MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY (CAN THO CAMPUS)
INFORMATION TECHNOLOGY DEPARTMENT

JAVA DESKTOP PRJ311 ASSIGNMENT
SOFTWARE ENGINEERING MAJOR

SUBJECT

AN EWALLET APPLICATION WIIBU PAY

Student: Nguyen Duc Tong
ID: CE140196
Student: Quan Duc Loc
ID: CE140037
Student: Tran Minh Thang
ID: CE140085

Can Tho, March 27th, 2020

SPECIAL THANKS

Mr. Luong Hoang Huong (Master of Information System) for giving guidance and supporting us in doing this assignment.

Mr. Quach Luyl Da (Master of Information System) for helping us to have more time to do this assignment.

Phan Le Trong Nghia for giving courage and supporting us in many ways, especially in giving the documentation guide video.

Tran Anh Van for designing application icons/images, which drastically improves the overall look and feel for the application.

Even though our group members have tried our best to complete the project satisfactorily, we still encounter many problems all the way to assignment completion due to the lack of knowledge regarding programming with Microsoft JDBC SQL Driver and Remote Method Invocation in Java. Hence, the subtle problems that have arisen during the process of assignment development is inevitable. Hopefully we will receive the valuable, constructive feedback from lecturers as well as our peers in order to make the assignment more perfect and usable.

Our group members are all grateful for everyone’s help in this project! We couldn’t have completed the assignment without all your help!

TABLE OF CONTENTS

TERMS, ACRONYMS

Terms/Acronyms	Meaning
DB	Database
UI	User Interface
OS	Operating System

LIST OF TABLES

LIST OF IMAGES

ABSTRACT

There are currently many advances in technology and the technology itself has impacted the world in various aspects, especially our daily lives. We do not need to use the products, let’s say cassette players, analog television, and fax machines, any longer. Such obsolete products are now almost completely replaced by MP3 players, digital television, and emails accordingly. The world around us provides us more and more options to replicate the current products, sometimes even by the state-of-the-art inventions, either by brand new start-ups, or renowned companies and incorporations, bring up the total amount of commodities for us to an endless extent.

However, regarding payment options, there are not really that abundance of alternatives to choose from, notably in online payments. Most of the time, there are scarcely any outstanding options to pick, making it virtually troublesome for users to decide. For instance, on Play Store, there are only a few options for you to pay: using Visa/Mastercard, via PayPal or your mobile career (abolished) and MoMo Ewallet. For young people, who are not yet at 18 to use other payment methods, MoMo is actually a viable pick. Nevertheless, it may be overwhelming for new users to work with, apart from using simple transaction functions such as Deposit, Withdraw and Transfer.

In order to solve one of the most critical issues regarding Ewallet, our group has decided to work on a brand new variation of Ewallet, which provides only a minimal amount of useful functions - an implementation simple enough that even the new users will find it pleasant to use. The assignment "An ewallet application Wiibu Pay" introduces all to an ewallet desktop application called "Wiibu Pay". This product - including 2 applications for Client and Server side - is written using Java programming language on NetBeans IDE 8.2, with Microsoft JDBC Driver 8.2 for SQL Server and Java Swing to design UI.

INTRODUCTION

I. Problem Statement

Payment methods have become more diverse for customers across the globe. People nowadays are not required to bring cash when going out to pay for the bills, or their shopping activities. There are many ways to replicate cash, to illustrate, there are other online payment options such as cards, PayPal, and methods using biometric data such as one's eyes or fingerprints to authenticate is no longer a sci-fi scene as it is more prevalent, exceptionally in developing countries such as China. In addition, not using physical cash can help people to reduce ways to get infected (remarkably during pandemic as current COVID-19), and also such payment methods are more secure in many other aspects, making purchases both easy and safe.

However, online payment has always been a problem for many users. There are not only just a few options, but also, observably, a number of them have completely stopped working, due to many reasons including financial problems, violating a nation's laws, etc. and some have complicated UI in accompaniment with abundance features, making the process of usage difficult. Those factors are what lead users to no choice but to use a payment method that they do not like.

Understanding the current scenario, to satisfy more people's needs, it is a must to implement more options in addition to current online payment methods, especially ewallets. This will help people to get rid of their concerns regarding online payment methods, particularly having only a minimal amount of useful functions.

II. Assignment Goals

This assignment goal is to develop and create a simple ewallet desktop application, which is user-friendly and provides UI for both Admin and User. The application will allow users to do transaction-related functions (deposit, withdraw, transfer, transaction history), account info management (change info, password, delete account, reset password) or other functions such as pay tuition, top up mobile account and see account balance. Likewise, in the admin UI, the application allows them to generate transaction reports (deposit/withdraw/transfer reports) and manage user account management (create new account for admin/user, suspend a user). All data will be stored using SQL Server 2016. UI will be designed using Java Swing. Database connection uses Microsoft JDBC Driver 8.2 for SQL Server. Code written using Java programming language on NetBeans IDE 8.2 for both Client and Server side.

Assignment product is an ewallet desktop application namely “Wiibu Pay”, which is an implementation of an ewallet online payment method.

III. Subjects and Scope of the Study

The subjects for this assignment are students or other subjects who have enthusiasm, passion or the demand to study about Java Swing, connecting to SQL Server databases using Microsoft JDBC driver 8.2, etc.

The scope of study for this assignment is the theoretical basis of creating a desktop application for both Client and Server sides using Java programming language on NetBeans IDE 8.2, designing its UI using Java Swing, connection to SQL Server 2016 database using Microsoft Microsoft JDBC driver 8.2.

IV. Research Content

Main content of the subject of this assignment:

- Program with Java programming language on NetBeans IDE 8.2.
- Designing UI using Java Swing.
- Connect to SQL Server database in Java using JDBC Driver.
- Access and modify database data using queries.
- Apache POI - Java API to export Java JTable data into Excel file.
- Add icons into buttons in Java

V. Main contribution of this assignment subject

V.1 About Theory

Contributions of assignment subject:

- Knowledge to program in Java using NetBeans IDE 8.2.
- Knowledge to program with Apache POI.
- Knowledge to connect to SQL Server database in Java using JDBC Driver.
- Knowledge to design UI using Java Swing
- Knowledge to access and modify databases via queries sent from Java applications.
- Knowledge to add icon into buttons in Java

V.2 About Product

Is an application (with server-side and client-side) designed to use for a small scope of users. Satisfy the demands regarding real life problem solutions.

VI. Document Layout

Document Layout consists of 3 parts:

- Introduction: includes Problem Statement, Assignment Goals, Subjects and Scope of the Study as well as Research Content and Main Contributions of this Assignment Subject.
- Content: contains 3 chapters - first chapter is Problem Definition, second chapter is Design and Implement Solution, third chapter is Testing and Assessment.
- Conclusion: states the achievements, shortcomings and development potentials of this assignment subject.
- Finally, there will be references and appendix.

CONTENT

This section aims at the following reader groups: designers, programmers, testers, maintenance staff and every other person who wants to dive deep into the operation of the product.

b. The Product's Scope

The product comprises a desktop application and database SQL file. It allows a user to login to server through/via client-side and access the functions dependent on whether he or she is a user or admin. For a user, there will be transaction-based functions and account info management. For an admin, there will be creating reports function and user management.

c. Document Layout

The layout of this document is composed of overall description, the outside application communication requirements, functions, non-functions of the product, solution choice and solution analysis.

Parts of the document are made to be appropriate for the following group of users' purposes:

- Group 1: programmers should read the communication requirements, functional requirements.
- Group 2: designers should read all parts of this chapter.
- Group 3: testers should read the application functions and non-functional requirements.
- Group 4: users should read the outside application communication requirements.

I.2.2 Overall Description

a. Product Background

The Ewallet applications have become more prevalent such as MoMo, ViettelPay, ZaloPay, etc. It is a breath of fresh air in mobile payment methods as well a trendy subject for passionate researchers or interested students.

The product consists of the Ewallet desktop application Wiibu pay for Client-side as well as Server-side, and SQL Server DB.

b. Product Functions

- For an ordinary user, the application grants them access to perform:
+ Transaction-related functions (deposit, withdraw, transfer, pay tuition, top up mobile account)
+ Account info management functions (change info, password, reset password, delete account, view transaction history)

+ Export transaction history into an Excel file.

- For an admin, the application will allow them to:
+ Create reports (deposit/withdraw/transfer reports)
+ Export the reports (deposit/withdraw/transfer) to Excel files.
+ User-management functions (create new user, suspend user).

c. User Characteristics:

Users must have basic knowledge for using desktop applications.

d. Operating Environment

- Application operates on computers with Intel Core processors and up.
- Operating System: Windows 8 and up.
- Installed NetBeans IDE or other IDEs that support Java programming language (for both the client and server sides).

e. Execution and Design Constraints

- Execution Constraints:
+ Server-side application must be connected to database when ran
+ Execution must be quick and accurate upon user requests.

- Design Constraints:
+ Friendly UI.
+ Appropriate and clear font size.
+ Graphical components are correctly aligned (visually appealing).

f. Constraint Assumptions

- Data in DB must be backed up frequently to avoid incidents causing data loss.
- Do not tamper with anything in installation path/program path or else the product may not work properly.

I.2.3 Outside communication requirements

a. Hardware Communication

User's desktop requirements: minimum Intel Core 2 1,8GHz and sufficient disk space for storing the product.

b. Software Communication

- Installed OS: Windows 8 and up.
- Pre-installed NetBeans IDE 8.2 or other IDEs that support Java programming language (for both the client and server sides).

c. Media Communication

I.3. Problem Analysis

I.3.1 Product Functions:

Categorizing priority of product functions into 3 levels: High, Medium, and Low.

- **High:** high priority functions are functions that play the foremost important role in the application. It directly affects the functionality of the product. If these functions are broken, the product could not work at all.
- **Medium:** medium priority functions play some important role in the functionality. If these functions are broken, the product could not work properly.
- **Low:** low priority functions play the least important role in the functionality. If these functions are broken, the product still works normally.

I.3.1.1 Create new account (user/admin)

- **Description:** This function allow people to create new user account base on the information included username, password, gender, phone, email, address.
- **Priority:** High
- **Factor:** Everyone (create new user account). Admin(create new user/admin account)

I.3.1.2 Login function

- **Description:** This function is used to authenticate username and password and open UI depend on type of account(user/ admin).
- **Priority:** High
- **Factor:** Registered user who have username and password.

I.3.1.3 Transaction method(Deposit, Withdraw, Transfer)

- **Description:** These functions are used to deposit money, withdraw money, and transfer money to another user in system. History of transaction is recorded into database
- **Priority:** High
- **Factor:** User login with account type user

I.3.1.4 Transaction method (Pay tuition, Top up mobile account)

- **Description:** These function are used to do some online purchases include Pay tuition and top up mobile account. History of transaction is recorded into database
- **Priority:** Medium
- **Factor:** User login with account type user

I.3.1.5 Change information, password

- **Description:** This function allow users to change their information and password. Update new information to database
- **Priority:** High
- **Factor:** User login with account type user and admin

I.3.1.6 Transaction History

- **Description:** This function allow users to show their personal transaction history. Included deposit, withdraw, transfer history
- **Priority:** High
- **Factor:** User login with account type user

I.3.1.7 Create Report(Transaction)

- **Description:** This function allow admin to create deposit report, withdraw report, or Transfer report of all user.
- **Priority:** Medium
- **Factor:** User login with account type admin

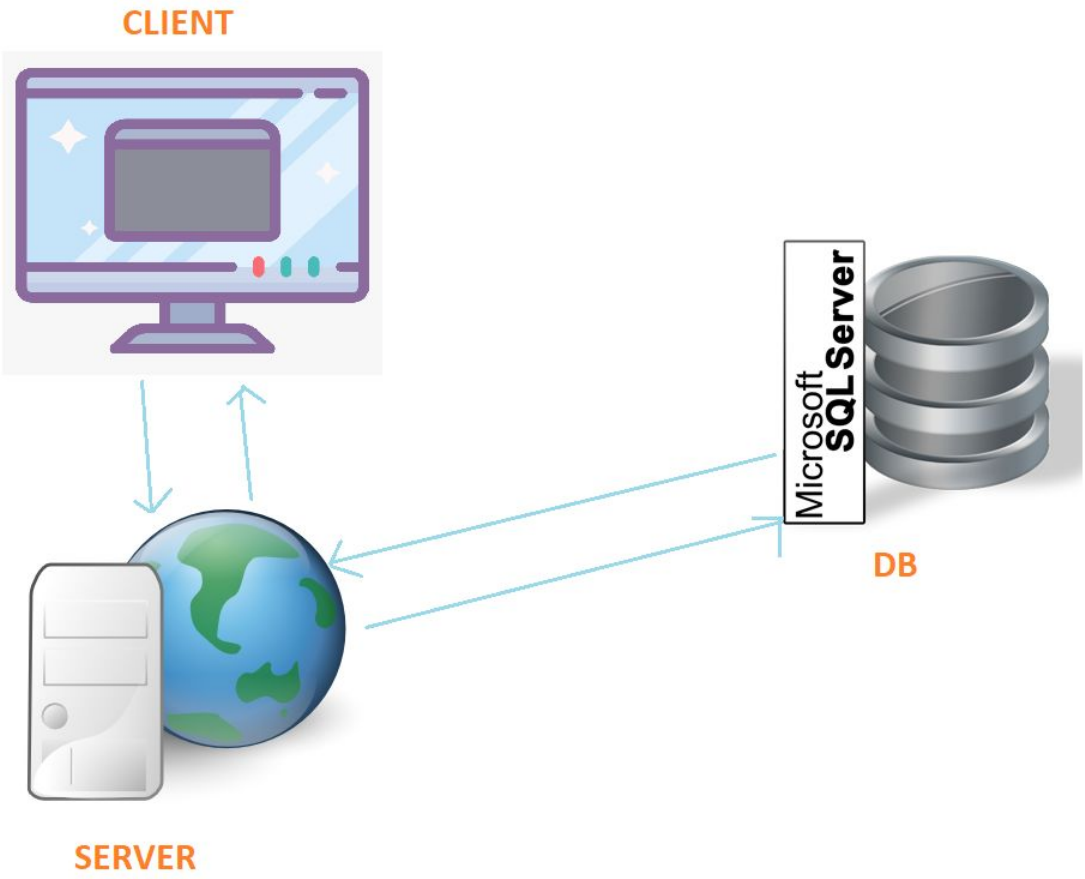
I.3.1.8 Suspend

- **Description:** This function allow admin to suspend user account.
- **Priority:** Medium
- **Factor:** User login with account type admin

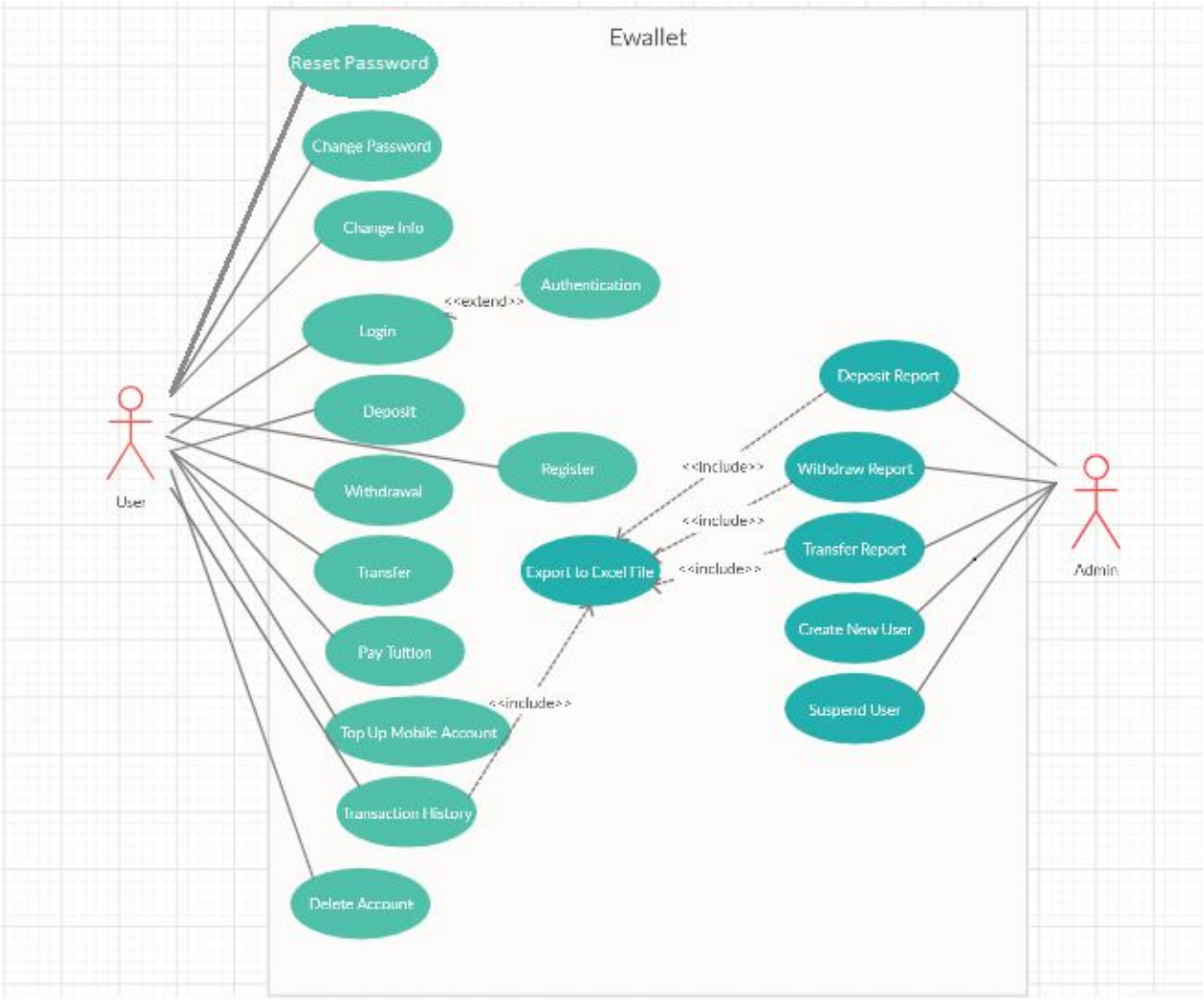
II. Chapter 2 - Design and Solution Implementation

II.1 Design system

II.1.1 Introduction



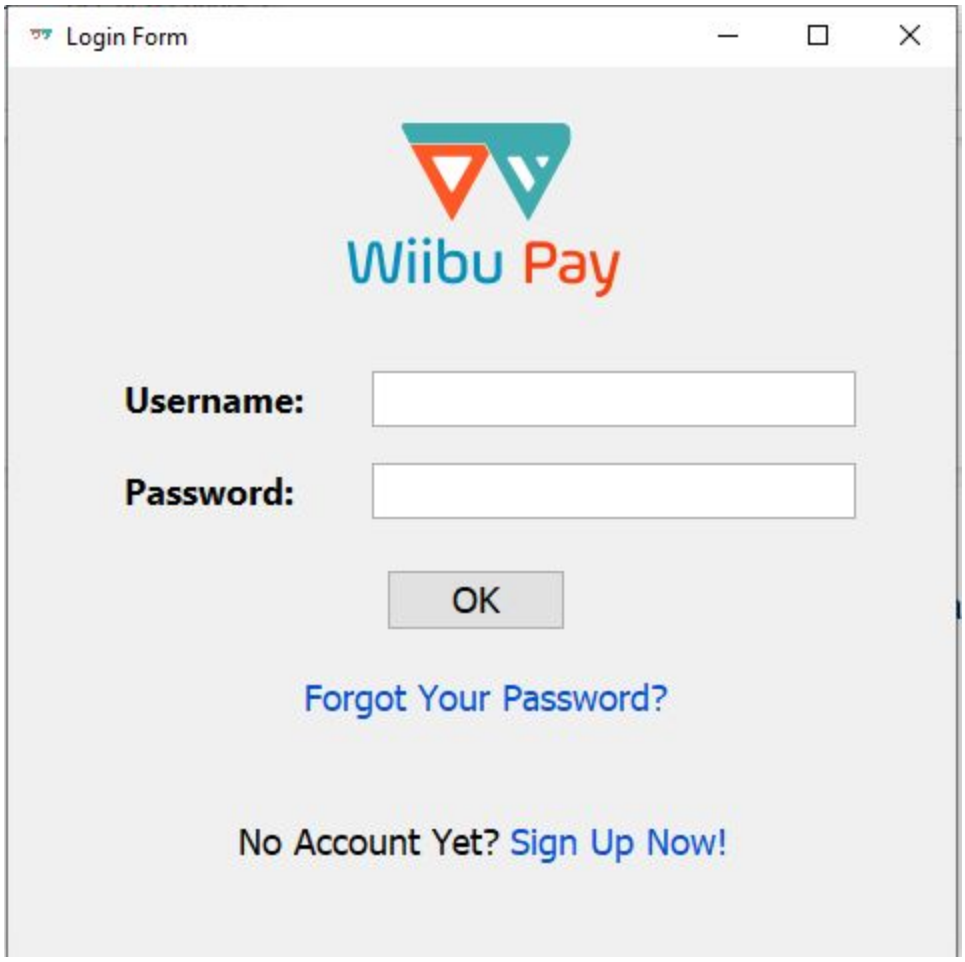
P1. Overall system architecture



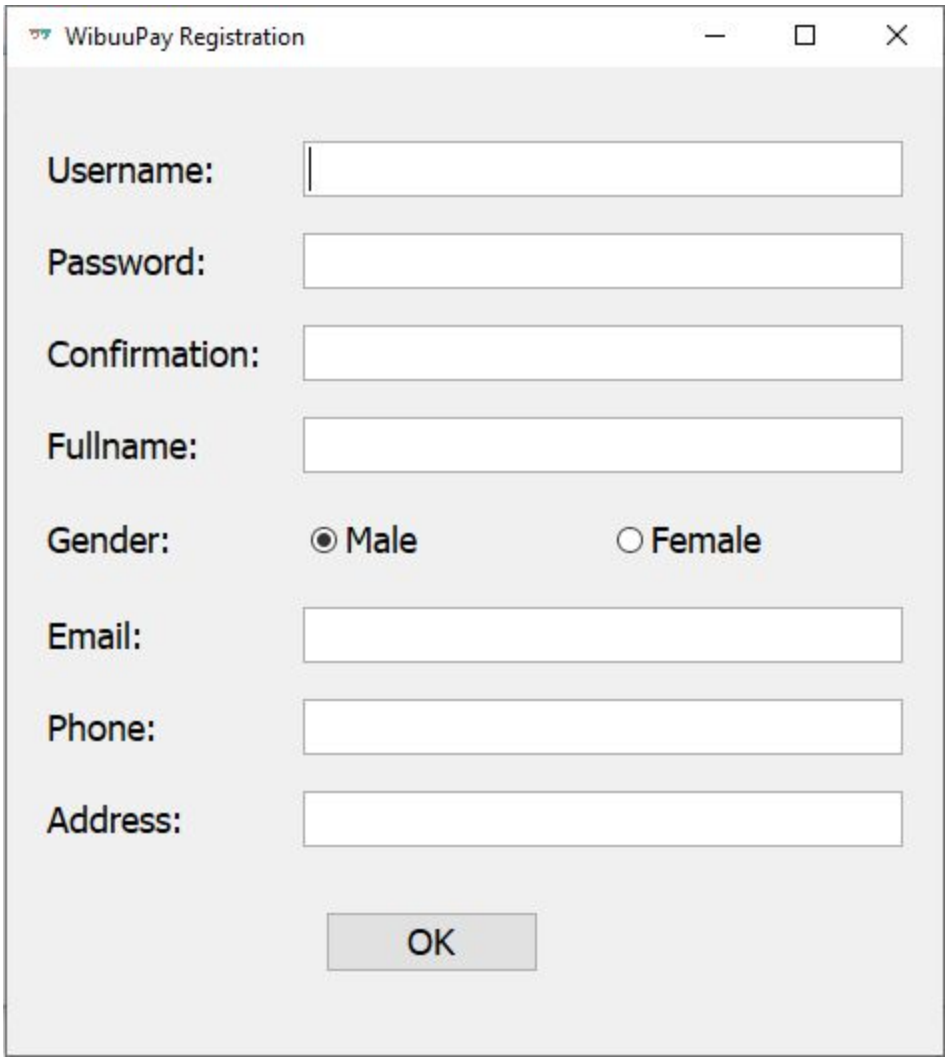
P2. Use-Case Diagram

II.2. Set up solution

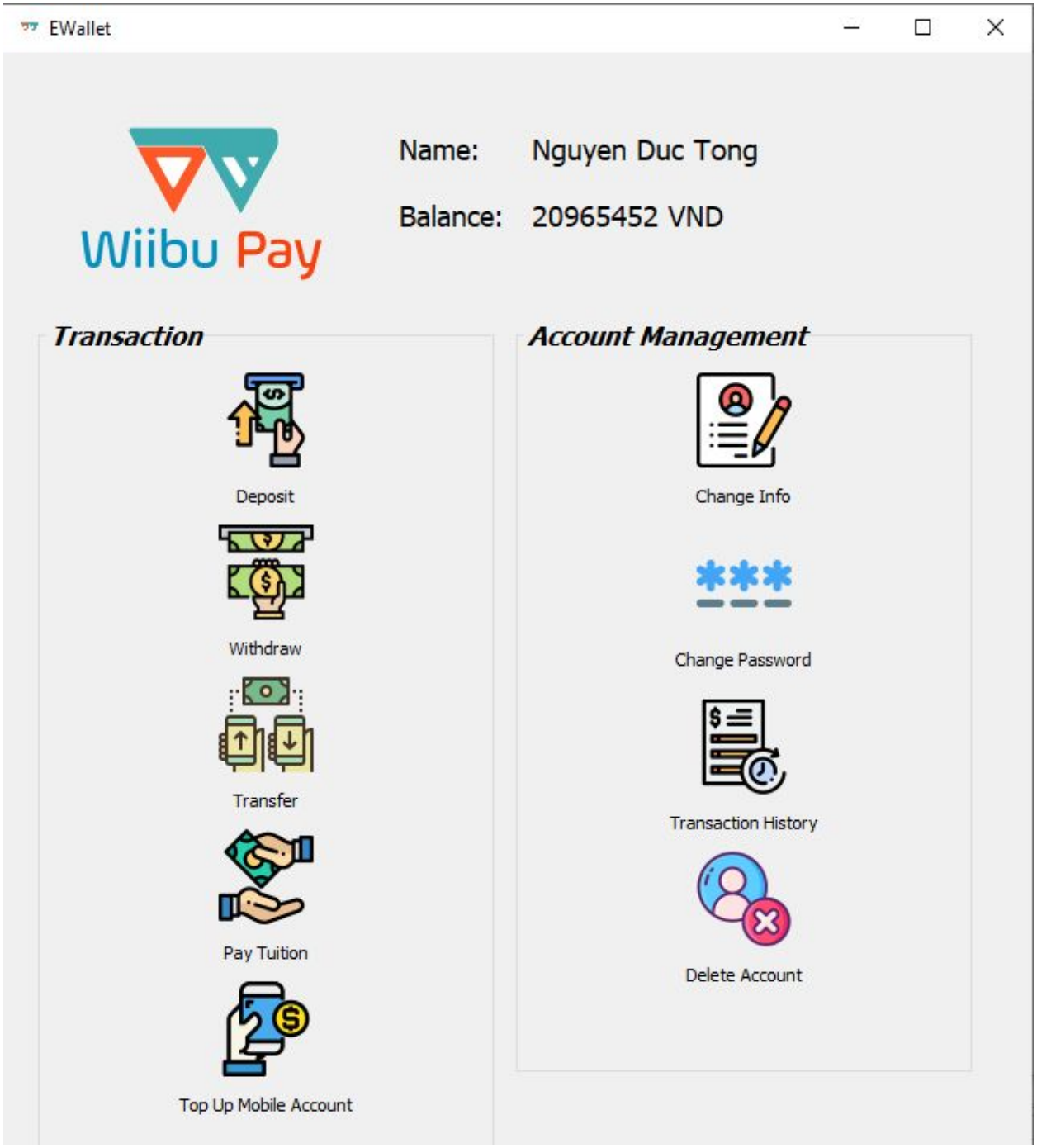
II.2.1. Main function of system



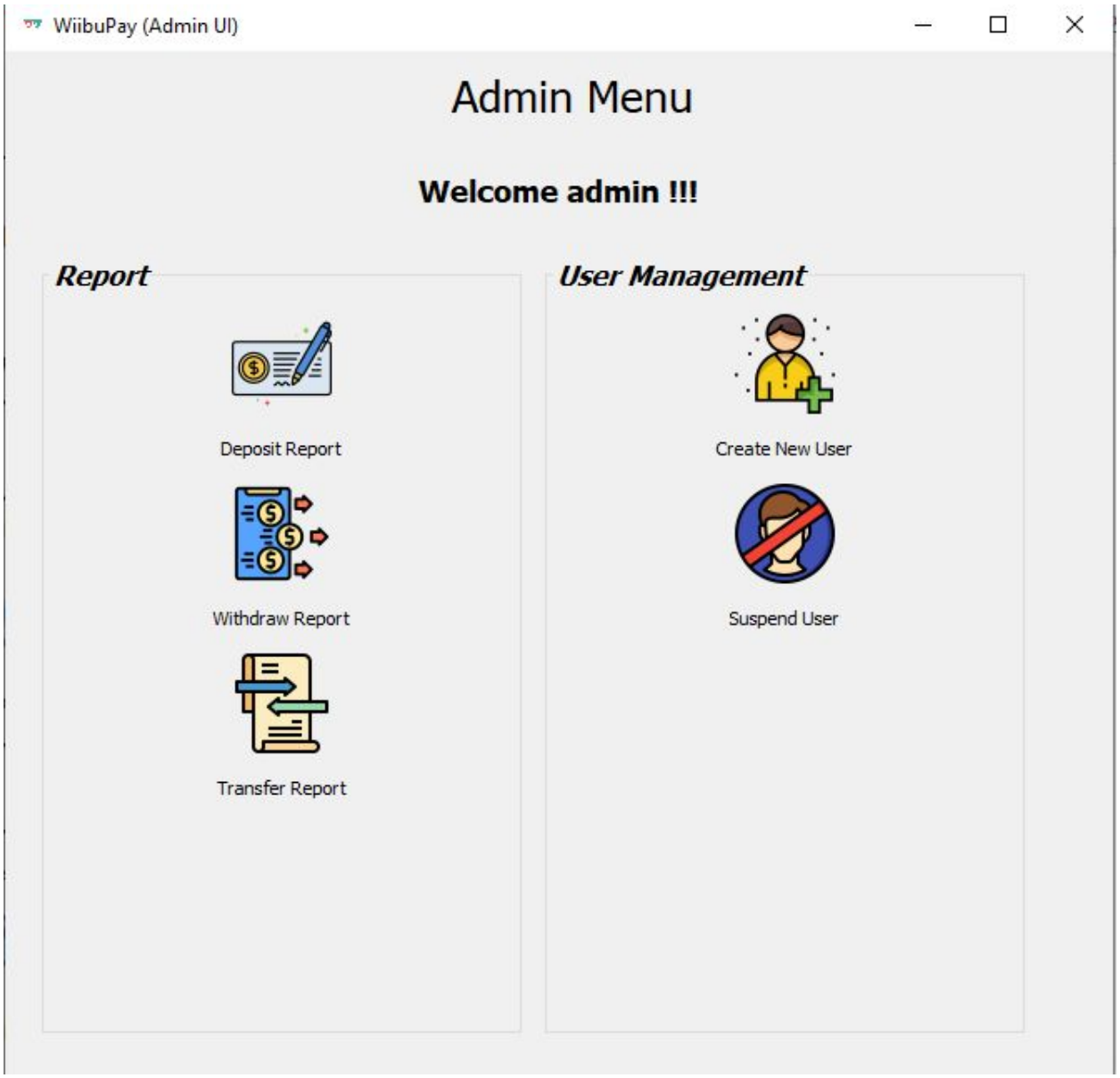
P3. Login Form



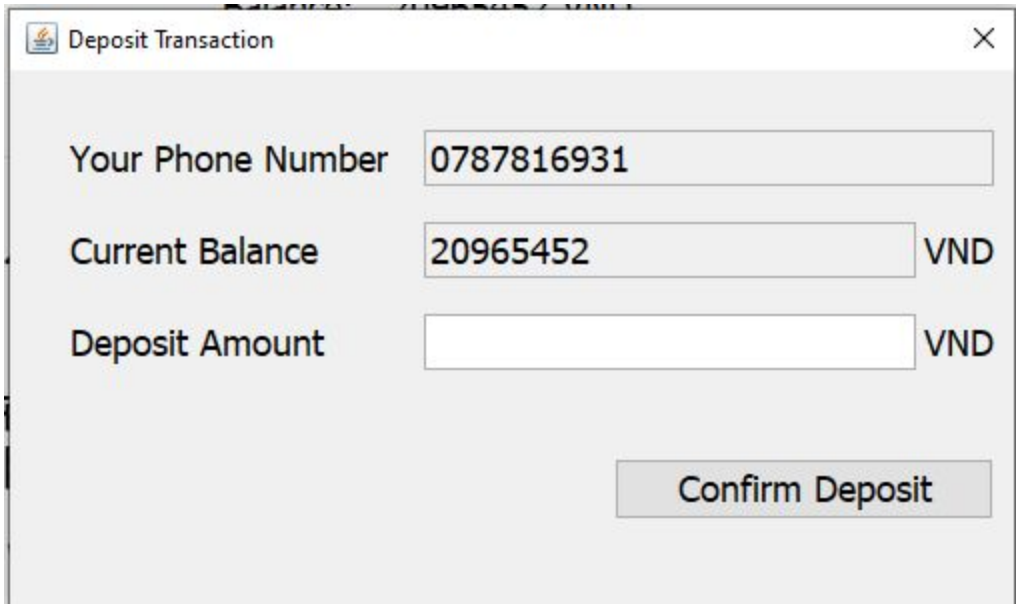
P4. Register Form


EWallet
Wiibu Pay
Name: Nguyen Duc Tong
Balance: 20965452 VND
Transaction
Deposit
Withdraw
Transfer
Pay Tuition
Top Up Mobile Account
Account Management
Change Info
Change Password
Transaction History
Delete Account

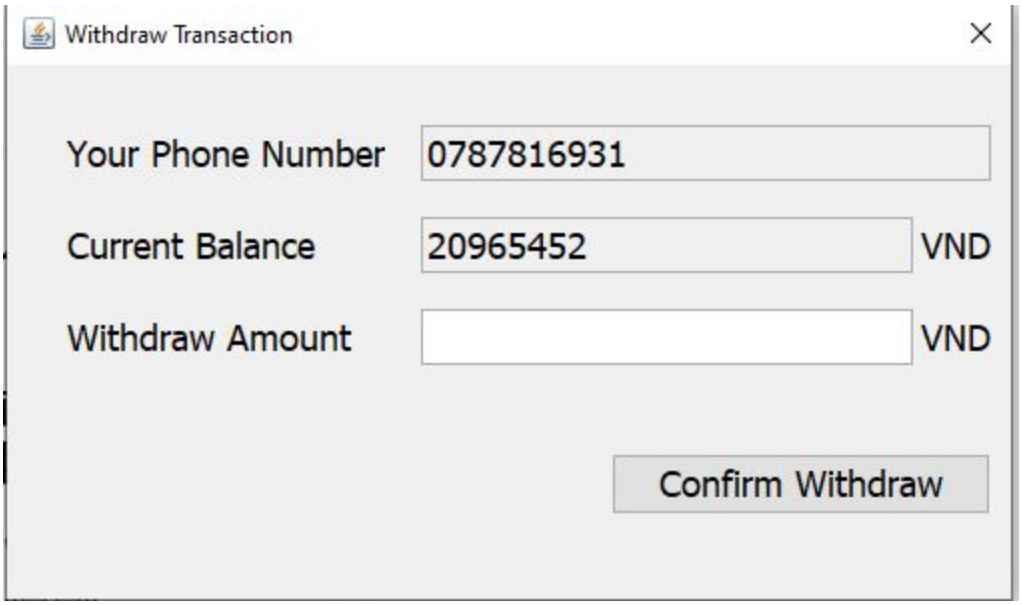
P5. User Menu



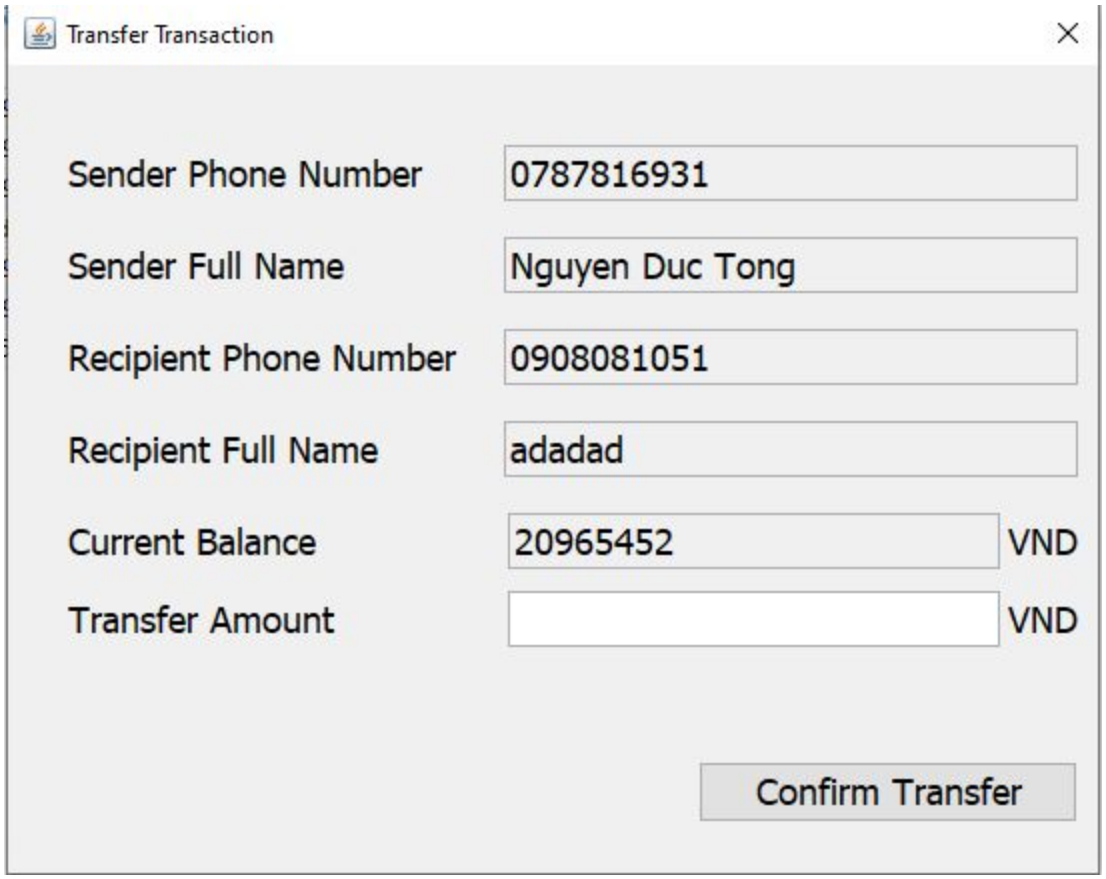
P6. Admin Menu



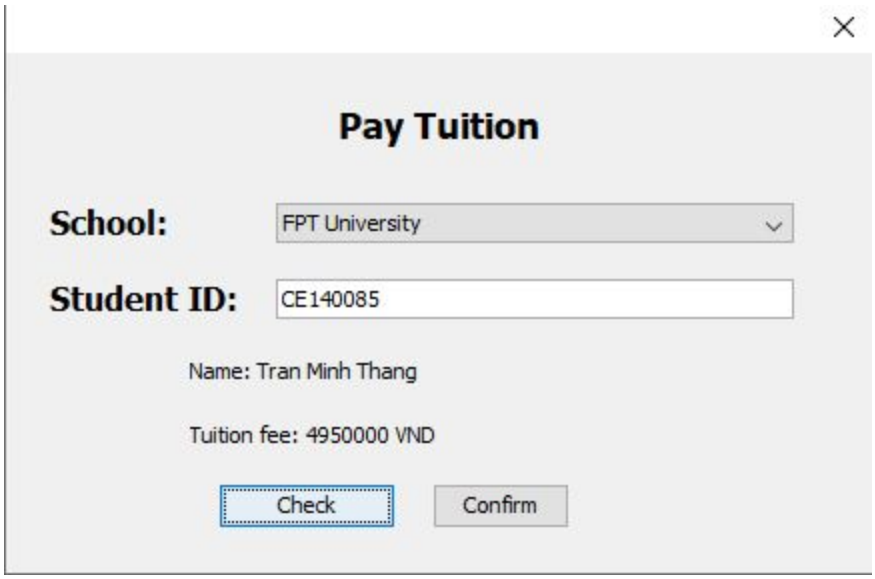
P7. Deposit Transaction



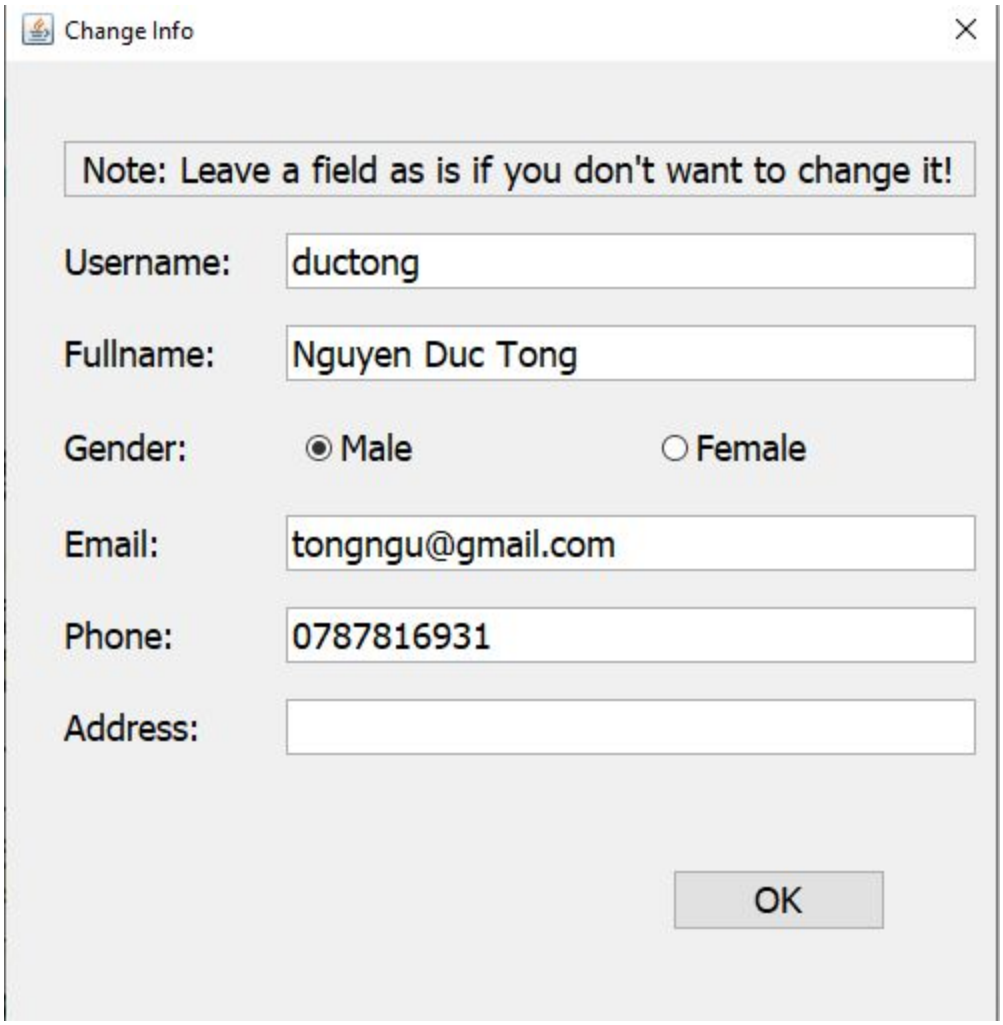
P8. Withdraw Transaction



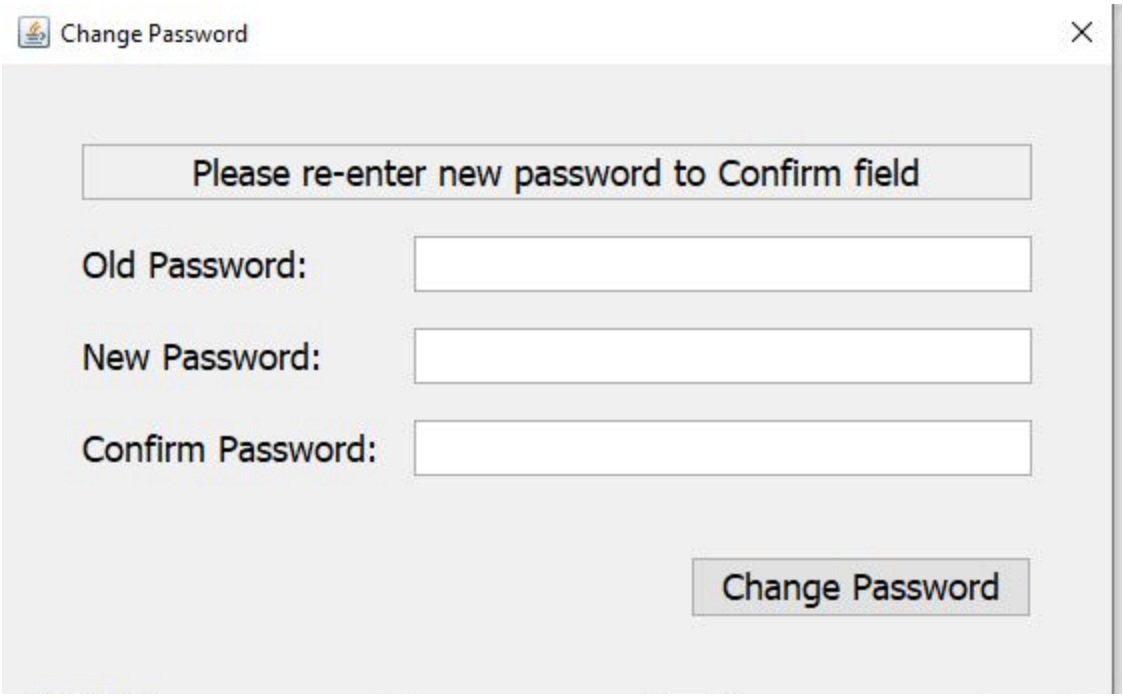
P9. Transfer Transaction



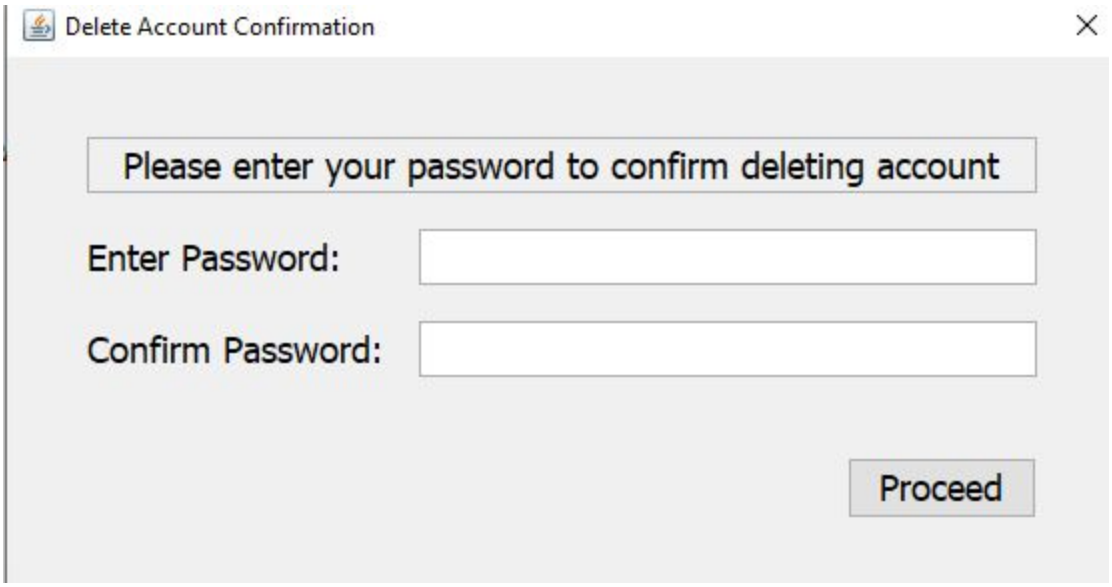
P10. Pay Tuition



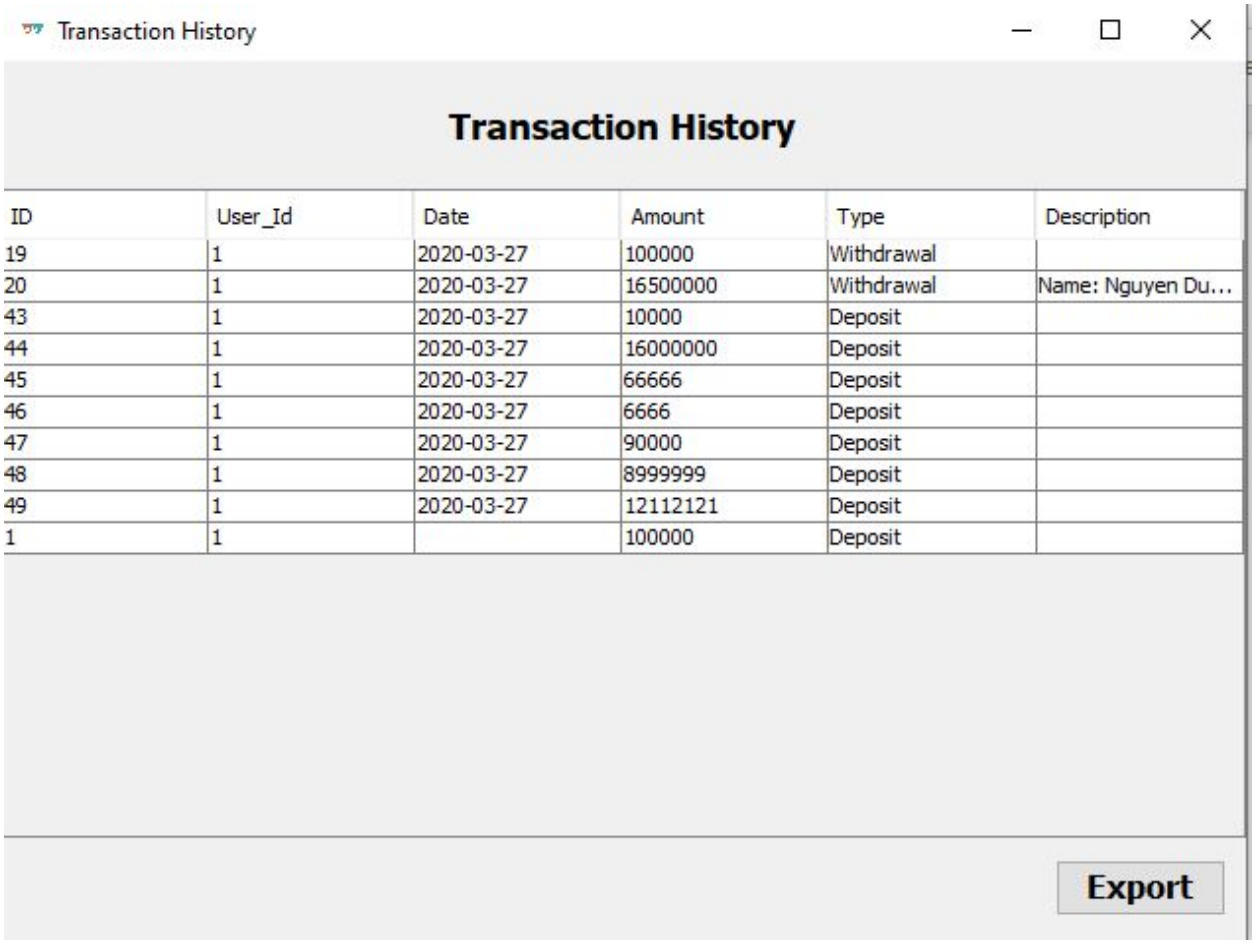
P11. Change Info



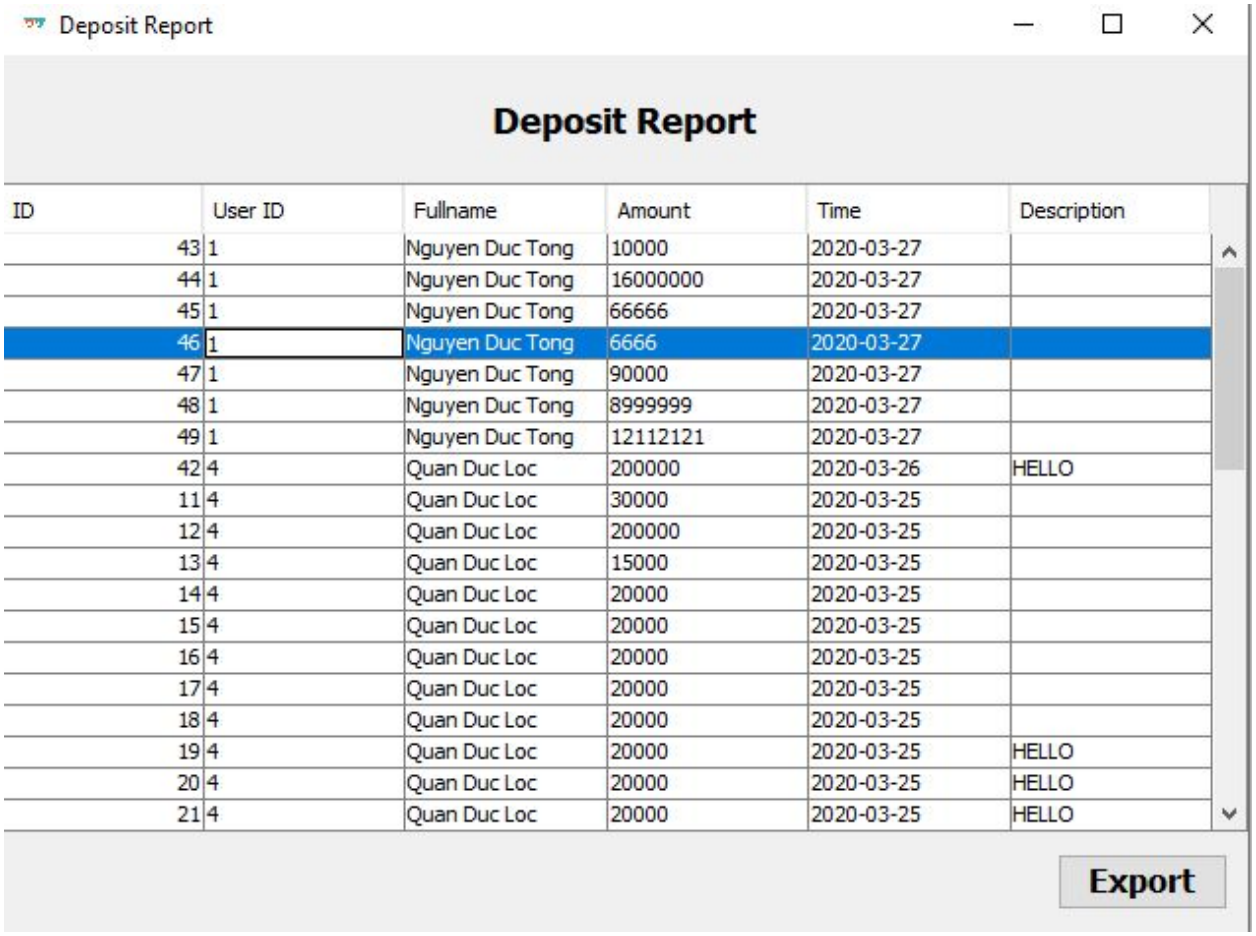
P12. Change Password



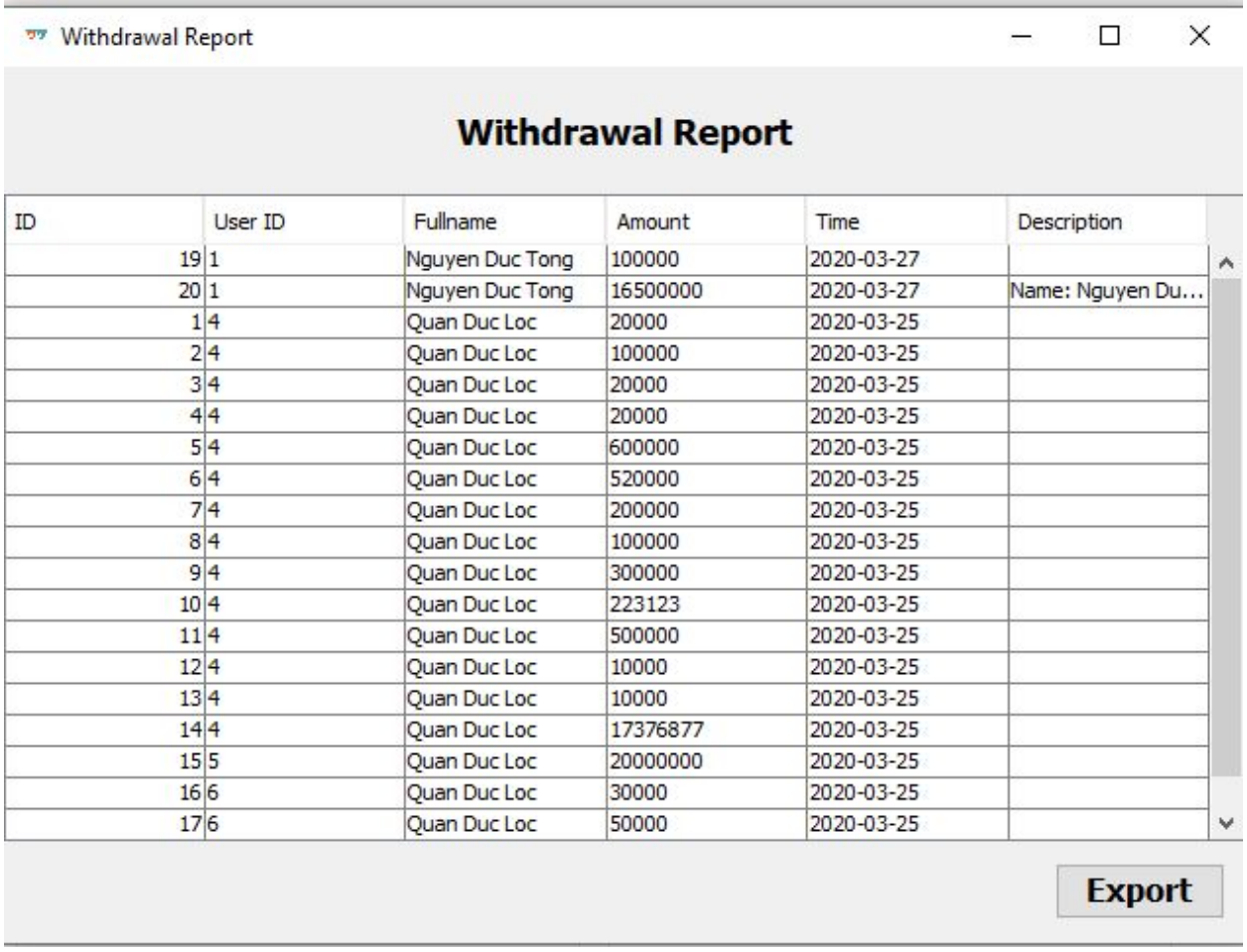
P13. Delete Account



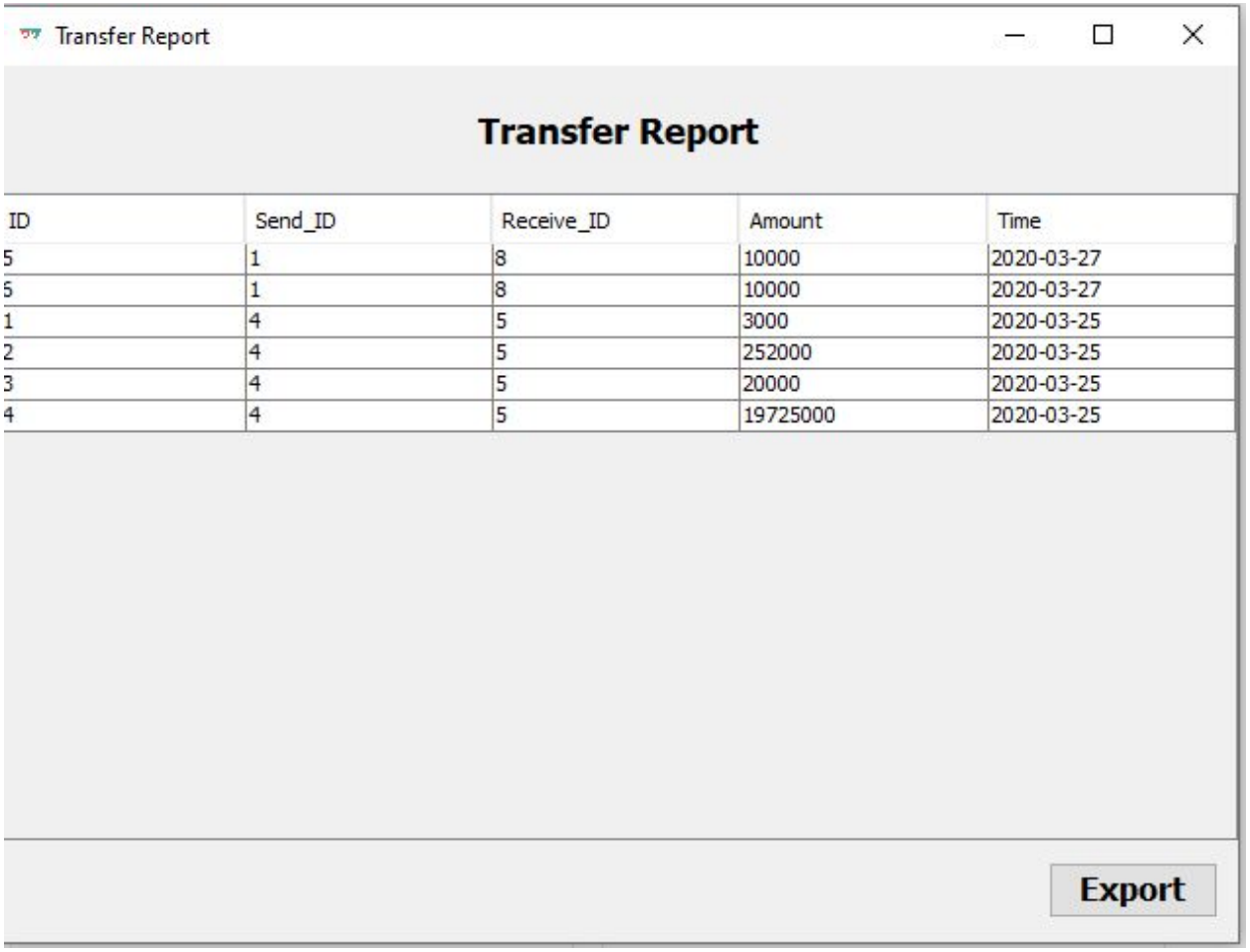
P14. Transaction History



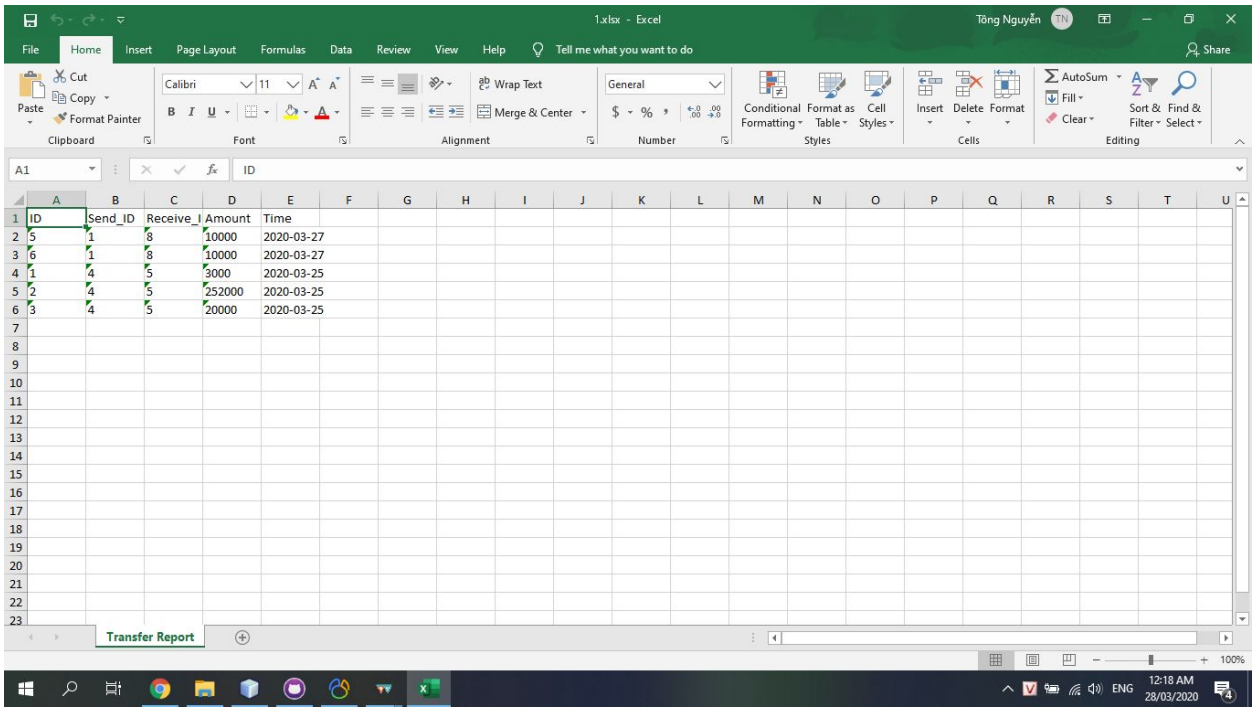
P15. Deposit Report



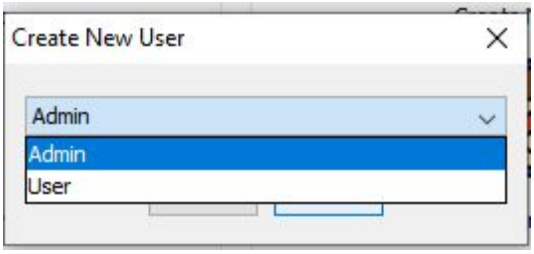
P16.Withdrawal Report



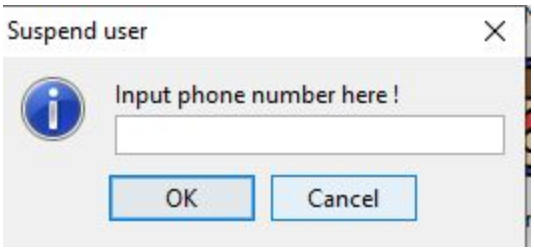
P17. Transfer Report



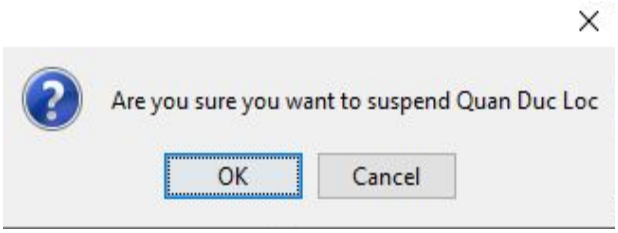
P18. Export to Excel



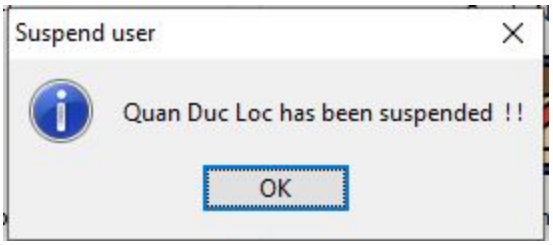
P19. Choose new account role when create new user(Admin)



P20. Suspend user



P21. Confirm if admin want to suspend account



P22. Message after suspended

III. CHAPTER 3 - TESTING AND ASSESSMENT

III.1 TESTING

III.1.1 Target

In order to check that the product is functioning according to the specifications, functions The feature works as expected or not. Presentation of test methods. Find solutions to overcome when an error occurs/

III.1.2 Scope

Create and test test cases corresponding to the internal functions specification documents and design documents.

III.2 TESTS

III.2.1 Test 1 : Login

a. Target

In order to find errors during system logins, see if there are any errors in the system if you enter the wrong account or password.

b. Input

Case	**Account**	**Password**	**Value**
1	ductong	123	TRUE
2	admin	123	TRUE
3	ductong	deptrai	FALSE
4	ductong		FALSE
5		khongcopass	FALSE
6	ductong	123	FALSE

Table 1. Input for test cases 1

c. Output

Case	Desired results	Success
1	You are logged in as a User	x
2	You are logged in as a Admin	x
3	Username or Password is Incorrect	x
4	The password must not be null	x
5	The username must not be null	x
6	Your account has been suspended	x

Table 2. Output for test cases 1

II.2.2 Test 2 : Register

a. Target

In order to find errors during the system registration process, see if there are errors in the case of entering the same username, phone number, missing information.

b. Input

Case	Inputted information	Value
1	NULL ALL	FALSE
2	Duplication phone number	FALSE
3	Duplication phone username	FALSE
4	FULL INFO	TRUE
5	Lengths of phone number different 10 ,not a number	FALSE

Table 3. Input for test cases 2

c. Output

Case	Desired results	Success
1	The username must not be null	x
2	Phone number already bound...	x
3	Username already bound...	x
4	Register successfully	x
5	Please enter phone number 10 digits	x

Table 4. Output for test cases 2

II.2.3 Test 3 : Deposit

a. Target

To identify and make sure the application works in all cases where deposit is invalid during the deposit process.

b. Input

Case	Inputted information	Value
1	Deposit value is NULL	FALSE
2	1000<Deposit value <50000000	TRUE
3	1000>Deposit value or Deposit value >50000000	FALSE
4	Deposit value is not a number	FALSE

Table 5. Input for test cases 3

c. Output

Case	Output information	Value
1	Please input deposit value	x
2	Deposit successfully	x
3	Please enter the value between 1000 and 50000000	x
4	Please input deposit value	x

Table 6. Output for test cases 3

II.2.4 Test 4 : Withdraw

a. Target

To identify and make sure the application works in all cases where withdraw is invalid during the withdraw process.

b. Input

Case	Inputted information	Value
1	Withdraw value is NULL	FALSE
2	1000<Withdraw value <Current balance	TRUE
3	1000>Withdraw value or Withdraw value >Current balance	FALSE
4	Withdraw value is not a number	FALSE

Table 7. Input for test cases 4

c. Output

Case	Output information	Value
1	Please input Withdraw value	x
2	Withdraw successfully	x
3	Please enter the value between 1000 and current balance	x
4	Please input Withdraw value	x

Table 8. Output for test cases 4

II.2.5 Test 5 : Transfer

a. Target

To identify and make sure the application works in all cases where transfer is invalid during the transfer process.

b. Input

Case	Inputted information	Value
1	Wrong phone number	FALSE
2	Input current user phone number	FALSE
3	1000>Transfer value or Transfer value >Current balance	FALSE
4	Transfervalue is not a number	FALSE
5	1000<Transfer value <Current balance	TRUE

Table 9. Input for test cases 5

c. Output

Case	Output information	Value
1	Please input phone number of 10 digits	x
2	Cannot transfer money to yourself	x
3	Please enter the value between 1000 and current balance	x
4	Please input transfer value	x
5	Transfer successfully	x

Table 10. Output for test cases 5

II.2.6 Test 6 : Pay tuition

a. Target

To identify and make sure the application works in all cases where paying tuition works wrong during the pay tuition process.

b. Input

Case	**Inputted information**	**Value**
1	Student ID null or Wrong	FALSE
2	Right Student ID	TRUE
3	Current balance <fee	FALSE
4	Current balance > fee	TRUE
5	Tuition fee =0	FALSE

Table 11. Input for test cases 6

c. Output

Case	**Output information**	**Value**
1	Student not found	x
2	Show Name and Fee	x
3	Not have enough money	x
4	Paid successfully	x
5	Student has paid fee	x

Table 12. Output for test cases 6

II.2.7 Test 7 : Top up mobile account

a. Target

To identify and make sure the application works in all cases where topping up mobile accounts works wrong during the top up process.

b. Input

Case	Inputted information	Value
1	Phone number null or wrong	FALSE
2	Right phone number	TRUE
3	Current balance < amount want to pay	FALSE
4	Current balance > amount want to pay	TRUE

Table 13. Input for test cases 7

c. Output

Case	Output information	Value
1	Please input phone number of 10 digits	x
2	Show Name and Fee	x
3	Not have enough money	x
4	Paid successfully	x

Table 14. Output for test cases 7

II.2.8 Test 8 : Change information

a. Target

To identify and make sure the application works in all cases where users change his/her info correctly or incorrectly.

b. Input

Case	**Inputted information**	**Value**
1	The inputted username has exist in database	FALSE
2	The inputted phone number has exist in database	FALSE
3	Username, full name, email or phone number is null	FALSE
4	Enter all or all but email and not the same username or phone number with other users	TRUE

Table 15. Input for test cases 8

c. Output

Case	**Output information**	**Value**
1	Username already exist	x
2	Phone number already exist	x
3	Error message	x
4	Change successfully	x

Table 16. Output for test cases 8

II.2.9 Test 9 : Change password

a. Target

To identify and make sure the application works in all cases where users change his/her info correctly or incorrectly.

b. Input

Case	Inputted information	Value
1	Old password correct but new password not match	FALSE
2	Old password incorrect	FALSE
3	Old password correct, new password matched	TRUE

Table 17. Input for test cases 9

c. Output

Case	Output information	Value
1	Confirm password not matched	x
2	Old password incorrect	x
3	Change successfully	x

Table 18. Output for test cases 9

II.2.10 Test 10 : Delete account

a. Target

To identify and make sure the application works in all cases where users delete his/her account.

b. Input

Case	Inputted information	Value
1	Old password correct	TRUE
2	Old password incorrect	FALSE

Table 19. Input for test cases 10

c. Output

Case	Output information	Value
1	Delete successfully	x
2	Old password incorrect	x

Table 20. Output for test cases 10

II.2.11 Test 11 : Report

Case	Type	Value
1	Deposit	TRUE
2	Withdraw	TRUE
3	Transfer	TRUE

Table 21. Result for test cases 11

II.2.12 Test 12 : Export Report

Case	Type	Value
1	Deposit	TRUE
2	Withdraw	TRUE
3	Transfer	TRUE

Table 22. Result for test cases 12

CONCLUSION

I. Achievements

- Meet this assignment's requirements, make a flawless combination of SQL Server 2016 Database Management, Microsoft JDBC Driver and Apache POI Java API.
- Uses the hardware optimally, with reasonable pricing, affordable.
- Researched and made a product which contains a server and client based on the theory about Java, Java Swing, Apache POI and SQL Server 2016.
- Complete ewallet applications (for both client and server sides).

II. Shortcomings

- The program lacks feature diversity.
- The JOptionPane's default icons are cropped when displayed for High DPI.
- The code behind somewhat lacks optimization.

III. Development Potentials

- Improve the feature diversity by adding more features, yet retain the minimalism as well as ease of use based on the demands of users/admins.
- Improve maintenance possibilities as well as code optimization and application scalability.
- Fix the High DPI icons problems by using native Windows icons as replacement to improve look and feel.

REFERENCES

[1] https://www.tutorialspoint.com/java/java_overview.htm
[2] https://www.tutorialspoint.com/apache_poi/apache_poi_overview.htm
[3] https://codesjava.com/jdbc-overview
[4] Can Tho University graduation essay, Software Engineering major, topic “"Pests Information Management of Can Tho City Software” - Author Nguyen Chi Thien (April 2016).
[5] https://www.flaticon.com/ (for providing free, exquisite and appealing icons for this assignment)