

## Group 11 T03 CPSC 233 Project Proposal

### Category: Form Based Application

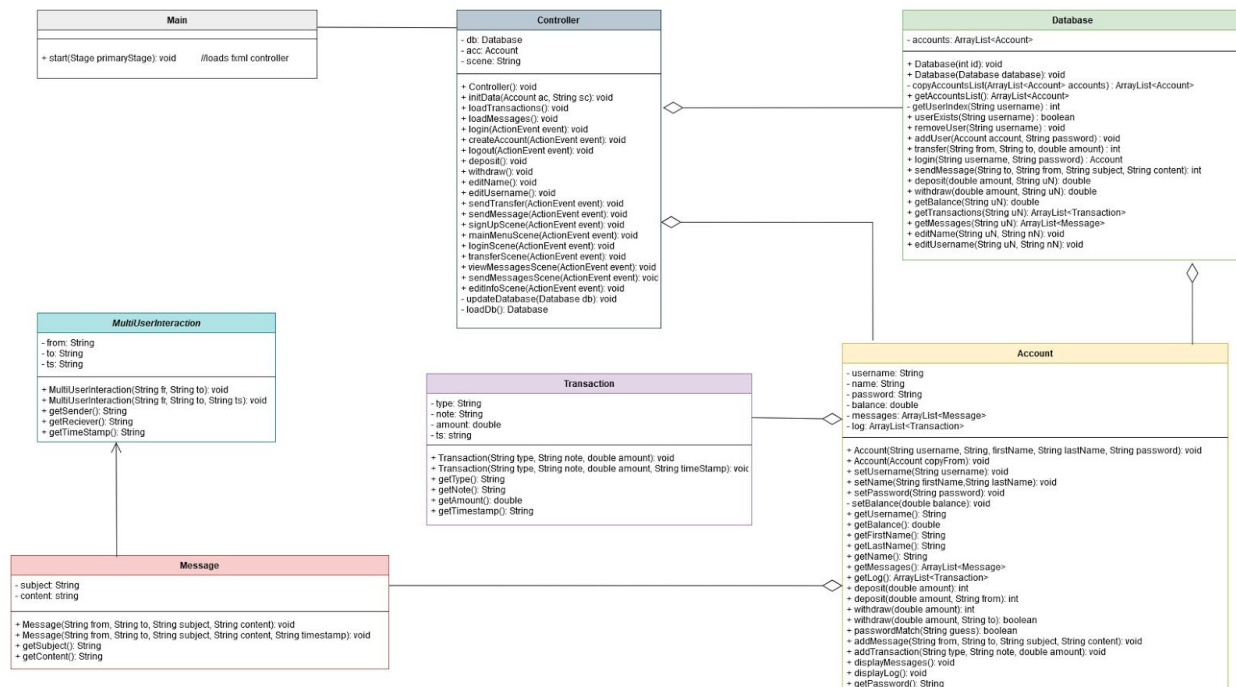
We plan to create an application that simulates online banking. Features include the transfer of virtual money between users, messaging, as well as a virtual deposit and withdrawal system.

Users will be able to manage the money within their account by manually entering the value of which they would like to deposit or withdraw.

Each user will have a unique banking number used to identify them which will be auto generated upon account creation. As well, they will be required to enter a unique username, and a password. Users will be accessible to one another through the use of a search bar; this lookup will allow users to send money to other users similar to an e-transfer. Error checking will be implemented to ensure users can't withdraw, send, or pay bills without the proper funds in their account. There will be an inbox type messaging service also available in which users can send messages to each other's inboxes. A log will also be available that records recent actions performed on a users account

Public information will include A user's first and last name, as well as their unique username. Private items will include balance, banking number, and password.

### T03 Group 11 UML Diagram



\*\*\*\*\*This UML was uploaded to the D2L Dropbox for your viewing convenience.\*\*\*\*\*

#### Demo 1:

For the first demo we will have a fully working text based version of the application. We will walk through how to create/edit accounts as well as how to use the many features we have implemented once an account has been created. Since we do not have access to a GUI we have created user friendly console menus that will allow us to clearly demonstrate the program without needing a graphical display. For the demo we also plan to demonstrate how two accounts will be able to interact with each other through messaging and funds transfers. After this demo we will outline our plan for the future including GUI implementation and then open the floor to any questions.

#### Demo 2:

For the second demo we will have completed and implemented the GUI interface of the application. Similar to the first demo, we will walk through all features: creating/editing an account, messaging and transfers between users, as well as deposits and withdrawals—this time focusing on the GUI rather than functionality. We will cover how we connected the new front end to our existing back end code, and outline adjustments that were required, should there be any. After this, we will briefly discuss our plans to save changes made to the database and its objects during runtime, and read saved data into the database from a save file. We will then open the floor to any questions.

#### Demo 3:

For the third demo we will have completed and implemented data loading and saving into our application. We will create multiple accounts with different stored values (ie. name, account balance, messages), and then close the program. Once opened for a second time, we will show that all previously created accounts are still accessible, and have maintained state. We will discuss the increased usability that data storage allows, and show how our program goes about writing to files. We will cover how we integrated this new functionality into our existing application, outlining any adjustments that were required, should there be any. We will briefly go over difficulties that we faced throughout the creation of this application, as well as possible adjustments that were made—should there be any—from our initial outline of the program, and why said adjustments were made. We will then open the floor to any questions.