C:\Users\XuanNguyen\Desktop\logoSoftech.gif

**e-Project Document**

**ATM System**

|  |  |  |
| --- | --- | --- |
| **Group 4** | **Member** | **Email** |
| **Doan Ngoc Van Khoa** | **bindizz88@gmail.com** |
| **Nguyen Minh Vuong** | **vuongnm@hey.com.vn** |
| **Nguyen Van Xuan** | **xuannguyen6489@gmail.com** |
| **Nguyen Duc Hoang** | **hoangnguyen91dn@gmail.com** |
| **Teacher** | |
| **Kieu Minh Canh** | |

Contents

[I. Problem Definition 5](#_Toc368394661)

[1. Introduction 5](#_Toc368394662)

[2. EXISTING scenario 5](#_Toc368394663)

[3. Proposed solution 5](#_Toc368394664)

[II. Customer Requirement Specification 5](#_Toc368394665)

[1. Admin use-case: 6](#_Toc368394666)

[2. Customer Use-case 8](#_Toc368394667)

[III. Project Plan 9](#_Toc368394668)

[IV. E-R Diagrams 10](#_Toc368394669)

[V. Table design 11](#_Toc368394670)

[Table User\_Group 11](#_Toc368394671)

[Table SYSTEMUser 11](#_Toc368394672)

[Table Customer 11](#_Toc368394673)

[Account 12](#_Toc368394674)

[Table CustomerTransaction 13](#_Toc368394675)

[Table BankInfo 14](#_Toc368394676)

[TABLE TRANSFER 14](#_Toc368394677)

[V. ALGORITHMS 14](#_Toc368394678)

[1. Administrator Module 14](#_Toc368394679)

[1.1 Login & Logout System 14](#_Toc368394680)

[1.2 CHANGE PASSWORD 16](#_Toc368394681)

[1.3 EDIT BANKINFO 18](#_Toc368394682)

[1.4 ADD NEW ADMIN 19](#_Toc368394683)

[1.5 EDIT ADMIN 20](#_Toc368394684)

[1.6 ADD NEW ACCOUNT 21](#_Toc368394685)

[1.7 EDiT ACCOUNT 23](#_Toc368394686)

[1.8 DELETE ACCOUNT 23](#_Toc368394687)

[1.9 CREATE CUSTOMER 24](#_Toc368394688)

[1.10 EDIT CUSTOMER 25](#_Toc368394689)

[1.11 DELETE CUSTOMER 26](#_Toc368394690)

[1.12 TRANSACTION 27](#_Toc368394691)

[1.13 RESET PIN 28](#_Toc368394692)

[2. CUSTOMER MODULE 29](#_Toc368394693)

[2.1 LOGIN 29](#_Toc368394694)

[2.2 LOGOUT 31](#_Toc368394695)

[2.3 CHANGE PIN 31](#_Toc368394696)

[2.4 WITHDRAW 33](#_Toc368394697)

[2.5 DEPOSIT 34](#_Toc368394698)

[2.6 TRANSFER 35](#_Toc368394699)

[VI. GUI DESIGN 37](#_Toc368394700)

[1. ADMINISTRATOR MODULE 37](#_Toc368394701)

[1.1. LOGIN 37](#_Toc368394702)

[1.2. ADMIN SCREEN 37](#_Toc368394703)

[1.3 ADMIN INFORMATION AND CHANGE PASSWORD 38](#_Toc368394704)

[1.4 CHANGE ADMIN PASSWORD 39](#_Toc368394705)

[1.5 ADMIN MANAGEr 40](#_Toc368394706)

[1.6 ADMIN CREATE NEW USER 40](#_Toc368394707)

[1.7 CREATE CUSTOMER 41](#_Toc368394708)

[1.8 CUSTOMER MANAGER 42](#_Toc368394709)

[1.9 CREATE NEW ACCOUNT 43](#_Toc368394710)

[1.10 ACCOUNT DETAILS 44](#_Toc368394711)

[1.11 ACCOUNT MANAGER 45](#_Toc368394712)

[1.12 DEPOSIT 46](#_Toc368394713)

[1.13 WITHDRAW 47](#_Toc368394714)

[1.14 WITHDRAWAL REPORT 47](#_Toc368394715)

[1.15 DEPOSIT REPORT 48](#_Toc368394716)

[1.16 ACCOUNT REPORT 48](#_Toc368394717)

[1.17 TRANSFER REPORT 49](#_Toc368394718)

[2. CUSTOMER 49](#_Toc368394719)

[1.1 LOGIN 49](#_Toc368394720)

[1.2 CONTROL 50](#_Toc368394721)

[1.3 CHANGE PIN 51](#_Toc368394722)

[1.4 WITHDRAW 51](#_Toc368394723)

[1.5 DEPOSIT 53](#_Toc368394724)

[1.6 TRANSFER 53](#_Toc368394725)

[1.7 BALANCE ENQUIRY 54](#_Toc368394726)

[1.8 BANK INFORMATION 55](#_Toc368394727)

[VII. TASK SHEET 55](#_Toc368394728)

[VIII. Testing Checklist 57](#_Toc368394729)

# I. Problem Definition

## 1. Introduction

* An automated teller machine (ATM), also known as a Cash Machine and by several other names is a computerized device that provides the clients the financial institution without the need for the cashier .The client is identified by inserting unique card number and some security information.A user can assess the ATM(Automatic Teller Machine) system / screen only after entering the correct user name and the password on the very first screen which is installed on the the ATM room.
* The user name should be the unique card number provided to the user and the password can be any numeric number of four digits which is called as PIN (Personal Identification number)

## 2. EXISTING scenario

Earlier all this transaction is done manually which requires lot of paper work . Manual works make the maintenance difficult and also increases possibility of the discrepancies in data. Even Data retrieving also becomes hectic with this system. To avoid the wastage of time company has approached you to make the Automated System.

## 3. Proposed solution

* Since the ATM System is designed for the existing user, creating the account would be done by admin only.
* For each time the user tries to enter the ATM need to use the same password which is entered at the very first transaction (unless the user changes the password with the option of the ATM system).
* Once the user name and the password are verified the ATM screen would appear.
* Each time a new user enters the ATM room are suppose to provide their authentication details.
* The initial screen of the ATM system should appear “Enter your card number” button along with the keypad.
* Once the card number is entered and button is clicked it should check its availability in the database and if it is not available then the error message should be generated stating the user is invalid else the password field would appear, user needs to enter the password. Once the password is verified, the screen appears with following buttons: (main menu):
  + Deposit
  + Withdrawal
  + Balance Enquiry
  + Change Password
  + Transfer money

# II. Customer Requirement Specification

## 1. Admin use-case:

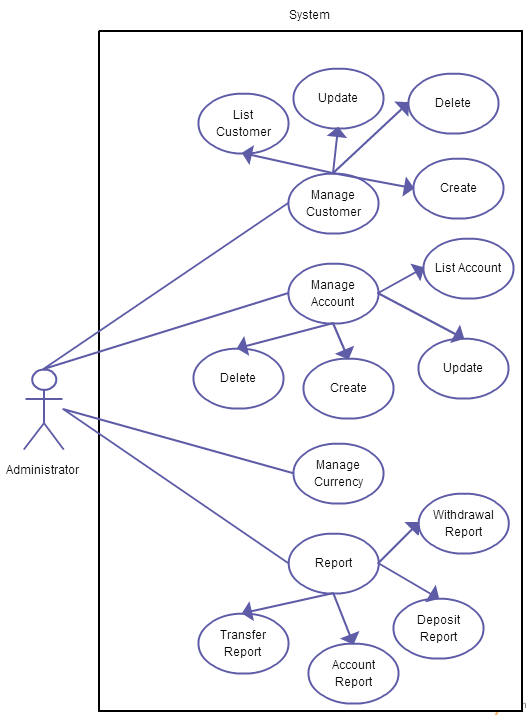
**Admin users are bank staff who has the duty and right to serve customers action. Admin has the highest right over the system and their action are described with rules following:**

* Creating the account of the user by providing a unique identification number and four digit PIN( Personal Identification Number) and also by filling the basic details like name, contact number, gender, address.
* Rules to be followed while performing transaction are:

1. DEPOSIT:- There should be verification / check for deposit : a user neither can deposit more than 25000 $ in a single day nor the user can deposit more than 5 times in a single day.
2. WITHDRAW :- There should be verification / check for withdrawal : a user neither can withdraw more than 25000 $ in a single day nor the user can withdraw more than 5 times in a single day
3. BALANCE ENQUIRY:- On click of this button user last balance should be stated . Along with this , user can view the last 5 transaction with a complete description such as date , debited, credited and particulars of the transaction.
4. CHANGE PASSWORD:- Here user can change the password For these three fields should appear : one for the old password and other two for the new password and confirmation password respectively. Once the password is changed the user page should log out automatically and then he/she is suppose to login with the new password .

* Reports :

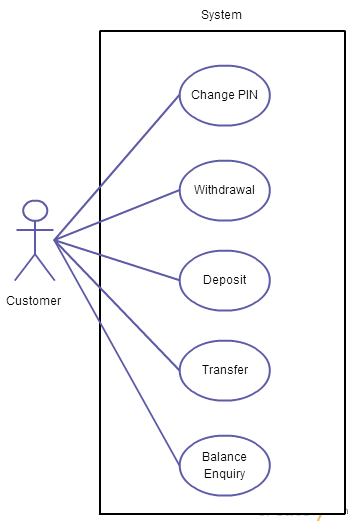
1. WITHDRAWAL REPORT: Here the Date is taken as input and all the withdraw transaction done on that day should be generated with the fields like name , unique number, withdraw amount , balance .
2. DEPOSIT REPORT: Here also the date is taken as input and all the deposit transaction done on that day should be generated with fields like name , unique number , Deposited amount , balance .
3. TRANSFER REPORT: Here also the date is taken as input all the transfer details report should be generated with fields such as Name of the debitor , Unique number of debitor moneytransfer , balance of the debitor, name of the creditor, unique number of the creditor , balance of the creditor.
4. ACCOUNT REPORT: Here the list of all the registered user should be displayed with details like name, unique identification number, password, contact number, gender, age .



## 2. Customer Use-case

**Customers are the main users of the system who use the bank systeem to store their money. They have the following usecase and Rules to be followed while performing transaction are:**

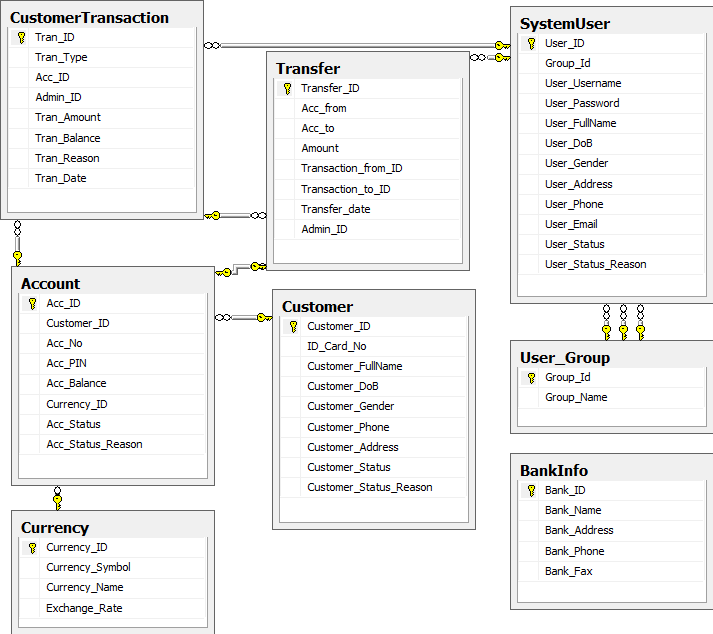
* DEPOSIT:- Every time user clicks on the deposit button a text field appears on which user can enter the amount he wants to deposit and then he needs to click on the ‘Enter” button to deposit the amount
* WITHDRAW :- Every time user clicks on the withdrawal button a text field appears on which user can enter the amount he/she wants to withdrawal and then needs to click on the ‘Enter” button to withdraw the amount. If the user is entering the amount more than his/her balance then the error message should be generated stating “ Invalid Balance”
* BALANCE ENQUIRY:- On the click of this button the available balance of the user should be on the display.
* CHANGE PASSWORD:- It would help the user to change / reset the password. Once the user clicks on the change password button three password fields should appear: one for the old password and the other two fields for new password and confirmation of the new password respectively. Once the password is changed the user should be lead to “very first screen” and needs to enter with the changed password.



# III. Project Plan

|  |  |  |
| --- | --- | --- |
| Date Time | Tasks | Members |
| 04/09 – 13/09 | Project Planning.  Analyze and Database design  Customer and Admin usecase specification  Write first document | The whole team  Document writer: Nguyen Van Xuan |
| 14/09 – 25/09 | Develop Admin Module   * Design admin interface * Develop function   Develop Customer Module   * Design customer interface * Develop function | The whole team  Document writer: Nguyen Van Xuan |
| 25/09 – 4/10 | Complete eProject   * Module admin * Module customer * Input data * Test | The whole team  Document writer: Nguyen Van Xuan |

# IV. E-R Diagrams

****

# V. Table design

### Table User\_Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Group\_Id | INT | No | Primari key | Group Id |
| Group\_Name | VARCHAR(100) | No |  | Group Name |

### Table SYSTEMUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| User\_ID | INT | No | Primari key | User ID |
| Group\_Id | INT | No | Foreign key | Group ID |
| User\_Username | VARCHAR(16) | No |  | User Name |
| User\_Password | VARCHAR(40) | No |  | Pass word |
| User\_FullName | VARCHAR(40) | Yes |  | Full Name |
| User\_DoB | DATETIME | Yes |  | Date |
| User\_Gender | VARCHAR(15) | Yes |  | Gender |
| User\_Address | VARCHAR(300) | Yes |  | Address |
| User\_Phone | VARCHAR(15) | Yes |  | Phone |
| User\_Email | VARCHAR(40) | Yes |  | Email |
| User\_Status | VARCHAR(15) | No |  | Status |
| User\_Status\_Season | Varchar(MAX) | Yes |  | Delete Season |

### Table Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Customer\_ID | INT | No | Primari key | Customer ID |
| ID\_Card\_No | VARCHAR(11) | No |  | Customer National Identity Number (Social Security Number in US) |
| Customer\_FullName | VARCHAR(40) | No |  | Full Name |
| Customer\_DoB | DATETIME | No |  | Date |
| Customer\_Gender | VARCHAR(15) | No |  | Gender |
| Customer\_Phone | VARCHAR(15) | No |  | Phone |
| Customer\_Address | VARCHAR(300) | No |  | Address |
| Customer\_Status | VARCHAR(15) | No |  | Status |
| Customer\_Status\_Season | VARCHAR(300) | No |  | Reason |

### Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Acc\_ID | INT | No | Primari key | Account ID |
| Customer\_ID | INT | No |  | Customer ID |
| Acc\_No | VARCHAR(19) | No |  | Account No |
| Acc\_PIN | VARCHAR(40) | No |  | Pin |
| Acc\_Balance | FLOAT | No |  | Balance |
| Currency\_ID | INT | No | Foreign key | Curency ID |
| Acc\_Status | VARCHAR(15) | No |  | Status |
| Acc\_Status\_Reason | VARCHAR(300) | No |  | Status Reason |

#### Table Currency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Currency\_ID | INT | No | Primari key | Currency ID |
| Currency\_Symbol | VARCHAR(5) | No |  | Currency Sumbol |
| Currency\_Name | VARCHAR(30) | No |  | Currency name |
| Exchange\_Rate | FLOAT | No |  | Exchange Rate |

### Table CustomerTransaction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Tran\_ID | INT | No | Primari key | Transaction ID |
| Tran\_Type | VARCHAR(15) | No |  | Transaction Type |
| Acc\_ID | INT | No | Foreign key | Acc Pri ID |
| Admin\_ID | INT | No | Foreign key | Admin ID |
| Tran\_Amount | FLOAT | No |  | Transaction Amount |
| Tran\_Balance | FLOAT | No |  | Transaction Balance |
| Tran\_Reason | VARCHAR(MAX) | Yes |  | Transaction Reason |
| Tran\_Date | DATETIME | Yes |  | Transaction Date |

### Table BankInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Bank\_ID | INT | No |  | Bank ID |
| Bank\_Name | VARCHAR(40) | No |  | Bank Name |
| Bank\_Address | VARCHAR(300) | No |  | Bank Address |
| Bank\_Phone | VARCHAR(15) | No |  | Bank Phone |
| Bank\_Fax | VARCHAR(15) | No |  | Bank Fax |

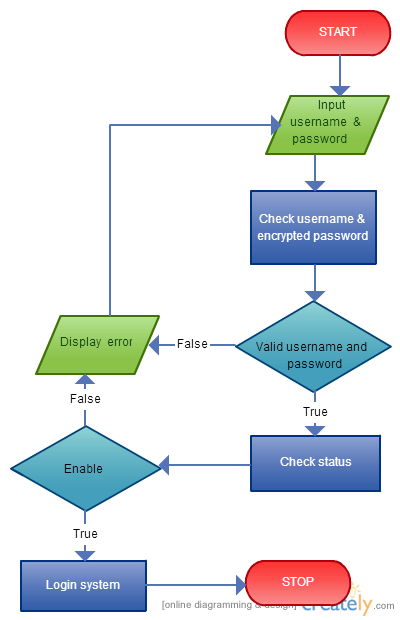
### TABLE TRANSFER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | **Type(Size)** | **Null** | **Constraint** | **Description** |
| Transfer\_ID | INT | No | Primari key | Transfer ID |
| Acc\_from | INT | No | Foreign key | Account from |
| Acc\_to | INT | No | Foreign key | Account to |
| Amount | FLOAT | No |  | Amount |
| Transaction\_from\_ID | INT | No | Foreign key | Transaction from ID |
| Transaction\_to\_ID | INT | No | Foreign key | Transaction to ID |
| Transfer\_date | DATETIME | No |  | Transaction date |
| Admin\_ID | INT | Yes | Foreign key | Admin ID |

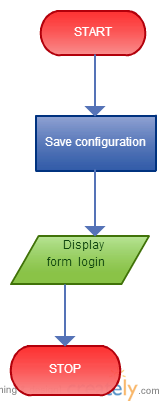
# V. ALGORITHMS

## 1. Administrator Module

### 1.1 Login & Logout System

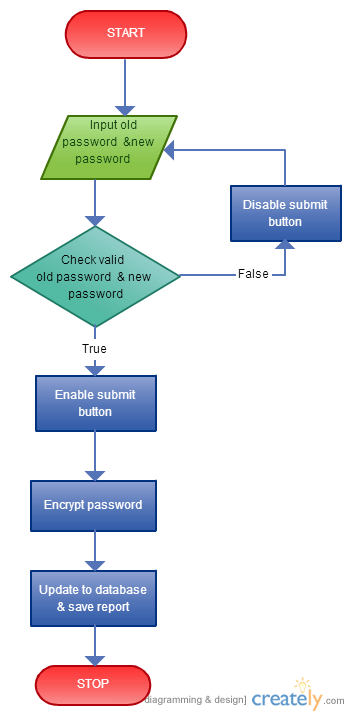


Description: Admin logins by their username and password, after authentication success, admin is allowed to enter the main screen.



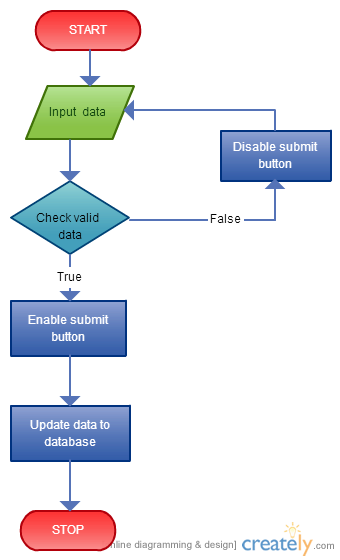
Logout: Exit from the system

### 1.2 CHANGE PASSWORD



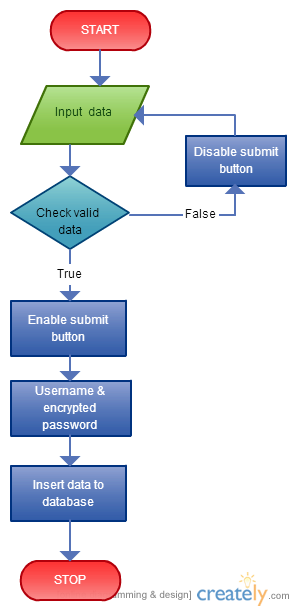
When changing password, admin user is forced to enter the old password again. This password is then hashed by MD5 algorithm and then compared with record in database (this process is the same as login). Then new password and re-enter password are check for equally before hashing and storing to database

### 1.3 EDIT BANKINFO



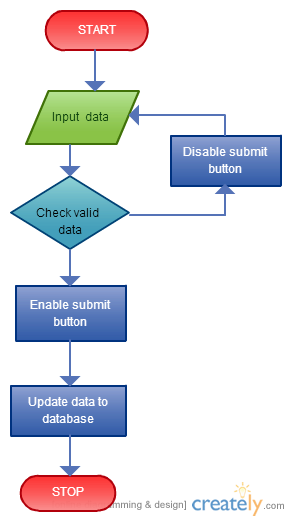
This process helps admin user change their bank info

### 1.4 ADD NEW ADMIN



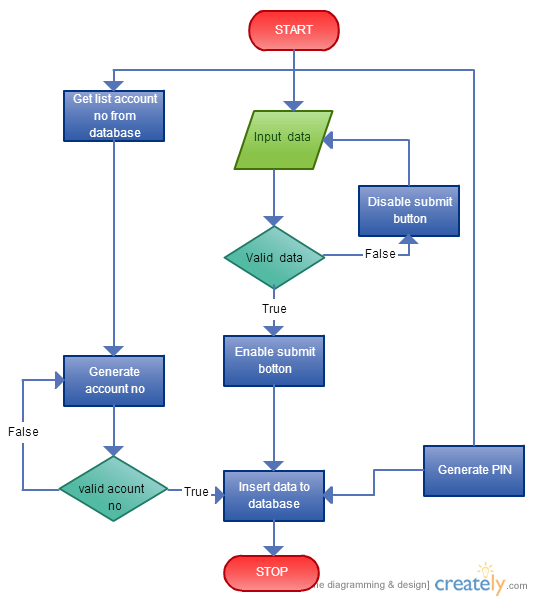
Create a new system user, check if all records are validity before storing to database.

### 1.5 EDIT ADMIN



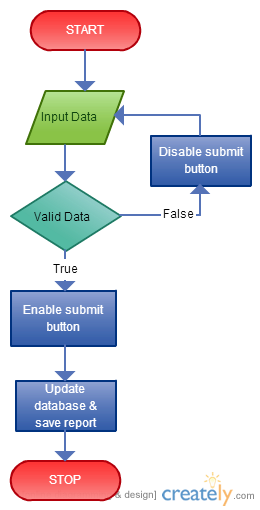
Change admin info: All data are check for validity (eg: length and required fields) before storing to database.

### 1.6 ADD NEW ACCOUNT



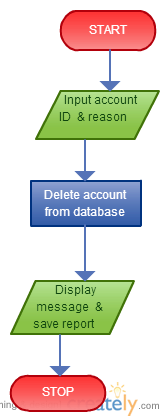
Create a new customer user, first data must be check for validity (eg: required fields, string length, pin code length) then storing to database.

### 1.7 EDiT ACCOUNT



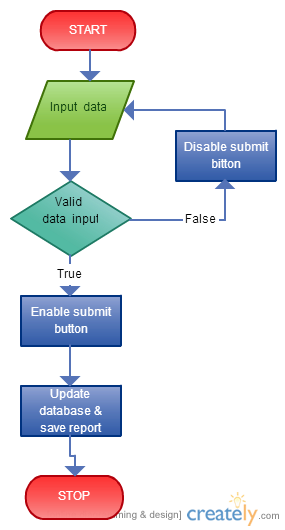
Change account information by first check for validity (require fields, string length) before storing to database.

### 1.8 DELETE ACCOUNT



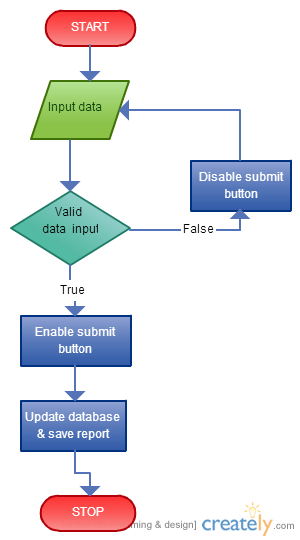
Change account status to inactive, admin user required to enter a “reason”

### 1.9 CREATE CUSTOMER



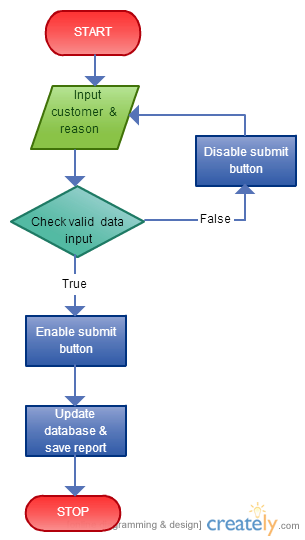
Create a new user by first check for input validity, then store record to database and update any relevant views/report

### 1.10 EDIT CUSTOMER



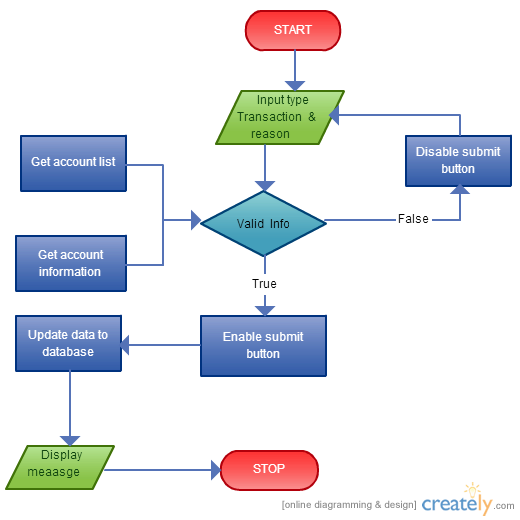
Change user info by first check for data validity and store to database on success.

### 1.11 DELETE CUSTOMER



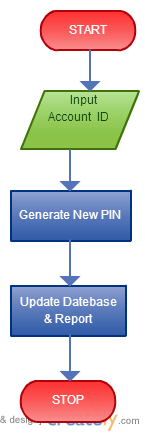
Change user status into inactive and here admin must give a reason.

### 1.12 TRANSACTION



Executing transactions like withdraw and transfer money from account to account, first all data are check for validity, and amount of money must satisfy business rule, like admin cannot withdraw more than 25000 and balance has a greater value than withdrawal amount. If all conditions meet then transaction is allowed to execute.

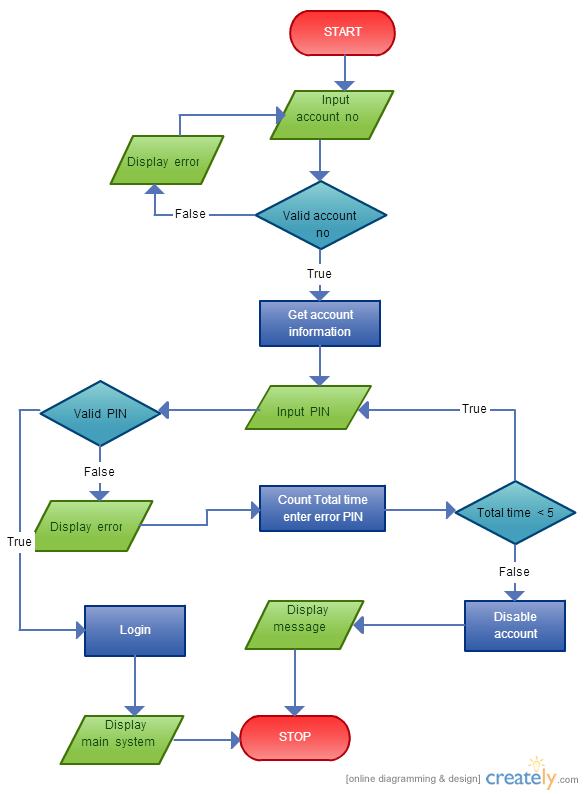
### 1.13 RESET PIN



Select account needed to be changed PIN code and update PIN code.

## 2. CUSTOMER MODULE

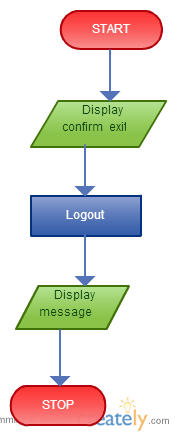
### 2.1 LOGIN



Login

User is asked to enter account number and PIN code to login. If PIN code not match, notification will be shown and after 5 fail attempts, user account will be temporary blocked.

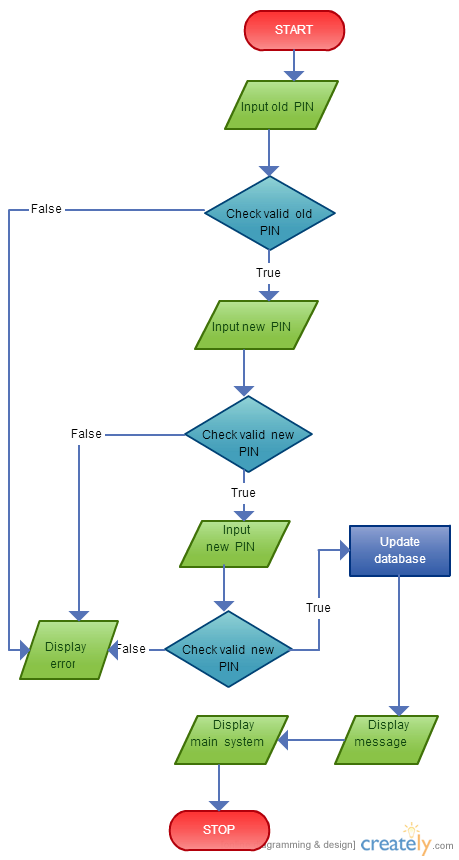
### 2.2 LOGOUT



Logout

Logout of banking system. Transaction finishs.

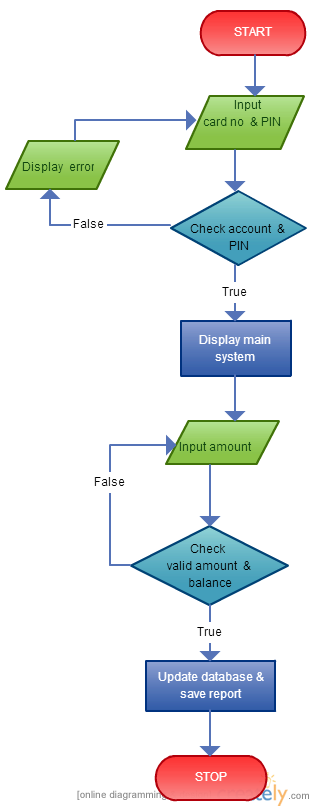
### 2.3 CHANGE PIN



Change PIN

User need to login before changing pin, Pin code is asked to enter twice for safety. If PIN code meets the condition then it’s allowed to change

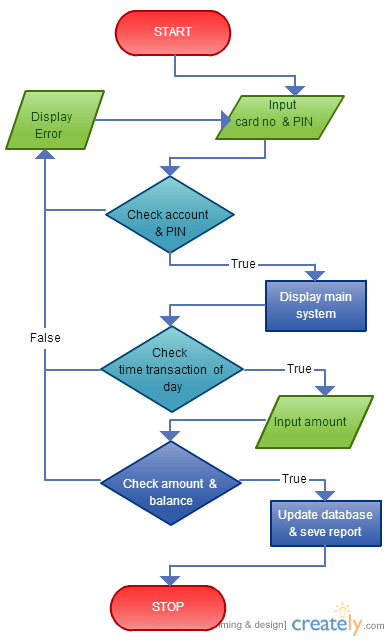
### 2.4 WITHDRAW



Withdraw

In withdraw screen, user must login by their account number and PIN code, If login success then the main screen appear with recent transaction and current balance. User can withdraw with an amount less than balance.

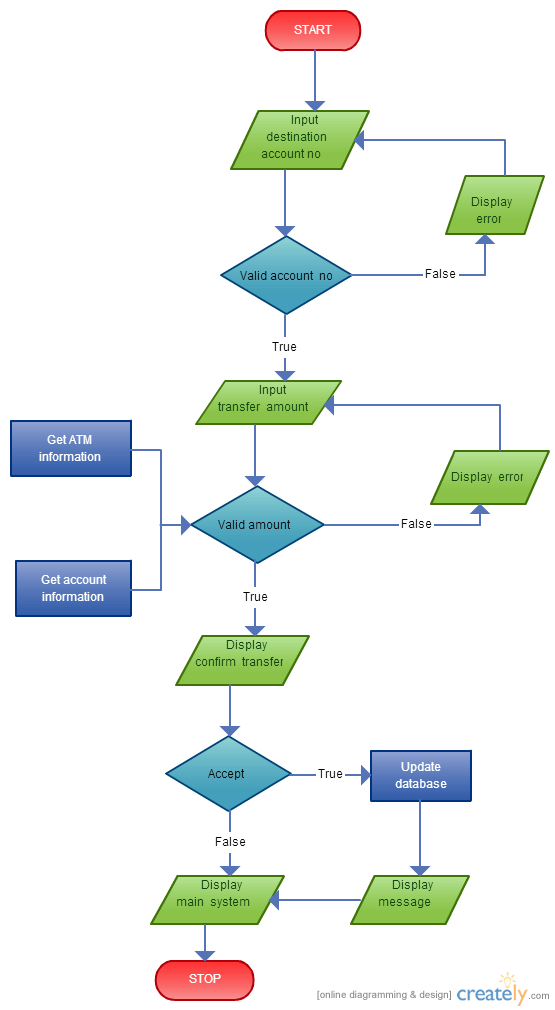
### 2.5 DEPOSIT



Deposit

In deposit screen, user has to authenticate by their account number and PIN code before they can add money to their account.

### 2.6 TRANSFER



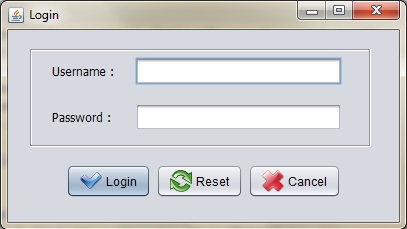
Transfer

This screen allows user to send money to another account. All data like amount and balance will be checked against each other to ensure money can be sent.

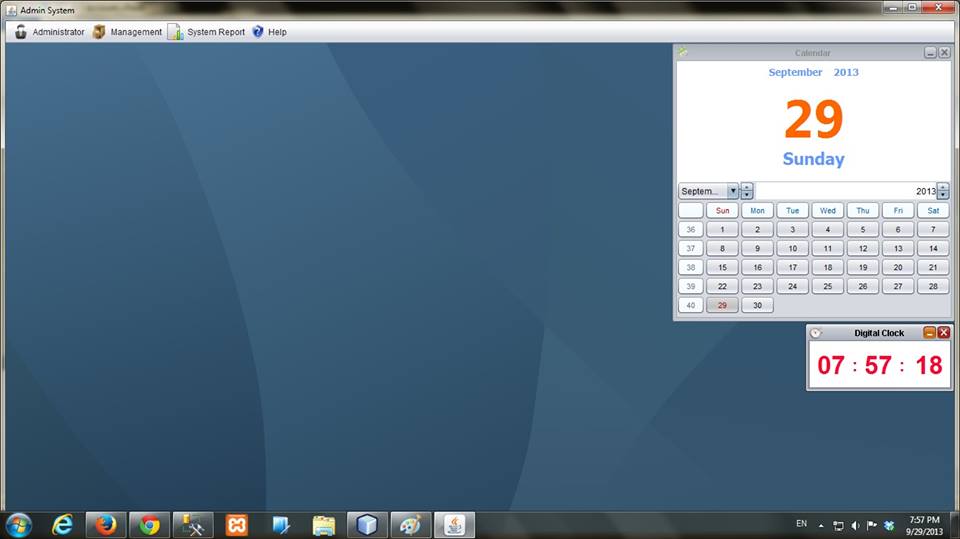
# VI. GUI DESIGN

## ADMINISTRATOR MODULE

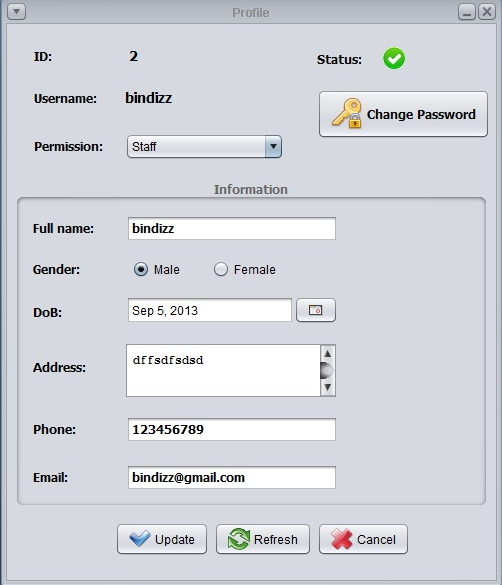
### LOGIN



### ADMIN SCREEN



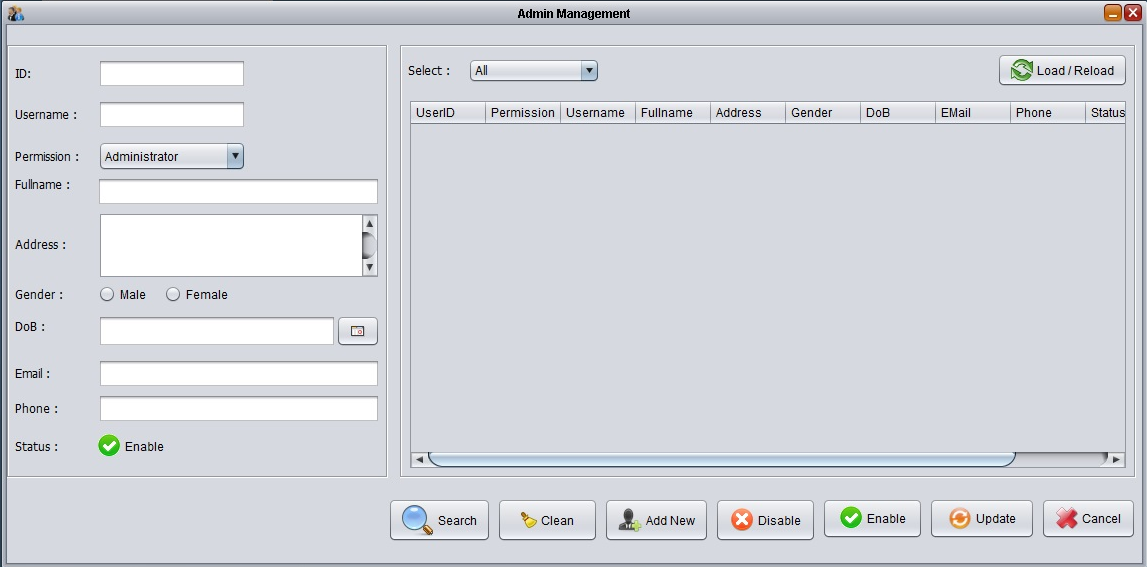
### 1.3 ADMIN INFORMATION AND CHANGE PASSWORD



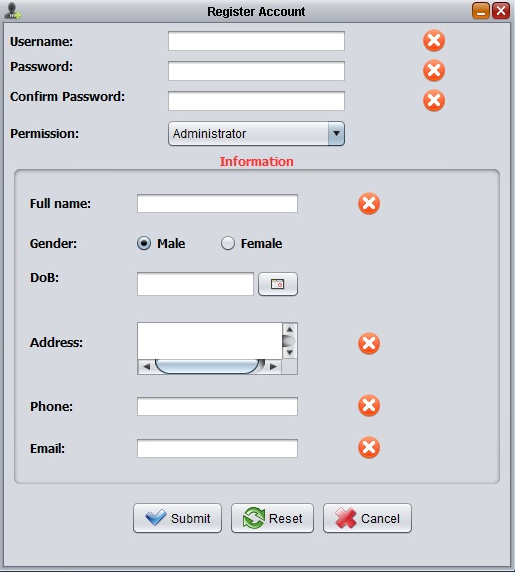
### 1.4 CHANGE ADMIN PASSWORD



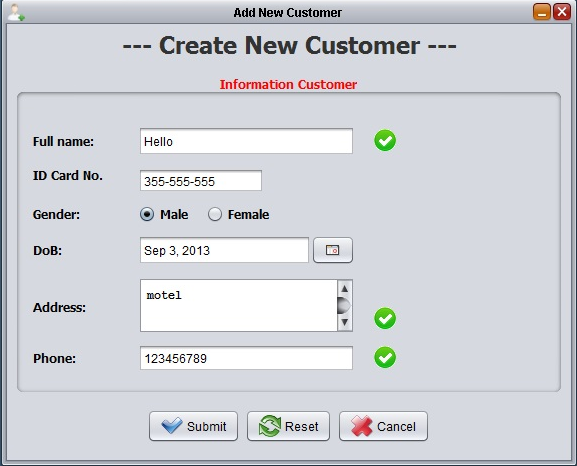
### 1.5 ADMIN MANAGEr



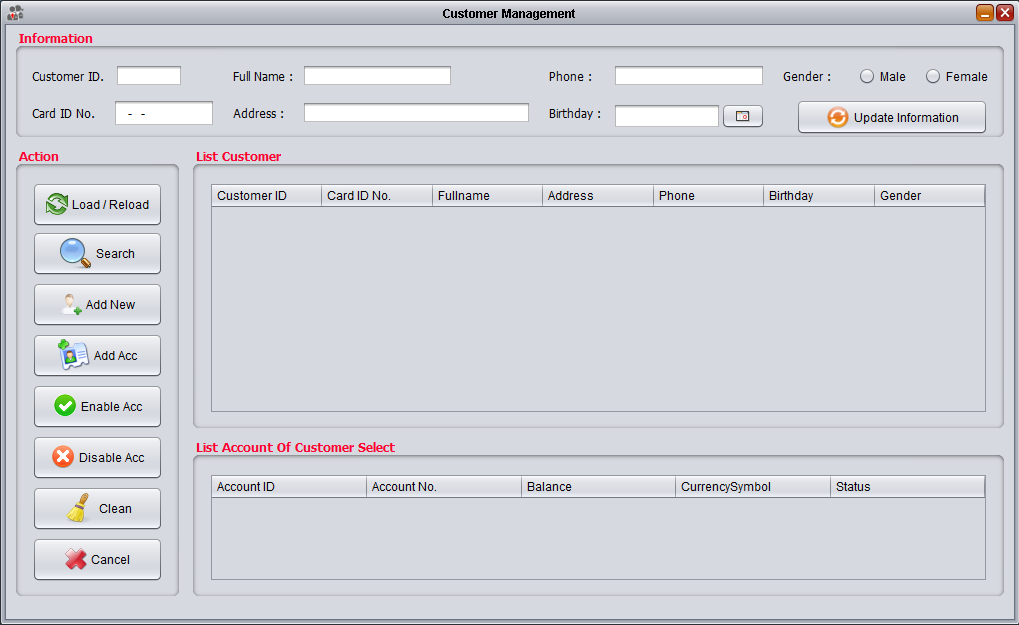
### 1.6 ADMIN CREATE NEW USER



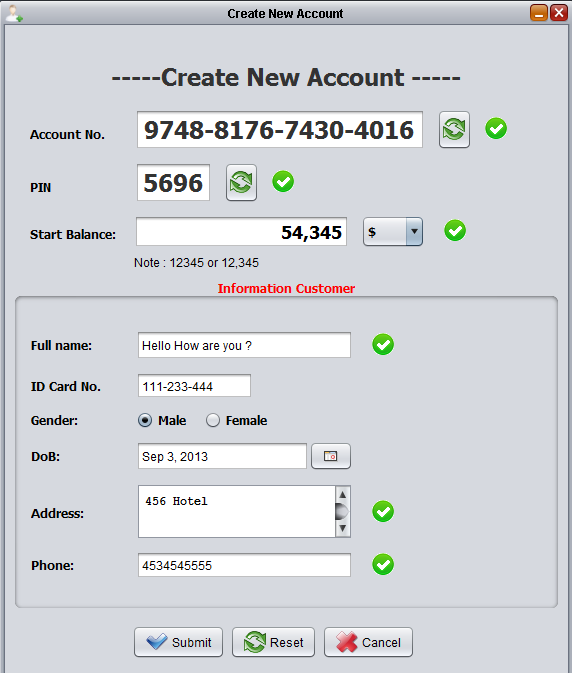
### 1.7 CREATE CUSTOMER



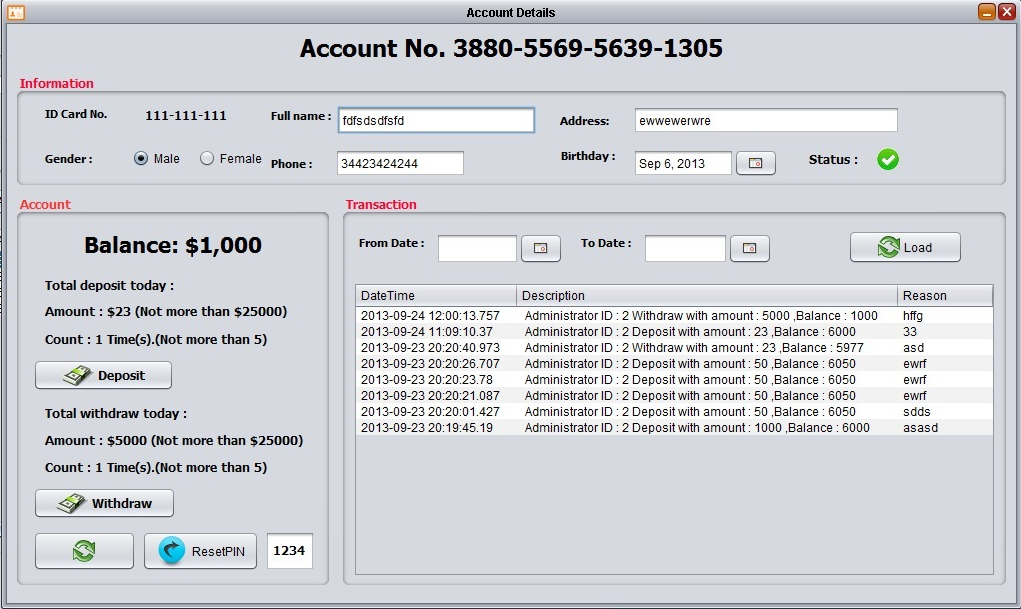
### 1.8 CUSTOMER MANAGER



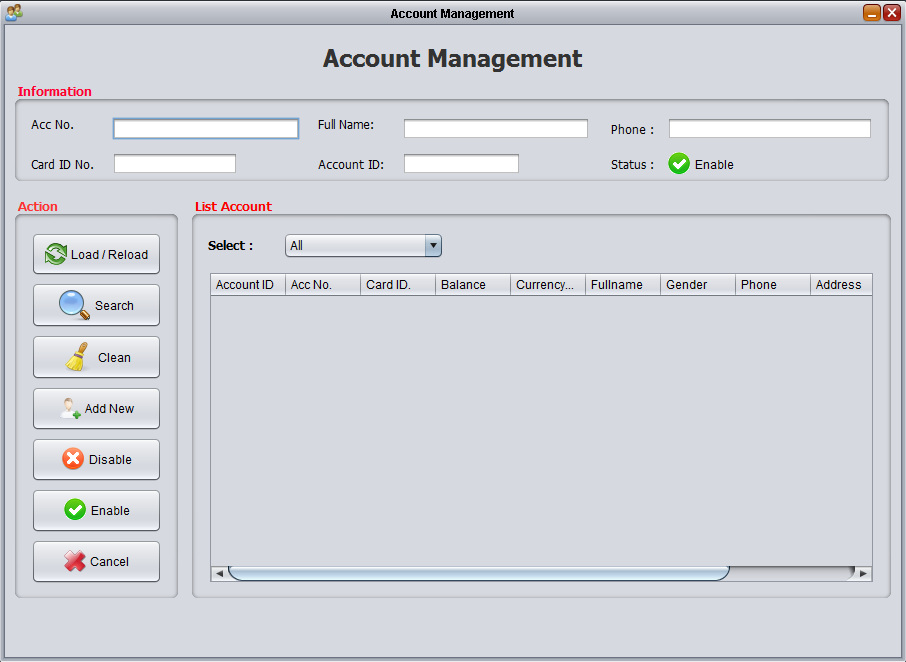
### 1.9 CREATE NEW ACCOUNT



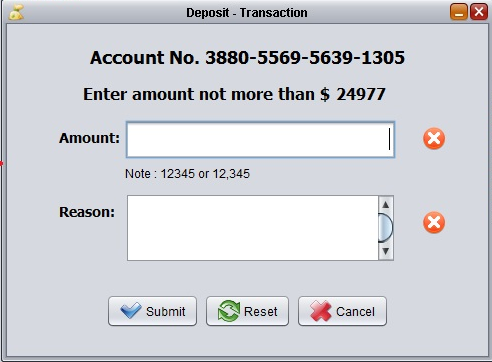
### 1.10 ACCOUNT DETAILS



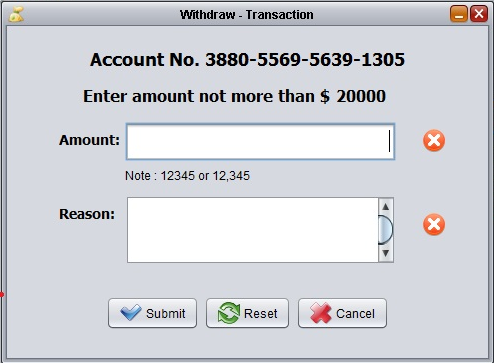
### 1.11 ACCOUNT MANAGER



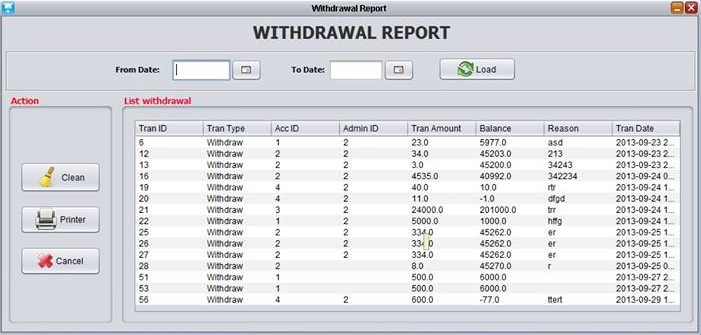
### DEPOSIT



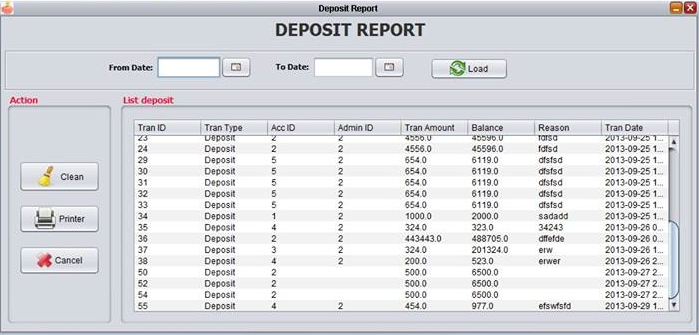
### WITHDRAW



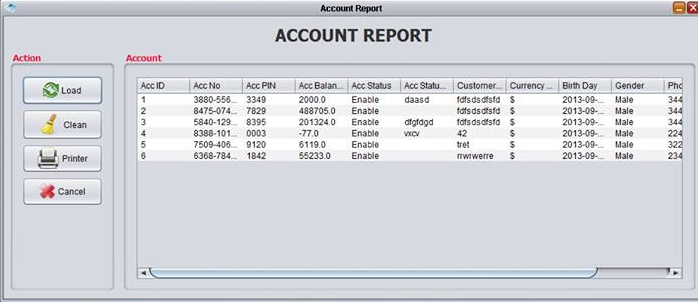
### 1.14 WITHDRAWAL REPORT



### 1.15 DEPOSIT REPORT



### 1.16 ACCOUNT REPORT

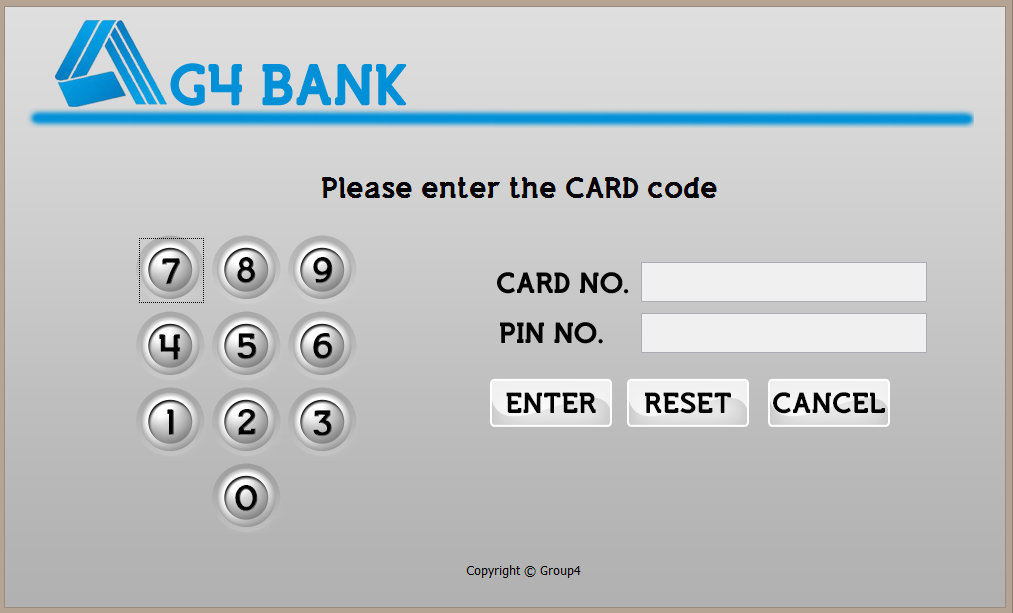


### 1.17 TRANSFER REPORT

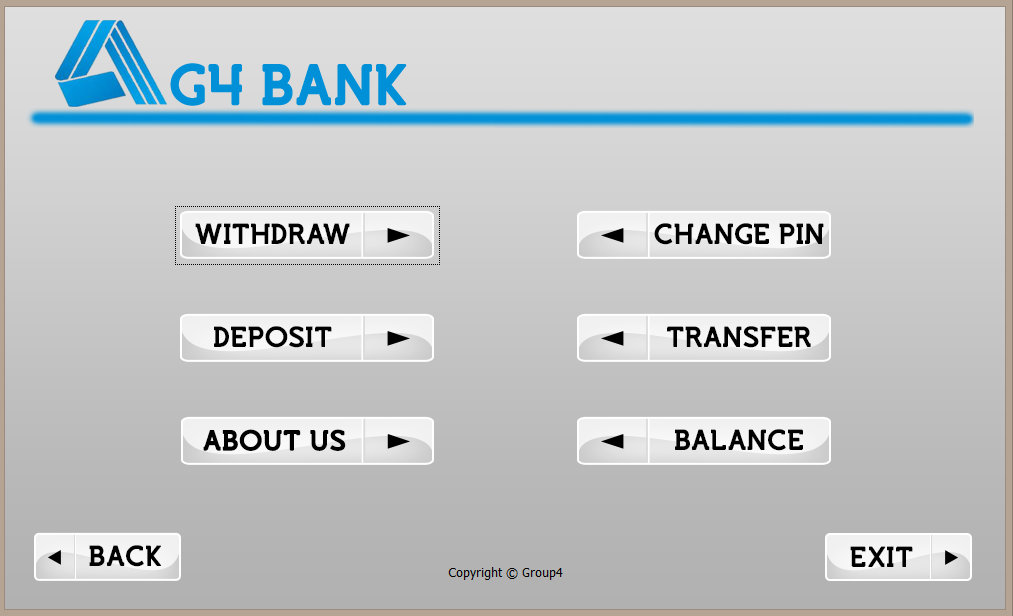


## 2. CUSTOMER

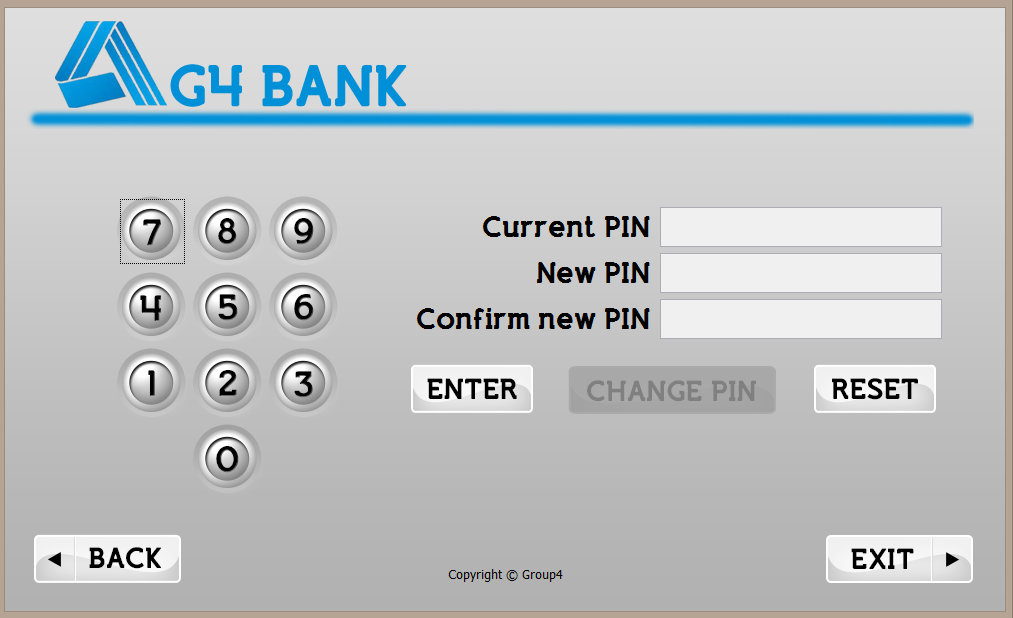
### 1.1 LOGIN



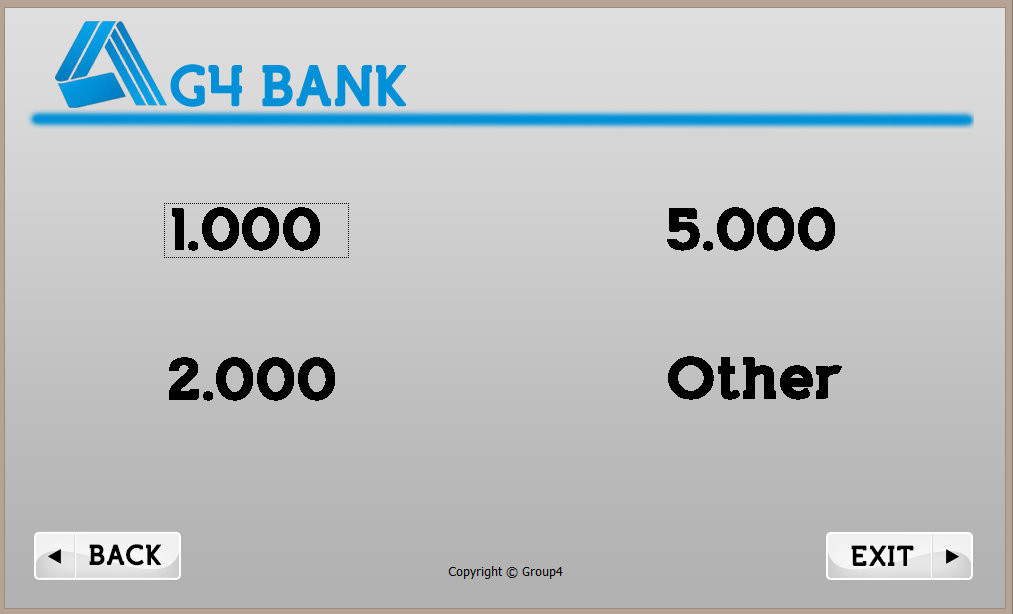
### 1.2 CONTROL

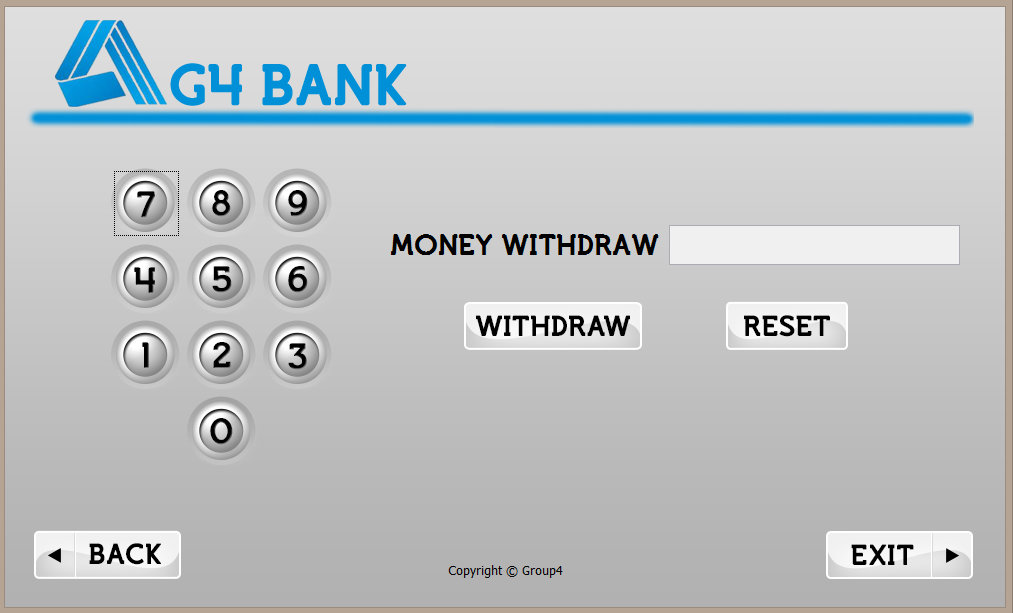


### 1.3 CHANGE PIN

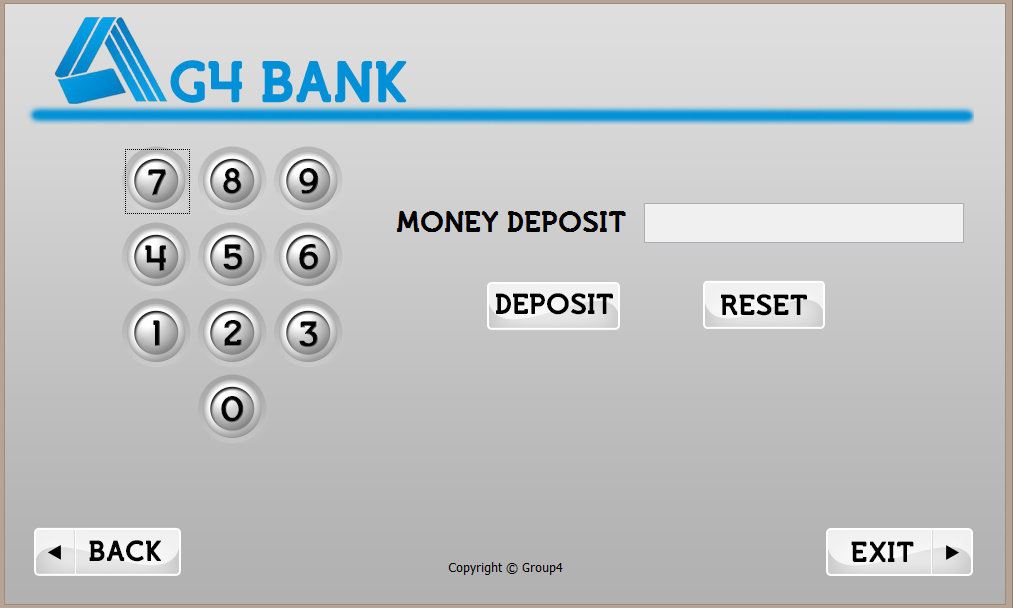


### 1.4 WITHDRAW

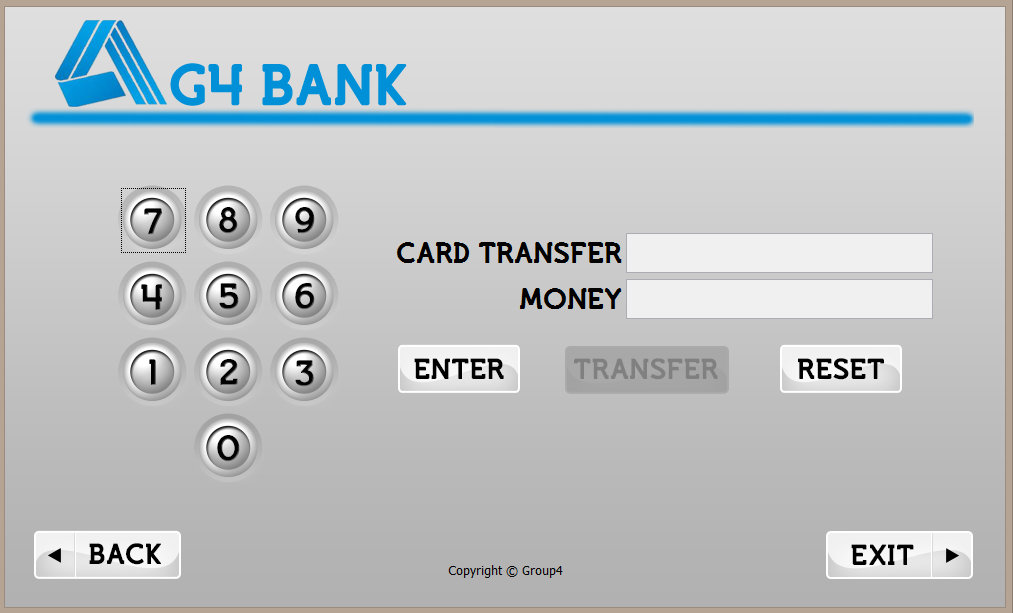




### 1.5 DEPOSIT



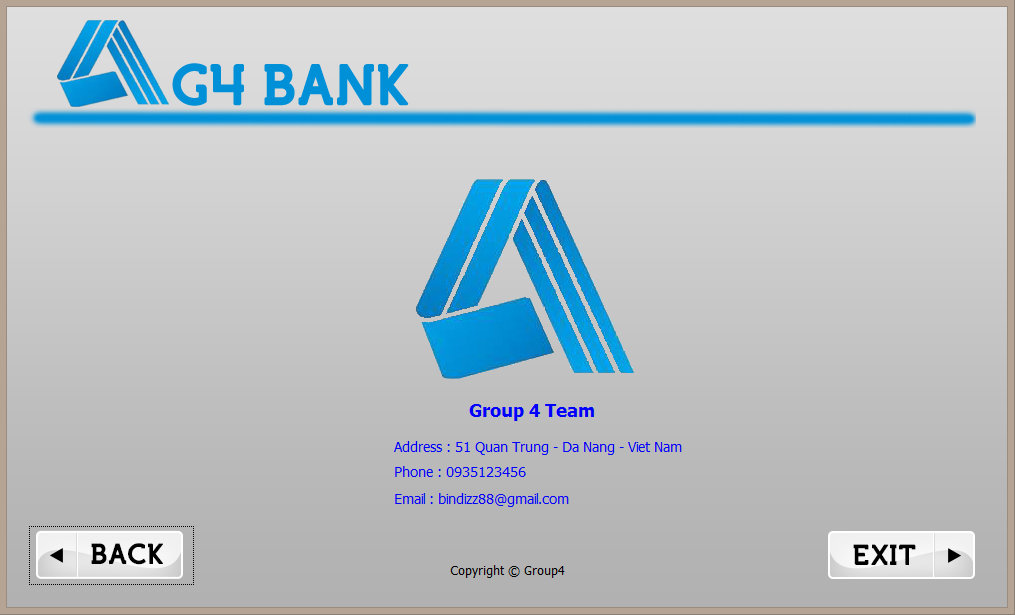
### 1.6 TRANSFER



### 1.7 BALANCE ENQUIRY



### 1.8 BANK INFORMATION



# VII. TASK SHEET

Each members of the team required to take part in any phase and any module of the projects. So the whole team together can complete and understand all tasks of this banking system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tasks** | **Member** | **Decription** | **Status** |
| **1** | Design database | Doan Ngoc Van Khoa | Design database for ATM system | Complete |
| **2** | Code database | Doan Ngoc Van Khoa  Nguyen Van Xuan | Create new database with SQL server | Complete |
| Study and create basic application structure | Nguyen Van Xuan | Study Hibernate and MVC model from web framework like Struts and tried to apply to Netbeans\_Swing project | Complete |
| **3** | Project Report and Documentation | Nguyen Van Xuan | Prepare and Write Eproject documentation.  Creating diagram, chart and UML modeling. | Complete |
| **4** | Nguyen Duc Hoang | Slide show | Complete |
| **MODULE ADMINISTRATOR** | | | | |
| **5** | Design Admin GUI | Nguyen Minh Vuong | Design interface for admin system | Complete |
| **6** | Doan Ngoc Van Khoa |
| **7** | Code model MVC Classes | Doan Ngoc Van Khoa | Create Model, controller, View package and classes | Complete |
| **8** | Adminstrator Management | Nguyen Minh Vuong | Login, logout, change password, create new admin, delete admin and edit admin information | Complete |
| **9** | Customer Management | Nguyen Van Xuan | Create new customer, edit customer, delete customer | Complete |
| **10** | Account management | Doan Ngoc Van Khoa | Create account, edit , delete account, withdrawal , deposit | Complete |
| **11** | Security admin information | Nguyen Van Xuan | Encrypt password admin | Complete |
| **12** | Report system | Nguyen Van Xuan | Report withdrawal, deposit, transfer, account, bank info | Complete |
| **13** | Bank information | Bank information | Complete |
| **MODULE CUSTOMER** | | | | |
| **14** | Design custmer GUI | Nguyen Duc Hoang | Design interface for customer system | Complete |
| **15** | Nguyen Minh Vuong |
| **16** | Code model MVC Classes | Nguyen Van Xuan | Create Model, controller, View package and classes | Complete |
| **17** | Login, Logout | Nguyen Minh Vuong | Input card no , PIN login system | Complete |
| **18** | Change PIN | Doan Ngoc Van Khoa  Nguyen Van Xuan | Change PIN | Complete |
| **19** | Transaction | Withdrawal, deposit, transfer |
| **20** | Balance enquiry | Balance enquiry |
| **21** | Test | Nguyen Minh Vuong | Test application | Complete |
| **22** | Input data | Input data to database | Complete |

# VIII. Testing Checklist

|  |  |
| --- | --- |
| **Checklist Item** | **Result** |
| Correct database design that support the requirements | PASSED |
| Coding style, naming convention, code documentation | PASSED |
| Data Validity on Object level | PASSED |
| Data validity on Form level | PASSED |
| Data length validity | PASSED |
| Required fields data validity | PASSED |

**THE END.**