**CST-235 Java Application Programming Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Topic 5 – Data Access Layer in Java EE | |
| **Date:** | 6/13/2021 | |
| **Revision:** | Version 3.0 | |
| **Team:** | 1. Vinson Martin | |
| 1. Paul Stevenson | |
| 1. Nathan Meyer | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *SQL script development* | *Paul Stevenson, Vinson Martin* | *7* | *0* | | *Product Display Module* | *Nathan Meyer* | *1* | *0* | | *Product Creation Module: Object Model* | *Vinson Martin* | *1* | *0* | | *Product Delete Module* | *Nathan Meyer* | *1* | *0* | | *Project Status and Design Report Update* | *Nathan Meyer, Vinson Martin* | *2* | *0* | | *Product Update Module* | *Vinson Martin* | *2* | *0* | | *Product Report Module* | *Vinson Martin* | *2* | *0* | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | <https://github.com/vm84039/cst-235milestone.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:**

*N/A*

|  |
| --- |
| **What Went Well** |
| **Everyone completed their tasks in a timely manner** |
| **The team assisted each other very well** |
|  |

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

**Design Documentation**

**Install Instructions:**

*Push file from GitHub, install into FireFly 12 web server, then run web server.*

**General Technical Approach:**

We decided to split our work into three equal portions, each team member working on a specific portion to then later merge into the final product of the Milestone. Meetings are on Discord. For Milestone 2, Vinson Martin is in charge of aggregating the code, main module design, site stylesheet design, and pushing to GitHub GIT. Paul Stevenson is in charge of the login module design and the creation of the user Bean used in this milestone. Minor alterations to the user Bean were made by Vinson Martin and Nathan Meyer. Nathan Meyer is in charge of registration module design, ER diagram design, and class diagram design with Vinson Martin.

**Key Technical Design Decisions:**

There are three frameworks for each tier of the application, the framework for the data tier which stores the database and database information will be JDBC, the framework for the logic tier which manages user login and registration information will be JSF which is used for security, and the framework for the presentation tier which manages the website appearance will be JSP. These frameworks were selected as they are included in the class’ syllabus and assignments, and thus would be easier to apply for this assignment.

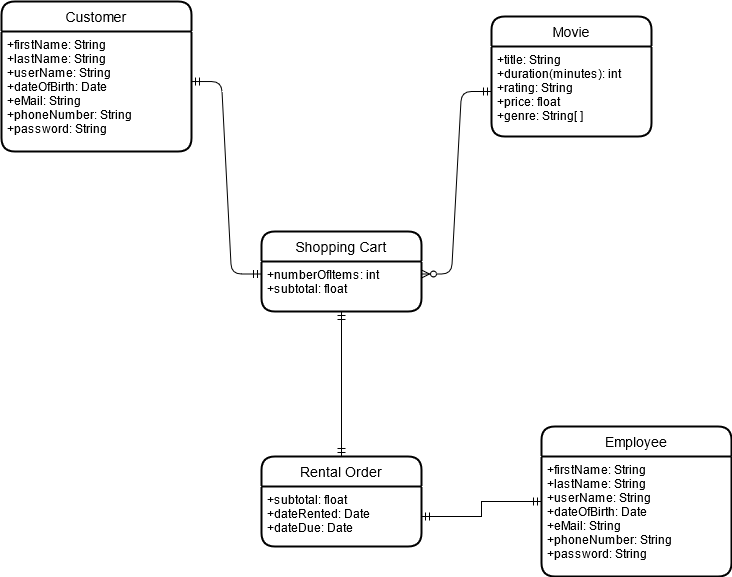
**Known Issues:**

*Delete Button & Edit Button have bugs. Not enough time to correct..*

**Risks:**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Details | Strategy for Risk **Avoidance** | Strategy for **Mitigation** |
| Project is too difficult to program | Advance features such as online checkout by customers, customer profiles and rental history are too difficult to program in one semester | Learn what the course covers and limit the features to covered topics | Keep project simple and relevant to the course |
| Teamwork failure | This project requires teamwork, and all participants must do their fair share for the | Delegate tasks to the teams’ strengths and keep a check progress often. | Check progress consistently and address issues early |
| Communication failure | Team does not communicate well which leads to confusion and incompleteness | Keep in contact with team members often and use a platform that allows for instant communication. Make tasks specific and go over every deliverable. | Be prepared to do extra work in case a task is missed and not assigned to a team member |
| Emergencies in the team | Sometimes life happens and a team member is not able to contribute as expected | Be prepared for anything and on standby for extra duties | Work ahead and stay ready. |

**ER Diagram:**

**

**DDL Scripts:**

CREATE DATABASE IF NOT EXISTS `MovieInfo`;

USE `MovieInfo`;

DROP TABLE IF EXISTS `Movies`;

CREATE TABLE `Movies` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`TITLE` varchar(30) NOT NULL,

`RYEAR` int(4) NOT NULL,

`GENRE` varchar(30),

`STAR` varchar(40) NOT NULL,

`STUDIO` varchar(30),

`DIRECTOR` varchar(30),

`LENGTH` int(4) NOT NULL,

`RENTAL` float NOT NULL,

`COST` varchar(20),

`REVENUE` varchar(20),

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=35 DEFAULT CHARSET=utf8;

**INSERT MOVIES**

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES ('Avatar',2009, 'Fantasy', 'Sam Worthington', '20th Century Fox', 'James Cameron', 162, 2.00, '237000000', '2847000000');

INSERT INTO MovieInfo.Movies (TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('Titanic',1997, 'Drama', 'Leonardo DiCaprio', 'Paramount Pictures', 'James Cameron', 195, 3.00, '200000000','2195000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('The Dark Knight',2008, 'Fantasy', 'Christian Bale', 'Warner Bros', 'Christopher Nolan', 152, 3.00, '185000000', '1005000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('Pulp Fiction',1994,'Crime','Uma Thurman', 'Miramax Films', 'Quentin Taratino', 175, 5.00, '237000000', '2847000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('GoodFellas',1990, 'Crime', 'Robert De niro', 'Warner Bros', 'Martin Scorsese', 150, 2.00, '2300000000', '3000000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('Jaws',1975, 'Horror', 'Roy Schneider', 'Universal Pictures', 'Steven Spielberg', 145, 4.00, '211000000', '2147000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('Dotherightthing',1989, 'Comedy', 'Spike lee', 'Universal Pictures', 'Spike Lee', 135, 6.00, '227000000', '2347000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('ET',1982, 'Adventure', 'Dee Wallace', 'Universal Pictures', 'Steven Spielberg', 185, 4.00, '257000000', '2557000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('The Karate Kid',1984, 'Drama', 'Ralph Macchio', 'Columbia Pictures', 'James Cameron', 125, 9.00, '287000000', '2457000000');

INSERT INTO movieinfo.movies(TITLE, RYEAR, GENRE, STAR, STUDIO, DIRECTOR, LENGTH, RENTAL,COST, REVENUE)

VALUES('Jurassic Park',1993, 'Adventure', 'Jeff Goldblum', 'Universal Pictures', 'Steven Spielberg', 179, 3.00, '245000000', '2797000000');

**Create User**

CREATE DATABASE IF NOT EXISTS `UserInfo`;

USE `UserInfo`;

DROP TABLE IF EXISTS `Users`;

CREATE TABLE `Users` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`FIRST\_NAME` varchar(30) NOT NULL,

`LAST\_NAME` varchar(30) NOT NULL,

`BMONTH` int(2) NOT NULL,

`BDAY` int(2) NOT NULL,

`BYEAR` int(4) NOT NULL,

`EMAIL` varchar(30) NOT NULL,

`UPASSWORD` varchar(30) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=100 DEFAULT CHARSET=utf8;

**Insert User**

*INSERT INTO UserInfo.Users(FIRST\_NAME, LAST\_NAME, BMONTH, BDAY, BYEAR, EMAIL, UPASSWORD)*

*VALUES('Boyce', 'Wickens', 5, 6, 1964, 'BWickens@example.com', 'This is Boyce Wickens');*

*INSERT INTO UserInfo.Users(FIRST\_NAME, LAST\_NAME, BMONTH, BDAY, BYEAR, EMAIL, UPASSWORD)*

*VALUES('Lannie', 'Turberville', 10, 2, 1972, 'LTumberville@example.com', 'This is Lannie Turberville');*

*INSERT INTO UserInfo.Users(FIRST\_NAME, LAST\_NAME, BMONTH, BDAY, BYEAR, EMAIL, UPASSWORD)*

*VALUES('Dawne', 'Bensen', 5, 27, 2002, 'DBensen@example.com', 'This is Dawne Bensen');*

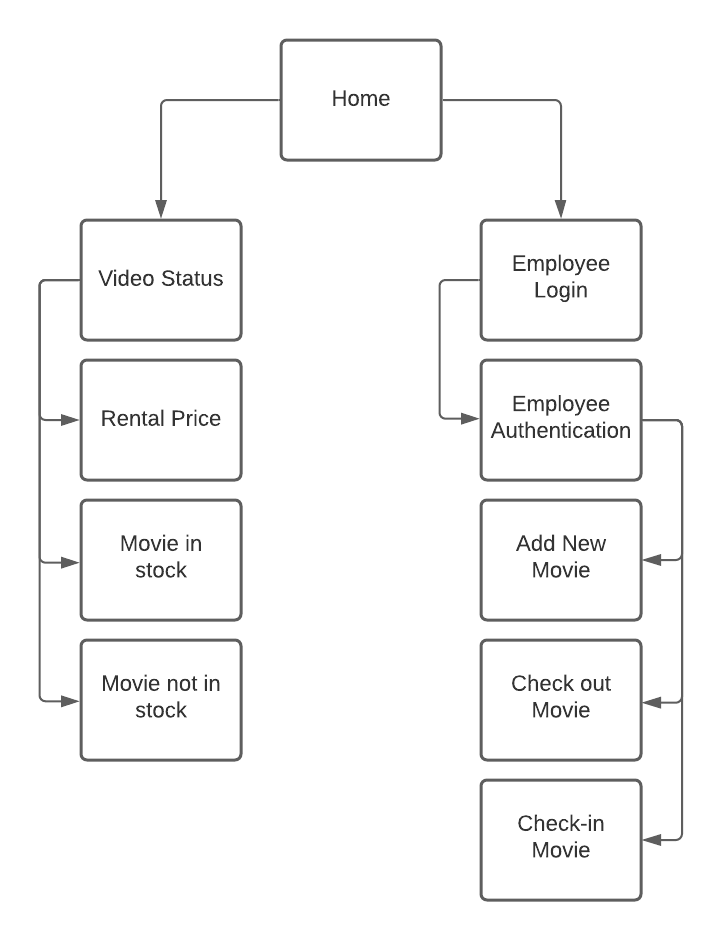
*INSERT INTO UserInfo.Users(FIRST\_NAME, LAST\_NAME, BMONTH, BDAY, BYEAR, EMAIL, UPASSWORD)*

*VALUES('Colin', 'Odonoghue', 4, 20, 1937, 'COdonoghue@example.com', 'This is Colin Odonoghue');*

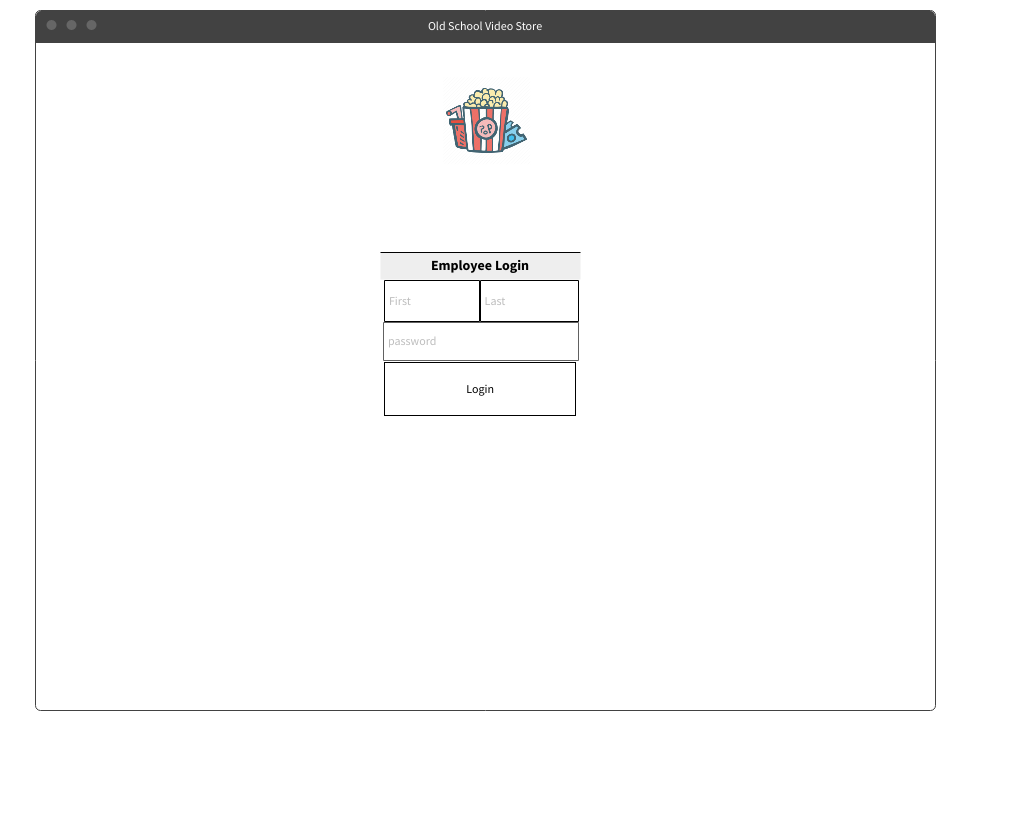
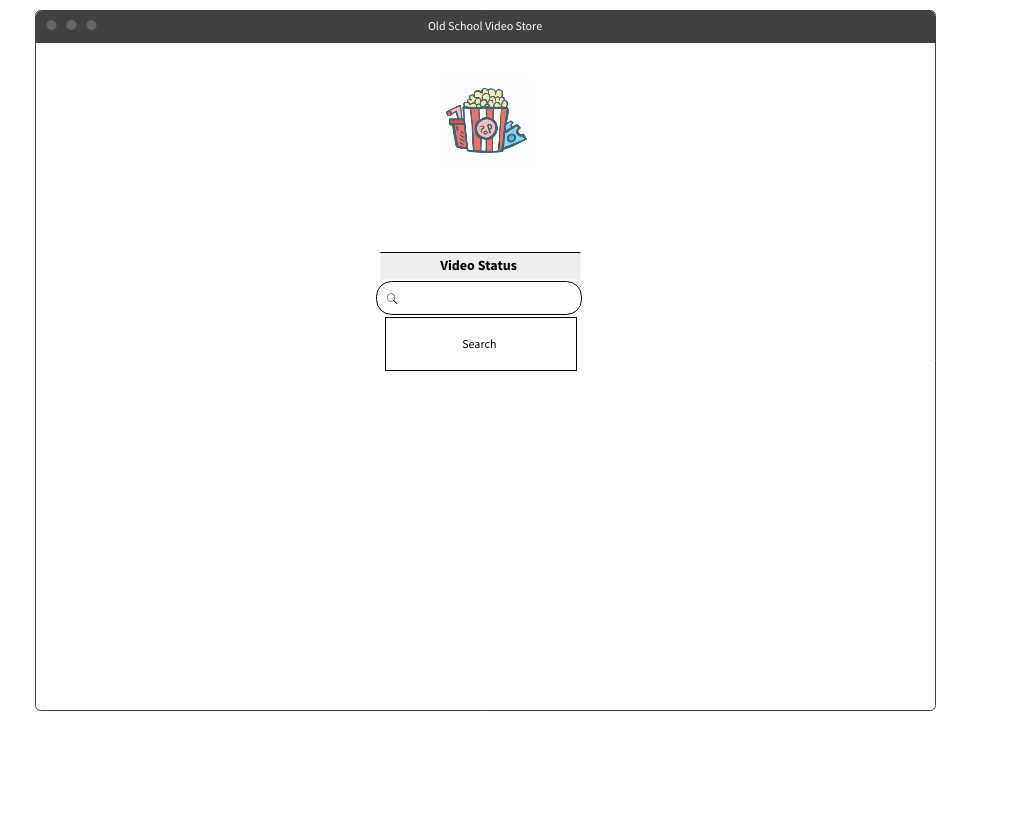
*INSERT INTO UserInfo.Users(FIRST\_NAME, LAST\_NAME, BMONTH, BDAY, BYEAR, EMAIL, UPASSWORD)*

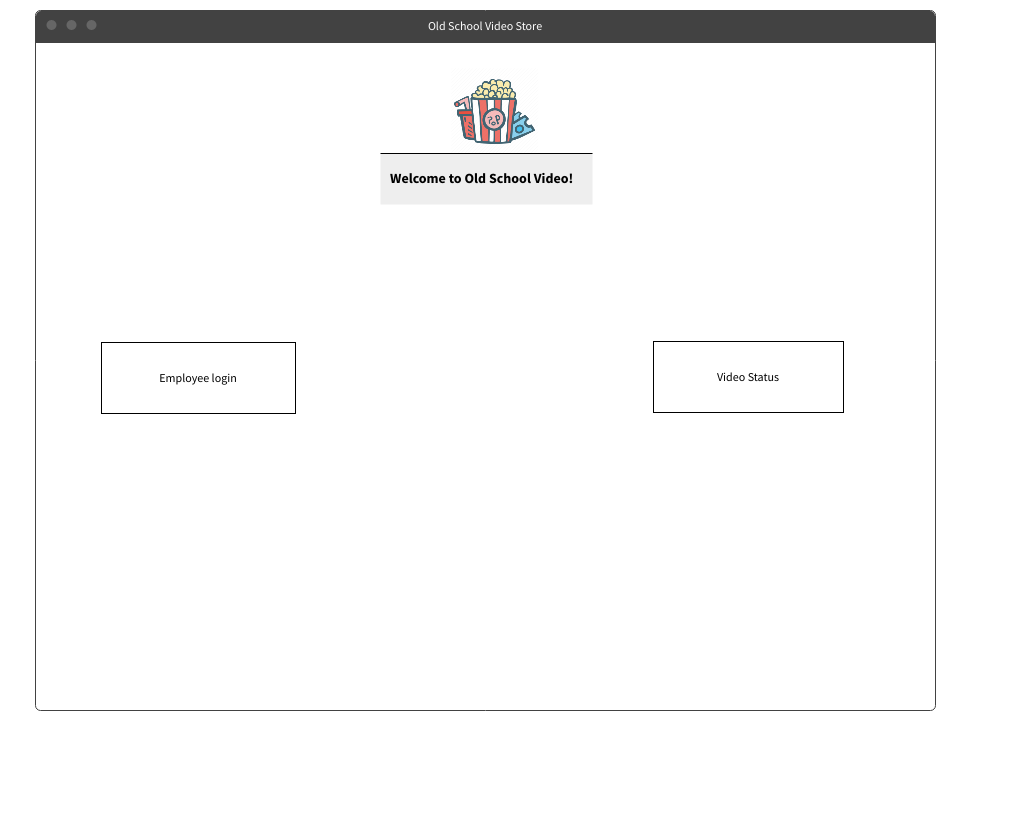
*VALUES('Opal', 'Rehm', 10, 21, 1978, 'ORehm@example.com', 'This is Opal Rehm');*

**Sitemap Diagram:**

**

**User Interface Diagrams:**





**Class Diagrams:**

**Graphical user interface, table

Description automatically generated**

**Service API Design:**

*N/A*

**Security Design:**

*N/A*

**Other Documentation:**

*N/A At this time there is no other additional documentation.*