510 Final Project phase 2 specs -150 points due date: 12/12 Sat. midnight

**Objective**: To implement your application idea initiated in phase 1 using JavaFX.

**Goal**: To use a JavaFX, a MVC wired framework, to accomplish your objective.

**Instructions**: Implement your application idea for your project

Basically the implementation of your app idea will follow a MVC style format to include at minimum the following:

4 classes

A database with at minimum 2 tables ( including a user table for logins)

If you are working in groups of two, include an abstract base class to be and an additonal table for your database

**Steps**:

Create a JavaFX project and add packages naming them controllers, models and views respectvely which serves as your MVC format.

Place files into your packages that make logical and organizational sense. Coding will be easier this way.

Example of an MVC format for a Bank.

**Models** (Bank domain) can be -Transactions, Customer, Users, Bank, Account

**Controllers** can be – LoginController, CustomerController, AdminController

**Views** can be – AddBankView, AdminView, CustomerView, LoginView, AccountsView

As you can see, it is good to have a similar naming convention to “glue” files together into an MVC style.

Code your program to include database interactons to hold pertinent data for the processing of data into information. Make use of functions to perform various CRUD operations.

Your classes should be organized into logical packages to perform the following deliverables:

**Model** classes should allow for queries to be performed and for data retrieval/storage.

**View** files (.FXML file format) should serve the needs for user interfaces i.e., basic IO needs.

**Controller** classes should help “glue” together, Model classes with View files.

Include the ability for your app to allow for ample multiple user roles. Example: One role should be for a regular user and another will be for an admin/manager.

For credit, include the following actions from within your application:

1. Add at least 10 records to your database added by some user(s).

Example: User(s) creates 10 transactions.

1. Delete one record from your database
2. Update one record from your database

Show snapshots for transactions in the following order:

1. Showing a successful login for a regular user.
2. Show a successful login for an admin.
3. Show a result of all 10 records added to your database from the database itself.
4. Demonstrate sufficient functionality for the regular user. Ex. User enters a deposit and can view their new balance.
5. Show some update performed by the admin for the first record entered into the db.
6. Show the deletion of the last record by the admin entered in the db.
7. Show a snapshot of your remaining records in some columnar view.

Example: Show result of transaction by some user id.

1. Show a snapshot of your user table (i.e., the user login credentials) from the database itself.

Feel free to have any of your snapshots to be from popup dialog boxes or from labels, table views, etc. showing the needed results.

Completion of project for credit. **\***Submit the following to Blackboard and to your repository account.

1. Copy all your project files into a pdf.
2. Include a pdf documentation file with the following:

-An abstract description of your app.

-Description of which file to execute at startup and what any **login credentials** are for regular users and for the admin.

-Sufficient snapshots of running app, labeled appropriately.

3. Include a working .**jar** file of your main application file.

4. Include a brief video of your working application. For teams, each member must submit a video.

of their portion of work.

**\***Submission of project files into BB or GitHub/Gitee can be submitted by one team member. Make sure your names for the group are included in a title page.

**\*\***Extra Credit options: to 15 max points given the following choices

1. Outstanding GUI (Charts, CSS style effects, menus, user interactions, functionality) (+5)
2. Class diagrams / Entity Relational Diagrams (+10)
3. Stored procedures (+10)
4. Rollbacks and commits (+5)
5. Password protection (password hashing) (+10)

**\*\***Make sure to include snapshots for EC, labeled appropriately.

Use the following credentials for your DB connectivity

url = "**jdbc:mysql://www.papademas.net:3307/510fp?autoReconnect=true&useSSL=false**";

username = "**fp510**";

password = "**510**";

Grading rubric:

|  |  |
| --- | --- |
| Criteria | 150 Points |
| Deployment / Database Creativity | **25 points** |
| Presentation video/Powerpoint | **25 points** |
| Code | **65 points** |
| GitHub | **10 points** |
| Report Documentation & Snapshots | **25 points** |
| Totals | **150 points** |
| Extra Credit max additional points | **15 points** |