**Programming in Java III**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *CLC Milestone 8* | |
| **Date:** | *12/11/21* | |
| **Revision:** | *Rev 8.0* | |
| **Team:** | 1. *Daniyar Abeuov* | |
| 1. *Charles Osiris* | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Add admin users display, add, delete functionality* | *Charles Osiris* | *4* | *0* | | *Fix homepage movie list to display properly* | *Charles Osiris* | *2* | *0* | | *Final Project Milestone 8* | *Charles Osiris* |  | *0* | | *Benchmark Presentation* | *Charles Osiris* |  | *0* | |  |  |  |  | | *Final Project Milestone 8* | *Danny Abeuov* |  | *0* | | *Benchmark Presentation* | *Danny Abeuov* |  | *0* | |  |  |  |  | | *Generated JavaDocs* | *Charles and Danny* | *2* | *0* | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | *https://github.com/cjosiris1* | |
| **Screencast URL:** | <https://www.loom.com/share/8900111f5c214f008d768cf0b8e74078> - *Charles Osiris* | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this Report and we agree to the approach we are all taking. |

**Planning Documentation**

**Initial Planning:**

*Generate JavaDocs for the project. Refactor code and Reformat webpage designs as seen fit for clearer use. Troubleshoot and polish website’s flow and ease of access. Do those parts, review, revise, and submit the assignment.*

**Retrospective Results:**

|  |
| --- |
| **What Went Well** |
| Created JavaDocs |
| Added admin user page display, add, delete functionality |
| Fixed list of movies displayed on homepage |
| Troubleshoot to ensure full page functionality |

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
| UML Class Diagram PDF incorrectly displays in Git, but the image has been attached in this report. |  |  |
|  |  |  |
|  |  |  |

**Install or Configuration Instructions:**

*Take the KingKong.zip file from GitHub or the provided repository, drag it onto the desktop, and unzip KingKong file. Load Spring Tools Suite 4. Select Open projects from File System, choose the import source’s directory. Click finish, run as configuration, and type in localhost:{port} into the browser. View the application.*

**Design Documentation**

**General Technical Approach:**

*Our general technical approach to this project was essentially setting up a “movie” web application which features a login, registration, view movies, about us, and contact us/support pages. An admin page will be displayed to users with permissions and allows for editing, adding, and removing movies from the list. From logging in, the user will be granted access to view selected movies. From the home page, users will be able to look for movies or tv shows, view the company’s core values, and contact support. All the class data is stored securely on the MySQL database.*

**Key Technical Design Decisions:**

*The Spring framework, Spring boot, and Maven were used for this web application. The Spring Framework was used because of its ability to be employed on all architectural layers in development. It allowed us to freely link modules, easily test them, and eliminate the need to independently create factory and singular classes. Spring boot and Maven were used to speed up the dependency management process as it packages required third party dependencies within its starter packages, and because of its automatic configuration.*

**Known Issues:**

*No known issues. Project flows smoothly.*

**Risks:**

*There are many risks that went with designing and implementing our “movie” web application. Some of the risks for this project included: meeting the deadlines, implementing some of the features beyond the assignment requirements like movie details and the ability to view a movie. Technical risks included: computer failure (HDD died or something similar) the project fails to build and/or compile due to the environment suddenly breaking, some technology we are going to use in this assignment not working due to our environment etc..*

*Some ways to minimize the risks or damage done would be to: always start working early on each milestone to give us enough room and time to finish the milestone requirements and start working on the additional features. Always have a backup of the project on multiple devices, including a git repository. For the environment, we can make sure it is set up correctly and that we have the proper versions of the technology that we are going to be using.*

**ER Diagram:**

Table

Description automatically generated with low confidence

**DDL Script:**

**A picture containing text

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Project Proposal:**

*Design and implement a “movie” web application. Users are able to register for a new account, log in and gain access to pages that are secured behind login. Each user has login credentials as their email, password they created, and role assigned to them. Users are able to view the movie catalog and click on a movie to watch and view its details. Each user is able to comment under each movie and respond to other comments. User will have an ability to rate the movie on a scale of 1-5. The movie class holds the following information: movie name, date released, user rating, list of primary actors, the movie genre, age restriction, producer name, country of origin. Each movie has a basic video player and the video player for the trailer. The application features an admin management system, where the user with proper role assigned to them is able to add new movies, view all movies, edit movies, or delete them. All the class data is stored on the MySQL Database. Access to the database is made using SpringJDBC, and all business logic/services uses SpringBeans. The application includes a Java API to access movie detail information and is secured with Spring Security (log in and proper permissions required.) All the data inputs are validated and have some sort of restraints which are displayed through incorrect input. (Example: password must be at least 8 characters long, have numeric characters and a special character.)*

**Draft Division of Work:**

*Each person worked on designing and implementing 1 “feature” each. We will discuss and decide who will do what based on the preferences and/or their knowledge of the technology used. In the case with the implementation of Spring Security, Danny worked on forming the REST API design. Charles worked on reformatting table designs, movie details, and hyperlink preparation for movies. During this sprint, the website was refactored for additional implementation for movies and a more aesthetic display. Both of us worked on website troubleshooting and fixing errors over discord to make sure the website remains streamline with no dead ends or fall outs.*

**Sitemap Diagram:**

Diagram

Description automatically generated

**User Interface Diagrams:**

**Diagram

Description automatically generated**

**Class Diagrams (attached):**

Diagram

Description automatically generated

**Wireframe Diagrams:**

**A picture containing chart

Description automatically generated**

**Chart

Description automatically generated**

**A picture containing table

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Chart, surface chart

Description automatically generated with medium confidence**

**Bar chart

Description automatically generated with medium confidence**

**Table

Description automatically generated with medium confidence**

**Surface chart

Description automatically generated with medium confidence**

**Graphical user interface, application, table, Excel

Description automatically generated**

**Service API Design:**<https://app.swaggerhub.com/apis/Daniyar/CST339-API-Documentation/0.1>

**Other Documentation:**

*No additional documentation.*