Group 9 Project Proposal

We propose to make a java application to replace the current ATM applications. Our ideas stem from the fact that most ATMs that are used internationally either use a microsoft operating system or a linux based system. So we would like to have a portable software application that would be able to be used by any system with only minor changes necessary. We would also like to create a standardized framework that ATMs can follow when deployed, making them easier to make and use for users.

This application would have the same functionality of an ATM as well as some other features. The user shall be able to check an account balance, make a withdraw from their account, and transfer money between accounts. We will also be making the personal identification number accept entries from both a keypad and from a virtual keypad displayed by the graphical user interface. This is to increase user security because then a dummy keypad cannot be used to gather personal information. We also plan on adding an international exchange rate to allow for transfers from two different currencies. There will also be a concept of ownership of the ATM, as it will add a transaction fee to accounts from banks that differ from the ATM’s owner.

There are two main pieces that are required for this project. The main focus of the first phase will be setting up a database and having it connect with a basic GUI. There we will do our best to ensure a safe and secure connection between the database and the application. The second piece will be setting up a more refined GUI. Should we choose to further develop this application after the semester, we could integrate it with hardware, such as a physical ATM, and have it connect to bank accounts.

During the first phase we will set up our database and establish the code to make a connection as well as send and receive information from the database as well as set up our functions that the GUI will use. During this time we will also set up a basic GUI so that we can properly test the unseen functions of our application. Also when testing this phase, we will also show the users the database so they understand how the system fully operates, and that it does.

The second phase will focus of refining the GUI to allow for the recognizable and easy use that is consistent with that of the current ATM that we are all used to. We will also implement an output to show that our application can implement printers as well as input devices for both checks and paper money at a later date.

The main skeleton of the application would consist of the GUI, a back end containing Java code to interact with the other parts of the application, and a SQL database. The GUI would be a user friendly interface that each user interacts with when they want to do a transaction, and all of their information is stored in a separate database. This database is accessed through Java code to assist in the integration, and will also establish an internet connection to retrieve exchange rates between currencies.

As for experience for each of the team members, while most team member have prior experience with high level programming languages, and basic database management, integrating them and adding on additional features may prove to be somewhat of a challenge since our experience is limited, particularly with portable applications. However much of our previous knowledge will be helpful in integrating the system as a whole.

With feasibility, on the surface the application would require several components, such as the GUI and database, to be created independently then integrated through different ways like JDBC. Within the scope of the course, the entire project would easily be created throughout the span of the semester, with room to increase functionality on. For example, within the course we wouldn’t have access to the technology and licensing to make a credit card scanner an easier way of input, but could be added in the future. Going back to the course, the first phase of implementation would include the majority of the back end and a basic GUI such that users can interact with the system. The next phase would be focused on adding more functionality in the front end that users can use and making it possible to be deployed, such as making a mobile version of the application. This way the ATM application could be accessed through more than one method.

When it comes to risks of the project, the biggest issue that could be noticed would be that we don’t have access to each bank account or hardware involved with an ATM, so we wouldn’t be able to actually withdraw money or deposit it. However, within the scope of this class, the intent is to create a framework that could become standardised for ATMs and make it easily deployable. However, this makes testing of the system in a real world aspect more difficult, since all we would have in the database are cards linked with accounts that the designers would manually input. However, when the system would get tested, we would supply the users with the necessary information.