CS383 Group Project

Software Design Document

(SDD)

for

Smart ATM

Version 2.0

Prepared by Group 2

Contents

Change History

Date	Version	Reasons for change
29/11/2021	1.0	Initial software design
08/12/2021	2.0	Adding points to Section 5 – Updating Software architecture

1 Introduction

1.1 Scope

The system that we are developing is basically an ATM with more advanced/smart features that do not exist in ATMs and can improve current ATMs, including: face recognition, fingerprint, opening an account, CCTV, Recognize a wanted person, Smartphone login, auto failure alert, currency exchange from any currency to Riyal.

1.2 Document Structure

-***System Overview***: An introduction to the system design.

-***System Architecture and Components Design***: Which covers Architectural Description, Component Decomposition Description, Detailed Components Design Description, Architectural Alternatives and Design Rationale.

-***Data Design***: Depicts entities and their relationships.

-***Design Details***: Describes the design using class diagrams.

-***Human Interface Design***: Visual representation of the user interface

-***Resource Estimates***: Estimate number of computer resource needed for the operating system

1.3 Constraints

None

2 System Overview

The main reason for the development of Smart ATM is to offer more services than the normal ATM, with the Smart ATM the user will have several options to access his or her account and these options are:

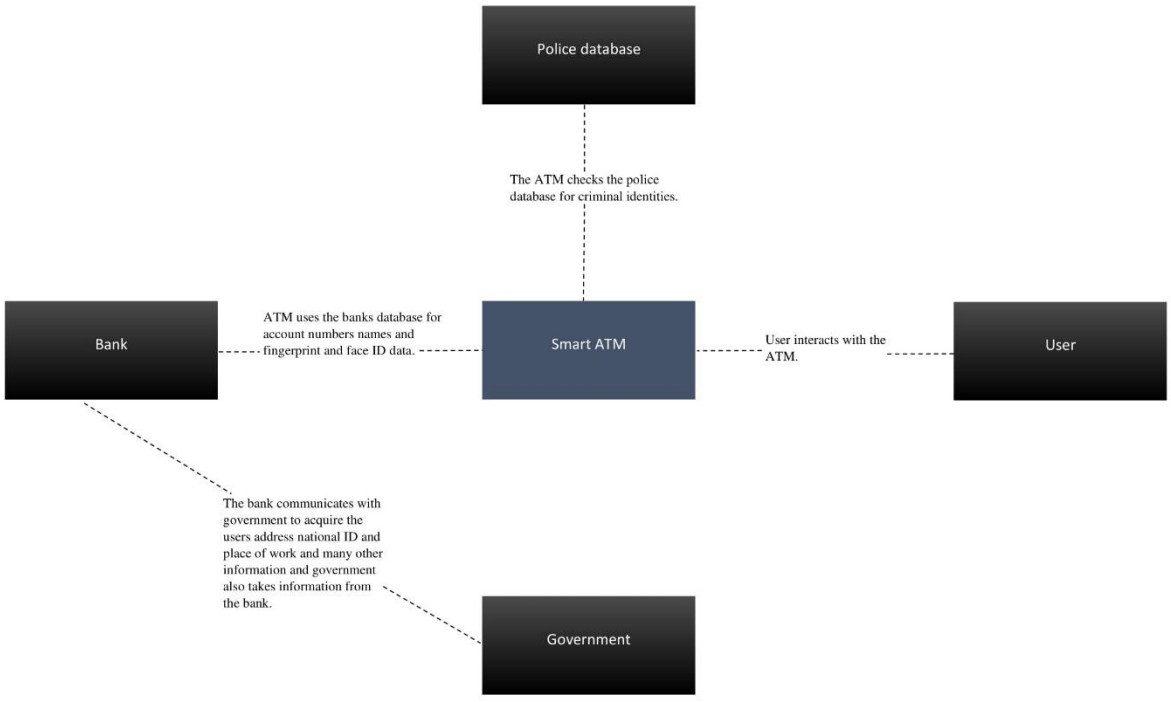
1- Face ID: The ATM is equipped with the latest face recognition sensors which will identify the user upon approaching the machine, also a bonus feature is the ability to identify wanted criminals and reporting the authorities

2- Fingerprint reader: the fingerprint reader is an option for the convenient of females who cover their faces.

3- Smartphone login: one amazing feature is the ability to access the account through the user NFC chip in their smartphone device.

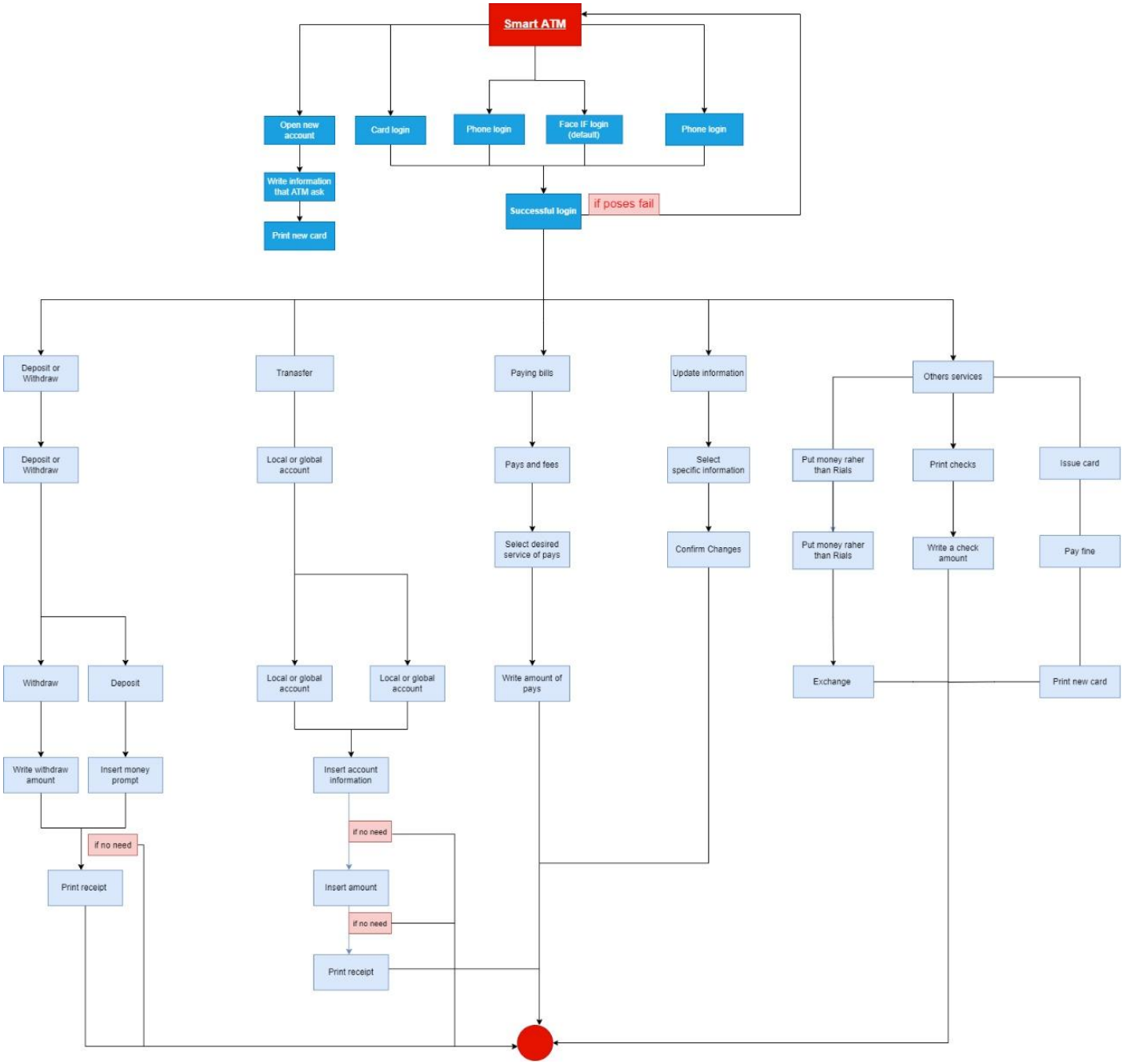
4- Currency exchange: the ability to exchange any currency to Riyal.

Also, here we have the Software architecture diagram:



3 System Architecture and Components Design

3.1 Process Architectural Description



3.2 Component Decomposition Description

When customer be front of smart ATM the smart ATM will stops its ads and take face recognition scan directly to logins customer account then if login in customer account services will display to customer, if not login in the ATM will shows choices of multiple logins in ways (Fingerprint login, Face ID login, phone or card login), so consider that customer login in his account then customer can select any services that he want which is (print check , Update information, Open a new account, Paying Bills, Transfer, Deposit or withdraw, issue a new card).

3.3 Detailed Components Design Description

• **Component Identifier**: face recognition.
• **Purpose**: to authenticate users.
• **Function**: authenticate the user by face recognition.
• **Subordinates**: used by camera.
• **Dependencies**: must have a camera.
• **Interfaces**: record the user and compare it to the face recognition database.
• **Data**: face ID database.

• **Component Identifier**: fingerprint.
• **Purpose**: to authenticate users.
• **Function**: authenticate the user by his fingerprint.
• **Subordinates**: used by the fingerprint sensor.
• **Dependencies**: must have a fingerprint sensor.
• **Interfaces**: capture the fingerprint and compare it to the fingerprint database.
• **Data**: fingerprint database.

• **Component Identifier**: smartphone login.
• **Purpose**: to authenticate users.
• **Function**: authenticate the user by his phone.
• **Subordinates**: used by phone.
• **Dependencies**: must have the bank application for smartphone login.
• **Interfaces**: get data from the application to authenticate the user.
• **Data**: accounts database.

• **Component Identifier**: print check.
• **Purpose**: if the user wants to print a check.
• **Function**: print a check.
• **Subordinates**: none.
• **Dependencies**: must have a check printer.
• **Interfaces**: get information from the user and put it in the check.
• **Data**: check layout

• **Component Identifier**: update information.
• **Purpose**: to make changes to your account.
• **Function**: change information of your account.
• **Subordinates**: none.
• **Dependencies**: none.
• **Interfaces**: access the account information and make changes.
• **Data**: accounts information database.

• **Component Identifier**: open a new account.
• **Purpose**: to open an account for new users.
• **Function**: opens an account for new users and prints the card.
• **Subordinates**: none.
• **Dependencies**: none.
• **Interfaces**: insert information about the new account.
• **Data**: insert and save the new account information in the account database.

• **Component Identifier**: pay bills.
• **Purpose**: to make the user pay his bills.
• **Function**: give the user access to his bill and he can pay them.
• **Subordinates**: none.
• **Dependencies**: none.
• **Interfaces**: none.
• **Data**: update the bill information of the chosen bill.

• **Component Identifier**: transfer.
• **Purpose**: transfer money between people.
• **Function**: lets you choose an account and transfer money to it.
• **Subordinates**: none.
• **Dependencies**: must be a valid account.
• **Interfaces**: update the amount of the account after transfer.
• **Data**: data about the account amount.

• **Component Identifier**: deposit or withdraw.
• **Purpose**: to put/get money in/from the account.
• **Function**: insert the money to deposit or choose the amount you want to withdraw.
• **Subordinates**: need a cash counter for deposit.
• **Dependencies**: none.
• **Interfaces**: update account information after process.
• **Data**: data about the account amount.

• **Component Identifier**: issue a new card.
• **Purpose**: print a new card.
• **Function**: ask for a new card then pay the fine to get the new card.
• **Subordinates**: none.
• **Dependencies**: need to have a card printer.
• **Interfaces**: none.
• **Data**: the user account information.

3.4 Architectural Alternatives

One architecture that could be considered is the Event-driven architecture because it is a good way to define the system's state at every point or event.

3.5 Design Rationale

The chosen architecture gives a clear image of the ATM processes at run-time; however, it is just a process architecture; it does not show the relationships between entities.

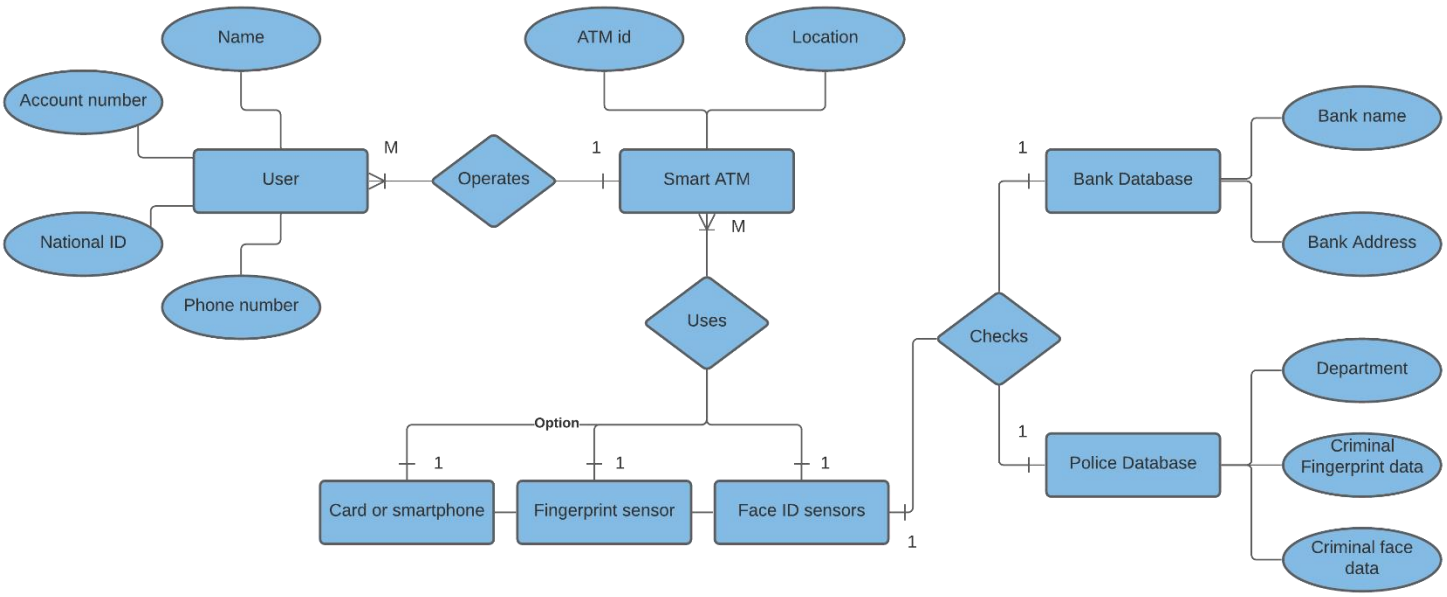
4 Data Design

Covers database description and data structures.

4.1 Database Description

In the initial meeting, the client suggested to add a Criminal recognition system, which will match each user's face id and fingerprint to the police's database and warn them if a match is found, then there are the normal bank database functions and they are Client name, account number, national id, address and phone number which the ATM will match with the bank's database.

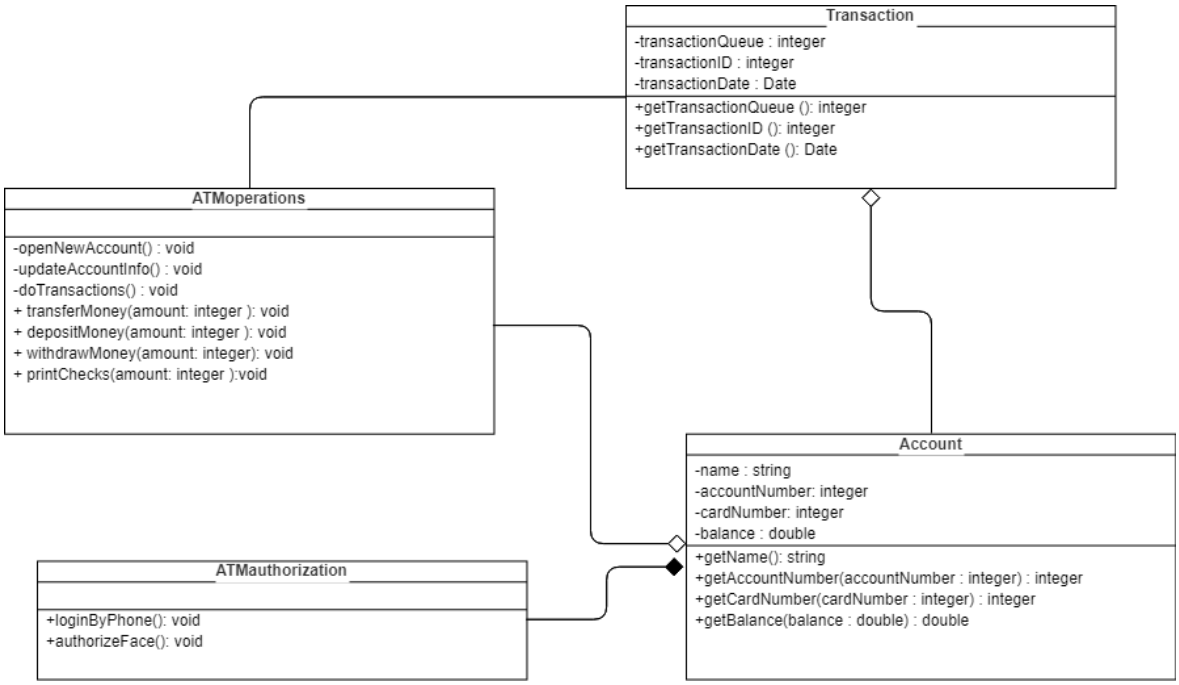
4.2 Data Structures



5 Design Details

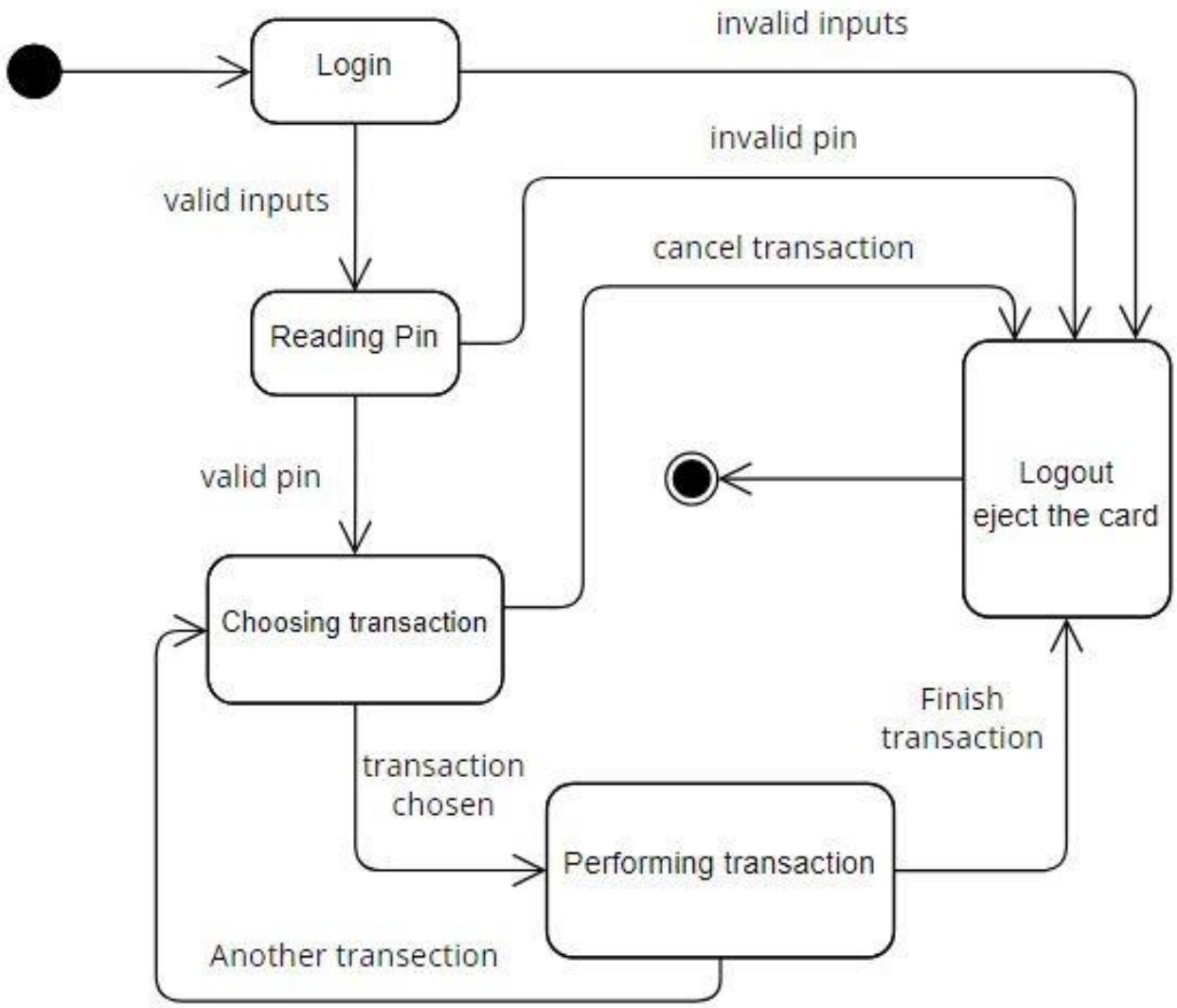
Describes design by using diagrams to reflect all the major requirements.

5.1 Class Diagrams

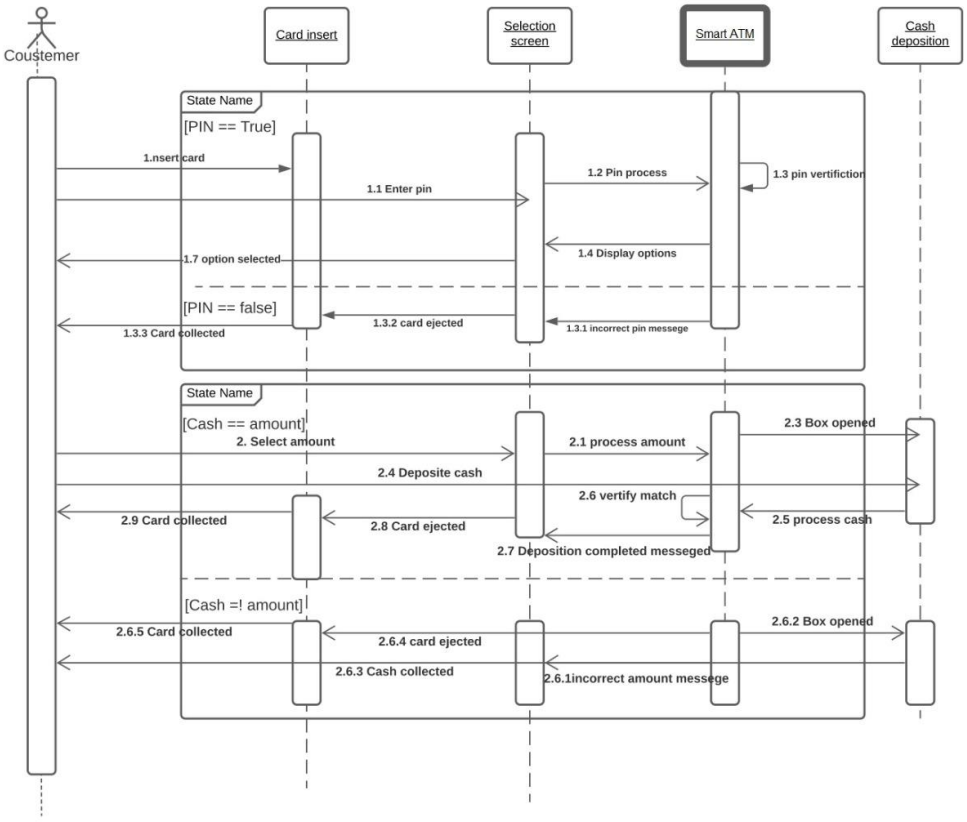


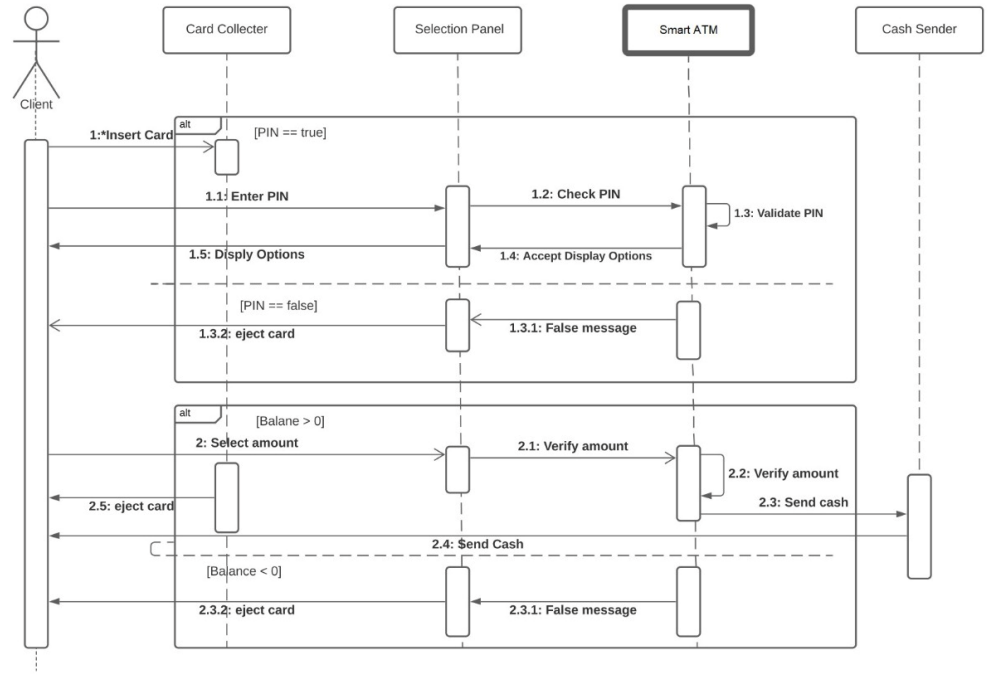
5.2 State diagrams

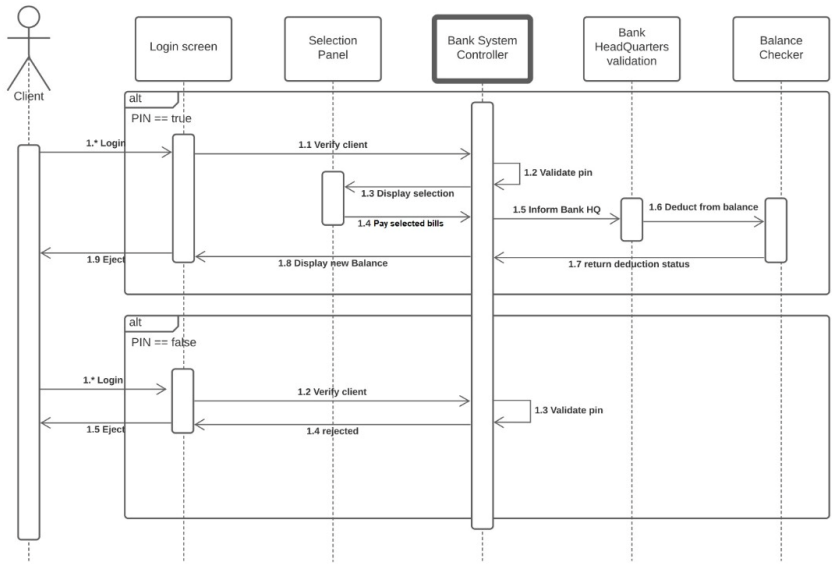
.

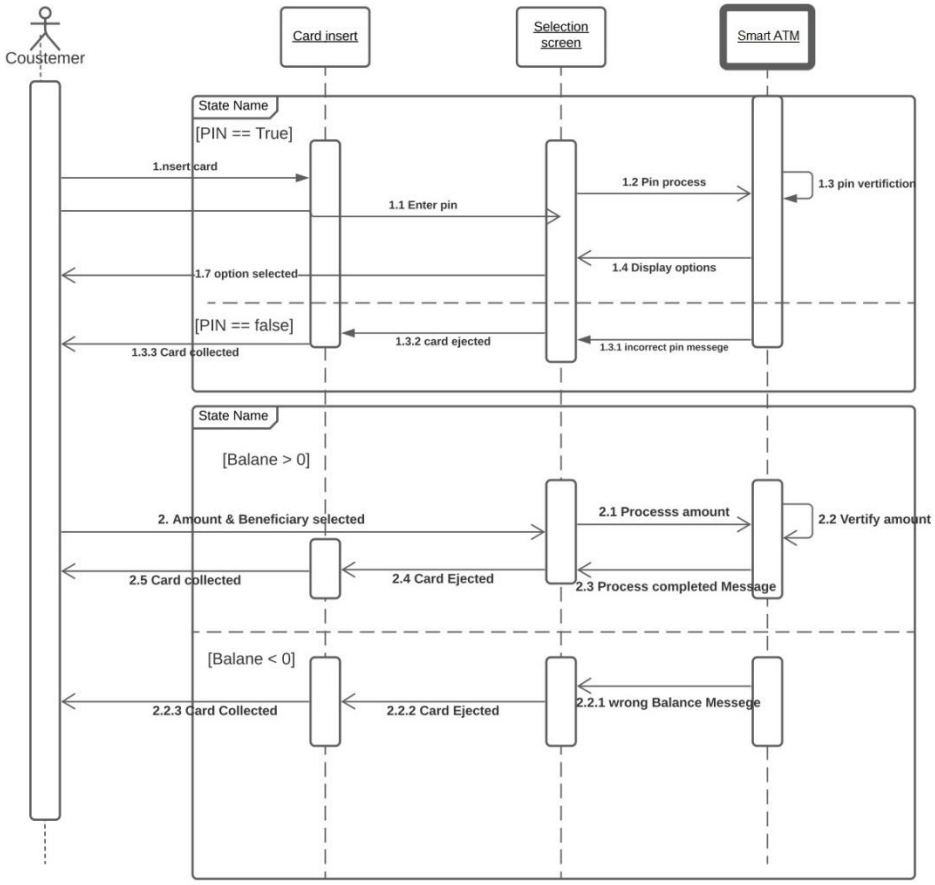


5.3 Sequence Diagram






Client
Login screen
Selection Panel
Bank System Controller
Bank HeadQuarters validation
Balance Checker
alt
PIN == true
1.* Login
1.1 Verify client
1.2 Validate pin
1.3 Display selection
1.4 Pay selected bills
1.5 Inform Bank HQ
1.6 Deduct from balance
1.7 return deduction status
1.8 Display new Balance
1.9 Eject
alt
PIN == false
1.* Login
1.2 Verify client
1.3 Validate pin
1.4 rejected
1.5 Eject


Coustemer
Card insert
Selection screen
Smart ATM
State Name
[PIN == True]
1.nsert card
1.1 Enter pin
1.2 Pin process
1.3 pin vertifiction
1.4 Display options
1.7 option selected
[PIN == false]
1.3.2 card ejected
1.3.1 incorrect pin messege
1.3.3 Card collected
State Name
[Balane > 0]
2. Amount & Beneficiary selected
2.1 Processs amount
2.2 Vertify amount
2.5 Card collected
2.4 Card Ejected
2.3 Process completed Message
[Balane < 0]
2.2.3 Card Collected
2.2.2 Card Ejected
2.2.1 wrong Balance Messege

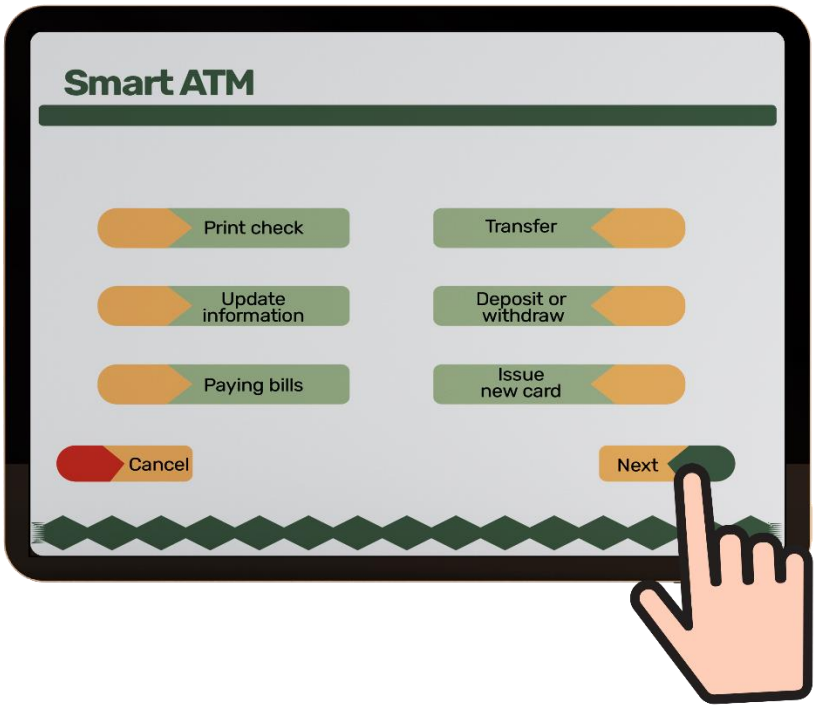
6 Human Interface Design

6.1 Overview of the User Interface

Describe the general functionalities of the system from end users' perspective.

These days, end user needs efficiency also simple and faster, our project tries to make that happen, for example how the user logs in is more than one way which user prefer like fingerprint or face ID or phone or card login.
For that propose our other function we try to make it simple, if user need to transfer money or he lost his card and he can issue a new one right away.

6.2 Detailed Design of the User Interface



Screen Objects and Actions:

Print check: To print a check for client.

Transfer: to transfer money to another account.

Update information: to update any information of client.

Deposit or withdraw to deposit money or withdraw money.

Paying bills: to pay bills or checks.

Issue new card: to print a new card if need it.

7 Resource Estimates

The Smart ATM requires a mid-tier system component, which will be more than enough to run the system and the communication with the different databases with relative smoothness and swiftness.

The estimated requirements for the system:

CPU: A CPU from 3 or 4 years back will be enough, the baseline is running multiple tasks in windows 10, also preferably an integrated GPU.

GPU: some CPUs do not have an integrated graphics, so if the CPU does not include one a very low-end GPU will suffice.

RAM: 8 gigabytes will more than enough for our purposes.

Storage: there is no need for high-capacity storage but, using an SSD is a must.

Power supply: It should be power efficient the reasoning behind that is the ATM will always be running.

Screen: A high resolution led screen with touch capabilities.