Requirements Engineering Document

Group 11

Alexander Wojtowicz-Forte – 500658355

Christopher Papanagiotou – 500613906

Edgar Safaryan – 500579814

Jermaine Ganado – 500624506

Joseph Chan – 500583984

Maaz Jawed – 500591818

Febuary 2, 2016

**Table of Contents**

|  |  |
| --- | --- |
| Non-Functional Requirements | ................................................................................................. 3 |
| Functional Requirements | ................................................................................................. 4 |
| Constraints | ................................................................................................. 5 |
| Prioritized Requirements | ................................................................................................. 6 |
| User Case Diagram | ................................................................................................. 7 |
| User Cases | ................................................................................................. 8 – 18 |

**Non-Functional Requirements**

**Speed**  
-not too much of an issue, but shouldn’t be able to freeze  
-Should not be too slow either, to allow other customers to use the machine without waiting too long.

**Size**  
-Not an issue either as long as the hardware has enough memory space to accommodate it.

**Ease of use**  
-organize the UI  
-Must be simple and easy to use for people with or without any technical experience.

**Reliability**  
-Should not fail during a transaction.

**Robustness**  
-Do not allow for data corruption. Save nothing unless confirmed proper operation.

**Portability**  
-not required past the single identical line of ATMs

**System Functional Requirements**

**Card reading**  
-system needs to be able to take card info and find a matching account  
-the provided pin must match the found account  
-no more than 5 attempts at entering correct pin before lockout  
 -display a get assistance message

**UI**  
-a welcome screen   
-options:  
 -see account information  
 -withdraw funds  
 -deposit funds  
 - print receipt  
 - transfer funds  
 -quit/end session  
-welcome screen needs a space for bank promos  
-each screen needs a back button  
-on exit there is a prompt for donating to charity

**Data**  
-confirm that cash was dispensed before changing funds when withdrawing  
-be able to use exception handling in case something goes wrong  
 -in which case don’t change the balance or dispense money  
-reduce the users balance by an additional 1% of their withdrawn amount  
-verify that there are enough funds available into the machine to satisfy the customer’s needs

**Security**  
-protect against system cracking  
-do not allow for any kind of withdrawing without a card, sufficient balance, or unauthorized accounts  
-in the case of insufficient funds, tell this to the user and prompt for a new amount. Don’t withdraw  
 -they can input a new amount or quit  
-in some way implement a protection for forced withdraws by muggers. Add a discrete security button that notifies authorities and security (it should do it without displaying any confirmation)

**Records**  
-save a copy of each transaction’s information (receipts) to a file showing amount withdrawn, remaining balance, and data and time along with a transaction number  
-receipts printed to the customer should say something nice

**Constraints**

-phone connection for alert button  
-card or manual identification entry necessary to access system  
-write code in Java, develop using Git

# Prioritized Requirements

#### Essential

* Reliability
* Data
* Security
* Robustness
* Read Card Use Case
* Validate Pin Use Case
* Start ATM Session Use Case
* Perform Transaction Use Case
* Withdraw Funds Use Case
* Deposit Funds Use Case

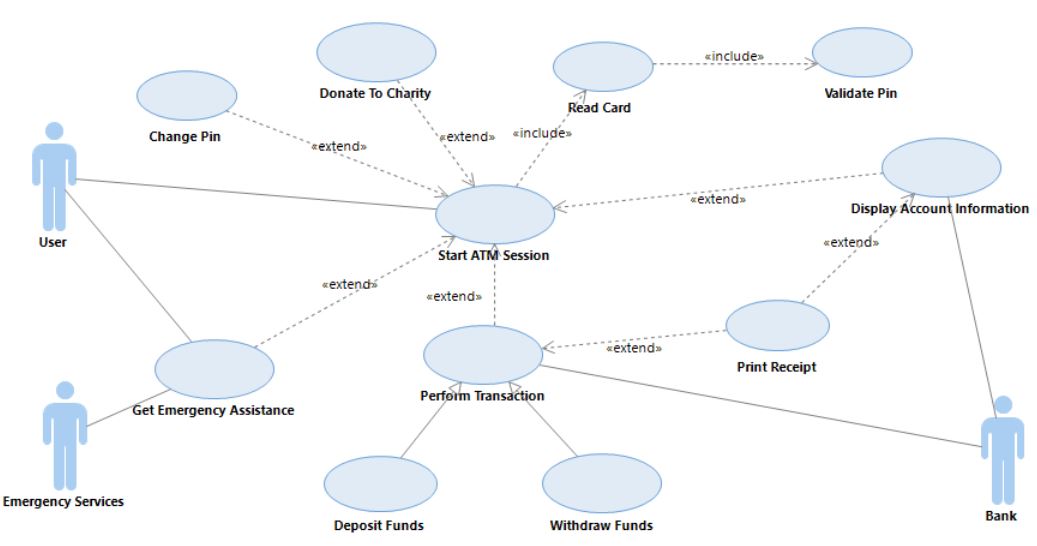
#### Desirable

* Records
* Ease of Use
* UI
* Change Pin Use Case
* Print Receipt Use Case
* Display Account Information Use Case
* Get Emergency Assistance Use Case

#### Optional

* Speed
* Size
* Portability
* Donate To Charity Use Case

**Use Case Diagram**



**PerformTransaction #3**

**Brief Description**

This use-case describes the process to begin and choose a transaction. This use-case is a generalization of [#7](https://github.com/Joey152/cps406_s3_group11_w16/issues/7) Withdraw Funds and [#10](https://github.com/Joey152/cps406_s3_group11_w16/issues/10) Deposit Funds. This use-case also includes [#11](https://github.com/Joey152/cps406_s3_group11_w16/issues/11) Print Receipt.

**Actors**

User(Primary)  
Bank(Secondary)

**Triggers**

User selects perform transaction on screen.

**Flow of Events**

Basic Flow - Perform Transaction  
1. Use-case beings when user selects Perform Transaction operation on screen.  
2. System displays Withdraw Funds and Deposit Funds.  
3. If User selects Withdraw Funds  
 3.1. extend from [#7](https://github.com/Joey152/cps406_s3_group11_w16/issues/7) Withdraw Funds  
4. If User selects Deposit Funds  
 4.1. extend from [#10](https://github.com/Joey152/cps406_s3_group11_w16/issues/10) Deposit Funds  
5. System updates records  
6. System waits for next input

**Alternative Flows and/or Subflows**

a. User selects back or cancel  
b. System returns to promo menu

**Special Requirements**

None

**Preconditions**

Card must be verifed  
User must access system

**Postconditions**

**Success Postcondition(s)**

A transaction is performed

**Failure Postcondition(s)**

A transaction is not performed

**Extension Points**

* [#7](https://github.com/Joey152/cps406_s3_group11_w16/issues/7) Withdraw Funds
* [#10](https://github.com/Joey152/cps406_s3_group11_w16/issues/10) Deposit Funds

# ValidatePin #4

Brief Description: This use case describes the operation of validating the number given from the user with the pin number on the bank card.This use case is included in [#5](https://github.com/Joey152/cps406_s3_group11_w16/issues/5) Read Card use case and includes [#14](https://github.com/Joey152/cps406_s3_group11_w16/issues/14) Get Emergency Assistance.

Actors:  
User(primary)

Triggers: The read card use case calls this use case to validate the pin number in order to access the bank account information of the user.

Flow of events:  
Basic Flow:  
1. This use case starts when the system wants to verify the pin given from the user.  
2. The user enters the pin number through the keypad into the system.  
3. The ATM checks to see if the pin number given from the user matches the pin on the card.  
3.1. If it doesn't match reduce number of tries(number of tries is 5) by one and ask the user to enter a valid pin.  
4. The ATM allows access to the bank account information.

Alternative flows and/or Subflows:  
1. If the user exceeds the number of tries   
1.1 The ATM holds the card  
1.2 Include [#14](https://github.com/Joey152/cps406_s3_group11_w16/issues/14) Get Emergency Assistance

Special Requirements: None

Preconditions: Card is inserted into the ATM

Success Postcondition(s): Card is validated

Failure Postcondition(s):Card is not validated

Extension Points:

Include [#14](https://github.com/Joey152/cps406_s3_group11_w16/issues/14) Get Emergency Assistance

# ReadCard #5

Brief Description: This use case describes the process of reading a bank card.

Actors:  
User (primary)  
Bank (secondary)

Triggers: Called from the perform transaction use case . User wants to perform a transaction.

Flow of events:  
Basic Flow:  
1. The use case starts when the user inserts his bank card into the ATM.  
2. The scanner inside the card reader reads the data from the magnetic stripe.  
3. The reader checks if the card is a valid bank card.  
3.1 If the card is a valid bank card.  
3.1.1 Include [#4](https://github.com/Joey152/cps406_s3_group11_w16/issues/4) Validate Pin  
3.1.2 The ATM then connects and receives the account information from the bank.

Alternative flows and/or Subflows:  
1. If the card is not a valid bank card  
1.1 The card is returned to the user  
1.2 The ATM displays a message on the screen that the user should enter a valid bank card.

Special Requirements: None

Preconditions: User must have a bank card.

Success Postcondition(s): The user successfully gains access to the account information.

Failure Postcondition(s):The user doesn't gain access to the account information.

Extension Points:

Include [#4](https://github.com/Joey152/cps406_s3_group11_w16/issues/4) Validate Pin

# DisplayAccountInformation #6

Brief Description:This use case describes the display account information process.  
  
Actors:User(primary)

Triggers:The user presses the check account information button .

Flow of events:  
Basic Flow:  
1. Use case starts when the user presses the check account information button.  
2. The user's bank account balance is displayed on the screen as well as a check transaction history button ,a print account information and a back button.  
3 If the check transaction history button is pressed.  
3.1 The system displays all the previous transactions done up until now.  
4. If the print account information button is pressed.  
4.1 extend from [#11](https://github.com/Joey152/cps406_s3_group11_w16/issues/11) PrintReceipt   
5. Use case ends when the user presses the back button and returns to the main user interface.

Alternative flows and/or Subflows:

Special Requirements:

Preconditions: User must have gained access into his bank account.

Success Postcondition(s): User gets informed about the state of his bank account.

Failure Postcondition(s): None.

Extension Points:  
This use case is a extention of #11 PrintReceipt  
This use case extends to [#12](https://github.com/Joey152/cps406_s3_group11_w16/issues/12) Start ATM Session.

# WithdrawFunds #7

Brief Description:   
This use case describes the process of withdrawing currency from an account.

Actors:  
User(primary)  
Bank(secondary)

Triggers:  
User chooses the option "Withdraw" from the graphics user interface.

Flow of events:  
Basic Flow  
1. The use case starts when the user chooses to Withdraw from the user interface.  
2. User chooses whether they want to withdraw from their "chequing" or "savings" account.   
3. The user chooses how much money is being withdrawn.  
4. User confirms the withdraw amount.  
5. Money is given to the user from the ABM.

Alternative flows and/or Subflows:  
1. User withdraws more than their current balance.  
2. ABM does not withdraw any money and shows an error message  
--------- or --------  
1. User wants to withdraw an amount that is not a multiple of 20  
2. ABM asks user to only choose to withdraw an amount that is a multiple of 20.

Special Requirements:

* Withdraw amount must be equal or less that total balance on account.
* Withdraw amount must be a multiple of 20.

Preconditions:

* User must have entered the correct PIN.
* User must have either a chequing or savings account.
* User should technically have money in account to withdraw; however if not, a message will display that tells the user that they have insufficient funds.

Success Postcondition(s):   
Successfully withdraw money from the bank account.

Failure Postcondition(s):   
Money is not withdrawn from the account.

Extension Points:  
This use case is a specialization of use case [#3](https://github.com/Joey152/cps406_s3_group11_w16/issues/3) Perform Transaction.

# DepositFunds #10

Brief Description:  
This use case describes the process of depositing currency into account.

Actors:  
User(primary)  
Bank(secondary)

Triggers:  
User chooses the option "Deposit" from the graphics user interface.

Flow of events:  
Basic Flow  
1. The use case starts when the user chooses to Deposit from the user interface.  
2. User chooses whether they want to deposit into their "chequing" or "savings" account.   
3. The user chooses how much money is being deposited.  
4. User confirms the deposit amount.  
5. Money deposits into account.

Alternative flows and/or Subflows:   
None

Special Requirements:

* Cannot deposit coins.

Preconditions:

* User must have entered the correct PIN.
* User must have either a chequing or savings account.

Postconditions:   
None

Success Postcondition(s):  
Successfully deposit money into bank account.

Failure Postcondition(s):  
Money is not deposited to account.

Extension Points:  
This use case is a specialization of use case [#3](https://github.com/Joey152/cps406_s3_group11_w16/issues/3) Perform Transaction.

# PrintReceipt #11

**Brief Description**

This use-case describes the print receipt process. This use case is called from the use cases Perform Transaction and Display Account Information.

**Actors**

User(Primary)  
Bank(Secondary)

**Triggers**

User selects print receipt operation during use case [#3](https://github.com/Joey152/cps406_s3_group11_w16/issues/3) Perform Transaction or [#6](https://github.com/Joey152/cps406_s3_group11_w16/issues/6) Display Account Information.

**Flow of Events**

Basic Flow - Print Receipt  
1. Use-case beings when user selects Print Receipt operation on screen.  
2. System prints out receipt.

**Alternative Flows and/or Subflows**

1. User selects back or cancel  
2. System does not print out receipt

**Special Requirements**

None

**Preconditions**

A transaction needs to occur  
User must access system

Postconditions

**Success Postcondition(s)**

A receipt is printed

**Failure Postcondition(s)**

A receipt is not printed

**Extension Points**

This use case is an extension of use case [#6](https://github.com/Joey152/cps406_s3_group11_w16/issues/6) Display Account Information and use case [#3](https://github.com/Joey152/cps406_s3_group11_w16/issues/3) Perform Transaction.

**StartATMSession #12**

**Brief Description**  
This Use Case Describes the start of the ATM session

**Actors**  
User(Primary)  
Bank (Secondary)

**Triggers**  
The User begins a session

**Flow of Events**  
Basic Flow — Start Up  
1. Use case begins when the user initiates a session with the ATM.  
2. The user is shown a welcome screen with a promotional advertisement.  
3. The user is asked to enter their pin.  
3.1. The user enters the wrong pin and is prompted up to 4 more times to re-enter.  
4. The user is then allowed to select an option.  
5. To end their session, the user selects quit.  
6. Use Case ends with the user selecting quit and being prompted with a donation request.

**Alternative Flows and/or Subflows:** None

**Special Requirements**  
User has a supported bank account

**Preconditions**  
User has card ready.

**Postconditions**  
The session ends and the system is ready to being a new session.

**Success Postcondition(s)**

Successful error free operation of ATM.

**Failure Postcondition(s)**   
Appropriate error message is displayed and the system is locked with all unverified changes not committed.

**Extension Points**  
This Use Case is extended by Use Case [#15](https://github.com/Joey152/cps406_s3_group11_w16/issues/15) Change Pin, [#13](https://github.com/Joey152/cps406_s3_group11_w16/issues/13) Donate To Charity, [#6](https://github.com/Joey152/cps406_s3_group11_w16/issues/6) Display Account Information, [#14](https://github.com/Joey152/cps406_s3_group11_w16/issues/14) Get Emergency Assistance, and [#3](https://github.com/Joey152/cps406_s3_group11_w16/issues/3) Perform Transaction.  
This Use Case includes Use Case [#5](https://github.com/Joey152/cps406_s3_group11_w16/issues/5) Read Card.

# DonateToCharity #13

Brief Description  
This Use Case describes the process of a user making a one dollar donation to charity.

Actors  
User(Primary)

Triggers  
User selects to quit their session

Flow of Events  
Basic Flow — Donation to Charity  
1. Use case begins when a prompt appears when the user selects to end their session.  
2. The user can select whether or not to make a donation to charity of one dollar from.  
2.1. Should the user select no, the use case ends  
3. The user selects yes and are prompted with which account to take the dollar from.  
3.1. Accounts with insufficient funds are not selectable.  
4. Use Case ends with the user selecting an account to donate from.

Alternative Flows and/or Subflows  
1. The user can end the use case by selecting cancel when selecting an account.

Special Requirements  
None

Preconditions  
The user has at least one dollar in

Success Postcondition(s)  
If a donation is made, a dollar is removed from the user's account and is put to charity.  
The user ends their session.

Failure Postcondition(s)  
The session ends

Extension Points  
This use case is an extension from Use Case [#12](https://github.com/Joey152/cps406_s3_group11_w16/issues/12) StartATM

**GetEmergencyAssistance #14**

**Brief Description**

This Use Case describes the means by which a user may contact the authorities.

**Actors**

User(Primary)  
Emergency Services (secondary)

**Triggers**

User presses emergency button on keyboard for emergency services.

**Flow of Events**

Basic Flow — Get Assistance  
1. Use Case begins when the user presses the emergency button.  
2. Use Case ends with plea for assistance sent to the authorities.

**Alternative Flows and/or Subflows**

None

**Special Requirements**

Auto-dialer for emergency services.

**Preconditions**

None

**Postconditions**

No message shall be displayed.

**Success Postcondition(s)**

The authorities will be notified with the location.

**Failure Postcondition(s)**

None

**Extension Points**

This use case extends use case [#12](https://github.com/Joey152/cps406_s3_group11_w16/issues/12) Start ATM Session

# ChangePin #15

**Brief Description**  
This Use Case describes the process of a user changing their pin. This use case is called from the DisplayAccountInformation Use Case.

**Actors**  
User (Primary)  
Bank (Secondary)

**Triggers**  
User selects option to change their pin.

**Flow of Events**  
Basic Flow — Change Pin

1. Use case begins when the user selects Change Pin.  
2. System prompts the user to re-enter their current pin as a security check.  
2.1. User incorrectly enters their existing pin and is re-prompted.  
3. The user correctly enters their pin and the system requests a new one.  
4. The system requests that the user enter their new pin a second time for verification.  
4.1. User incorrectly enters their new pin, returns to c.  
5. Use Case ends with the user correctly re-entering their new pin and the changes are saved.

**Alternative Flows and/or Subflows**  
1. User selects back... Use case ends

**Special Requirements**  
None

**Preconditions**  
User must access system

**Postconditions**

**Success Postcondition(s)**   
The pin number to this account is changed to the new one.

**Failure Postcondition(s)**   
The pin number remains unchanged

**Extension Points**  
This use case is an extension of use case [#12](https://github.com/Joey152/cps406_s3_group11_w16/issues/12) Start ATM Session