

UCDF1909ICT(SE), UCDF1909ICT(ITR), UCDF1909ICT, UCDF1909ICT(DI)

HAND OUT DATE: 4th 2021

HAND IN DATE: 6th 2021

WEIGHTAGE:

NAME: Shawn Lee Wen Xuen TP057258

**INSTRUCTIONS TO CANDIDATES:**

**1 Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**

**2 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**

**3 Cases of plagiarism will be penalised.**

**4 The assignment should be bound in an appropriate style (comb bound or stapled).**

**5 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**

**Table of Contents**

[1.0 Execution Outputs 4](#_Toc79357877)

[1.1 Staff Login 4](#_Toc79357878)

[1.2 Staff Home 5](#_Toc79357879)

[1.3 Browse Client 6](#_Toc79357880)

[1.4 Client Registration 7](#_Toc79357881)

[1.5 Client Login 9](#_Toc79357882)

[1.6 Client Home 10](#_Toc79357883)

[1.7 Withdrawal & Deposit 11](#_Toc79357884)

[1.8 Showing Client QR Login Code 12](#_Toc79357885)

[1.9 Client Editor 13](#_Toc79357886)

[2.0 Object-Oriented Concepts & Java Features 14](#_Toc79357887)

[2.1 Polymorphism 14](#_Toc79357888)

[2.1.1 Method overloading 14](#_Toc79357889)

[2.1.2 Method overriding 15](#_Toc79357890)

[2.2 Data Abstraction 16](#_Toc79357891)

[2.2.1 Private 16](#_Toc79357892)

[2.3 Classes 17](#_Toc79357893)

[2.3.1 Inheritance 17](#_Toc79357894)

[2.3.2 Static Methods & Attributes 18](#_Toc79357895)

[2.4 Threads 19](#_Toc79357896)

[2.5 Event Handling 20](#_Toc79357897)

[2.6 Exception Handling 21](#_Toc79357898)

[3.0 Additional Features 22](#_Toc79357899)

[3.1 Password Encryption 22](#_Toc79357900)

[3.2 QR Login 23](#_Toc79357901)

[3.3 Password Changing 26](#_Toc79357902)

[3.4 Transaction History 27](#_Toc79357903)

[3.5 Browse Client 29](#_Toc79357904)

[4.0 Assumptions 30](#_Toc79357905)

[5.0 References 31](#_Toc79357906)

# Execution Outputs

## Staff Login

Figure 1.1.a: Staff login page

From the main access page, the Staff Login page, staff inputs their and registered username and password to verify their identity.

## Staff Home

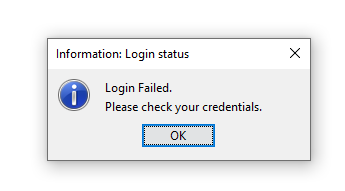
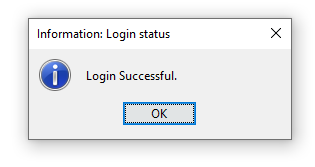


Figure 1.2.a: Login status

Figure 1.2.b: Staff Home Page

If the login credentials match the record in the text database of staff, the staff home window will be shown to the user as the login page is disposed. The admin’s username will be shown in the window and have access to “browse clients”, “register new client”, “login client” and “QR login”, all with their respective functions stated in the name.

## Browse Client

Staff can see the records retrieved from the client’s database, showing their account number, type, name, contact methods, to allow staff to easily contact the clients if needed.

Figure 1.3.a: Browse Client window, accessed by staff

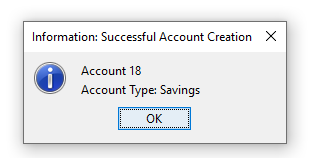
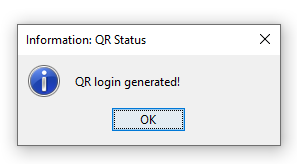
## Client Registration

Figure 1.4.a: Client Registration with invalid input, disabled “Create Account” button

Figure 1.4.b: Client Registration with valid input, enabled “Create Account” button

The registration page will be shown with newly generated account number based on the last record of clients in the database. Staff inputs the details of the clients with input validations in format sensitive fields like Identification Number, Contact Number and Email. Staff can also choose an account type (“Savings” or “Current”), as decided by the client from the dropdown box.

Figure 1.4.c: Successful QR login code generation and account creation.



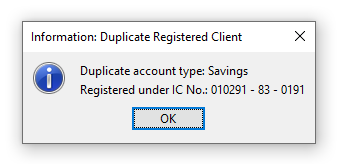
Successful account creation notification will be shown to staff if no duplicate record was inserted into the database, followed with the generation of QR login code for the client.

Figure 1.4.d: Failed Account Creation: Duplicate Registered Client

Unsuccessful account creation notification will be shown to the staff if the account type already existed for the client.

## Client Login

Figure 1.4.a: Staff manually logging in client with client inputting their own credentials

Staff will allow the client to have access to the physical input devices to input their own credentials and will be prompted notifications based on the login status of the client.

Figure 1.5.b: Login successful notification

Figure 1.5.c: Login failed notification

## Client Home

Figure 1.5.a: Client Home

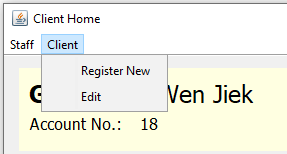
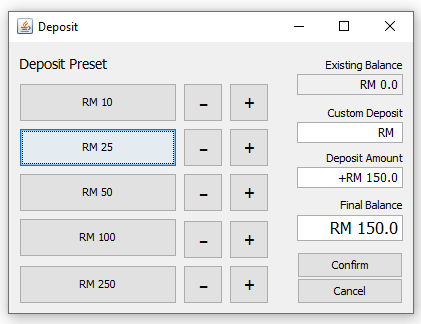
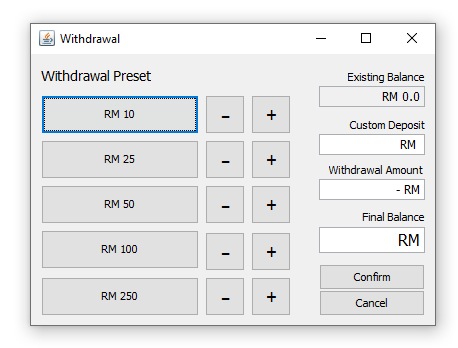
After validating the client input credentials with records in database, client home window will be shown to the staff with shown name, account number, type, balance, and recent transactions. The buttons have functionalities as respective to their name.

Figure 1.6.b: Client menu dropdown

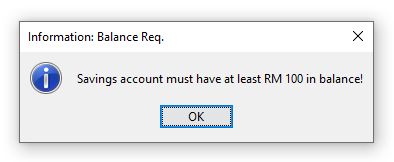
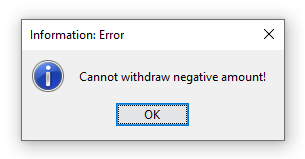
## Withdrawal & Deposit

Figure 1.6.a: Withdrawal & deposit window



After pressing “withdraw”, withdrawal window will be shown to the staff with amount displacement buttons, custom deposit textfield, confirm and cancel buttons. Staff can add withdrawal or reduce withdrawal amount by clicking on the respective buttons or using the custom deposit textfield. Staff can confirm the withdrawal amount by clicking the “confirm” button or cancel the transaction by clicking “cancel”. This applies to the deposit window as well.

Figure 1.7.b: Withdrawal error messages



Staff will be prompted with error notifications if there are logical error in the withdrawal or deposit amount or unmet account balance requirements.

## Showing Client QR Login Code

Figure 1.7.a: QR image file for client with account no. 18

After staff clicks on “show QR” in the client’s home, the QR image file of the respective client will be shown in the default system window to allow printing or easy access of the image file to be sent using email.

## Client Editor

Figure 1.8.a: Edit menu dropdown

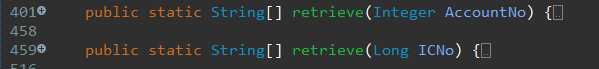
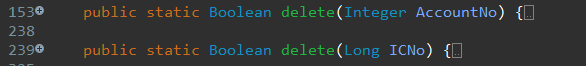
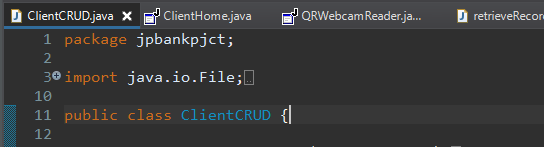
Figure 1.9.b: Client editor window

After clicking “edit” from the client menu bar in Client Home window, the client editor window will be shown to the staff. The editor form is prefilled with information of the client, and has fields that are changeable such as the contact information. The staff can also change the account password of the client by clicking on the “change password” button and delete the account using the “delete” button. From the “edit” menu dropdown, staff also has access to “reset” to reset to original form and “delete client” to delete every record of the client in the database.

# Object-Oriented Concepts & Java Features

## Polymorphism

### Method overloading



In the ClientCRUD class, which is mainly handling CRUD (add, edit, delete, view operations) for client, method overloading is applied to methods like delete() and retrieve(). Method overloading allows the methods with the same name to carry out different functions or tasks.

Figure 2.1.a: Collapsed view of ClientCRUD class

For example, the method delete() in line 153, will delete clients based on the received parameter of the client’s account numbers; whereas in line 239, the delete() method is overloaded to delete clients based on the received of the client’s Identification number instead.

### Method overriding

Figure 2.1.b: Overriding of mouseEntered() and mouseExited() methods in the MouseAdapter object, created from the addMouseListener()

The password field, txtPass has the addMouseListener() method, and under the class of MouseAdapter, there are default methods of mouseEntered() and mouseExited(), which does not contain any customized coded tasks.

In order to customize the task when the method is invoked, method overriding is required to replace the default method body with customized method body, for example, in line 63, a line of code is added into the mouseEntered() method, so that the echo character can be disabled when the mouse enters the field.

## Data Abstraction

### Private

Figure 2.2.a: Private declared variables

The attributes or variables, serialVersionUID, clientHome, contentPane and retrievedClientRecord, are declared as private attributes to ensure the attributes are only accessible by the parent class and not the other classes, increasing the security of the program.

## Classes

### Inheritance

Figure 2.3.a: Collapsed view of ClientRegistration class

ClientCRUD class contains many methods and variables that will be reused in other classes. In order to support the reusability of members and methods of ClientCRUD, for example, in line 39, the ClientRegistration class is set to inherit the methods and attributes of ClientCRUD using the “extends” keyword.

The new relationship between the two classes after the inheritance are ClientRegistration (subclass) inherits ClientCRUD (superclass).

### Static Methods & Attributes

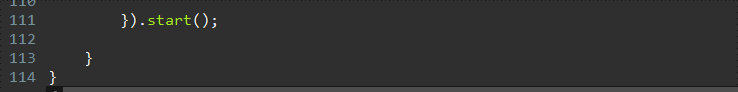
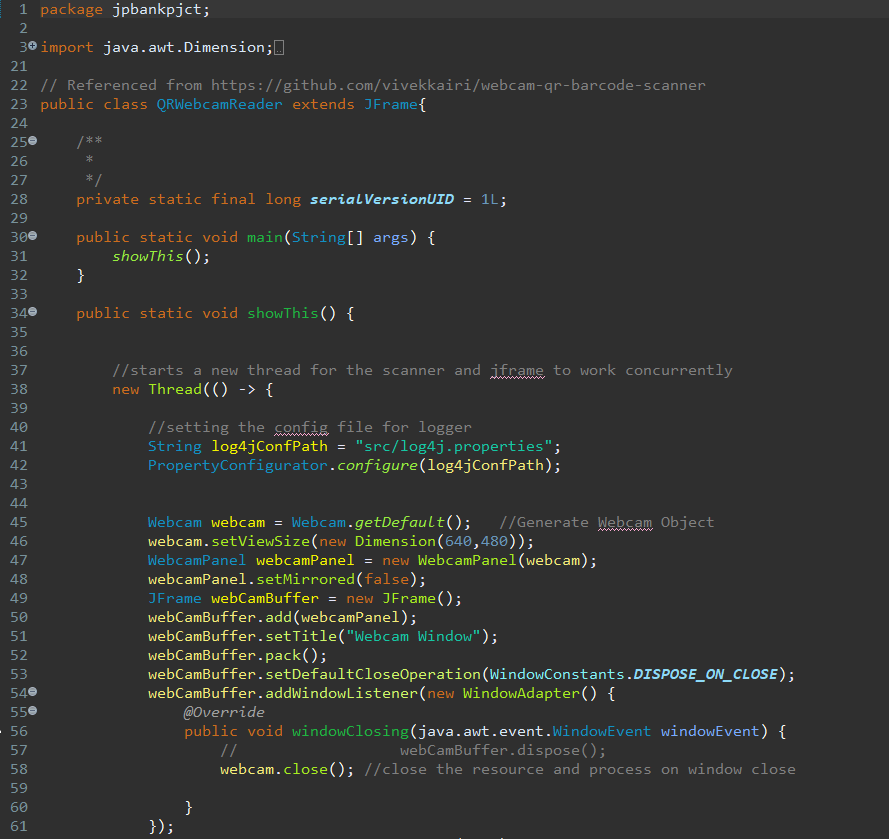
Figure 2.3.b: Methods in ClientCRUD class

In the ClientCRUD class, the methods are declared as static methods because the methods are not using instance attributes and the codes are independent off instance creation. The static methods are also not meant to be changed or undergo polymorphism like method overriding. The static method such as create(), can also access the attributes and methods of the same class, ClientCRUD, or other classes, but its only limited to static methods and attributes only, with the exclusion of non-static methods and attributes.

## Threads

In the class of QRWebcamReader, a frame and the display of running camera feed are required to run concurrently for the camera frames in the window to update constantly. In order to run the codes concurrently, a new thread is made in line 38 and the thread is stared on line 111 in order to run the code effectively in the program without interrupting the main thread, which can cause the program to have issues.

Figure 2.4.a: Thread usage in QRWebcamReader class



## Event Handling

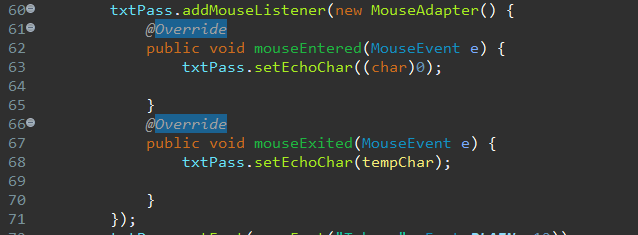
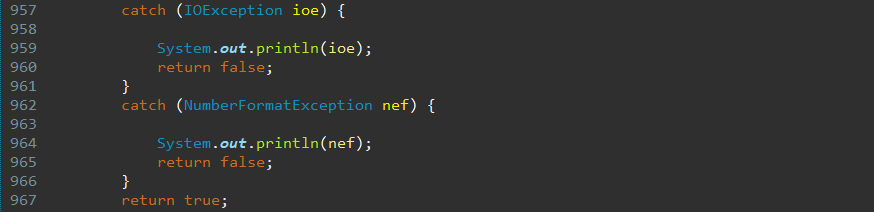
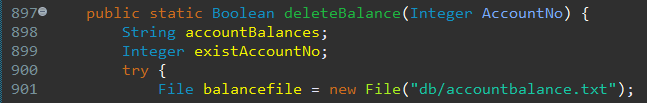
Password field, txtPass, is added with event handling method addMouseListener(), which is the listener for mouse events, so that txtPass will listen for mouse events. In this example, for specific events like mouseEntered() and mouseExisted(), the body of the methods are overriden with inclusion of customized codes, and when the event is fired by user by entering or exiting the password field, the respective codes will be ran.

Figure 2.5.a: Event handling for txtPass, a password field

## Exception Handling

In the method deleteBalance(), the body is wrapped with a try/catch statement in order to catch exceptions (error) that might be thrown while running the program. By applying proper exception handling, when an error occurs, for example:

Figure 2.6.a: Try/Catch statement in method deleteBalance()



in line 957, input output errors,   
in line 962, string to numeric types error,

specific codes will be executed to handle the error instead of the program crashing because of unhandled exceptions. This will ensure the program to run consistently even when an error or exception is thrown.

# Additional Features

## Password Encryption

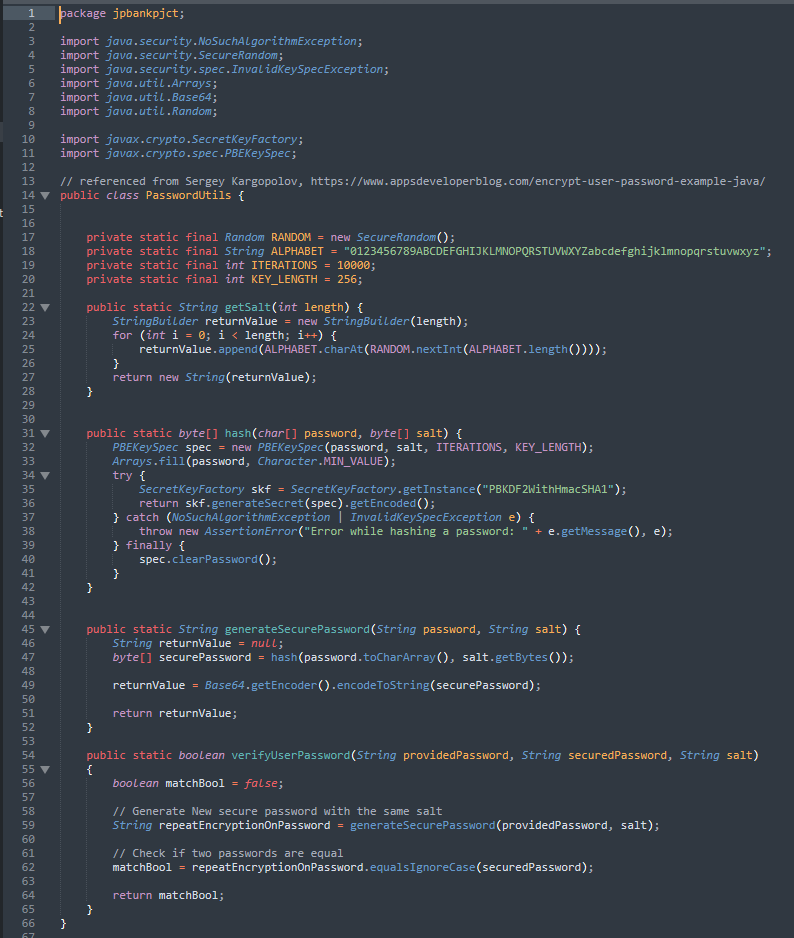
Password provided by clients are encrypted with sha1 hash generator and salting of password to increase the security of the password with prevention of reverse-engineering. The password can be then verified by decrypting it with the generated encrypted pass word and saved password salt value from the database.

Figure 3.1.a: PasswordUtils Class

## QR Login

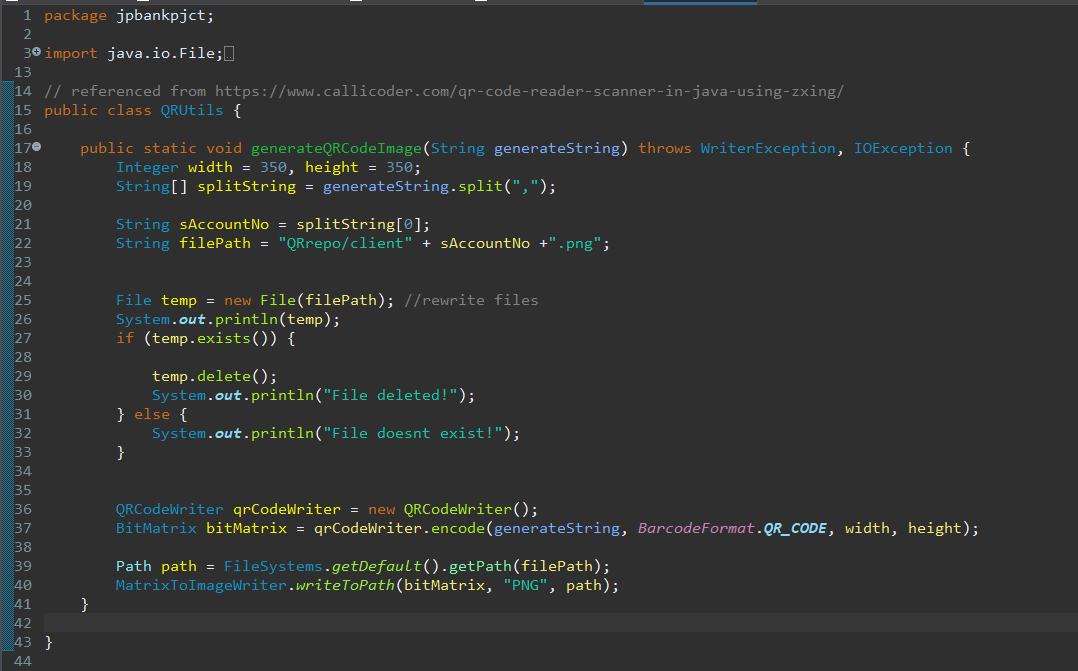
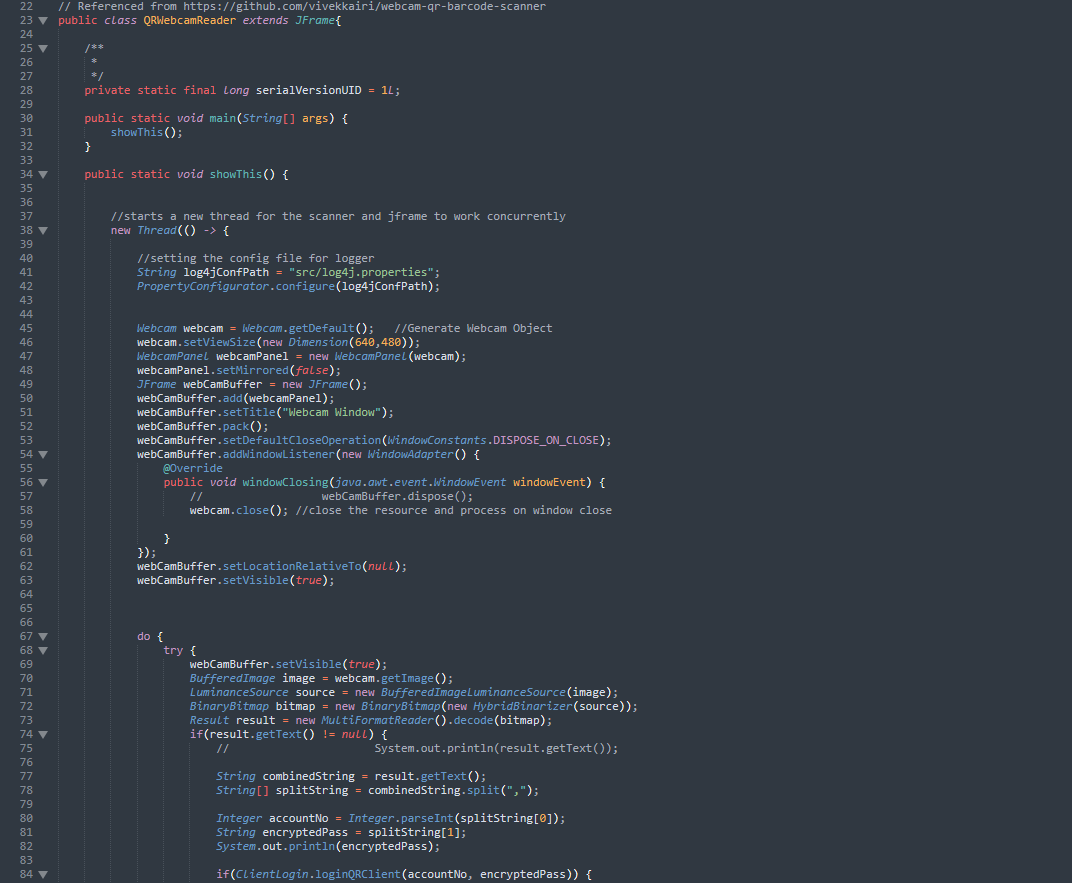


Figure 3.2.a: Code Snippet for generating the QR login code for clients

Figure 3.2.b: Code Snippet for Webcam QR Reader for Client QR login



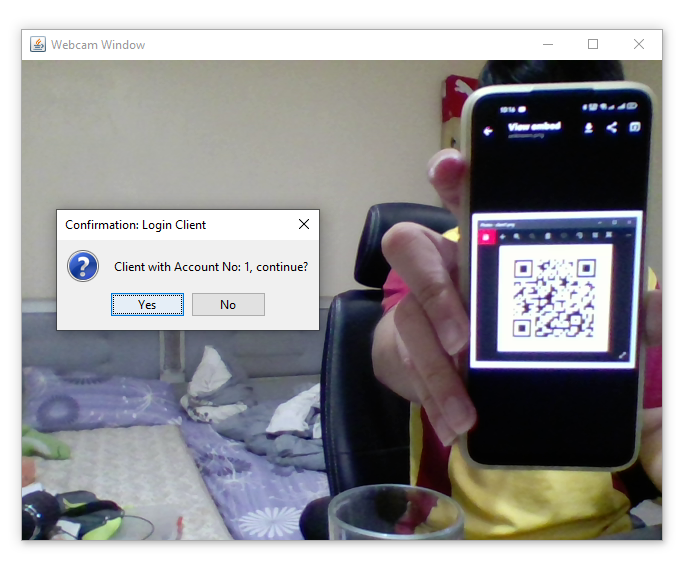


Figure 3.2.d: example of client logging in with his QR login code on his phone

Figure 3.2.c: Generated QR login code for client 1

QR login code are generated in png format whenever a new client registers or password changes. The QR code generation is achieved by utilizing external package from ZXing (Zebra Crojssing) library and the reading of QR code using webcam is achieved with sarxos’ library that includes webcam capture API combined with QR decoder from ZXing package.

## Password Changing

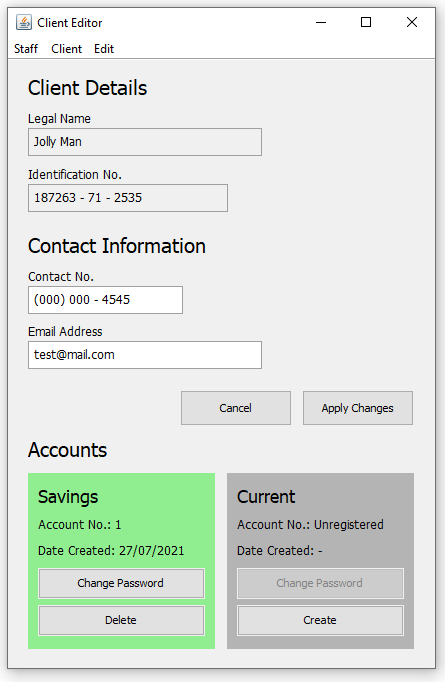


Figure 3.3.a: Client Editor opened for Client: Jolly Man

Clients are allowed to change individual passwords for both type of account if applicable. New QR login code will be generated, replacing the old QR, using the new encrypted password to ensure the client can login using the QR method.

Figure 3.3.b: Change Password Form

## Transaction History

Figure 3.4.a: Transaction History of Client: Jolly Man

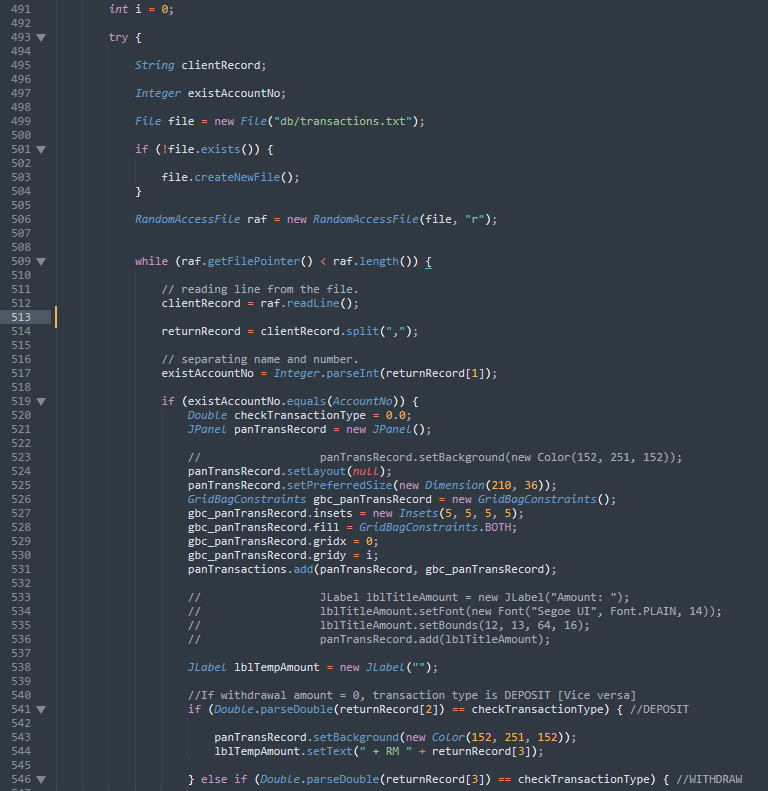


Figure 3.4.b: Code snippet for generating the transaction rows, Part 1



Figure 3.4.c: Code snippet for generating the transaction rows, Part 2

Transaction history of clients are shown in client home to provide easy viewing of the details of the transaction at a glance. Green rows indicate deposits and red rows indicate withdrawals.

## Browse Client

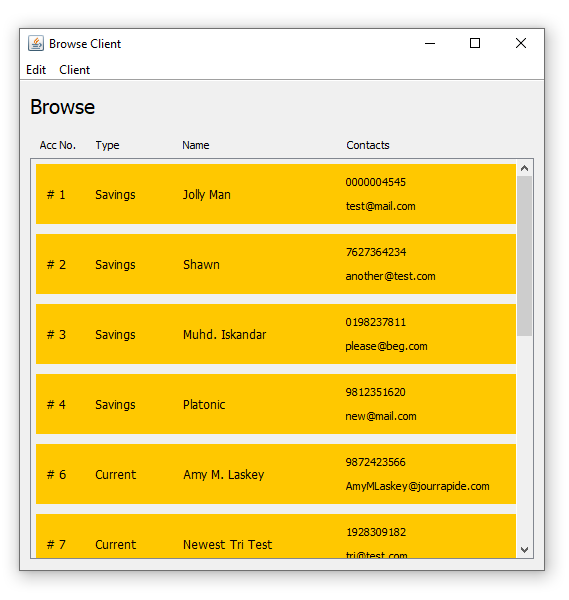
From this window, using the same generative GUI while loop as the Transaction History, client’s non-sensitive information are shown instead, allowing staff to have easy access to information like contacts details for easier job operations.

Figure 3.5.a: Browse Client window, accessed by staff

# Assumptions

External libraries, ZXing, webcam-capture, apache commons validator are expected to be included in the class paths and database folder are expected to have client, staff, transaction and accountdetails database files. QRepo folder is expected to be created in the root folder.

For this banking system, every unique client are identified by their unique National Identification number, and every client can have only two types of account at once, which have different account numbers.

Staffs are assumed to have administrative control of the client account after logging in with client’s credential and can perform administrative operations like editing and removing the client or individual accounts.

Clients are assumed to have direct control of the input methods, keyboard, and camera, from the control of the staff to input password for registration, login, change of password as well as QR login.

For QR login, staffs are expected to print out or send the generated QR login code as attached image for newly registered client, when client specifically requested for a QR login code, when client changes password for any type of existing account.

For client withdrawal, client with Savings account type are required to have at least RM100 and Current account type are required to have at least RM500. Negative withdrawal and deposal amounts are prohibited.

Clients are assumed only the privilege to edit personal contact information and cannot edit sensitive information like their Identification Number and Legal Name for security measures.

# References

Kargopolov, S., 2017. *Encrypt User Password Example in Java.* [Online]   
Available at: https://www.appsdeveloperblog.com/encrypt-user-password-example-java/

sarxos, 2018. *webcam-capture.* [Online]   
Available at: https://github.com/sarxos/webcam-capture

SINGH, R., 2020. *Write a QR Code Reader in Java using Zxing.* [Online]   
Available at: https://www.callicoder.com/qr-code-reader-scanner-in-java-using-zxing/

zxing, 2020. *zxing.* [Online]   
Available at: https://github.com/zxing/zxing

# Appendix

[Figure 1.1.a: Staff login page 4](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358062)

[Figure 1.2.a: Login status 5](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358063)

[Figure 1.2.b: Staff Home Page 5](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358064)

[Figure 1.3.a: Browse Client window, accessed by staff 6](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358065)

[Figure 1.4.a: Client Registration with invalid input, disabled “Create Account” button 7](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358066)

[Figure 1.4.b: Client Registration with valid input, enabled “Create Account” button 7](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358067)

[Figure 1.4.c: Successful QR login code generation and account creation. 8](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358068)

[Figure 1.4.d: Failed Account Creation: Duplicate Registered Client 8](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358069)

[Figure 1.4.a: Staff manually logging in client with client inputting their own credentials 9](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358070)

[Figure 1.5.b: Login successful notification 9](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358071)

[Figure 1.5.c: Login failed notification 9](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358072)

[Figure 1.5.a: Client Home 10](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358073)

[Figure 1.6.b: Client menu dropdown 10](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358074)

[Figure 1.6.a: Withdrawal & deposit window 11](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358075)

[Figure 1.7.b: Withdrawal error messages 11](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358076)

[Figure 1.7.a: QR image file for client with account no. 18 12](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358077)

[Figure 1.8.a: Edit menu dropdown 13](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358078)

[Figure 1.9.b: Client editor window 13](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358079)

[Figure 2.1.a: Collapsed view of ClientCRUD class 14](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358080)

[Figure 2.1.b: Overriding of mouseEntered() and mouseExited() methods in the MouseAdapter object, created from the addMouseListener() 15](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358081)

[Figure 2.2.a: Private declared variables 16](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358082)

[Figure 2.3.a: Collapsed view of ClientRegistration class 17](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358083)

[Figure 2.3.b: Methods in ClientCRUD class 18](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358084)

[Figure 2.4.a: Thread usage in QRWebcamReader class 19](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358085)

[Figure 2.5.a: Event handling for txtPass, a password field 20](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358086)

[Figure 2.6.a: Try/Catch statement in method deleteBalance() 21](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358087)

[Figure 3.1.a: PasswordUtils Class 22](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358088)

[Figure 3.2.a: Code Snippet for generating the QR login code for clients 23](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358089)

[Figure 3.2.b: Code Snippet for Webcam QR Reader for Client QR login 24](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358090)

[Figure 3.2.d: example of client logging in with his QR login code on his phone 25](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358091)

[Figure 3.2.c: Generated QR login code for client 1 25](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358092)

[Figure 3.3.a: Client Editor opened for Client: Jolly Man 26](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358093)

[Figure 3.3.b: Change Password Form 26](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358094)

[Figure 3.4.a: Transaction History of Client: Jolly Man 27](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358095)

[Figure 3.4.b: Code snippet for generating the transaction rows, Part 1 27](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358096)

[Figure 3.4.c: Code snippet for generating the transaction rows, Part 2 28](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358097)

[Figure 3.5.a: Browse Client window, accessed by staff 29](file:///C:\Users\AverageDude\Desktop\Drive\Diploma\SEM%205\(JP)%20Java%20Programming\documentation.docx#_Toc79358098)