Project Plan

Team Sudoers

Raysean Jones-Dent, Tonye Andre Martial, Matthew Mitchell, Kristine Dudley, Woo Choi, Justin Kim

March 31, 2019

CMSC 495 7982

Dr. Janell Robinson

University of Maryland University College

**Table of Contents**

[Introduction 3](#_Toc4950864)

[Requirement Specifications 3](#_Toc4950865)

[Systems Specifications 4](#_Toc4950866)

[Software Management 4](#_Toc4950867)

[Functionality 4](#_Toc4950868)

[User interface 4](#_Toc4950869)

[Miscellaneous 5](#_Toc4950870)

[Languages 5](#_Toc4950871)

[Security 5](#_Toc4950872)

[Team Roles 6](#_Toc4950873)

[Key Milestones 7](#_Toc4950874)

# Introduction

The goal of this Project Plan is to clearly define our project goals and specifications. This includes all project functionalities and design aesthetics. Our team will develop an Automated Teller Machine (ATM) application with cutting-edge properties, reliable security measures, and multiple language options.

# Requirement Specifications

Our ATM project will simulate a virtual ATM machine where users may conduct ATM transactions such as: withdrawing, depositing, and transferring money. Our GUI application will be developed using Java 8.0. The program will start with a login page where the user may input their phone number, PIN, account number, and last 4 digits of their SSN. The homepage will also provide an option to “Change Language” and “Create New Account”. Once logged in, the GUI will display the balances for both the Checking and Savings accounts. This menu will have buttons where the user may select the transactions listed above as well as an option to “View Transaction History” or “Logout”.

Our program will include a .xml file with a list of active users and their information so that we can effectively test our program. When a new account is created, it will add the new user’s information onto the .xml file. When logging in with an existing user, the program will import that user’s data from the .xml file in the project folder. We will also conduct tests on various scenarios which will utilize all the program’s functions. This will include stress tests such as multiple transactions.

# Systems Specifications

Our GUI application will be developed using Java 8.0 with the NetBeans 8.2 IDE. We will use the JavaFX platform to build the user interface for the program. The documentation will be written via Microsoft Word. We will also use Google Drive and GitHub’s repository to share work amongst the team members.

# Software Management

Version control will be managed through our group’s GitHub repository:  
 <https://github.com/team-sudoers>

# Functionality

## User interface

* Home page functions for currently logged in user
  + How does the user pick the account they want to perform the transaction on?
    - Banking function buttons will have a prompt when selected
  + How many accounts can we display?
    - Checking and savings
  + What exact banking functions are allowed?
    - Two account types (checking and savings)
    - Balance inquiry (visible at log-in)
    - Withdrawal (checking and savings)
    - Deposit (checking and savings)
    - Transfer
* Message prompts
  + What gets displayed after the user performs an account function? Return to homepage?  Display some status prompt?
    - Prompt will ask which account you want to use
* Input methods
  + Should we create an on-screen virtual keyboard or is touch/mouse and keyboard acceptable?  Would have to consider multiple languages if we choose an onscreen keyboard.
    - Use mouse and keyboard

## Miscellaneous

* Receipts
  + After a transaction, the system will provide an option to select email and/or text receipts.

## Languages

* English, Spanish, Chinese, French, German, Korean
* Translations will be made using Google Translate

## Security

* PIN
* The program should obscure the PIN before storing in the XML file using an industry standard hash function such as SHA256.
* Inactivity Timeout
* The program should have an “inactivity timeout” for logged in users.  When the inactivity period expires, the program will automatically display a warning message. If the user does not select the option to continue or chooses to log out, the system will log off and return to the login page.
  + Prompt that will remind you that you are about to timeout. Should ask if you want to continue or exit.
  + 30 seconds of inactivity will trigger the “inactivity timeout” function.

# Team Roles

Weekly leads have been assigned and can be seen in the Key Milestones table. Work will be evenly distributed throughout the project and each member will maintain constant communication with the rest of the team. Weekly meetings will be held every Tuesday night via Skype.

# Key Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Start Date** | **End Date** | **Lead** |
| **Develop Project Plan- Establish requirements, determine weekly leads & roles** | March 25, 2019 | March 31, 2019 | Justin |
| **Develop User Guide & Test Plan, establish test cases** | April 1, 2019 | April 7, 2019 | Kris |
| **Peer Review 1** | April 1, 2019 | April 7, 2019 | N/A |
| **Project Design- Develop coding framework & UI design, UML class diagram, XML entities** | April 8, 2019 | April 14, 2019 | Dre |
| **Peer Review 2** | April 15, 2019 | April 21, 2019 | N/A |
| **Phase 1 Source- Begin initial GUI coding & documentation** | April 15, 2019 | April 21, 2019 | Ray |
| **Phase 2 Source- Complete GUI & begin coding backend (ATM transactions, language setting)** | April 22, 2019 | April 28, 2019 | Matt |
| **Phase 3 Source- Finalize system specifications, begin testing & debugging.** | April 29, 2019 | May 6, 2019 | Woo |
| **Peer Review 3** | May 7, 2019 | May 12, 2019 | N/A |
| **Final Deliverables- Organize documentation, finalize deliverables (code, citations, etc.)** | May 7, 2019 | May 12, 2019 | Justin |