Peer Review 1

Raysean Jones-Dent, Tonye Andre Martial, Matthew Mitchell,

Kristine Dudley, Woo Choi, Justin Kim

7 April, 2019

CMSC 495 7982

Dr. Janell Robinson

University of Maryland University College

## Student 1

1. Will anything be done for security? Information such as SSN could be sensitive and harm the person if it gets breached.
   * This feature has been removed.
2. Will the account history have any filters or searches? For example, if I wanted to search for a transaction of a certain amount of all transaction in the past 7/30 days, will there be any way to do that?
   * The only account history that would be shown at the ATM would be the last 7 days.
3. Finally, will users be able to change their username or password?
   * This would be something that should be done on a mobile banking site and not at the ATM.

## Student 2

1. If I can make some suggestions, just for the readability, I would try to start certain sections in an entirely new page, like when the Test Plans table starts or section on How to Transfer.
   * We will add page breaks to keep sections together.
2. I would try to add a confirmation pages for deposit, withdrawal and transfer to make it more realistic since most ATMs have that feature just as a final check.
   * We had discussed this, but the message boxes were forgotten on the initial user guide. They have been added.

## Group 1

### User Guide:

1. When the user completes a transaction such as a deposit, transfer, or withdrawal, there is a message displayed if the transaction fails but no message is displayed if the transaction is successful.  It would make sense for there to be a message stating “Transaction Successful” on successful completion of any transaction.
   * We will be adding a transaction successful pop-up.
2. For the “How to Check the Balance” section, some more clarity could be added rather than just stating: “the screen.”  This could be replaced with “the Main Menu screen” or something to that effect.
   * We think this is pretty clear.
3. For the sake of security since this is a financial application, it would make sense for the user to be locked out of attempting to log-in for a short time on some amount of failed log-in attempts.  For example, maybe after 5 failed attempts, the user should have to wait 5 minutes to try again.
   * This is a good point. We will add that after 3 failed attempts the user will be locked out.

Test Plan:

1. If you are going to include a feature to change the language, then you will need to test the entire application in every optional language to make sure all of the dialogs are correctly displayed for each language.  It is not clear from the guide how many languages are going to be available, but they should all be tested. Any step where the expected result is some sort of dialog being displayed should be tested in every available language. Also, the mechanism that changes the current language should be tested to make sure it functions as expected.
   * Yes, we will be doing all tests in every language we include. The Login page picture was not correctly added. That will be addressed, which will have the languages on it.
2. The non-numeric account balance tests are not clear to me.  How is it possible that the account balance could be non-numeric?  This does not seem to be a value that should be editable by the user.  I would think a non-numeric transfer amount test would make more sense.
   * This would be something that a teller would have entered and is stored in the XML file. This test is to ensure that only numbers are read and not letters.
3. There are some back-end type things that may need to be tested as well, depending on how that piece functions.  For example, if the password is being encrypted in the backend, that could be tested. Since there doesn’t appear to be functionality to create users or change passwords, this recommendation may not apply to this project.
   * There is no server communication at all. No admin functions to test. It is assumed that the XML file already has the passwords encrypted
4. This may be beyond the scope of this project, but if this is meant to be a multi-user application, then it would make sense to test two users (such as co-owners of the account) logging into the same account at the same time.  Should the second user be blocked and not allowed to log-in? If not, what happens if one of the users completes a withdrawal and then the other also completes a withdrawal shortly after? Does the application recheck the balance at the time of transaction or does it rely on the stored value of the balance which is displaying on the screen for the user (potentially allowing the balance to go into the negative)?
   * This is good feedback but it’s beyond the scope of this project.

## The Friday Group

### Application Overview

1. Provide a Description of What this Application Is

What sort of application is this? It appears to be a mix of an ATM and some sort of web application. Do ATMs have an authentication screen, requiring a username and password? Can you withdraw physical money from a bank web application? A clear description of this application would be very helpful.

* We included this with the Project Plan. However, we will rename the username/password fields to avoid further confusion of a cardless ATM machine.

1. Unknown First Name on Login Screen

On the login screen, consider removing "Welcome [User First Name]" since the system cannot know the first name until the user clicks the login button.

* This was just an added visual to the User Guide and would obviously not start with welcoming the user that is signing in.

### Application Security

1. Use security question instead of SSN

Create a way for users to answer a pre-determined security question. Give multiple set questions to answer such as the street they grew up on.

* This is an ATM simulator not a web application.

1. Never use Personally Identifiable Information (PII) for Account Security

You have chosen to use the last four digits of one’s social security number as part of account security. What was the thought process behind this decision? What will you do if a customer doesn’t have an SSN? Why are you storing an SSN in plaintext in your XML file? Consider using a six digit randomly selected number for MFA.

* These are valid points. True MFA is beyond the scope of this project, but would be a useful requirement to add for a real ATM. We have removed the SSN MFA feature.

1. Perform Some Threat Modeling

I do not see any sort of threat modeling as part of the Test Plan. ATMs/financial web applications are regular targets for attackers, for obvious reasons. Performing some modeling specific to what this application is. Identify the threats and show mitigation steps.

* Will discuss at next meeting.

1. Adhere to the PCI Standards

The Payment Card Industry (PCI) has standards for financial transactions. PCI has ATM specific security standards. Follow the guidance in those standards.

* This is beyond the scope of this project.

1. Perform Logging

I do not see anything addressing logging in the Test Plan. Logging should always be part of securing software. Logging needs to be performed both for security issues, but also for internal and external auditing.

* For the scope of this project the account history is the audit log.

1. Identify Which Sort of Sensitive Data is to be Retained

Only the minimum amount of sensitive data needed to perform a transaction should be retained. The group should identify what that data is.

* Per the project requirements the first and last name will be kept for the greeting. The email and phone number are not required and could be taken out.

1. Use an Account Protection Mechanism

There doesn’t seem to be anything that protects an account after a specified number of authentication failures. For a financial application there should be an account lockout, the need to change a user’s password, etc.

* We have decided that after 3 attempts the user will be locked out.

1. Clarify the MFA to be Used

I don’t see anything about how the MFA works, other than using the last four SSN digits. Is this needed for an ATM software?

* This feature will be removed, since true MFA is beyond the scope of this project. It has since been removed.

1. Securely Transmit Data

I don’t see anything about how data is transferred. Strong cryptography should be used to transmit data. If nothing else, list how data will be securely transferred.

* Server communication is beyond the scope of the project. It is assumed the XML file will exist locally.

1. Use a Recognized Cryptographic Algorithm

What sort of algorithm is the group using for this application? The group should use an algorithm that is recognized by government/industry and note the algorithm in use.

* Already addressed in the Project Requirements

1. Security Testing

I did not see any testing for security involved in the Test Plan. When dealing with such personally identifiable information (PII) such as a Social Security Number or bank account information high level security measures must be taken to protect user information from attacks. Some more information about the security testing would benefit the overall product. Guidance may look to industry standard influence such as those set by the Bank Secrecy Act (BSA), Regulation E (the Electronic Fund Transfer Act), and the overarching Bank Protection Act (BPA).

* For security testing we have identified the test cases for login, but the BSA and BPA standards are beyond the scope of this project.

1. Personally Identifiable Information Should Not Be Stored in the XML File

By having the last four digits of a user’s Social Security Number stored in the XML file your security is not only in danger, but you have made identity theft of your users that much easier.

* Refer to response in 2.2.2

1. Users Should Be “Locked Out” After a Certain Number of Failed Login Attempts

I see that the failed SSN authentication through the two-factor identification will send the user back to the login, but there should be some sort of maximum attempts. Otherwise a brute-force attack is not only effective, but imminent. By establishing a “locked out” period, those types of simple attacks can be made nearly obsolete.

* Refer to response in 2.2.7

1. Logout on Close

I imagine the program will log a user off if the application is closed mid-session. This may be worth mentioning in the User Guide in the “How to Logout” section, and even worth including on the Test Plans possibly. If the program does not logout on close, then account information could be at risk when another user tries to log in from the same ISP.

* There is no server communication, session management is beyond the scope of this project. Once the application is closed the user is logged out.

1. SSN for MFA

Use of SSN seems a bit questionable for MFA. Possibly use something different?

* Refer to response in 2.2.8

### Application Functional/Non-Functional Items

1. Add Overflow Test Case

Add a test case for putting in large numbers and characters that could cause overflow errors in transactions or login errors in login fields.

* This will be an added test case.

1. Add Test for Withdrawal Amount

Add a test case for attempting to withdraw more money than is in the ATM at the time.

* This will be an added test case.

1. Use Some Sort of Transaction Identifier

I see that there is a date/timestamp being used for each of the transactions in the XML file, but there should be an identifier beyond just the date/timestamp. In a large application there would be multiple transactions that could all have the same date/timestamp, so being able to clearly identify each transaction is very important.

* This will be discussed at a later design meeting.

1. Test Withdraw, Transfer, and Deposit of a Negative Number

TODO: Add attempted withdraw, transfer, and deposit of a negative number to your Test Plans. While your test cases include the attempt of a non-numeric entry, the testing of a negative number does not appear in you test cases.

* This will be an added test case.

1. Number Format

Will there be a restriction for deposit and transfer amounts to no more than 2 decimals?

* Yes, there will be. This is going to be added the requirements.

1. Test Case to Simulate Failure Mid-Transaction

Transferring money usually entails withdraw from one account and deposit into another (or in reverse).  It is important to test a failure case when the system fails between these two operations. It's suggested to add a test case that simulates failure mid-transaction to verify that the transfer operation is atomic.

* Since all the data is stored locally in a XML file, a failure mid-transaction is beyond the scope of this program.

### Document Formatting

1. Remove the Contents Listing “Sudoers’ User’s Guide and Test Plans”

TODO: Removing the Contents listing, “Sudoers’ User’s Guide and Test Plans,” the document still clearly contains both your User’s Guide and your Test Plans. You may also choose to remove the header on the page 3 similarly labeled “Sudoers’ User’s Guide and Test Plans.”

* This has been removed.

1. Increase Test Case Readability Using Defined Borders

TODO: Use a table with defined borders to separate your Test Plan cases.  Some of the Test Case text runs together, making it hard to decipher when one test ends and the other begins.

* Will make the table more readable by adding double horizontal lines to separate test cases.

1. Test Case readability

I would suggest using borders for the table (at least horizontal ones) in order to increase readability.

* Refer to response in 2.4.2