|  |
| --- |
| CS 307 – Software Engineering |
| ATM 2.0 |
| Project Requirements Backlog |

|  |
| --- |
| **Team 25**  Anthony Goeckner  Krutarth Rao  Austin Reed  Harold Smith  9-5-2016 |

**Problem Statement**

Currently, automated teller machines (ATMs) are subject to fraud by use of stolen bank cards and information. Our solution, the ATM 2.0, involves the use of three-point biometric and traditional PIN authentication, which will effectively negate this risk by requiring fingerprints and facial recognition in order to dispense money. Such ATM systems are not used commercially in the United States at this time.

**Background Information**

ATM technology reached a plateau in 1970’s. The machines have continued to use PIN and card to authenticate a user. During the same time fraudsters and hackers have continued to grow the arsenal of tools they have available to penetrate these systems.

**Audience**

ATM2.0 hopes to provide banks a more secure way to dispense cash to its customers. The users of the ATM and the bank will have an edge over criminals conducting ATM fraud. Through 3-point biometric authentication, the bank can be sure that the intended customer is collecting the cash and the customer can be sure that no adversary can withdraw cash from their bank account without their presence.

**Similar Applications**

Conventional ATM machines exist throughout the world. Currently the machines are owned and operated by the banks. Although the existing machines serve the same purpose i.e. withdraw cash, check account balance, print receipt, etc. we aim to provide this functionality in a more secure way to reduce loss to the bank or the customer through fraud/theft.

**Existing Limitations**

Conventional ATM machines are limited by their authentication techniques. Originally ATMs were created because banks wanted to have tellers 24 hours a day, this was proving to be quite expensive. Keeping this in mind, current ATM machines should provide at least as much security as a human teller. However, a fraudster today only needs a PIN that he/she can acquire through a multitude of attack vectors on the user’s digital data and the card can be acquired through physical theft. Then on, if the thief were to walk up to a human teller with a card and PIN, a teller would be able to detect that the person doesn’t match the photo id on record and prevent any fraudulent withdrawals. The machines don’t provide this level of security but ATM2.0 does through facial recognition.

**Functional Requirements**

* As a user, I need to withdraw specific amounts of money.
* As a user, I would like a touch based interface.
* As a user, I need an intuitive interface.
* As a user, I need to deposit money.
* As a user, I need to manage my security preferences, such as enabling/disabling biometric authentication.
* As a user, I need to view my account balance.
* As a user, I should be able to set my PIN.
* As a user, I should be able to enter a backup password received from the bank.
* As a user, I need to transfer funds between accounts.
* As a user, I would like to select my language.
* As a user, I would like a “Support” button in case I run into problems.
* As a user, I would like the ability to print a receipt.
* As a user, I would like the ability to make multiple transactions.
* As a user, I need the ability to log out of my account.
* As a user, I need the ability to cancel a transaction.
* As a user, I would like the ability to withdraw using a “fast cash” feature.
* As a user, I would like the ability to set my “fast cash” amount.
* As a developer, I would like to have code that can be easily understood
* As a developer, I would like code that is built to be updated later
* As a developer, I would like an easily accessible source control system.
* As a developer, I would like to use Triple DES level encryption on the authentication data collected from the user before being sent to the network layer.
* As a developer, I would like USB ports disabled to prevent unauthorized access.
* As a developer, I would like Windows to be the operating software for interoperability with Azure.
* As a developer, I would like the connection between the Raspberry pi and the server to be over a VPN.
* As a developer, I would like a robust and customizable framework for the graphics user interface.
* As a developer, I would my customer database to be easily scaleable.
* As a developer, I would like to create an interface for the bank to add/modify customers details in the customer database.
* As a developer, I would like to use Azure services to host my customer database, authentication protocol and banking interface.
* As a developer, I would like to use C# and LINQ queries to modify my database.
* As a customer, I should be able to provide a backup passwords in case the hardware fails and the user of the ATM machine calls.
* As a customer, I should be able to perform routine checks on each hardware device individually for
* As a customer, I need the software to run efficiently
* As a customer, I would like ATM 2.0 to be compatible with standard ATMs.
* As a customer, I would like my logo to be displayed on the screen.
* As a customer, I would like remote terminal to the machine.

**Non-Functional Requirements**

* blah