

**NAME : RAHIL SHA**

**GRADE :11B**

**INFORMATION TECHNOLOGY**

**PAT: Encryption and Decryption**

**PHASE 1**

# TABLE OF CONTENTS

page:

Scenario and scope: 3-4

defining the task

User Requirements: 4-5

Navigation and Flow

Diagram: 6-7

## DATASTRUCTURES

Database design: 7-8

Text files an Arrays: 9

Design of GUI: 9-16

## Software tools

Data Input: 16-20

Data processing: 21-23

Data output: 24

# Scenario and scope – Define the task:

## Topic: Online Banking

With technological achievements ever advancing, becoming greater and greater it is no surprise that old reliable systems are getting replaced with new more advanced systems, with the introduction of the internet there are more opportunities to do things like shopping or banking without having to leave the comfort of your own house.

The problem with these ever-advancing systems is that they bring about new sets of issues and problems never seen before, an example of one of these systems is online banking.

## The Problem:

The main issue of online banking comes down to human error. Many older people or people who are not as exposed to technology find these online systems too complicated to use, but hard to use interfaces aren't the only concern in terms of online banking. There is also a concern of data theft and people hacking into your account to gain your financial details. This is made very clear and abundant in people not as exposed to technology as they are more susceptible to types of social engineering such as phishing because they were never exposed to these things before and aren't as educated or experienced on how to handle it.

## Solution and Scope:

I will have to create a banking app with a simple and easy to use interface to cater to people not as experienced in online banking and ensure even the newest of people to online banking can have an enjoyable experience. In case of a security breach to maintain the security of a client's financial information I will have to implicate a two-step verification system that if a user's login details are acquired by someone else the user's data will be encrypted unless a third login requirement is met, an example of what this

third login requirement could be is a security question. A piece of information that is different for everyone and doesn't have to fit in as a normal login requirement.

i.e. 'what's your mother's maiden name" or "Name of your high school"

If this third requirement is met then the user's data will be decrypted and the user will be free to check their financial information and edit their finances. There will also be certain types of users such as admins who will only be able to check the information of everyone on the system and not edit.

## User requirements

There are two types of users that will use this software:

---

=====

☺ Clients- people who create and use accounts

---

=====

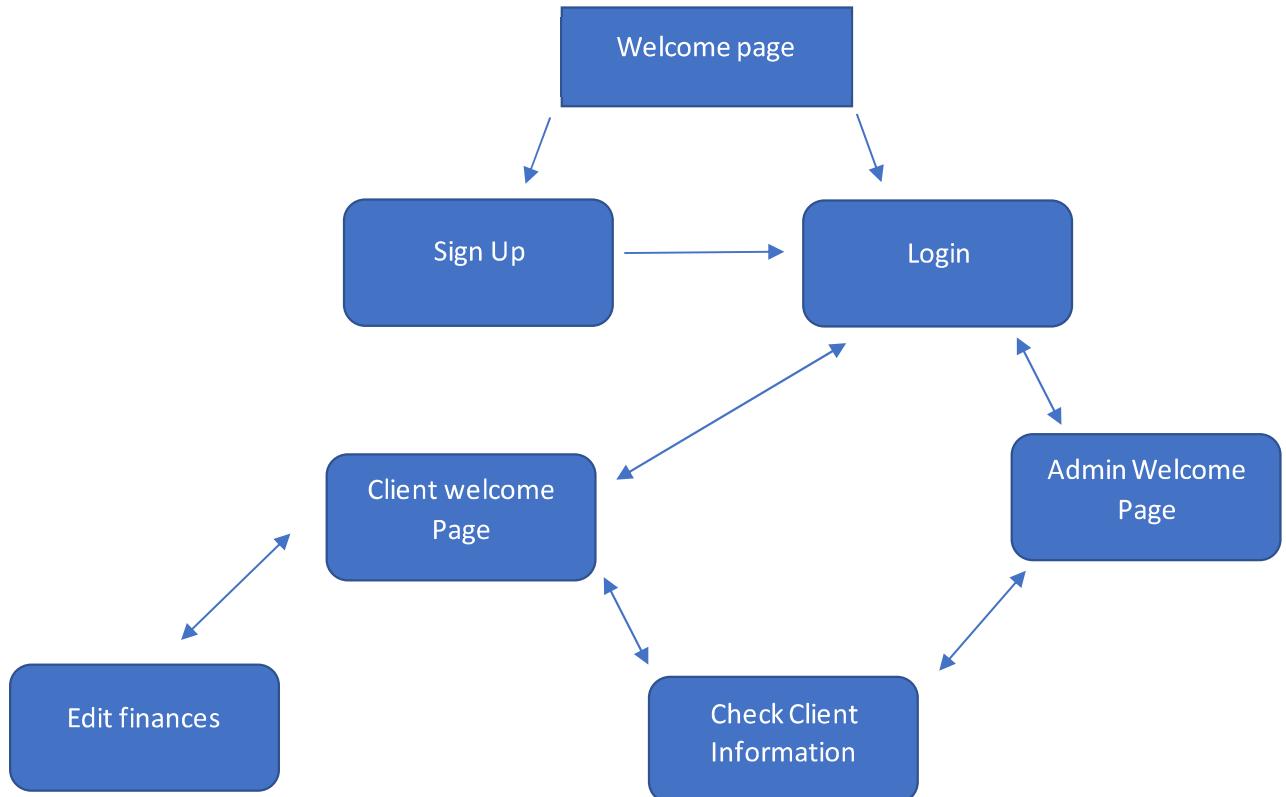
☺ Admins- people who work for the banking company and can monitor and view client's accounts

---

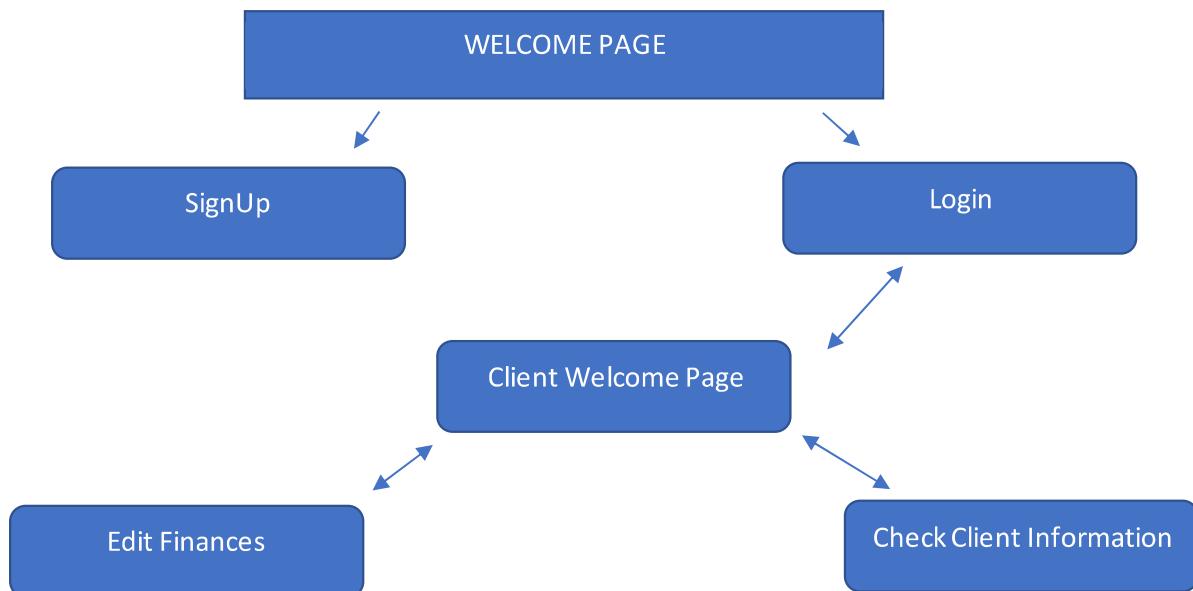
<u>USERS</u>	<u>ACTIVITIES</u>	<u>LIMITATIONS</u>
<u>CLIENTS/ ACCOUNT HOLDERS</u>	Clients will be able to create their own account and are able to change certain information about their account at anytime. clients will be able to	Clients will only be able to access information of their own account, there will be certain information that once users input that will not able to be changed.

	<b>add or withdraw finances from their account at will, and Accounts can be deactivated by contacting the bank.</b>	<b>Depending on the age of the client there will be a limitation on the amount of money you can withdraw at a time.</b>
<b><u>ADMIN</u></b>	<b>Admins will be able to access the data of others and look at their details using an admin pin. Admins will be able to delete users account at will.</b>	<b>Admins will only be able to view other peoples information. They will not be able to alter information of users in anyway and will not have access to extra confidential information such as :</b> <ul style="list-style-type: none"> <li>• <b>Passwords</b></li> <li>• <b>Answers to security questions</b></li> </ul>

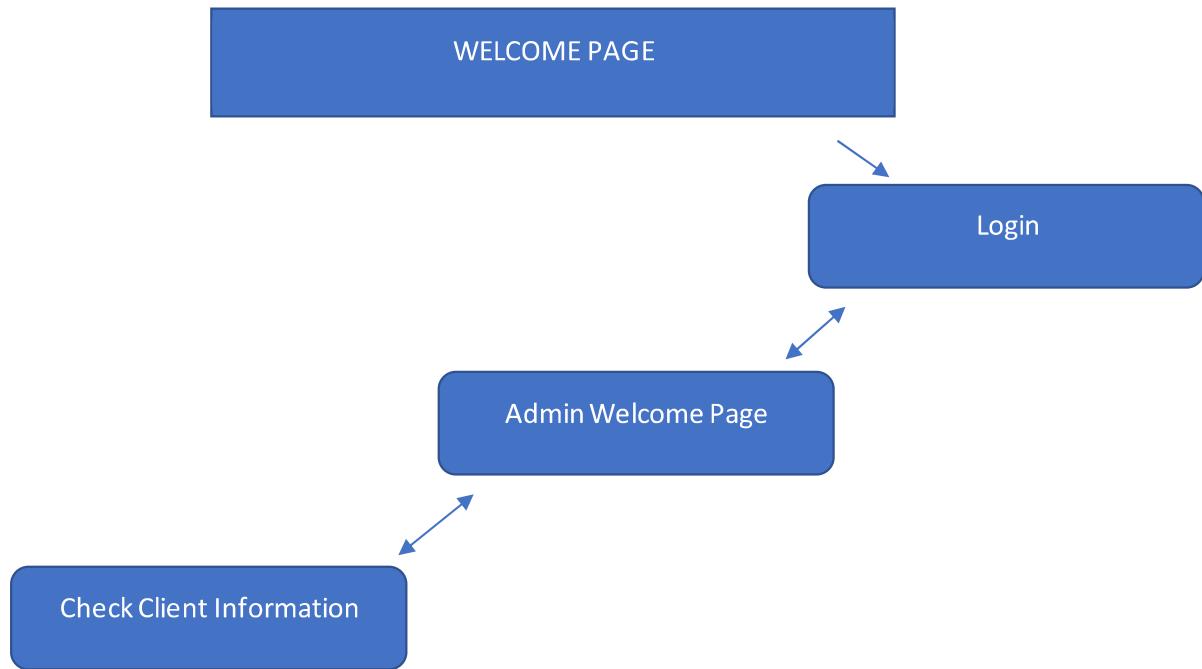
# Navigation and flow diagram



## Client's view:



## Admins view:



## DATA STRUCTURES:

### DATABASE DESIGN

#### TblLoginDetails

Field Name	Data Type	
AccNum	Number	account position in system
FirstName	Short Text	field size: 15
Surname	Short Text	field size: 15
UserName	Short Text	field size: 20
Password	Short Text	field size: 20
DateOfBirth	Date/Time	age of user
Admin	Yes/No	Only me

This table contains all the login information of the users and admins on the system. It stores the:

- Position of the account in the system
- Name of user or admin
- Age of user or admin
- Passwords
- Status of user whether they are a client or admin

Information such as usernames and passwords will be able to be changed. Admins will be able to delete users from the system.

## **tblFinancialDetails**

Field Name	Data Type	Description
AccNum	Number	Position of account in system
AvailableBalance	Currency	Amount of money in account
Deposit	Currency	Last amount deposited in account
DateOfLastDeposit	Date/Time	Date of last deposit
Withdrawal	Currency	Last amount withdrawal from account
DateOfLastWithdrawal	Date/Time	Date of last withdrawal

This table contains all the financial information of the clients and admins on the system. It stores the:

- Position of the account in the system
- Balance of account holder
- Amount last deposited into account
- Amount last withdrawal from account
- Date of last deposit
- Date of last withdrawal

Details such as Available balance, Deposit, Withdrawal, date of last deposit and date of last withdrawal change every time money is added or taken out of account.

\*Depending on age, the amount that can be withdrawal at one time is restricted

# Text files and Arrays:

## Text Files:

My program will use text files in multiple areas to help give and gain information to and from users. Examples of these areas are:

- On the welcome page there will be an area that you can click on to view the company's terms and conditions and before a user can create an account they are notified to go over the terms. It is made this way cause terms and conditions can be very long and I aim to make a neat and compact program .
- On the welcome page there will be an area that if the user has come across any problems with the program. They can send their feedback and this information will be saved on a text file to view for later.
- It will also be used to keep second verification information

## Array(s):

At this point in time there is only one use for an array in my program and it is:

- In the signup screen when a user is entering their date of birth it helps with the storing of the data and validating that the date of birth entered is actually a realistic one.

# Design of GUI

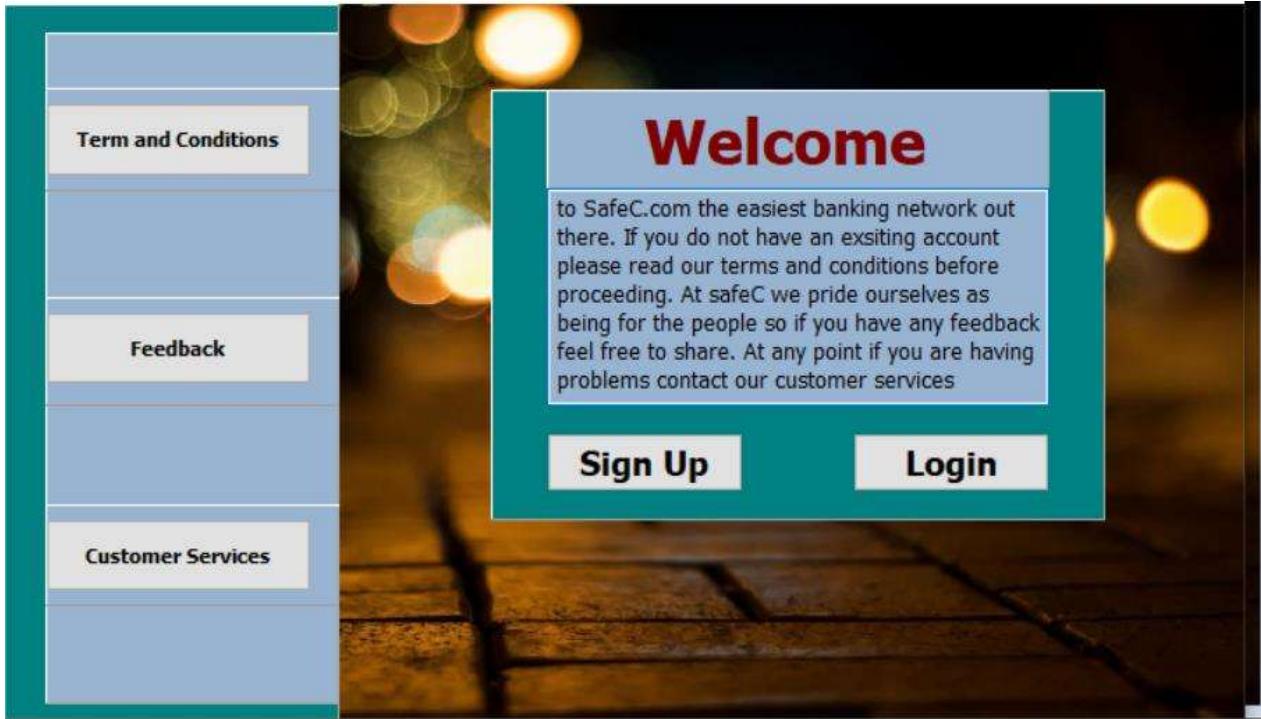
**please note: all GUI shown is subject to change**

I have to keep in mind that not every single user has the same level of experience with technology, so to ensure that every user has an enjoyable experience.

The screenshots provided are rough prototypes of the main forms of the program. These screenshots will include all components used and explain the functions of all the important components.

**\*Component's uses that are not specified are for aesthetic only**

## Welcome Screen



This is the first page that users will see. It is the page that contains all the important information about the program (e.g. Contact information and Terms and conditions). This is where users can sign up or if they already have an existing account, login. Any problems experienced by users can be shared from here through the feedback button. The GUI consists of:

- 5 buttons
- 1 Memo
- 1 label
- 1 image component
- 7 panels

Component	Function
btnSignUp	Takes user to sign up page

<b>btnLogin</b>	Takes user to Login page
<b>ImgBackground</b>	Image for background
<b>btnTerms</b>	Shows User terms and conditions
<b>btnFeedback</b>	Allows user to give feedback
<b>BtnCustomerService</b>	Shows contact details
<b>Memo</b>	Welcomes user and provides introduction
<b>panels</b>	For aesthetic only

## Sign Up page

The screenshot shows a 'Sign Up' dialog box. The title is 'Sign Up'. A red warning message at the top states: '\*You may not proceed if all fields are not filled in\*'. The form contains four input fields: 'First Name' and 'Username' in the top row, and 'Surname' and 'Password' in the bottom row. Below these is a 'Date of Birth' field with placeholder text '(eg 1 september 2004)'. At the bottom are 'Sign Up' and 'Cancel' buttons.

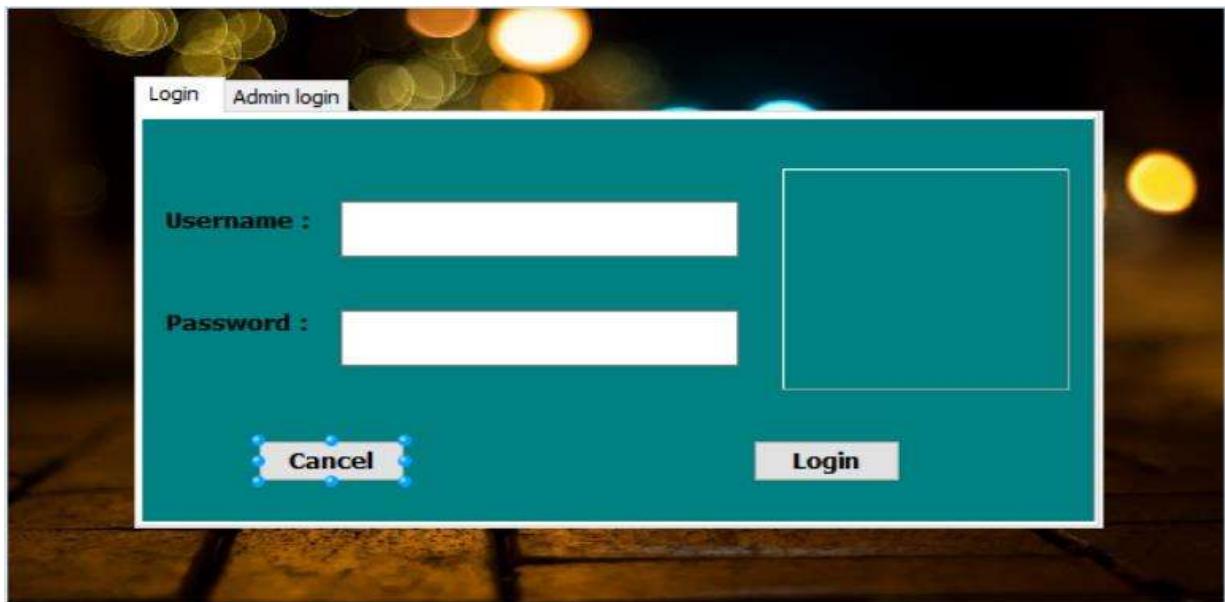
This is the page users go to if they do not have an existing account. Users will have to enter their information and if all fields are not filled they will not be able to proceed. After sign up is complete users will be asked a security question. The GUI consists of:

- 2 buttons
- 6 labels
- 1 image component
- 5 edit boxes
- 2 panels

Component	Funtion
-----------	---------

lblPassword	Indicates where to enter password
lblUsername	Indicates where to enter username
lblDOB	Indicates where to enter Date of birth
lblSurname	Indicates where to enter surname
lblName	Indicates where to enter name
btnCancel	Takes user back to welcome screen
btnSignUp	Creates account for user
edtName	Extracts user's name
edtSurname	Extracts user's surname
edtDOB	Extracts user's date of birth
edtUsername	Extracts user's username
edtPassword	Extracts user's password

## Login page



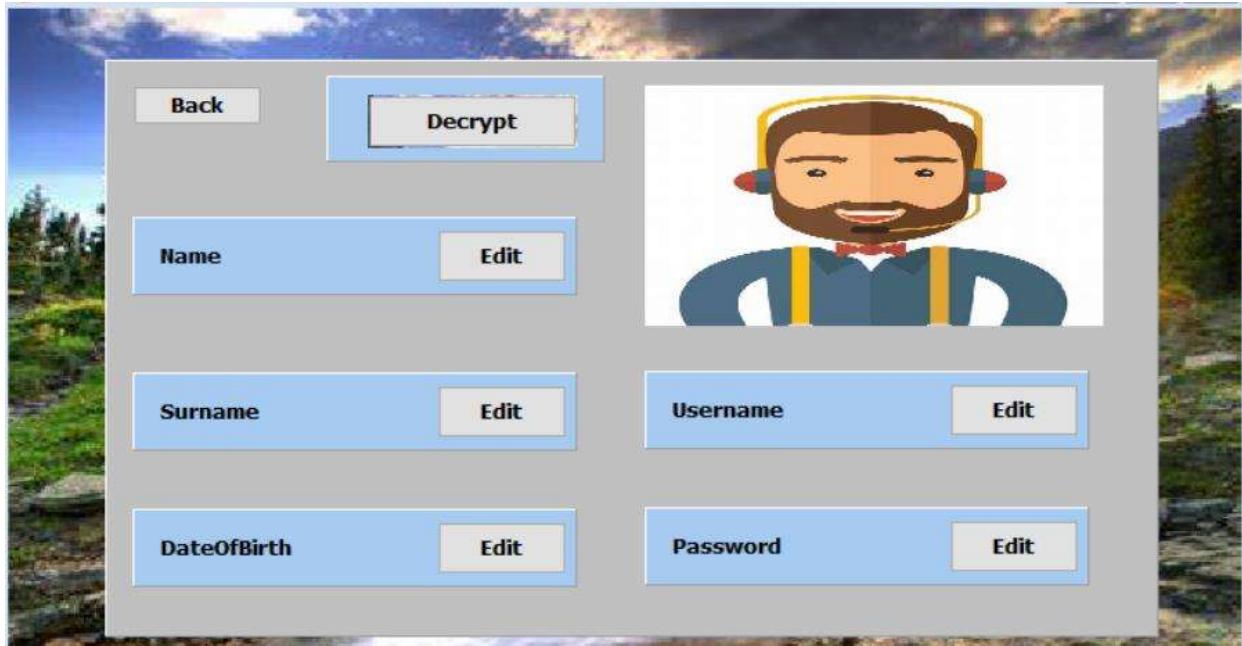
After successfully signing up the user is sent to the login screen or users who already had existing accounts can access the login screen from the welcome screen. This page contains two login areas one for clients and one for admins. After logging users can access the rest of the program. The GUI consists of:

- Page control (2 tabs)
- 4 buttons
- 4 panels

- 4 edit boxes
- 4 labels
- 1 image component

Components	Function
btnLogin	Takes user/admin to user/admin welcome page
btnCancel	Takes user/admin back to welcome page
lblUsername	Indicates where to enter username
lblPassword	Indicates where to enter password
pnlDisplay	Displays if information is wrong
edtUsername	Extracts username
edtPassword	Extracts password

## Client info page



This is where users can view and edit their personal information. Before the information can be viewed or edited it has to be decrypted. The GUI consists of :

- 1 image component

- 7 buttons
- 7 panels
- 5 labels

Component	Function
lblDOB	Changes to User's date of birth
btnDOB	Edits user's date of birth
btnName	Edits user's name
btnSurname	Edits user's surname
btnUsername	Edits user's username
btnPassword	Edits user's password
btnDecrypt	Decrypts users information
btnBack	Takes user back to client welcome screen
lblName	Changes to User's name
lblSurname	Changes to User's surname
lblUsername	Changes to User's username
lblpassword	Changes to User's password

## Finances page



From this page users will be able to view their finances. After they decrypt their information by entering a pin they will be able to view their available balance, amount of last deposit, amount of last withdrawal, Date of last deposit and date of last withdrawal. They will also be able to add and take out funds from their account at will although depending on age of the user there might be a restriction on the amount they can take out at one. The GUI consists of:

- 10 labels
- 4 buttons
- 2 image components
- 2 panels

Component	Function
btnDecrypt	Decrypts users information
btnWithdrawal	Subtracts funds from users account
btnDeposit	Adds funds to users account
BtnClose	Takes user back to client welcome page
IblDOLW	Changes to user's date of last withdrawal

<b>lblDOLD</b>	<b>Changes to user's date of last deposit</b>
<b>lblWithdrawal</b>	<b>Changes to user's amount of last withdrawal</b>
<b>lblDeposit</b>	<b>Changes to user's amount of last deposit</b>
<b>lblAvailableBalance</b>	<b>Changes to user's amount of money in account</b>

## **Software Tools –IPO:**

### **INPUT:**

#### **tblLoginDetails**

Input	variable	Data Type	Format	Source	component	Validation	Error Message
First Name	sFName	string	Letters (Aa-Zz)	Keyboard	EditBox	Checks name consists of letters only.	Please type name in correct format
Surname	sSurname	string	Letters (Aa-Zz)	Keyboard	EditBox	Checks Surname consists of letters only and is not the same as the first name	Please type Surname in correct format or First name and surname are the same
Username	sUsername	string	Letters (Aa-Zz) And	Keyboard	EditBox	Checks if username matches any on	Login: username or password

			Alphanumeric			System for login and sign up	d is incorrect Sign Up: Username already exists
Password	sPassword	string	Letters (Aa-Zz) And Alphanumeric	Keyboard	EditBox	Checks if password matches any on system	Password or username is incorrect
Date of birth	sDOB	string	Letters (Aa-Zz) And Alphanumeric	Keyboard	EditBox	Checks if sDOB is entered in correct format and the date is a plausable one	Please enter correct date of birth in correct format

## tblFinancialDetails

Input	Variable	Data Type	Format	Source	Component	Validation	Error Message
Date of last withdrawal	sDOLW	string	Letters (Aa-Zz) And Alphanumeric	Keyboard	InputBox	Checks if sDOLW is entered in correct format and the date is a plausible one	Please enter correct date in correct form
Date of last Deposit	sDOLD	string	Letters (Aa-Zz) And Alphanumeric	Keyboard	InputBox	Checks if sDOLD is entered in correct format	Please enter correct date in correct form

						and the date is a plausible one	
Deposit	rDeposit	Real	Currency	keyboard	InputBox	Deposit must be a number	Invalid Deposit
Withdrawal	rWithdrawal	Real	Currency	keyboard	InputBox	Withdrawal must be a number and you can't withdraw more than you have in your balance	Invalid Withdrawal

## Data Validation

### Types of data validated:

- String
- Boolean
- Real

Minimum of at least FOUR instances of data validation in my program:

1. Username and password
2. Details of user
3. Status of user
4. Withdrawal and deposit

## **1.Username and password (string)**

### **Sign Up:**

When a username is entered in the sign-up page the program will search through the database to check if that username is already in use, if that username IS in use the user is notified to pick another one, otherwise they may not proceed

### **Associated error message(s):**

"This user name is already in use please pick another one"

### **Login:**

When logging in, the user has to enter their username and password. The program searches through the database for the username and password entered in. If the username AND password is not found, the user is notified that the account they are looking for does not exist. If the username entered is found but the password entered is not or vice versa the user is notified they entered the wrong details

### **Associated error message(s):**

"The account you are looking for does not exist"

"The username or password is incorrect"

## **2.Details of user (string)**

When signing up the user has to enter details such as Name, Surname and date of birth. Each of these fields has individual validation requirements as follows:

### **Common for all- if field is left blank an error message is shown**

- Name- Program checks if the name entered contains any numbers or symbols if it does error message is shown
- Surname- Program checks if the Surname entered contains any numbers or symbols or if it is identical to the first name if so error message is shown

- **Date of Birth-** Program has to first check if the date entered is in the correct format an example is shown on the sign-up page so users know the correct format if it is not in the correct format an error is shown. If the format is correct the program still has to check is the date they entered is a plausible one if it is they may proceed is it is not an error is shown

**Associated Error message(s):**

"Please fill in all blank fields", "Name is invalid", "Surname is invalid", "Date of birth is in incorrect format"

**Status of user (Boolean)**

When logging in there are two tabs, one for clients and one for admins if the client tries to login in the admins tab the data base will have to go check if the username that was entered has admin privileges if it does not the user is noticed they logged in on the wrong tab vice versa

**Associated error message(s):**

"status does not correspond with login page"

**4. Withdrawal and deposit (real)**

When depositing or withdrawing into/from an account the program has to check that the amount entered didn't contain any letters or symbols an that the amount entered is rounded to two decimal places at most. When withdrawing the program also has to check whether the amount being withdrawal is more or less than the balance if it is less, they may not withdrawal

**Associated error message(s):**

"Invalid deposit", "Invalid Withdrawal"

# **DATA PROCESSING**

## **What needs to be processed:**

Minimum of 8 being listed:

### **1.Verifying usernames and passwords**

The program will have to go through that data base and check if BOTH the username and password entered are valid or real.

### **2.Encrypting and decrypting user data**

The program needs to be able to encrypt a user's data before it can be viewed and decrypt the data after entering a pin.

### **3.Searching for user's names**

Admins need to view the data of clients to help with any problems, to do this the program first needs to be able to search for a client's data.

### **4.Calculating Available Balance**

The program needs to able to calculate the available balance of a user's account after they deposit and withdrawal from their account.

### **5.Editing user details**

Users shouldn't be restricted personalisation of their account after they create it. Users are able to edit details about their account and this updates their information on the database.

## **6.Deleting of user accounts**

At any time if a user wants to terminate their account or their account has suspicious activity the admins need to be able to delete accounts.

## **7.Restricting access of users**

Users should not be able to enter their login information and get in admin welcome. The program will not allow this.

## **8. Determining age of user**

There are certain things that users under the age of 18 shouldn't be able to do. The program should be able to determine the age of a user from their date of birth.

# **HOW WILL THE PROCESSING BE DONE?**

At least 4 :

## **1.Validating usernames and passwords**

In order to validate usernames and passwords the program has to:

- ✓ Extract username and password from editboxes
- ✓ Search for username in database
- ✓ Compare the corresponding database password to the one entered in

## 2.Determine age of user

An example date of birth (1 September 2004). The program will have to:

- ✓ Check format and validity of date of birth
- ✓ Find the length of the date of birth to copy the last 4 characters
- ✓ Convert the last four characters into integer and minus it from current year

## 3.Calculate the available balance

To calculate the available balance. The program will have to:

- ✓ Find the users balance according the username
- ✓ Determine the amount a user wants to deposit
- ✓ Available balance + deposit = new available balance
- ✓ Determine the amount a user wants to withdrawal
- ✓ Available balance - withdrawal = new available balance

## 4.Encrypt and decrypt users data

This is the main purpose of the program to protect a user's data in the case that someone other than them or an admin is trying to gain unauthorised access of their data. The program will have to:

- ✓ Get the data from the database
- ✓ Convert that data to its ASCII values
- ✓ Add or subtract a set amount form the ASCII values
- ✓ And To decrypt data when a pin is entered by subtracting or adding the set amount to the ASCII values to get the original character

# **DATA OUTPUT**

## **tblLoginDetails**

<u>Output</u>	<u>Variable</u>	<u>Format</u>	<u>Source of Output</u>	<u>Component</u>
First Name	sFName	string	Database	EditBox
Surname	sSurname	string	Database	EditBox
Date of birth	sDOB	string	Database	EditBox
Username	sUsername	string	Database	EditBox
Password	sPassword	string	Database	EditBox

## **tblFinancialDetail**

<u>Output</u>	<u>Variable</u>	<u>Format</u>	<u>Source of Output</u>	<u>Component</u>
<u>Date of last withdrawal</u>	sDOLW	string	Database	EditBox
<u>Date of last Deposit</u>	sDOLD	string	Database	EditBox
<u>Deposit</u>	rDeposit	real	Database	EditBox
<u>Withdrawal</u>	rWithdrawal	real	Database	EditBox
<u>Available balance</u>	rAvailableBalance	real	Database	EditBox