Braxton Harris

IST 411, Section 001

Professor Reitter

03/22/2018

Project Proposal

GitHub URL: https://bwh5316.github.io/CashShare/

Description

My proposed project idea is to create a money-sharing / retrieving application, as suggested by one of the examples presented in class on February 20th. The idea is to allow users to send money electronically to a central storage (server) location, similar to PayPal, where another user may retrieve the funds vs sending money from one bank account to the next. First, users will create an account with the following information:

- First Name
- Last Name
- Date of Birth (DOB)
- Gender
- Username
- Password
- Email Address

After an account has been created, users will login using only the username and password that they created. Once logged in, the following options will be available for selection:

- Profile - Edit bio and/or view profile
- Request Funds – Send another user a request for a specified amount
- Send Funds – Send another user money with a specified amount
- Add Friend – Add a user by their username
- Settings – Edit privacy, notifications, and/or personal information
- Log Out – Log out of account

Beyond the above selections, I do not know what else I will implement but they are what I am expecting to have. They may also be subject to revisions and be expanded upon over time.

Design

The design for my project is expected to be simplistic but appealing. I want the design to be user-friendly so many users can easily understand and navigate while also providing users with an appealing and consistent design. The login portal is the initial view any user will see. It will show the project name, Cash Share, as the header and the following text fields:

- Username (enter username)
- Password (enter password)

The dashboard / homepage will be displayed after a user successfully signs in. This page is where the selections mentioned above (Profile, Add Friend, Settings, etc.) will be available for utilization.

Technology-wise, I will use Spring MVC as I am fairly comfortable with it. I would also like to use Java FXML as I am familiar with it and I am sure that I can enhance the appearance and capabilities of my project using the language. I would use Java FX's Scene Builder to feasibly arrange my views. It is hands-on for repositioning buttons and other objects without having to use hard-code. The problem may be that I cannot integrate FXML with Spring MVC or Spring Boot, but I am doing some research. If not, then I will only be using Spring MVC. The technologies I EXPECT to use are as follows:

- NetBeans IDE
- Scene Builder

NetBeans and Scene Builder can be integrated as I have used them both before. My main concern is that using Spring MVC while trying to integrate a JavaFX FXML project would bring complications may not work. With Scene Builder, work made to a Java FXML project can be exported and then imported into NetBeans. For more details on the integration of NetBeans and Scene Builder, refer to the following link:

https://docs.oracle.com/javafx/scenebuilder/1/use_java_ides/sb-with-nb.html

Schedule

My expectations for each checkpoint, while there will be three checkpoints before completion of the project, are as follows:

1. **Checkpoint 1:** Create 'Login' portal view (Username and Password text fields, submit button, etc.) and construct the layout for the dashboard / homepage. This may also include some working functionality of the buttons such redirection of pages. This may be subject to be completed by Checkpoint 2 based upon potential complications.
2. **Checkpoint 2:** Create and implement a database. Store sample user information, create and test sample users with login authentications. Familiarize myself with Map and Reduce and work on creating some usages for sorting and organizing information with such. Ensure that the dashboard / homepage is fully functional.
3. **Checkpoint 3:** Ensure that users can create accounts and login effectively (authentication = true), all buttons perform specified actions, ensure that the all the views a theme (consistent colors, fonts, etc.), ensure that each page has its own URL, set up the notifications -- alert a user that they have been sent funds.

Once I approach the final stages of the project, it is expected to look similar to Venmo or PayPal (if you are familiar with both). It is still early so my speculations will be continuously changing and updated.

Justification

My project's idea may be interesting as it may create convenience for many people who wish to send their family and/or friends money without the hassle of physical exchange. This will save time and energy for many people.

I don't think my project is drastically different than other existing platforms as there are already many money-sharing platforms on the market, but I plan to differentiate mine as much as physically possible. The goal is to make the design, the privileges, the prompts, and the overall project as innovative and different as possible.

I think the overall complexity of the project is fair. I will create room for me to learn and better understand many different frameworks and topics of programming. I will take some of the topics covered in class and use them. I believe I will become more comfortable with working on server-based programs, databases, Spring MVC, HTML, etc. I will also learn how to focus on human-computer interactions. There is definitely a lot of things to be learned!