SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;*

*SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;*

*SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';*

*-- -----------------------------------------------------*

*-- Schema mydb*

*-- -----------------------------------------------------*

*-- -----------------------------------------------------*

*-- Schema aawpey19h7mt74gk*

*-- -----------------------------------------------------*

*-- -----------------------------------------------------*

*-- Schema aawpey19h7mt74gk*

*-- -----------------------------------------------------*

*CREATE SCHEMA IF NOT EXISTS `aawpey19h7mt74gk` DEFAULT CHARACTER SET utf8 ;*

*USE `aawpey19h7mt74gk` ;*

*-- -----------------------------------------------------*

*-- Table `aawpey19h7mt74gk`.`artist`*

*-- -----------------------------------------------------*

*CREATE TABLE IF NOT EXISTS `aawpey19h7mt74gk`.`artist` (*

*`artistId` INT(11) NOT NULL AUTO\_INCREMENT,*

*`name` VARCHAR(100) NULL DEFAULT NULL,*

*PRIMARY KEY (`artistId`))*

*ENGINE = InnoDB*

*AUTO\_INCREMENT = 51*

*DEFAULT CHARACTER SET = utf8;*

*-- -----------------------------------------------------*

*-- Table `aawpey19h7mt74gk`.`genre`*

*-- -----------------------------------------------------*

*CREATE TABLE IF NOT EXISTS `aawpey19h7mt74gk`.`genre` (*

*`genreId` INT(11) NOT NULL AUTO\_INCREMENT,*

*`name` VARCHAR(100) NULL DEFAULT NULL,*

*PRIMARY KEY (`genreId`))*

*ENGINE = InnoDB*

*AUTO\_INCREMENT = 4*

*DEFAULT CHARACTER SET = utf8;*

*-- -----------------------------------------------------*

*-- Table `aawpey19h7mt74gk`.`table\_user`*

*-- -----------------------------------------------------*

*CREATE TABLE IF NOT EXISTS `aawpey19h7mt74gk`.`table\_user` (*

*`ID` INT(11) NOT NULL AUTO\_INCREMENT,*

*`firstname` VARCHAR(45) NULL DEFAULT NULL,*

*`lastname` VARCHAR(45) NULL DEFAULT NULL,*

*`email` VARCHAR(25) NULL DEFAULT NULL,*

*`phone` VARCHAR(11) NULL DEFAULT NULL,*

*`username` VARCHAR(25) NULL DEFAULT NULL,*

*`password` VARCHAR(25) NULL DEFAULT NULL,*

*`gender` TINYINT(4) NULL DEFAULT NULL,*

*PRIMARY KEY (`ID`))*

*ENGINE = InnoDB*

*AUTO\_INCREMENT = 684*

*DEFAULT CHARACTER SET = utf8;*

*-- -----------------------------------------------------*

*-- Table `aawpey19h7mt74gk`.`track`*

*-- -----------------------------------------------------*

*CREATE TABLE IF NOT EXISTS `aawpey19h7mt74gk`.`track` (*

*`trackId` INT(11) NOT NULL AUTO\_INCREMENT,*

*`name` VARCHAR(100) NULL DEFAULT NULL,*

*`albumId` INT(11) NULL DEFAULT NULL,*

*`genreId` INT(11) NULL DEFAULT NULL,*

*`composer` VARCHAR(100) NULL DEFAULT NULL,*

*`milliseconds` INT(11) NULL DEFAULT NULL,*

*`unitPrice` FLOAT NULL DEFAULT NULL,*

*PRIMARY KEY (`trackId`))*

*ENGINE = InnoDB*

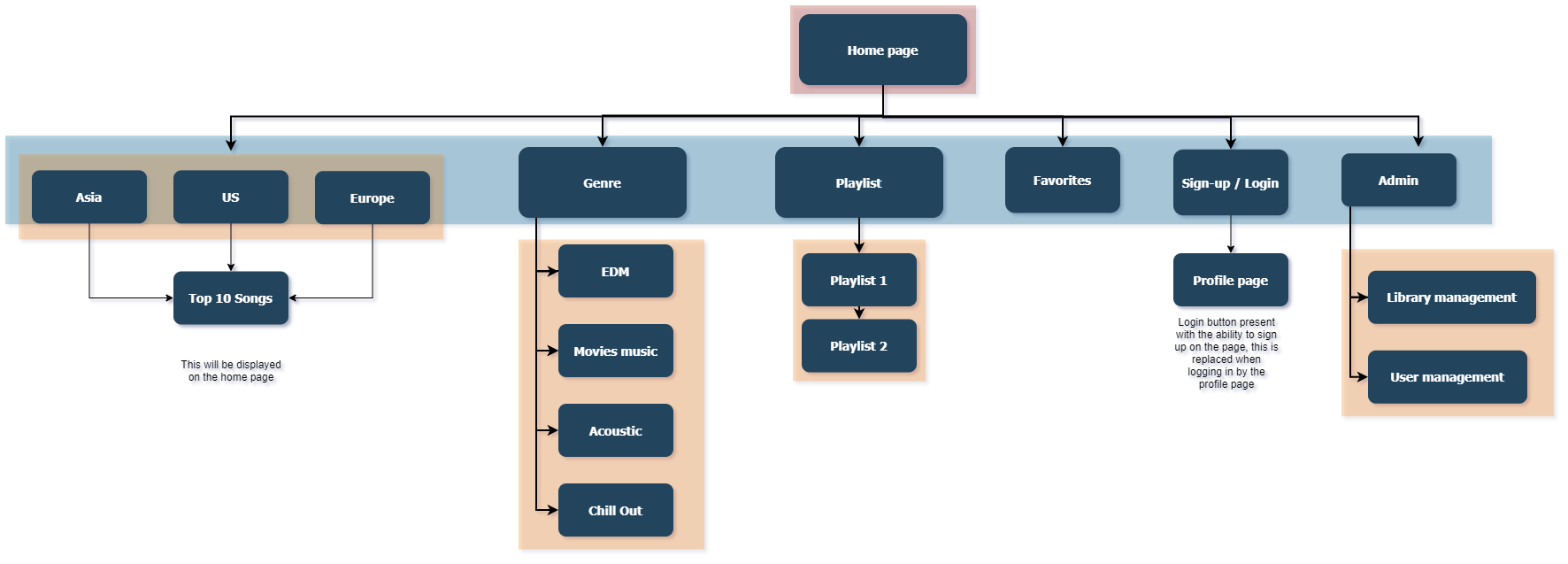
*DEFAULT CHARACTER SET = utf8;*

*SET SQL\_MODE=@OLD\_SQL\_MODE;*

*SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;*

*SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;*

**Sitemap Diagram:**

****

**User Interface Diagrams:**

*N/A.*

**Class Diagrams:**

*N/A.*

**Service API Design:**

*N/A*

**Security Design:**

N/A

**Project Outline:**

Our project is a music management system designed to sit over a large database of music. The application will have an easy to use interface that will allow the users to search through (potentially) thousands of songs sorted by artist, genre, and location. The product produced will be utilizing a much smaller database that will cover the basics and allow an adequate demonstration of the project without overburdening the team with mindless additions to the database. In addition to searching music the app will show the current top 10 songs for the month in each of the three main regions, North America, Europe, and Asia. One of the biggest draws of the app will be the ability to add songs to playlists and link those playlists to your account.

Open source licenses research.

The music application is going to be built up based on the open source tools. Team member requires to comprehend what is the characteristic open source licenses. Open source licenses refer to everyone about the freedom in software using and software development. Using open source license, programmers have the freedom to run the software, study how the software works and modify the software to adapt to their expectation, copy, or delivery the software to others. It does not essentially mean freeware. There are several open source licenses that impact the level of software freedom in use, modification, distribution software that programmers should consider such as Apache License 2.0, Berkeley Software Distribution License, and General Public License. These licenses are similar interns of allowing user to modify the product and self-distribution. Each license grants a different copy right.

The Apache License 2.0 authorizes users the freedom of changing, distributing the software for any purpose if they comply with the license terms. It means users cannot remove or replace the existing copyright from the original software license, developer must declare any changes of open source software to distribute to others. If these requirements are guarantee, user will not have any further obligation to the license. Apache license allows users to close their open source and commercializing their products with a copy right.

Berkley Software Distribution License

The Berkeley Software Distribution License is the oldest license terms in open source license. BSD has 2 versions licenses that are known as BSD 3-Clause “New” and BSD 2-Clause “Simplified”. BSD 3-Clause states that if the user redistributes the software, they must remain the copy right terms, conditions, and disclaimer of responsibility of the open source. The product developed with this open source license software can be commercialized without the permission of people who have contributed to the license. BSD 2-Clause “simplified” required the permission from people who own the license before commercializing their products. BSD licenses do not require the redistribution to be the open source.

GNU General Public License

General Public License (GPL) is another opensource license that provide to freedom to run, study, and modify the software. It requires the original copy right remain the same for every distribution. This license was created to protect the FNU software from becoming a proprietary software. Known as a copyleft license, anyone can write software that contains any GPL component, but the code must be released as open source. The advantage of this license is I can still be used it for commercial or proprietary purposes. Developers cannot get a copyright on the final product if it is based on software using this license.

The Apache License

Known as a permissive license, this allows you to use, modify, and distribute any apache licensed product. This license is very popular and has a large online following. Unlike copyleft licenses, this does not require a derivative work of the software or modifications of the original to be distributed with the same license. Apache Tomcat is an open source server run time that is release under the Apache License Version 2. It is a project intended to be a collaboration of the best-of-breed developers from around the world. During the Milestone development, the our team can contribute to this open source project by sending in bugs and logs so that this open source project is more completely developed.

The Eclipse Public License

Another copyleft license, EPL allows you to use the code if you allow the source code to become available upon request. EPL makes it clear that using a plugin does not mean you are using their code and do not need to disclose that. If you redistribute a program that uses an ELP component you must include the full license terms and copyrights. EPL protects authors from lawsuits or damages caused if a company uses that component commercially.