**Deliverable 2: Domain analysis and requirements - A textual description of the domain. - A use case diagram of the system. We did not cover use cases in the class because I assume that everybody had taken the prerequisite classes. For those of you who do not know about use cases, this is the chance to learn these concepts. - A list of functional requirements of your system.**

**Overview of the system**

The ATM system is used for cash dispense and other bank services without visiting the bank.

This system would accept a bank’s credit/debit card. After the system reads the card, it detects

the associated bank and then would redirect to the interface of the associated bank only. Apart

from the traditional features supported by the ATM, this system would be supporting additional

security features. The system features are listed as follows:

● A user can withdraw cash.

● A user can change PIN of the card.

● A user can make balance inquiry.

● A user can deposit cash.

● A user will have to go through an extra security check which will include biometric of his

right-thumb, applicable to specific bank if supported.

● Linking several banks into a single system with additional layer of security to protect

security breach within different banks.

● A user can request for receipt.

● A user can transfer funds to another bank account.

● A user can choose option to select language: either English or French.

**UC-1: Get Authentication (for registered user):**

|  |  |
| --- | --- |
| **Number** | UC2 |
| **Name** | Withdraw |
| **Summary** | User can withdraw money from there account |
| **Priority** | 1 |
| **Preconditions** | 1 Validate Card  2 Check Account balance |
| **Post Conditions** | 1 Dispense money  2 Update Account Balance |
| **Primary Actor(s)** | Customer |
| **Supporting Actor(s)** | Bank Server, Bank Database, Money Dispenser |
| **Trigger** | Customer |
| **Main Scenario** | |  |  | | --- | --- | | **Steps** | **Actions** | | 1 | User Enters the amount of money to be withdrawn | | 2 | System Checks the Balance | | 3 | System dispense the money if Balance is greater than entered withdraw amount. | | 4 | System Updates the Account balance via bank server in database | | 5 | System generates the balance receipt. | |
| **Extensions** | |  |  | | --- | --- | | **Steps** | **Actions** | | **2.A** | **If Balance is insufficient in account** | | 2.A.1 | System displays the error message. | | 2.A.2 | System displays the account current balance. | | **4.A** | **System fails to update new balance in database** | | 4.A.1 | System retries to update the new balance | | 4.A.2 | System updates correct balance and proceed to receipt printing | |
| **Open Issues** | |  |  | | --- | --- | | 1 | Internet stops working when User confirms the withdraw after adding the amount. | | 2 | System don’t have cash in it to dispense, amount debited from user account but ATM doesn’t dispense cash. | |

|  |  |
| --- | --- |
| **System:** | iTax |
| **Identifier** | NUC(iTax)03 |
| **Author(s):** | Arnav, Sumit and Sravani |
| **Version:** | 0.1 |
| **Priority:** | High |
| **Related Use Case(s)** | None |
| **Name:** | Steal Personal Information |
| **Precondition(s):** | None |
| **Postcondition(s):** | Misuser successfully steals the personal information. |
| **Trigger:** | Misuser he logs in to the system using the tax payer details. |
| **Normal Flow:** | 1. The Misuser can collect data from cookies while the tax payer is getting registered into the system 2. Then, he logs in and collects all the personal information. |
| **Exceptional Flow(s):** | Alternate Flow:  None  Recovery Flow:  None  Failure Flow:  None |
| **Related Actor(s):** | Primary Actors:   1. Hacker   Secondary Actor: |
| **Related Use Case(s):** | Include:  None  Extends:  None  Generalize:  None |
| **Summary** | Information is stolen from the cookies while the taxpayer is registered. |