PROJECT REPORT

THE CITY OF LIVERPOOL COLLEGE

01/05/2018

James Kendrick, Lewis Morris, Leandro Ruiz Boyle

Contents

Introduction

The motivation behind the Project

The most important thing in creating something new is finding a purpose for what is going to be made. In this case the main focus was in challenging ourselves in trying to build an application from scratch and be able to implement on our owns some skills that we have learned while coursing our module with The Liverpool City College.

We were also thrilled by the idea of implementing encryption as we learned in one of our modules. It was for both of us a very enjoyable topic from the course. We thought that by combining our knowledge learned so far we could be able to build a solution to a problem.

Structure

The report has been created following the bullets points that were handed out with the assessment brief. It is been built to provide guidance to the technical team that will be working with our application. For that purpose there will be some support information with the structure of the application, the flow diagrams and more. The diagrams were the technical team will use to know what they will be working with in order to be able to maintain the code.

Some design documents will be amended to this report. It is important to understand the algorithms and the processes that have been implemented to understand the application in a more visual way.

We have attached a basic testing plan and its results to prove that the application works correctly and has been tested against the main functionalities. One of the things that we will talk about in the improvements in testing.

Then after that we will not only share a personal evaluation from each team member, but also we will discuss future improvements that could be made to prepare our application to today's market standards.

Although it has not been asked for a user guide will be provided to help the users quickly learned to do basic tasks that the application offers.

Finally it provides all the application source code with its relevant comments to help the understanding of it.

Goals to achieve

As already described in the previous report we aim to provide a solution to a real problem. And we think that we can help out people that have plenty of passwords and can't remember their credentials. It is a real deal in terms of a business productivity when you there is staff losing hours a day trying to find the passwords or having to reset them. By creating a Password Locker we want to reduce the chances of being locked out from a service provider or having to reset their specific passwords because of not remembering them.

We want to be able to introduce more than one type of encryption process in our application. Also, to implement an advanced security encryption. The report per se is an opportunity to build a small project that could be potentially used as experience towards a recruiter in a job interview.

Moreover, we wanted to create two small applications that had to work together on the same database. The goal here is to be able to implement an interface customized to a specific role: users or administrators. We don't want users to be able to have full functionality, same as we do not want administrators to see sensible data.

Tools and material used

In order to develop our application we have used **NetBeans IDE** to help us out with the programming aspect of the project. It has been the tool that we have used the most. We have taken advantage of its debugger and the GUI builder to lighten the workload in code. Although the mayor complexity was built without any assistant.

To test the local server we have worked alongside with **XAMPP** to allocate the port 3306 to our local database.

MySQL Workbench has been or SGDB to connect to our database and create our database with our needs and manage all the data structures in the database. We have also highly used this application to check whether the application was contacting and working smoothly with the database. It has played a big role in the testing that we have done the application to run queries and verify the insertion of passwords were correct and the encryption has been made correctly.

We have used **Microsoft Publisher 2013** to create all diagrams in our report. It has been a very useful software application with a very easy way to use it towards the diagrams.

Free Vector Icons it is a website (https://www.flaticon.com/) that provides a wide range of artwork completely free. We have used some icons from this source because it did not demand any reference to the icons and they are free. Highly recommended for anyone that needs icons.

Microsoft PowerPoint 2013 is the tool we have chosen to build our presentation of the project. Following the bullet points on the assignment brief we are delivering a presentation document that fulfills all requirements requested, alongside with our short speech.

Microsoft Word 2013 has been the tool used to join all the contents made and edit them into a professional report. This is an essential piece of software that plays an important role in the whole project.

Technologies involved

JAVA is was selected to do this project because of various reasons. This is an object orientated language, which made the job much smaller to code and easier for the team skills set. The team has a better knowledge of JAVA so we thought it was the right choice. Java has great libraries for encryption and hashing and as we researched we found that it is one of the most common languages in cybersecurity.

SQL. Sequel Query Language has been the only language to build and set our small database. The language was easy to use and it covered the needs for our database.

Encryption – RSA. This is one of the easiest asymmetric types of encryption to implement. It works with two keys instead of one. One key, called public key, is used to build the encryption directly and a second key is built in relation to the encryption made. This will be used to decrypt the message in a decryption process. This algorithm creates the keys from a number that comes from the combination of two big random prime numbers. This makes it a hard computational task for any computer that tries to crack the algorithm it. To crack the security system a computer would need to answer the next question: "Which two random prime numbers have been multiplied to get this big number as a result?" And from this number the keys are generated. The keys should be stored to be able to decrypt the message. Currently RSA has not been proven to be vulnerable. So it has a high level of security. This technology has been used in the application to save and recover the users´ passwords.

A. Sánchez Campos, J. Montes Sánchez (2013)

Hashing Sha-512. Has been used to save the username and the main password of the accounts in the database without being able to transform the hash into the original message. SHA 512 creates an array of 128 hexadecimal characters. This is used in the application to find a match to the username input and password with the hashed data in the database.

Anyone that retrieves the database will not be able to query the passwords or usernames, because they will be all hashed. And the user's passwords will be encrypted.

A. Sánchez Campos, J. Montes Sánchez (2013)

Time Management

Produce a list of activities, identifying the estimated duration of each task and any preceding tasks

1. Assign Parts for the project (See Allocate tasks section)
2. Design documentation - 2 weeks (Finished by 1st May 2018)
3. Creation of the project - 12 weeks (Finished by 4th June 2018)
4. Time Management - 12 weeks (On going until everything is finished)
5. Test Plan - 1 day (Completed 14th May 2018)
6. Test Log - 1 day (Completed once the program is fully functional)
7. Documentation - 12 weeks (Completed by 4th June 2018)
8. Work placed into one single report - 1 day (Completed by 4th June 2018)

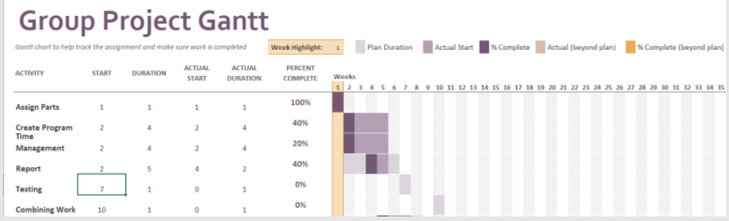
Produce network and Gantt charts for your project

For helping with time management we decided that it would be a good idea to use Gantt charts to help keep track of all our work, we initially set out to complete the project in 4 weeks, however due to the magnitude of the project and the fact that we had other projects to finish, this plan ended up being scrapped due to the fact that by the time we got to the original deadline we had set for ourselves we had only 40% of the program completed and 20% of the time management full designed.

Due to this oversight on our behalf wee extended the assignment by another 4 weeks (See amended Gantt), this gave us double the amount of time to complete the work, however we still fell short on what we wanted to do, because of this we made a group decision that we would extend the deadline of the project by another 4 weeks this would take us up till a week before the official deadline for the assignment.

After we decided to extend the deadline up to 12 weeks in total we manged to fully flesh out our assignment and make sure that everything was completed, we found that by doing so gave us plenty of time to complete everything that we needed, the project and the time management section took the whole 12 weeks assigned, however the report was completed on week 6 meaning that we had more hands on when it came to the time management section. The testing was completed on time without any issues as can be seen and that left us with the final task of combining the all the work into one final report to submit.

Original Gantt

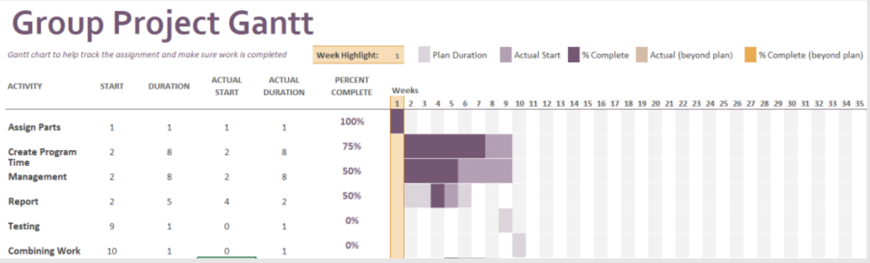


Group Project Gantt

Gantt chart to help track the assignment and make sure work is completed

ACTIVITY	START	DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Assign Parts	1	1	1	1	100%
Create Program Time	2	4	2	4	40%
Management	2	4	2	4	20%
Report	2	5	4	2	40%
Testing	7	1	0	1	0%
Combining Work	10	1	0	1	0%

Amended Gantt

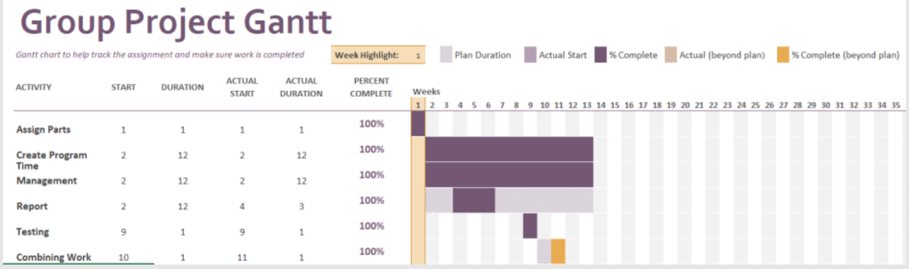


Group Project Gantt

Gantt chart to help track the assignment and make sure work is completed

ACTIVITY	START	DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Assign Parts	1	1	1	1	100%
Create Program Time	2	8	2	8	75%
Management	2	8	2	8	50%
Report	2	5	4	2	50%
Testing	9	1	0	1	0%
Combining Work	10	1	0	1	0%

Final Gantt



Group Project Gantt

Gantt chart to help track the assignment and make sure work is completed

ACTIVITY	START	DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Assign Parts	1	1	1	1	100%
Create Program Time	2	12	2	12	100%
Management	2	12	2	12	100%
Report	2	12	4	3	100%
Testing	9	1	9	1	100%
Combining Work	10	1	11	1	100%

Assign resources to each task

Due to the nature of this project we will be using different types of software and resources to help complete everything.

For the program itself we will be using NetBeans for the IDE, we believe this is a solid choice to use as it is easy to use and allows us to easily program using the JAVA language an added benefit to this is also that the software is free meaning we don't have to worry about paying for it.

We will also be using MYSQL Workbench for the program, this will be used for the database we found that using MYSQL Workbench was the best solution and easiest piece of software for us to gain what we wanted, the software just like NetBeans was also free meaning we are able to bring the cost of the development down again.

Finally, we used a piece of software called XAMPP, this was used to help connect to the 3306 ports in MYSQL, this software made it easier to get the database up and running and was again a free piece of software.

As for making sure we managed our time correctly we used a couple different pieces of software and online resources, firstly to track our tasks we used an online resource called Trello, this allowed us to manage all our tasks simply and make sure that we were completing everything by the deadline, because of this we were able to easily manage everything at once without any issue.

To make a Gantt chart we decided it would be best to make it using Microsoft Excel this is because the way the application is designed allowed for an easy to use Gantt chart that was very simple to read.

Allocate tasks to individual team members if not already done.

This project will be split into two different assignments, each person will have their own sections to complete by the deadline date this means that everyone must make sure that they are managing their time correctly and thus meaning that they don't miss a deadline as this can and will impact the other members of the group.

For the first assignment we were tasked with creating a proposal for the company that we will be making our project for. This meant that we had to assign each member a section of the proposal to complete, before we added them together for a single report, to do this we made sure that all the work was split evenly.

Lewis – Assigned to complete:

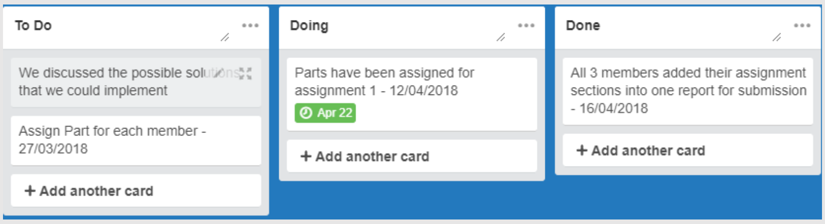
- Identifying the problems and requirements highlighted by the staff within the chosen company
- Provide more than one possible solution
- Provide a recommended solution

James – Assigned to complete:

- Document clearly the terms of reference
- Provide a summary of the findings and recommendations
- Document briefly the current operation of the chosen company

Leandro – Assigned to complete:

- Justify the recommendation of the solution
- Describe the opportunities which the proposed system offers the chosen company
- Provide an outline cost benefit analysis for the recommended solution



To Do

We discussed the possible solutions that we could implement

Assign Part for each member - 27/03/2018

Doing

Parts have been assigned for assignment 1 - 12/04/2018

Apr 22

Done

All 3 members added their assignment sections into one report for submission - 16/04/2018

Once we had finished the work individually we then combined our work into one report and then went over it and made sure that where was no mistakes and that the layout was acceptable before we finally submitted it.

For the 2nd phase we started looking into creating the application for the Hospital, this meant that we had to start creating the application, we had already decided that we would be making the project using the language JAVA, and to implement this we decided to use NetBeans as our IDE, to create the database we decided it would be best to use MYSQL Workbench as this offered all the functions that we required.

For this section of the project we split the work differently instead of assigning the work based on sections we decided to give it out based on what people were comfortable with, this was for both making it easier for the group but also because it meant people could work on something they are strong at.

Leandro – Assigned to complete:

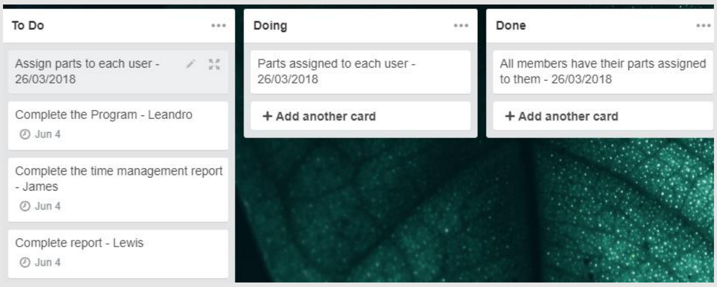
- Implement the solution using NetBeans (Created the program)
- Created the designs for the program

James – Assigned to complete:

- Identify the project deliverables and customer acceptance criteria
- Produce a list of activities, identifying the estimated duration of each task and any preceding tasks
- Produce network and Gantt charts for your project
- Assign resources to each task
- Allocate tasks to individual team members if not already done

Lewis – Assigned to complete:

- Identify any risks to the project and evaluate those risks. For the most serious risks identify contingency plans should those risks occur
- Identify suitable stage boundaries and any criteria that must be met before that stage is considered complete. (This is a Quality issue).
- Produce a detailed plan of your project.
- Identify the project deliverables and customer acceptance criteria



To Do

Assign parts to each user - 26/03/2018

Complete the Program - Leandro
Jun 4

Complete the time management report - James
Jun 4

Complete report - Lewis
Jun 4

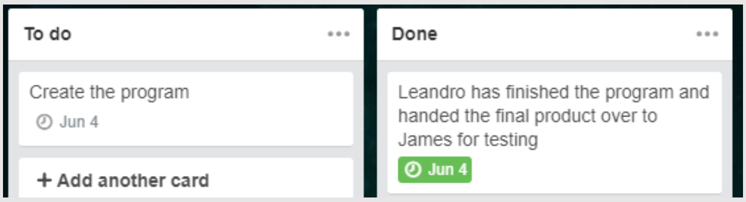
Doing

Parts assigned to each user - 26/03/2018

Done

All members have their parts assigned to them - 26/03/2018

Additional Trello Screenshots



To do

Create the program

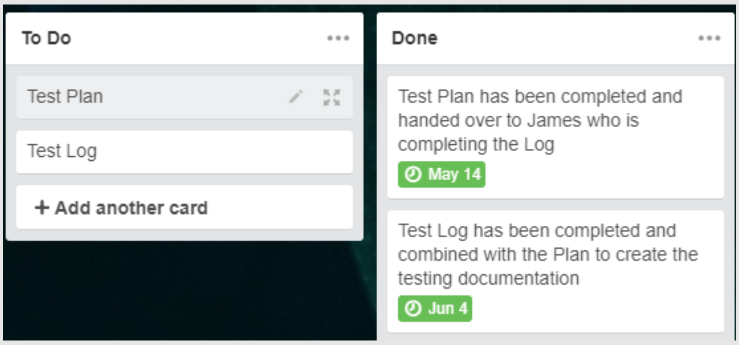
Jun 4

+ Add another card

Done

Leandro has finished the program and handed the final product over to James for testing

Jun 4



To Do

Test Plan

Test Log

+ Add another card

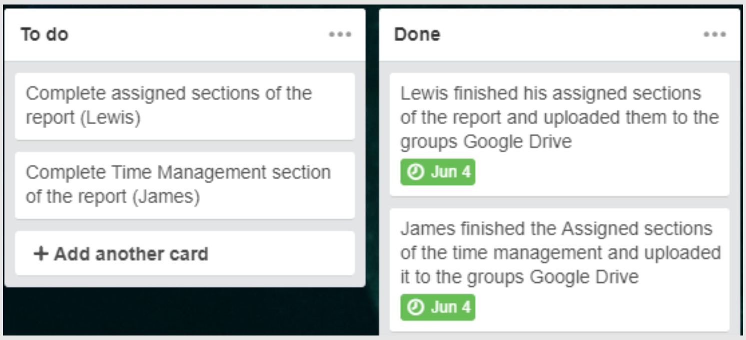
Done

Test Plan has been completed and handed over to James who is completing the Log

May 14

Test Log has been completed and combined with the Plan to create the testing documentation

Jun 4



To do

Complete assigned sections of the report (Lewis)

Complete Time Management section of the report (James)

+ Add another card

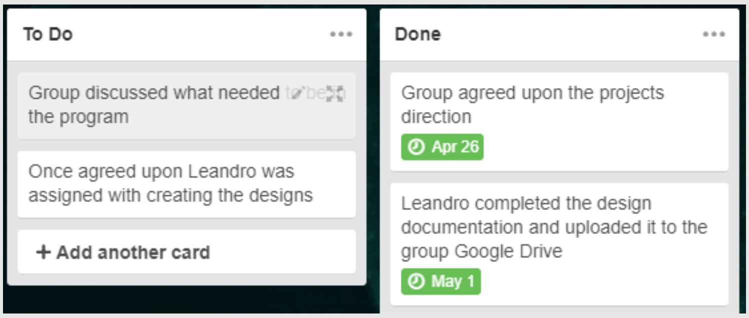
Done

Lewis finished his assigned sections of the report and uploaded them to the groups Google Drive

Jun 4

James finished the Assigned sections of the time management and uploaded it to the groups Google Drive

Jun 4



To Do

Group discussed what needed to be the program

Once agreed upon Leandro was assigned with creating the designs

+ Add another card

Done

Group agreed upon the projects direction

Apr 26

Leandro completed the design documentation and uploaded it to the group Google Drive

May 1

Identify any risks to the project and evaluate those risks. For the most serious risks identify contingency plans should those risks occur

It is important for any software to have protection over security risks. So, it is important to be aware of any risks that may happen to the software and how to deal with them.

First are physical risks. The physical risks for our software are relatively simple. If the user of our application writes down their password for their login it is a huge risk as it could be viewed by any staff member or anyone of the public depending on the sector and area of the user is currently located. It is strongly advised that the users do not write down their passwords and if found should be punished for creating security risks which may affect people using the software and protentional trailed in court for misconduct of mismanagement with sensitive information which could cause a breach within security. This should be aware with any use who access the software and should be reminded by their managers to keep the security as high as possible.

Another threat that could prone a risk are hackers. Hackers can try to access someone's account by guessing the password or by other means such as applications who try multiple password signatures with password methods. These types of hacks should be less of a risk compared to physical risks as the user's password is randomly created with numbers, symbols and letters which means that it is more secured compared to a password a normal employee would use which is usually a child's name, family dog or something simple and easy to remember.

The third type of risk is the software potentially breaking. This type of risk could be major if it affects multiple users as they have all the passwords required for other software stored into the application. This could cause huge downtime for the company while they fix it. It is possible for the company to call the team which will respond within the hour to fix any issues that may have occurred with the software.

Identify suitable stage boundaries and any criteria that must be met before that stage is considered complete

Storyboard, flowchart, screen designs, DB designs completed?

All project designs have been completed to start the project

Has a programming language been chosen?

We have chosen to use Java to complete our project

How many applications will be used to create the project?

We will create two applications for our project. First is the user application which will be used to log in and story passwords. Second will be used for the admin team at the company. This will manage user accounts and admin accounts. Users will be able to reset their login passwords too from the help of the admins.

Has the user application design been coded into java?

All the designs we created have been implemented into the user's side of the application.

Has the admin application design been coded into java?

All the designs we created have been implemented into the admins side of the application.

Can a user create a new record that will can be viewed and is stored inside the application to be viewed by only that user?

The record can successfully be seen inside the application for only the user that is logged in.

Can the new password record be modified and deleted if the user wishes?

The user can delete, modify any of the passwords records they add into the application.

Are the password records secure and encrypted?

All the records created by the user cannot be viewed outside the application by viewing a text file and is correctly encrypted.

Can a user be created through the admin application and be able to successfully log into the user application after creating the account?

The user can create an account through the admin application. This is the only way an account can be created and it requires a forename, surname and date of birth of the user. It also requires the user to create three security answers from security questions has to fill in to create the account. The account will then be created and the user will be given a randomly generated password which the user will use to log into the user application.

Can a user reset their login password for their account and be able to login with the newly created password?

In order to create a new password, the user will need to request for the admin to reset their account. The user will then have to correctly answer the security questions with the answers they used while creating the account. After this is completed they will be given a new randomly generated password.

Can an admin be created inside the admin application?

An admin can be created through the admin application.

Can an admin reset their current password through the admin application?

The admin can reset their current password through the admin application.

Do the applications work and running correctly as intended to mark the project as completed?

Both the user application and the admin application are running correctly as intended by our designs and ideas of the project.

Produce a detailed plan of your project

For our assignment my group and I have decided to do our project on encryption. We thought it was a great idea to do something unique, that none of the other groups would chose to do because of the difficultly level and because we wanted to do something special. We based our project on a business that would be able to use encryption effectively, and that would benefit them due to security issues companies can face.

After thinking about different types of encryptions we had the choice of making a program what would encrypt files and folders which would need a key to open. But we noticed Microsoft Windows already has a built-in system that would allow you to encrypt files and folders already.

So, we decided to chose our second idea. Our second idea is to create an encryption tool that could be used for any type of company that deal with large amounts of software on their computers. But for our current project, we decided to focus on a hospital setting business. Because of this we noticed that employees can have a difficult time remember all of their passwords for every software they need to log-in for. This would cause people to start writing their passwords down on pieces of paper which anyone inside the office could see. This can cause a major security floor for companies as anyone who walked into that office could see the passwords for software the employees used.

We noticed that this isn't a selective group of companies this could happen, this could affect every company. As people are liable to use unsafe passwords or passwords which would be difficult to remember, which would make them write down the password. This made us think that we should make a program that could combat this major security breach.

We decided on creating a unique encryption application that would allow the user to store their password all of their software into. This would then be protected by an encryption algorithm which would be impossible to decipher.

But while we were evaluating how to make the application (user application) we noticed that if we allow them to set up their own password for this software, they could potentially create one which is liable to be solved easily by black hat hackers. Because of this we decided to force passwords to be randomly generated by the application so the users are not allowed to create their own.

This means that there is a less chance for the password would be cracked as the random generator which mix symbols, numbers and letters together.

When the user signs up to use the software, they are required to fill in three security questions these questions will be directly personal so that only the user who is creating the account would know. We added these security questions so if someone was to try and reset the password for an account, they would need to know these three exact answers to the security questions otherwise the request would be declined. This is a way to combat security breaches that may happen to accounts if we didn't add these security questions as they are a second line of security for the users account.

The user who is requesting for a password reset would also require a signed form created by the manager of the department in order to review the legitimacy of the request as to identify the person who is requesting for a reset, in case any security breaches were to happen. These requests would then require the person to be present while the security questions are being answered and when the password is being reset as the password is randomly generated so the user will have to remember the newly created password.

Once the user has logged into the account they will be able to create, update and delete stored passwords as they wish. These passwords will require a username of the software, the password and a description of the software (for example, the name of the software) so that the user would not forget what the password is used for. These passwords will be encrypted into its own unique file which is not shared or accessed by anyone else meaning the file is completely secured.

Our group has also decided to create a second application (admin application) which is target towards to the security team within the company as the application will allow them to create the user accounts that will use the original software and any admin accounts that are required to access the second application. The second application will also manage and be the source to identify the users who are using the application to store their passwords. This includes the forename, surname and date of birth of the user. This information is only required for security reasons only and will not be used in any other means.

The admin application can also be used to create new admins which will be able to access and view all the current users that are registered to use the user application. An admin can also reset their own password through the admin application if needed.

Identify the project deliverables and customer acceptance criteria

Once we have finished our product we must make sure that it delivers the required services that the Hospital needs, we must make sure that the application is able to:

- Let people store passwords that are encrypted
- Have multiple users and admins
- An admin form that allows the creation and deletion of new admins and new users
- An admin form that allows the admin to reset and change passwords if needed
- A user form which allows them to browse the records

- A user should be able to Browse, add, modify and delete records as needed
- All records should be encrypted where needed

We are making this product for a Hospital and because of this we must make sure that the standard and quality of the final product is high because a Hospital holds a lot of sensitive information we need to make sure that the final product is working in every way that we have designed, because of this we have made sure that we go through everything at least twice before we are happy to submit it, this is because we need to make sure that the product that we make is not only acceptable to us, but is also acceptable to the customer as they are paying for this product in the end.

Improvements

The purpose of this work was to implement certain security techniques on the application developed. Due to this, the main improvement that we would advocate would be in the same line of security. So we would first seek to continue researching on what improvements in terms of security we could implement on the application that is already developed.

One of the proposals would be protection against SQL injections, but it may have been solved after encoding all the information that the user enters to gain access to the application. Although all sensitive information is encoded within the base, a little SQL injection could achieve.

When trying to request access to the application, it would also be a good practice to limit the number of attempts the user has to access their account. An example would be to allow 3 attempts and if the correct password has not been entered, then block the possibility of entering new passwords for the specific user for a limited time.

Block cells and functionalities to reduce human error of users. This is from blocking the JFrame so that its size is not distorted. The block of cells of the table is so that they can not be modified from the main page and can create a data entry error.

To put filters and to use tools of threads of characters to reduce the human error. Currently if the user typing a space by mistake will make the user fail to recognize that the user exists.

Finally and with regard to security. It would be fascinating to be able to create the functionality of connecting to a remote server and connecting it through a secure protocol such as SSL. To increase the security of the entire system.

On the other hand, one of the most important things is to create a better appearance and erase all the ballasts from the default values. The use of icons instead of text strings on buttons. Play with the colors and modify the default values. It must be known that people consume first by the eyes and then by other things. Therefore, the visual part and the level of usability must always be improved.

As for the interface it is preferable to have several warnings before making changes to the database or the application. Notices that clearly expose us in a text the consequences of the action that will be executed. As much as for warnings of possible irreversible changes as guides for the user to be informed.

To improve the user experience an EXE file (executable) should be created to avoid having the user look for the .JAR file in the project.

In addition, one of the improvements to the application would be to enter help information in the application in visual or text form to support the application. The content would be something of frequently asked questions with your answers or a usage guide.

Currently the application lacks functionalities such as being able to delete user accounts or update them in the administrator's application. That currently can only be created.

Finally, among the ideas that remained in the draft when choosing the project was to take advantage of the code of the application and move it to an Android environment using the memory of the phone as a memory source to store all the passwords. It would be an ideal version of working. An application for an individual and not for a system of an organization.

Conclusion

The main objective of the project was to offer a solution to the problem of the hospital where employees relied heavily on the technical team to reset each of the passwords they used because they were forgotten. Now with the given solution they will only have to remember a single password and they will be able to check the password of the systems in which they work.

With the solution we give, we meet the objective. We have tested it and it works. It has been a challenge to unite all the technologies together, but very satisfying for both to see it finished.

The most complex part during the development of the application was the part of the design, when having to pose for the first time a solution to a real problem and translate it into a scheme with which we can work. Due to inexperience, many aspects of design had to be modified when coding them. The contingencies generated more than one reset, having to change the structure of the database or parts of the application itself that did not work. Inexperience.

The other complex part was the implementation of Hashing SHA-512 and RSA. Both technologies were new and understanding their bookstores has taken us a little more time to understand their concepts. Especially with RSA encryption that we did not know that each encrypted data chain had its own key to be able to decipher later. This took us several days when it finally worked and we understood it.

We would like to be able to carry out a more exhaustive test in order to dismantle basic functionality faults and, on the other hand, to improve the security of the application. A battery of black box type tests. To focus on the logic failures resulting from interacting with the application.

Team evaluation

We have done the project with very good communication and understanding on the part of both team members. We have organized ourselves in a short time and there was great confidence on the part of the partner about the work of the other.

The time management has been good. Although not giving room for error we have been able to assign the available work hours with the points we had to develop on the project.

We are happy with the work done.

Self-Evaluation – James Kendrick

Overall during this project, I believe that I did well when it came to managing my time for the work, I believe that when it came keeping track of what the group was doing I did well as I knew what was needed when it came to creating a Gantt chart which meant I was able to make it fairly easily due to past experience making them.

However, I believe that there is room for improvement, this can be in form of communication with the rest of my team, I believe I could have done better when it comes to updating them on where I was up to whilst we were working on the project.

Self-evaluation Lewis Morris

I am overall impressed in how we completed this project. At first, I was worried that this project would be too difficult as encryption can be difficult to make it work correctly. But I was surprised by the fact we actually got the feature to work inside our program. The group worked well together and all had ideas to use in order to complete the project. As James and Leandro thought the idea to do a project about encryption (which I sadly missed as I wasn't in college for the discussion) while I thought of the idea to do an application about a password manager. I believe we completed the required targets with good communication with each other which allowed us to complete the project without any issues. I believe I could have communicated more when I came to the design as I had ideas on certain areas that I thought could have been an upgrade but, in the end, I was happy with the results of the project.

Self-Assesment Leandro Ruiz Boyle

The project was a great experience. I was able to push to the limit my time and abilities in learning new technologies and quickly addaptem in what I have been learning durong the course. I was lucky to have worked on a type of project that exited me.

I am happy with the results reached. The project meets the requierements and was delivered in time. The group worked well and everyone added their own strengths to the project and it was all combine well. Talking with other groups I realised we clearly did a good job of splitting the work to be able to perform a good amount of work.

Looking forward to improve this application and be able to present it as part of my portfolio in future job seeking.

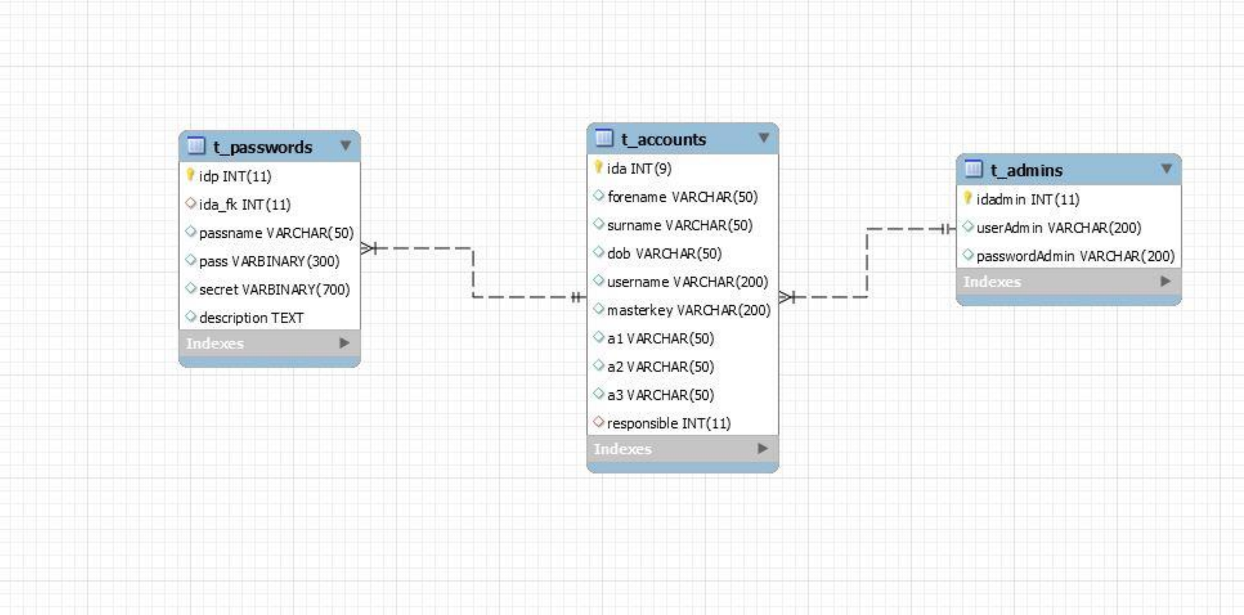
Reference

RSA Encryptation

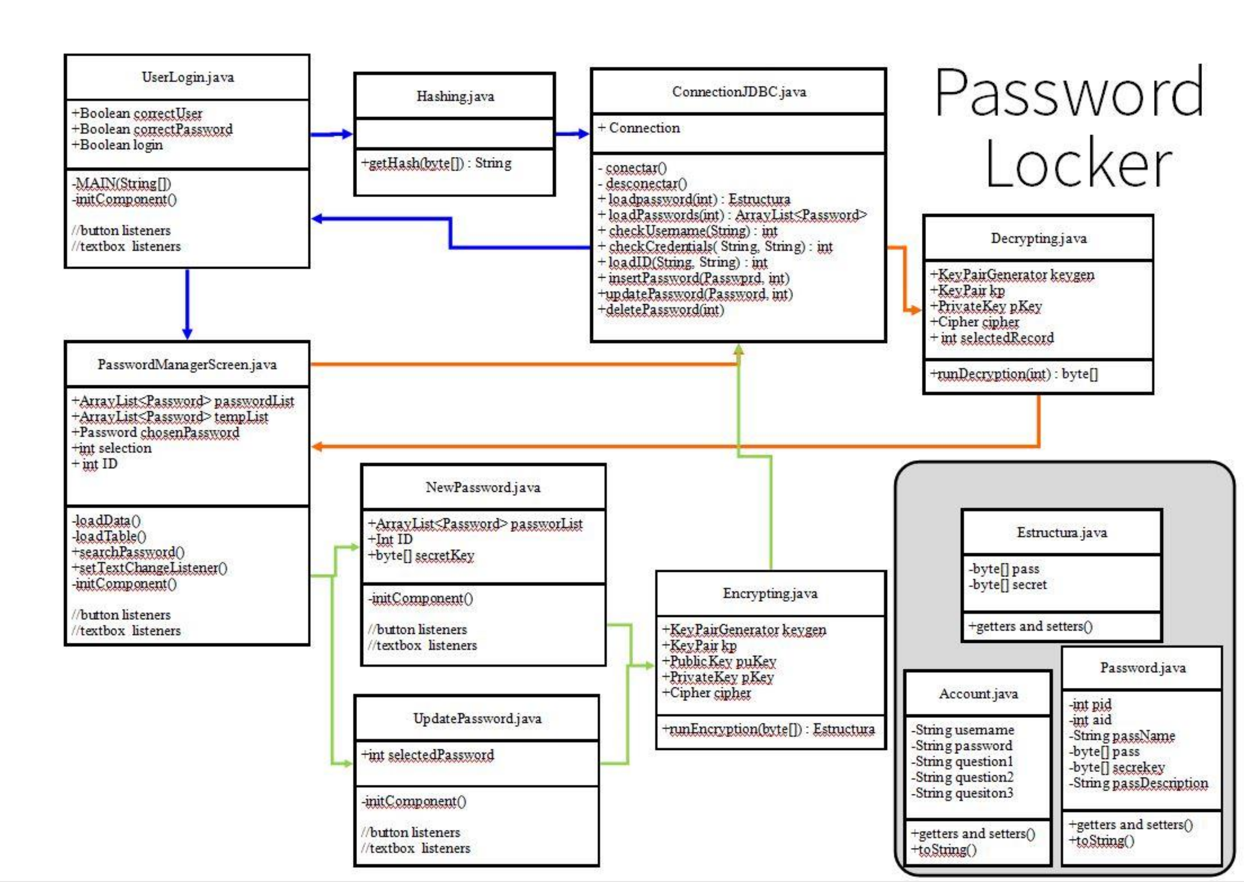
A. Sánchez Campos, J. Montes Sánchez (2013) Programación de Servicios y Procesos, RA-MA, (España, Madrid)

Appendix

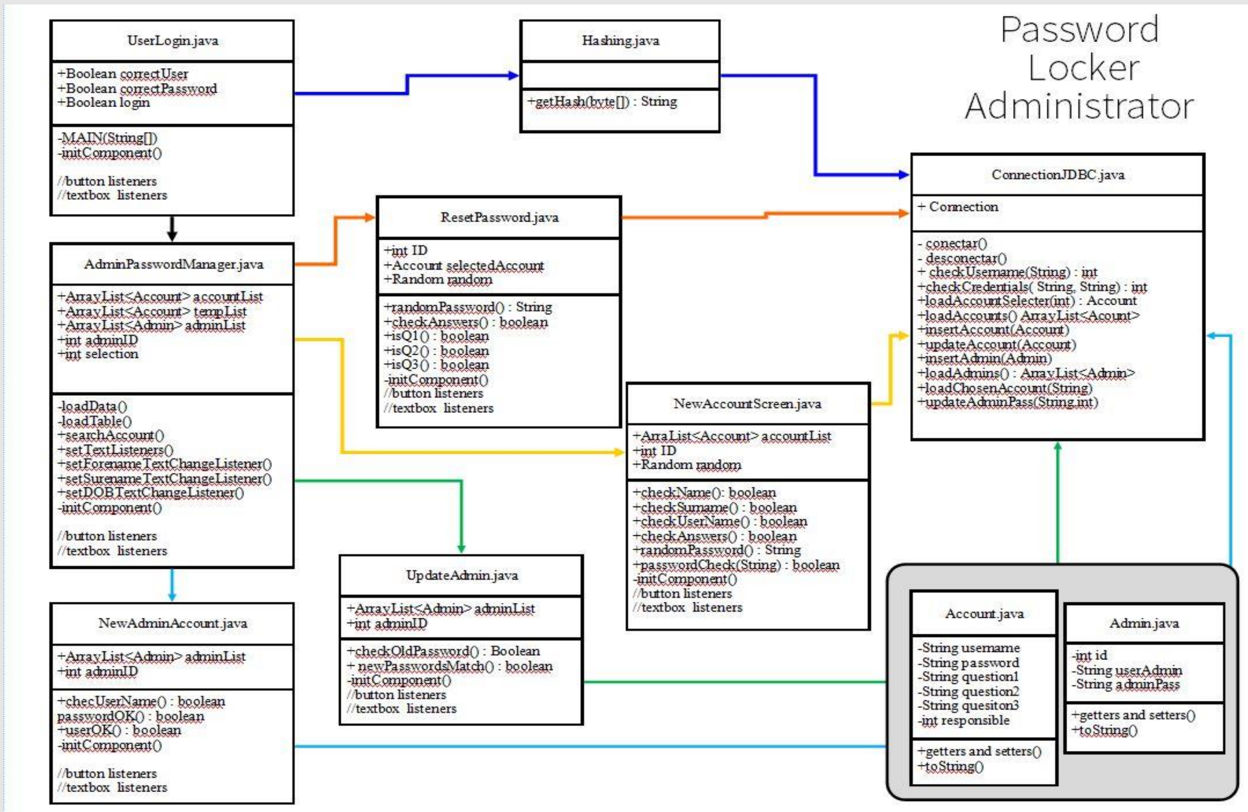
Database Diagram



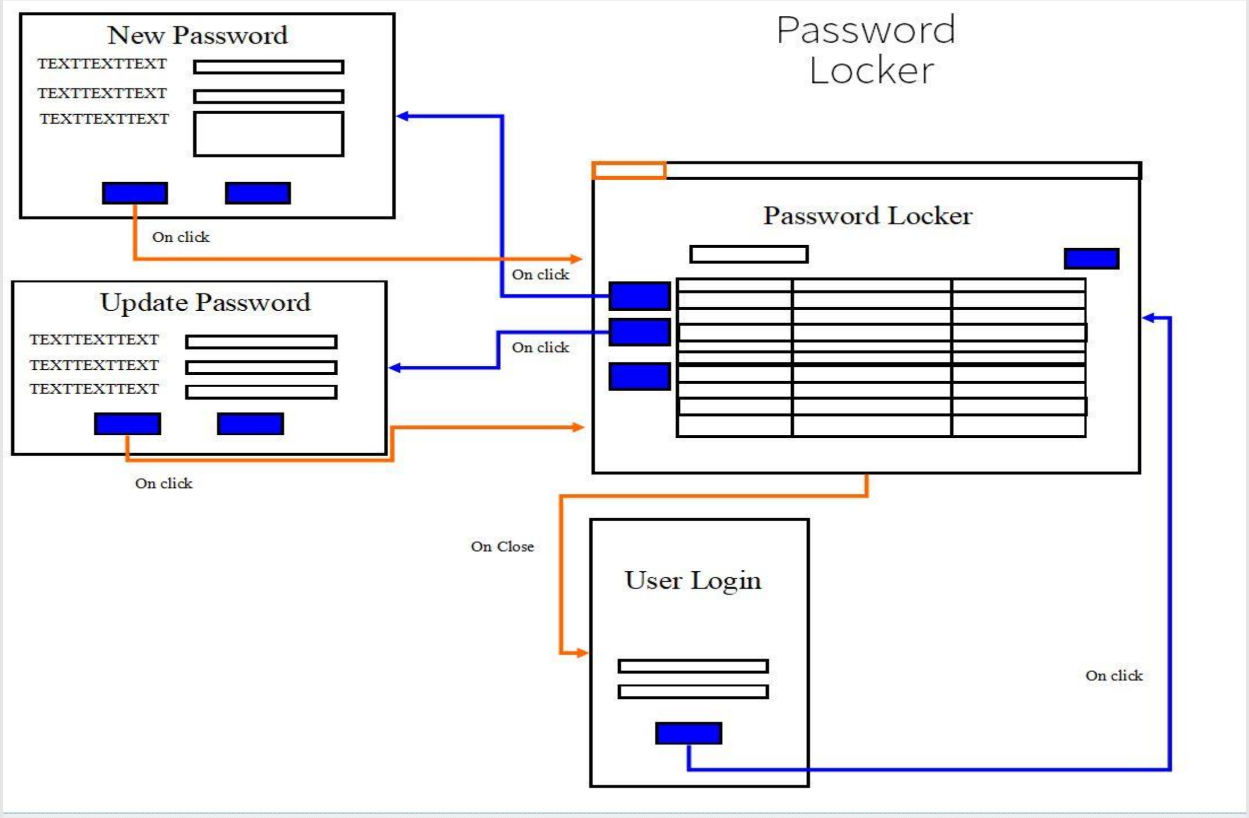
Password Locker Class Diagram



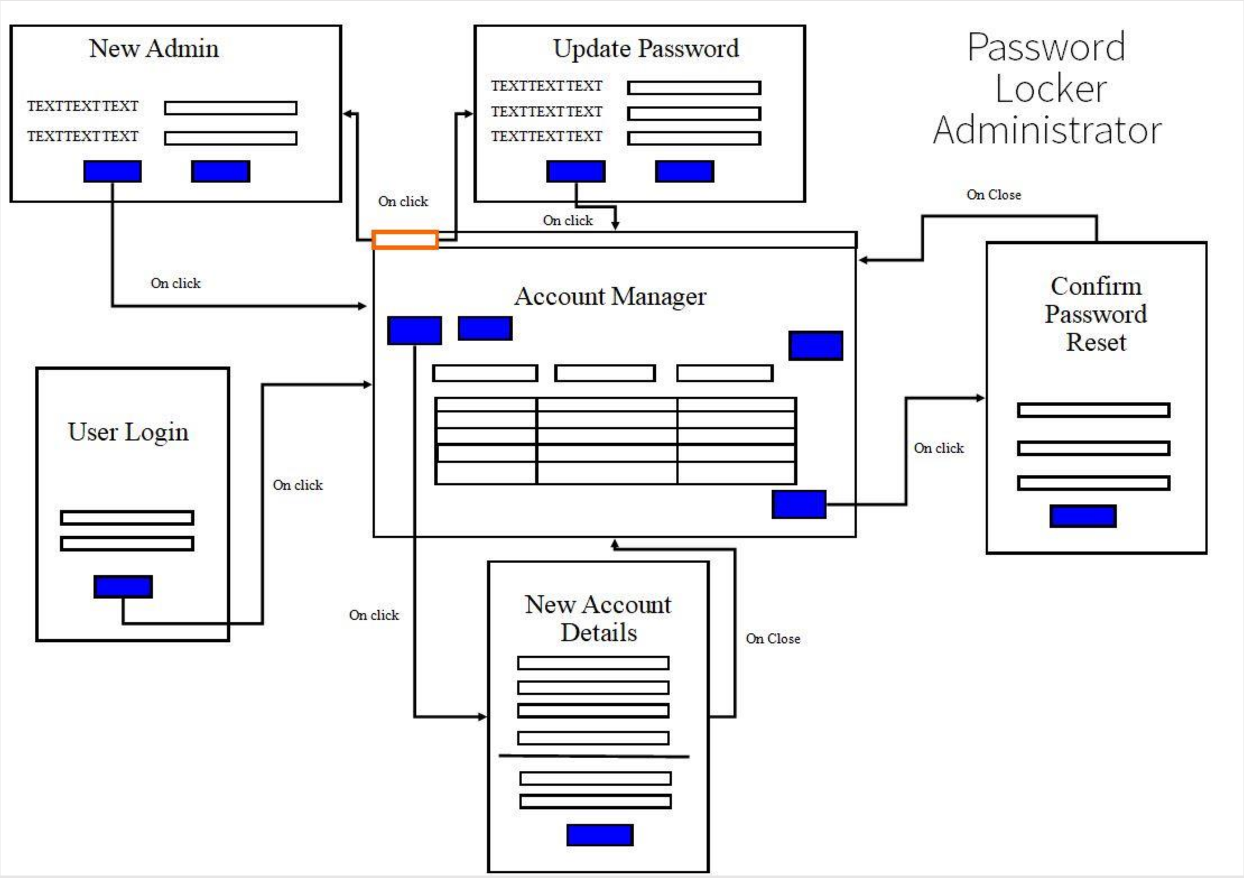
Password Locker Administrador Class Diagram



Password Locker Flow Diagram

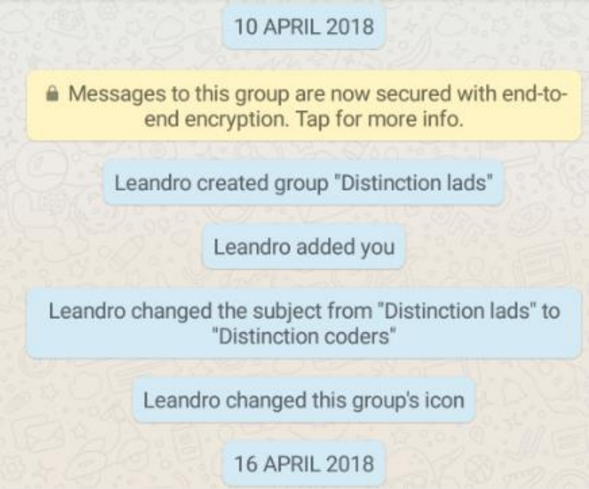


Pasword Locker Administrador Flow Diagram



Evidence of communication (Minutes)

As a group we decided the best way to keep in contact with each other would be by using a whatsapp group that we all had access to, this led to use creating a group where we could make sure we all knew what we where doing and when it had to be due in for. Due to the fact that we are only in college two days a week we did almost all of our communication through whatsapp. Below are some screenshots of the whatsapp group showing some of the communication between us:



10 APRIL 2018

🔒 Messages to this group are now secured with end-to-end encryption. Tap for more info.

Leandro created group "Distinction lads"

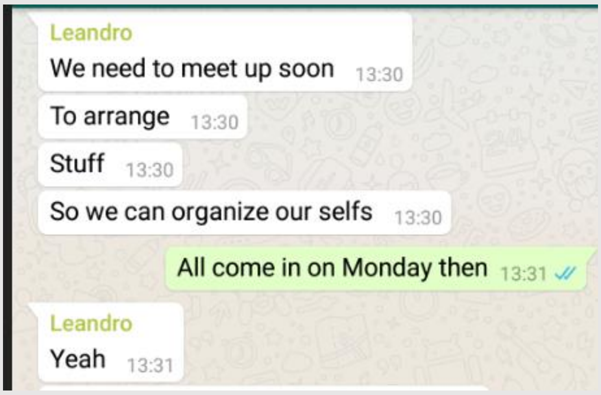
Leandro added you

Leandro changed the subject from "Distinction lads" to "Distinction coders"

Leandro changed this group's icon

16 APRIL 2018

Creation of the group



Leandro
We need to meet up soon 13:30

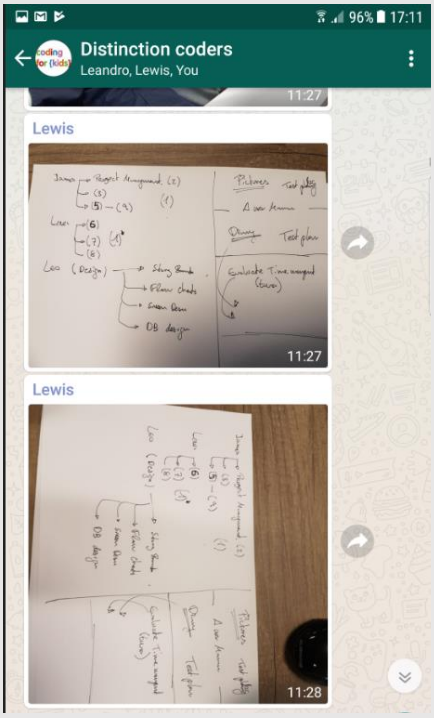
To arrange 13:30

Stuff 13:30

So we can organize our selfs 13:30

All come in on Monday then 13:31 ✓✓

Leandro
Yeah 13:31



Distinction coders
Leandro, Lewis, You

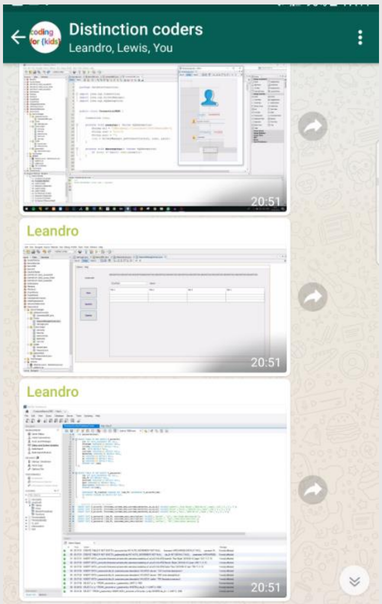
11:27

Lewis

11:27

Lewis

11:28



Distinction coders

Leandro, Lewis, You

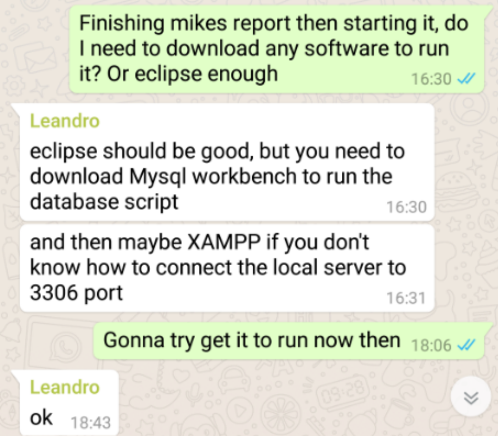
20:51

Leandro

20:51

Leandro

20:51



Finishing mikes report then starting it, do I need to download any software to run it? Or eclipse enough 16:30

Leandro

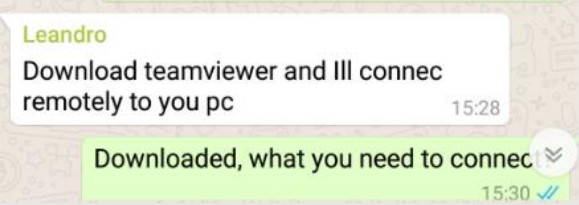
eclipse should be good, but you need to download Mysql workbench to run the database script 16:30

and then maybe XAMPP if you don't know how to connect the local server to 3306 port 16:31

Gonna try get it to run now then 18:06

Leandro

ok 18:43

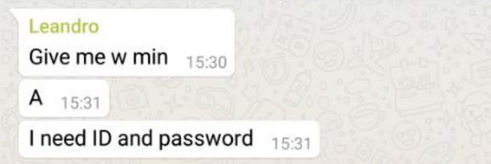


Leandro

Download teamviewer and Ill connec remotely to you pc 15:28

Downloaded, what you need to connec

15:30

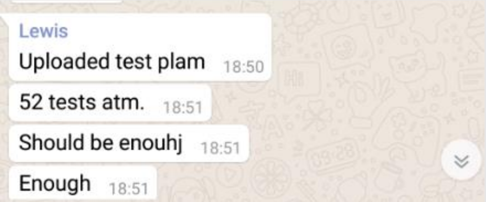


Leandro

Give me w min 15:30

A 15:31

I need ID and password 15:31



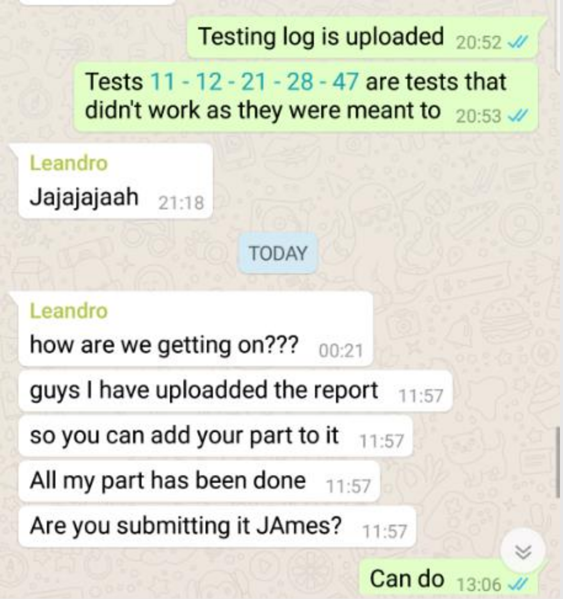
Lewis

Uploaded test plam 18:50

52 tests atm. 18:51

Should be enouhj 18:51

Enough 18:51



Testing log is uploaded 20:52

Tests 11 - 12 - 21 - 28 - 47 are tests that didn't work as they were meant to 20:53

Leandro
Jajajajaah 21:18

TODAY

Leandro
how are we getting on??? 00:21

guys I have uploadded the report 11:57

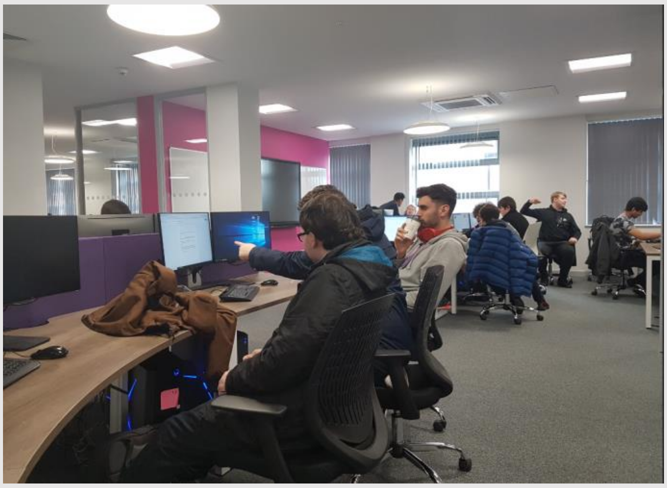
so you can add your part to it 11:57

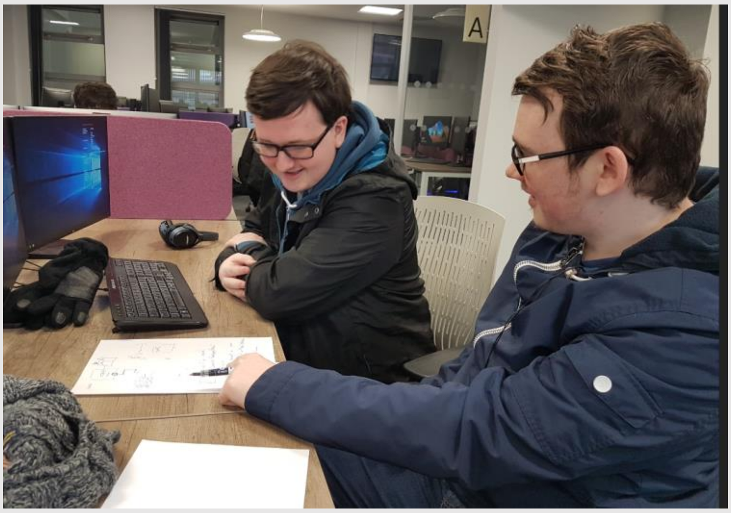
All my part has been done 11:57

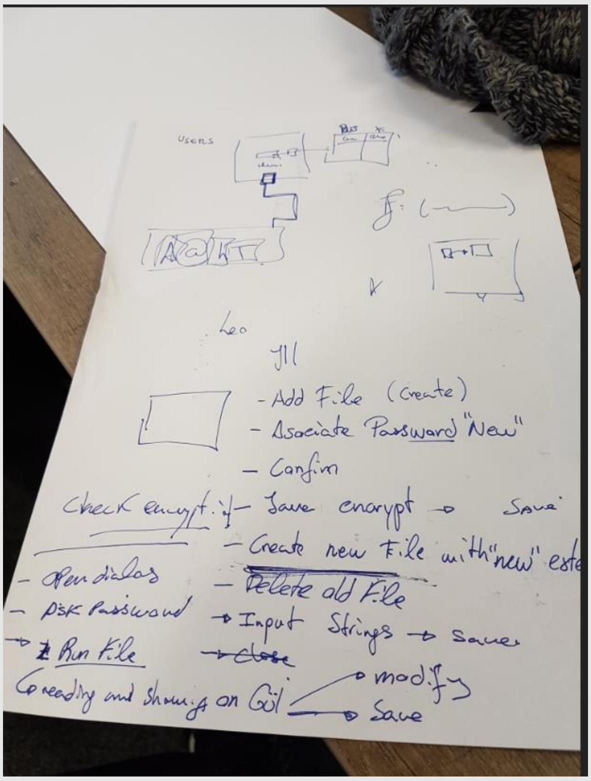
Are you submitting it JAmes? 11:57

Can do 13:06

We also have some photographs from working in college when we were initially planning what we would do, see below:







Leo

Jill

- Add File (Create)
- Asociate Password "New"
- Confirm

Check encrypt. if – Save encrypt → Save

– Create new File with "new" exte

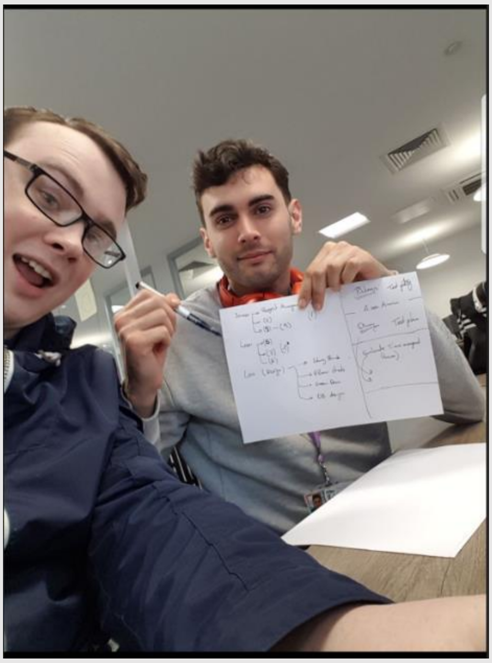
– ~~Delete old File~~

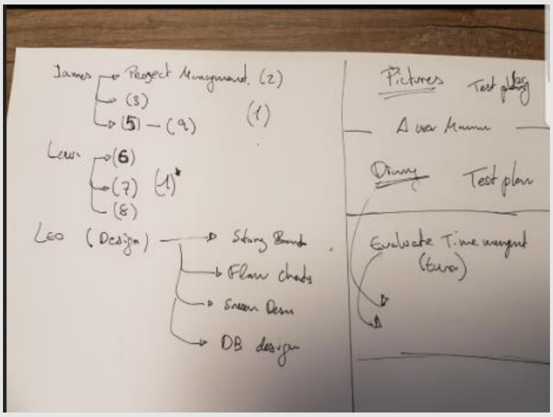
- Open dialog
- Ask Password
→ ~~A~~ Run File

→ Input Strings → Save

→ ~~Close~~

Go reading and showing on GUI → modify → Save





Diary

February 26th

We did a brainstorm and communicated all are skills and interests between the team. One of the members was absence

March 1st

All the team came with two new ideas once we already had all knowledge of skills and of a possible company that we could help. We discussed strengths and weaknesses of each idea.

We also asked for opinion to members of staff which had no IT knowledge. To see if it was understandable and simple.

March 5th

we have talked to our tutor and explained a all possible solutions for a real world issue. We presented the idea of helping out to cut labour hours in the IT department in a hospital.
We have chosen the idea that we are developing. We have decided to develop the solution or either in JAVA or in PYTHON.

And arrange the week’s work.

March 12th

We have assigned each part of the first assessment to each member of the group. We discussed each ones skills and knowledge for this task.

We have said that we are using an application that will monitor our group work.

March 22

The previous day to the deadline we have organized to put all information together and edit the end report.

We discussed the different options between human generating passwords and computer automatic password generating.

We reviewed the work done up to the the date.

We have discussed solutions to resetting passwords on the application. We had to make some changes to problems that we have encountered when doing more putting together all our research.

March 30th

We made sure everyone knew what they would be doing for the project, and assigned each person a section that they must complete.

Everyone was happy with what they were assigned and we started working towards the final product.

April 14th

We chose this day to check up on where everyone was up to and top make sure each member of the group knew what they were doing, this allowed us to see if anyone needed help with what they were doing and allowed us to check how long it would be before we believed we would finish.

April 30th

On this day we once again did a routine check to make sure that all the work was being completed on time and that we were sticking to our schedule, at this point we decided as a group that we would extend our target of a finishing date by 4 weeks giving us till the end of May as we were close to finishing but not quite there, this meant that we had to create a new Gantt chart showing the new estimated finishing date.

May 14th

With the whole project looking closer to being finished we started to focus on making sure that all the report was completed properly and that the time management was all ready to be inserted into one report, we gathered all of our evidence into one place, this included photos of us working and screenshots of us communicating as well as screenshots of our Trello tables which helped us keep track of our work.

June 1st

The final day for us working on this assignment was today, we had finished the program and tested it a lot, we then proceeded to combine all our work together to create our final report before we finally submitted the assignment at the end of today or the start of tomorrow June 2nd.

Test Plan

Test	Date	Purpose of Test	Input (test data)	Expected Output
1	25/05/2018	To see if the program will sign in if the user name is correct but the password is incorrect. (Passwordlock)	Username: el Password: Test	Unable to sign into the application
2	25/05/2018	To see if the program will sign in if the username is incorrect but the password is correct. (Passwordlock)	Username: Test Password: el	Unable to sign into the application
3	25/05/2018	To see if the program will sign in if the username is correct and the password is correct. (Passwordlock)	Username: el Password: el	Sign into the application

Test	Date	Purpose of Test	Input (test data)	Expected Output
4	25/05/2018	To see if you can create a new registry into the database by using the menu tab and selecting new password. (Passwordlock)	Open menu and select new password	Allows you to create a new password
5	25/05/2018	To see if a new password is created when filling in the text boxes required	Write a related name: Test Enter the password: Password Description: This is a test	Stores the new password into the database. Message box appearing indicating that the password was created
6	25/05/2018	To see if the new password appears when pressing the refresh button on the application	Pressing the refresh password button on the application	refreshes and shows new password.
7	25/05/2018	To see if you can create a new registry into the database by using the application and selecting new. (Passwordlock)	Select New on the application	Opens the menu to create a new password
8	25/05/2018	To see if a new password is created when filling in the text boxes required.	Write a related name: Jake14	Stores the new password into the database.

Test	Date	Purpose of Test	Input (test data)	Expected Output
		(Passwordlock)	Enter the password: 234fdf@£qwe Description: Used for Photoshop	Message box appearing indicating that the password was created
9	25/05/2018	To see if the new password appears when pressing the refresh button on the application. (Passwordlock)	Pressing the refresh password button on the application	refreshes and shows new password.
10	25/05/2018	To see if you can search for the name of the password and only the listed password appears. (Passwordlock)	Search by name: Test	Only shows the inputted password inside the database.
11	25/05/2018	To see if the inputted "search by name" clears and the database shows all records when pressing the refresh button after searching for a record. (Passwordlock)	Pressing the refresh button on the application	Clears the "search by name" textbox and shows all the records inside the database

Test	Date	Purpose of Test	Input (test data)	Expected Output
12	25/05/2018	To see if the help menu appears when selecting the help menu inside the help tab. (Passwordlock)	Pressing the help menu button inside the help tab	Opens the help menu
13	25/05/2018	To see if the update tab appears when selecting a record and pressing update. (Passwordlock)	Selecting a record and pressing update	Opens the update menu
14	25/05/2018	To see if the record is updated when changing the name of the password from "test to "Robot". (Passwordlock)	Changing the name from "Test" to "Robot" and pressing the update button	Updates the record and shows a message box saying the password record has been updated
15	25/05/2018	To see if pressing the refresh button shows the newly updated password records. (Passwordlock)	Pressing the refresh button	Refreshes and shows the updated password records

Test	Date	Purpose of Test	Input (test data)	Expected Output
16	25/05/2018	To see if a password record is deleted when selecting the record and pressing the delete button inside the application. (Passwordlock)	Selecting a record and pressing the delete button inside the application	Deletes the record
17	25/05/2018	To see if the database is updated when the record is deleted and changes the number lists inside the database to show the record has been deleted. (Passwordlock)	Pressing refresh	Shows that the record above the deleted record takes its place on the number of records list.
18	25/05/2018	To see if the update menu appears when selecting the update password button inside the options tab. (Passwordlock)	Pressing the update password button on the options menu tab	Opens the update menu
19	25/05/2018	To see if the record is deleted when selecting the delete password on the options tab. (Passwordlock)	Selecting the delete password button on the options menu tab	Deletes the record and updates the number list of the current records.

Test	Date	Purpose of Test	Input (test data)	Expected Output
20	25/05/2018	To see if the user logs out of the application when pressing the log out button inside the options tab (Passwordlock)	Pressing the log out inside the options tab	Logs the user out of the application
21	25/05/2018	To see if the user can re-log into the application after logging off by the login menu without a username and password (Passwordlock)	Pressing the sign in button after logging out	Shouldn’t log into the account
22	25/05/2018	To the if the record is added inside the database for the password locker application (Passwordlock)	Insert the password and viewing the SQL database	Password record inserted inside the SQL database
23	25/05/2018	To the if the record is updated inside the database for the password locker application (Passwordlock)	Update the password and viewing the SQL database	Password record updated inside the SQL database

Test	Date	Purpose of Test	Input (test data)	Expected Output
24	25/05/2018	To the if the record is deleted inside the database for the password locker application (Passwordlock)	Delete the password and viewing the SQL database	Password record deleted inside the SQL database
25	25/05/2018	To see if the program will sign in if the user name is correct but the password is incorrect. (Passwordlockadmin)	Username: admin Password: Test	Unable to sign into the application
26	26/05/2018	To see if the program will sign in if the username is incorrect but the password is correct (Passwordlockadmin)	Username: Test Password: admin	Unable to sign into the application
27	26/05/2018	To see if the program will sign in if the username is correct and the password is correct. (Passwordlockadmin)	Username: admin Password: admin	Sign into the application
28	26/05/2018	To see if the help menu appears when selecting inside the help tab.	Pressing the help menu tab	Shows help menu

Test	Date	Purpose of Test	Input (test data)	Expected Output
		(Passwordlockadmin)		
29	26/05/2018	To see if the new account button works inside the options menu. (Passwordlockadmin)	Pressing the new account button on the options tab	Opens the new account menu
30	26/05/2018	To see if the new button works and opens the new account screen. (Passwordlockadmin)	Pressing the new button on the application	Opens the new account menu
31	26/05/2018	To see if a new account is created when filling in the required fields. (Passwordlockadmin)	Filling in the textbox fields to create a new account and selecting "create account" button	Creates a new account and gives you a unique password for your account
32	26/05/2018	To see if the applications shows the newest records when pressing the refresh button. (Passwordlockadmin)	Pressing the refresh button on the application	Updates the application records and shows the new added records.

Test	Date	Purpose of Test	Input (test data)	Expected Output
33	26/05/2018	To see if it locates a record when typing the forename of the account into the name filter box. (Passwordlockadmin)	Typing “Flash” into the name filter box	Shows the record
34	26/05/2018	To see if it locates a record when typing the surname of the account into the surname filter box. (Passwordlockadmin)	Typing “Gorden” into the name filter box	Shows the record
35	26/05/2018	To see if it locates a record when typing the date of birth of the account into the date of birth filter box. (Passwordlockadmin)	Typing “07/01/1934” into the name filter box	Shows the record
36	26/05/2018	To see if reset password menu appears if you select a record and press the reset account password button the on the options tab. (Passwordlockadmin)	Pressing the reset account password button on the options tab	Shows the reset account password

Test	Date	Purpose of Test	Input (test data)	Expected Output
37	26/05/2018	To see if the password is reset for the account if the required fields are completed incorrectly. (Passwordlockadmin)	Inputting two correct questions but inputting one incorrect question into the fields and pressing reset	Shows an error on the incorrect field and will not let you reset the password.
38	26/05/2018	To see if the password is reset for the account if the required fields are completed correctly. (Passwordlockadmin)	Inputting the correct answers into the fields	Shows you a new unique random password for the account
39	26/05/2018	To see if the new admin account menu appears when pressing the new admin account button in the options tab. (Passwordlockadmin)	Pressing the new admin account button inside the options tab	Opens the new admin account menu
40	26/05/2018	Creates a new admin account when inputting a username and password into the fields and pressing the “create admin” button. (Passwordlockadmin)	New username: James Enter the password: 123456 Pressing the “create admin button”	Creates a new admin account

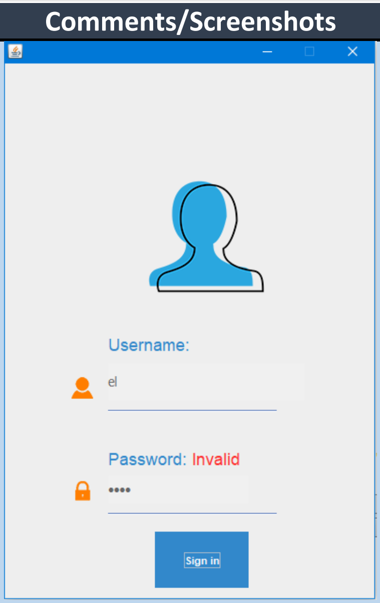
Test	Date	Purpose of Test	Input (test data)	Expected Output
41	26/05/2018	To see if the change admin password menu opens when pressing the change admin password button on the option tab. (Passwordlockadmin)	Pressing the change admin password button on the option tab	Opens the change admin password menu
42	26/05/2018	To see if you can log into the admin password locker account with the new admin account. (Passwordlockadmin)	Username: James Password: 123456	Logs into the admin password locker account
43	26/05/2018	To see if the admin password is changed when inputting the correct old password but not matching the two new passwords. (Passwordlockadmin)	Old password: 123456 New password: 654321 Confirm new password: test	Bring up an error message saying the new passwords do not match
44	26/05/2018	To see if the admin password is changed when inputting the incorrect old password but matching the two new passwords.	Old password: test New password: 654321	Bring up an error message saying the password does not match

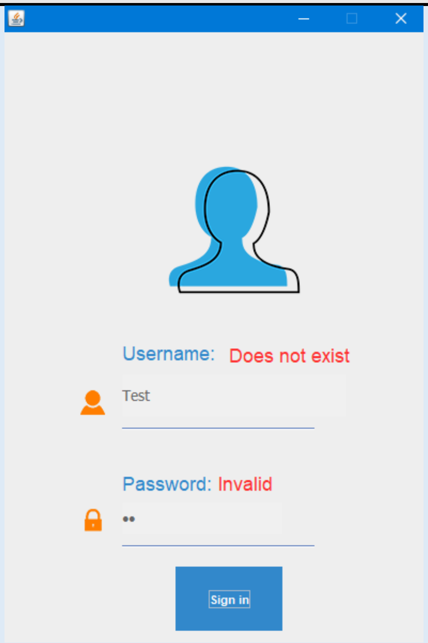
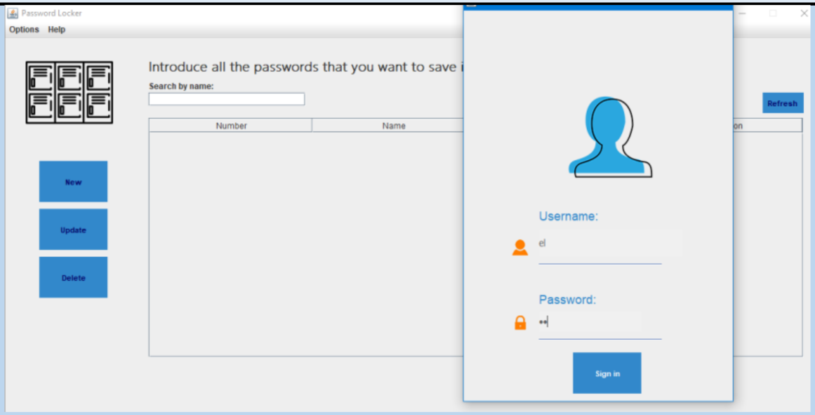
Test	Date	Purpose of Test	Input (test data)	Expected Output
		(Passwordlockadmin)	Confirm new password: 654321	
45	26/05/2018	To see if the admin password is changed when inputting the correct old password and matching the two new passwords. (Passwordlockadmin)	Old password: 123456 New password: 654321 Confirm new password: 654321	Message box appears saying the password has been changed
46	26/05/2018	To see if the user can access the admin password locker with the new updated password at the log in screen. (Passwordlockadmin)	Username: James Password: 654321	Logs into the application
47	26/05/2018	To see if you can log out of the admin password locker by pressing the log out button on the options tab. (Passwordlockadmin)	Pressing the log out button on the options tab	Logs out of the application
48	26/05/2018	To see if you can log into the password locker with the new account created	Username: Flash Password: q0a7tT	Logs into the application

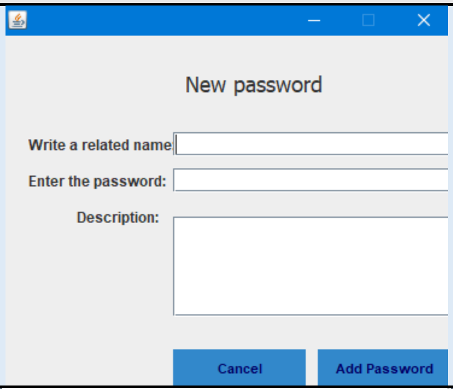
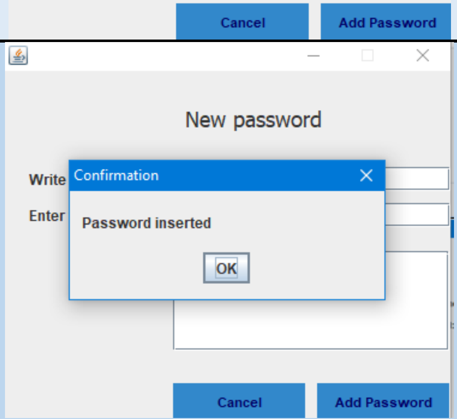
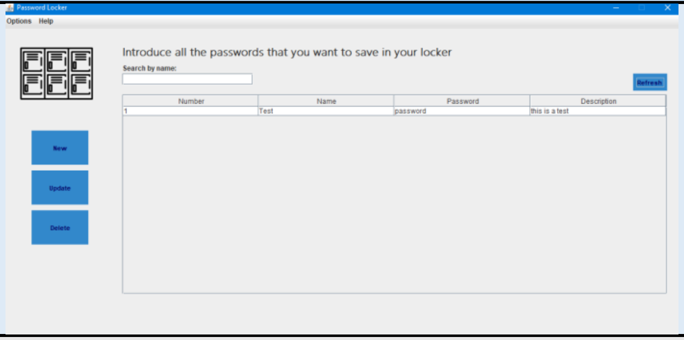
Test	Date	Purpose of Test	Input (test data)	Expected Output
		through the admin password locker with the correct username and password (Passwordlock)		
49	26/05/2018	To see if you can log into the password locker with the new account created through the admin password locker with the incorrect username and correct password. (Passwordlock)	Username: test Password: q0a7tT	Message saying the user does not exist
50	26/05/2018	To see if you can log into the password locker with the new account created through the admin password locker with the correct username and incorrect password. (Passwordlockadmin)	Username: Flash Password: test	Message saying the password is invalid
51	26/05/2018	To the if the record is added inside the database for the admin password locker application.	Insert the account and viewing the SQL database	account record inserted inside the SQL database

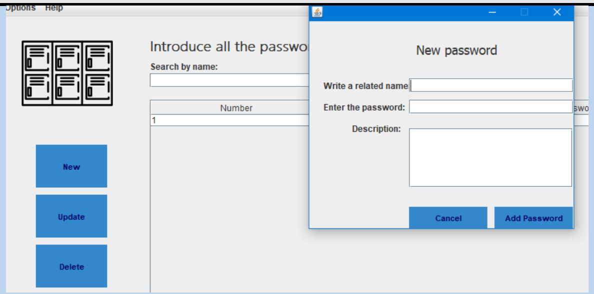
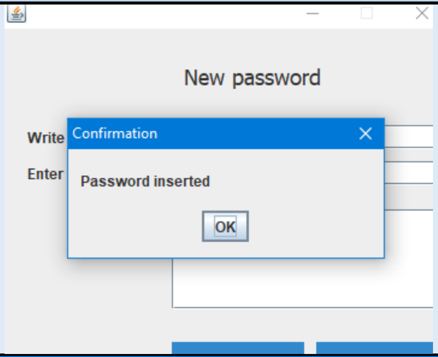
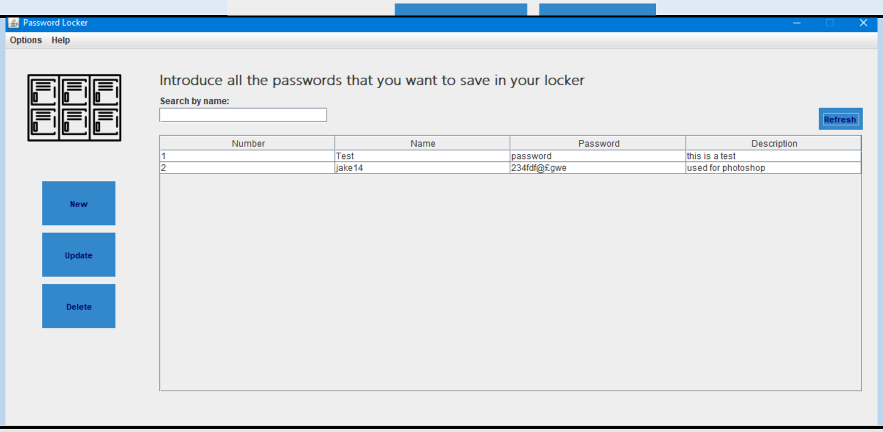
Test	Date	Purpose of Test	Input (test data)	Expected Output
		(Passwordlockadmin)		
52	26/05/2018	To the if the record is updated inside the database for the password locker application. (Passwordlockadmin)	Update the account password and viewing the SQL database	Password record updated inside the SQL database

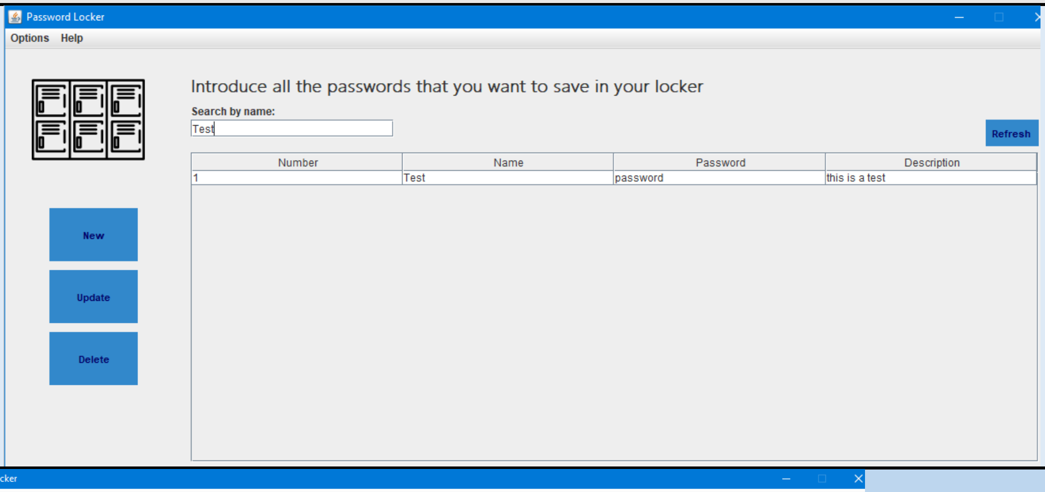
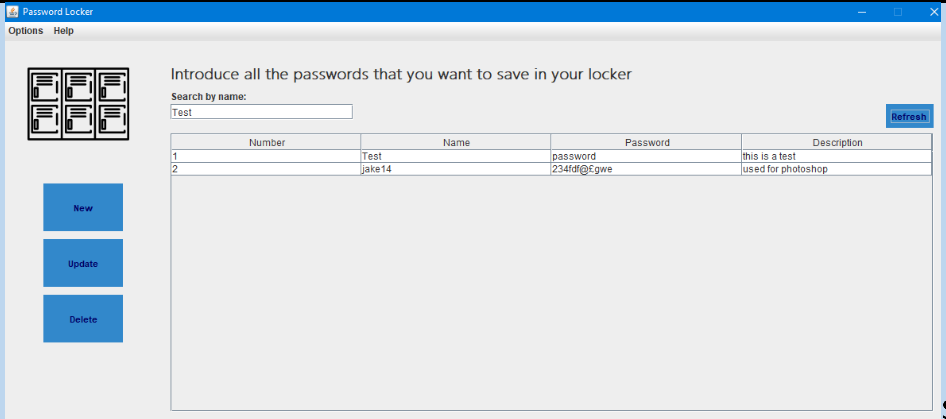
Test Log

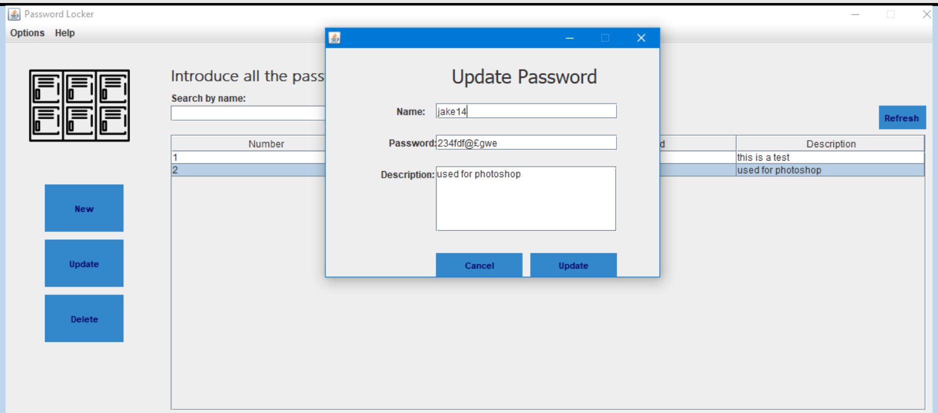
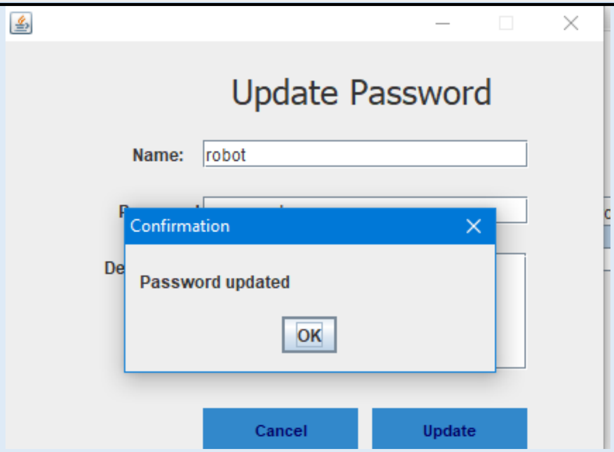
Test	Date	Actual Output	Success	Comments/Screenshots
1	22/03/2018	User cannot login as expected	Yes	 Username: el Password: Invalid •••• Invalid message displayed informing user they cant use the password

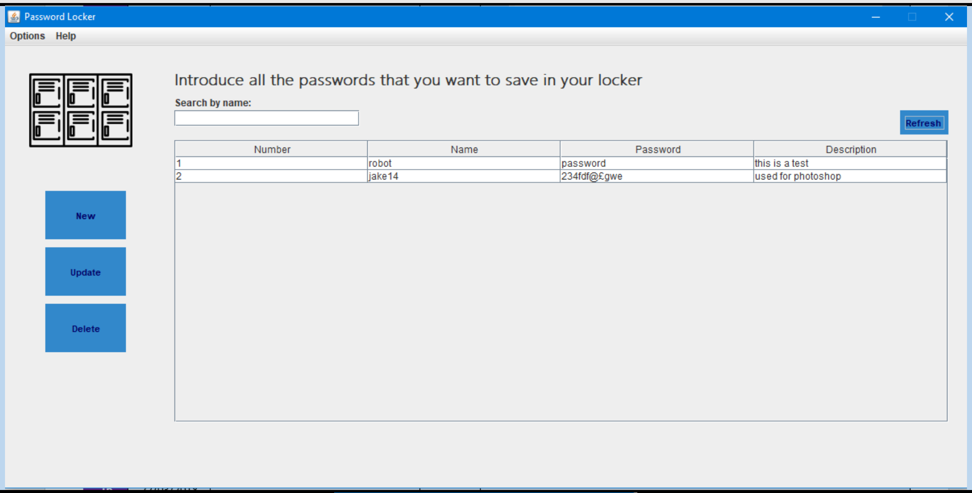
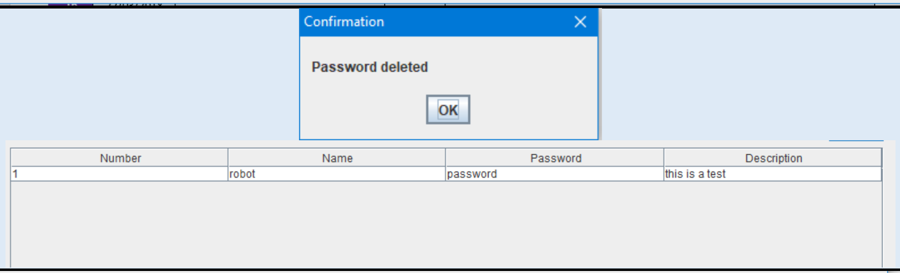
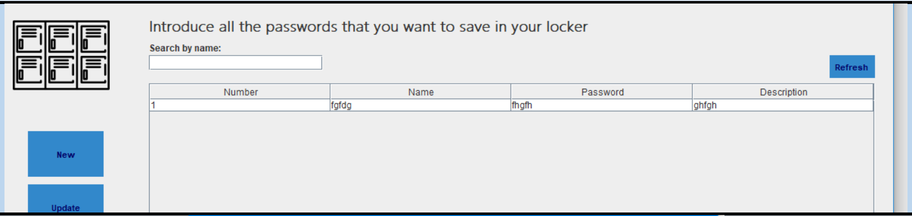
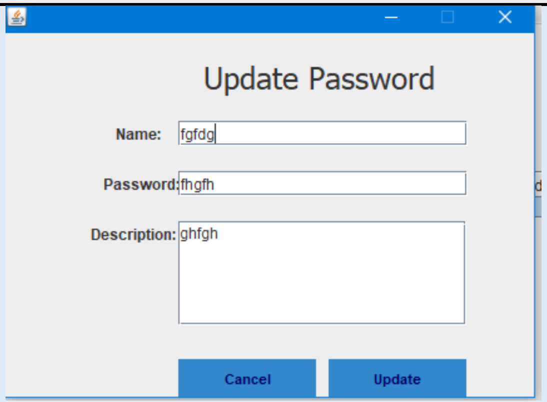
2	22/03/2018	User cannot login and message displays informing them of such	Yes	 Username: Does not exist Test Password: Invalid •• Does not exist message displayed informing user they cannot use username
3	22/03/2018	User can login using correct information	Yes	 Password Locker Introduce all the passwords that you want to save i Search by name: Username: el Password: •• user was able to login and access the form with the correct information provided

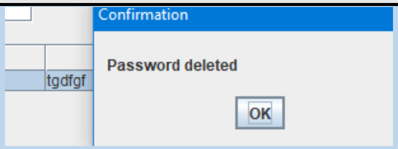
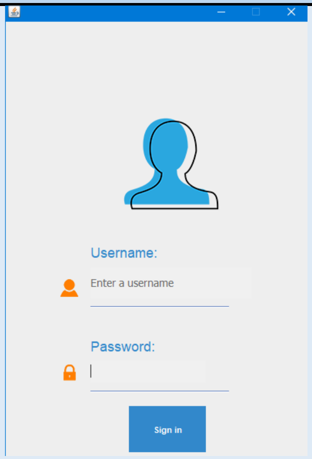
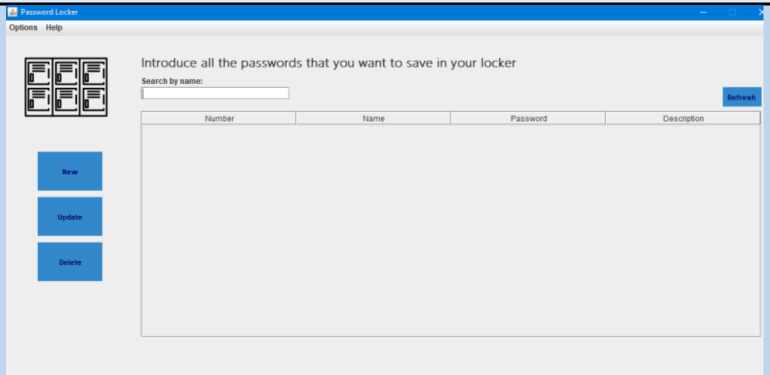
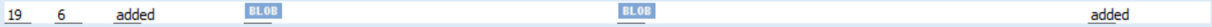
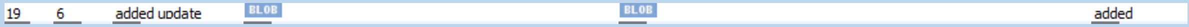
4	22/03/2018	User can open the new password menu	Yes	 New password Write a related name: Enter the password: Description: Cancel Add Password
5	22/03/2018	User can create a new password by completing the fields and click the add password button	Yes	 New password Write Enter Confirmation Password inserted OK Cancel Add Password
6	22/03/2018	Pressing the refresh button displays any new information	Yes	 Password Locker Options Help Introduce all the passwords that you want to save in your locker Search by name: Refresh Number | Name | Password | Description 1 | Test | password | this is a test New Update Delete

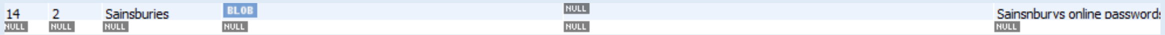
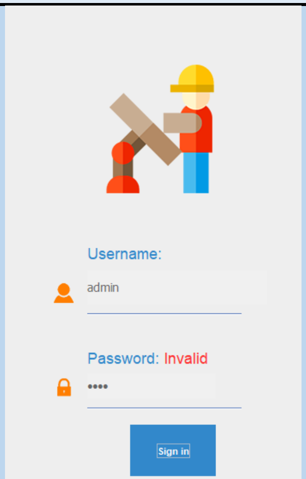
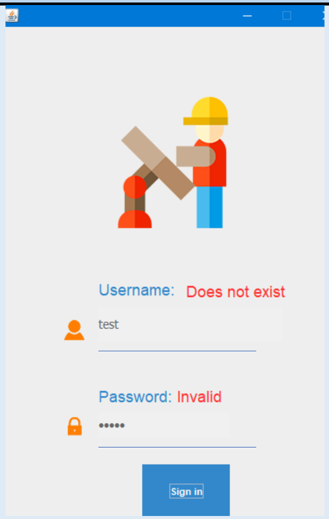
<table>
<tr><td>7</td><td>22/03/2018</td><td>User can open the new password form via the button on the main form</td><td>Yes</td><td>
Options Help
New
Update
Delete
Number
Cancel
Add Password
Introduce all the passwo
Search by name:
1
New password
Write a related name:
Enter the password:
Description:</td></tr>
<tr><td>8</td><td>22/03/2018</td><td>User can enter password without any issues</td><td>Yes</td><td>
New password
Write
Enter
Confirmation
Password inserted
</td></tr>
<tr><td>9</td><td>22/03/2018</td><td>User can use the refresh feature to display the new information</td><td>Yes</td><td>
Password Locker
Options Help
New
Update
Delete
Refresh
Introduce all the passwords that you want to save in your locker
Search by name:
<table>
<tr><th>Number</th><th>Name</th><th>Password</th><th>Description</th></tr>
<tr><td>1</td><td>Test</td><td>password</td><td>this is a test</td></tr>
<tr><td>2</td><td>jake14</td><td>234fdf@£gwe</td><td>used for photoshop</td></tr>
</table></td></tr>
</table>

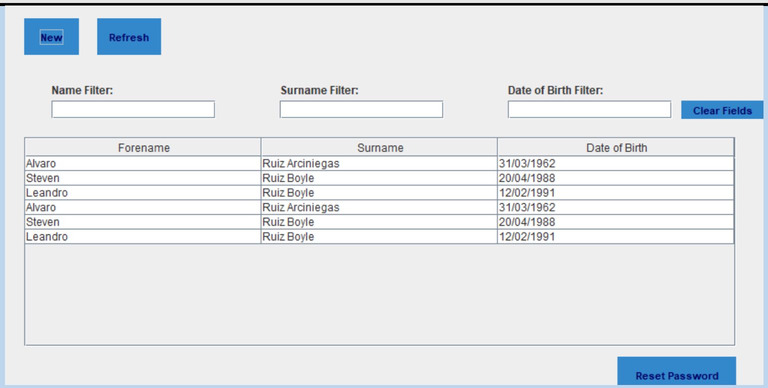
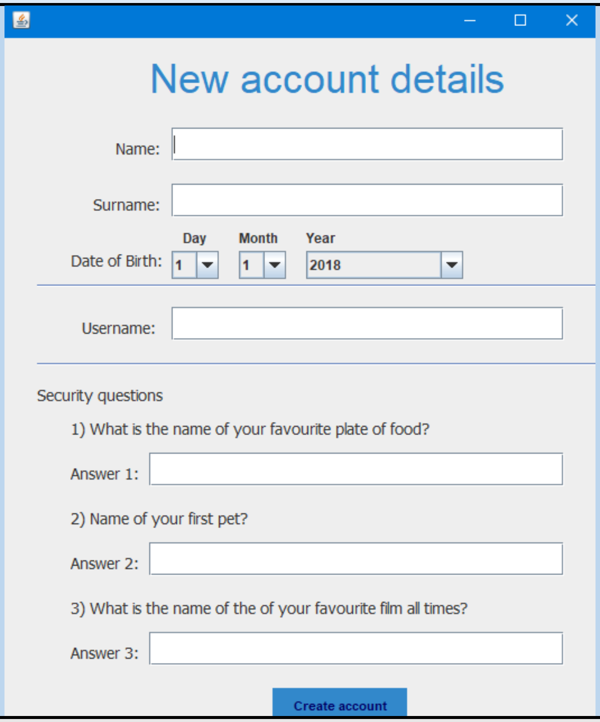
10	22/03/2018	Search function is working as intended and shows the data which matches the search field	**Yes**	 Password Locker Options Help Refresh New Update Delete Introduce all the passwords that you want to save in your locker Search by name: Test Number | Name | Password | Description 1 | Test | password | this is a test
11	22/03/2018	Refreshing the page resets, it back to the default layout displaying all information. However, the search field doesn't reset, and the input is present	**Yes/No**	 Password Locker Options Help Refresh New Update Delete Introduce all the passwords that you want to save in your locker Search by name: Test Number | Name | Password | Description 1 | Test | password | this is a test 2 | jake14 | 234fdf@£gwe | used for photoshop Search by name field doesn't clear, however the table does return to its original state.
12	22/03/2018	Help menu doesn't open as expected	**No**	Help menu doesn't open, button leads to nowhere.

13	22/03/2018	Update form opens as expected allowing user to update any data	Yes		 Password Locker Options Help Introduce all the pass Search by name: Refresh New Update Delete Cancel Update Update Password Name: jake14 Password:234fdf@£gwe Description: used for photoshop Number … Description 1 … this is a test 2 … used for photoshop
14	22/03/2018	User can change the data in the table as expected	Yes		 Update Password Name: robot Confirmation Password updated Name was originally Test, was changed to Robot.

15	22/03/2018	Refreshing shows the updated information as expected	Yes	 Password Locker Options Help New Update Delete Refresh Introduce all the passwords that you want to save in your locker Search by name: Number / Name / Password / Description 1 / robot / password / this is a test 2 / jake14 / 234fdf@£gwe / used for photoshop
16	22/03/2018	User is able to delete a record without any issues	Yes	 Password deleted Number / Name / Password / Description 1 / robot / password / this is a test
17	22/03/2018	Refreshing the form shows that the record has been deleted	Yes	 Introduce all the passwords that you want to save in your locker Search by name: Number / Name / Password / Description 1 / fgfdg / fhgfh / ghfgh
18	22/03/2018	User can use the update button under the options menu as well	Yes	 Update Password Name: fgfdg Password: fhgfh Description: ghfgh

19	22/03/2018	User can use the delete button from the options menu as well	Yes	 Password deleted
20	22/03/2018	User can log out of the application	Yes	 User is returned to this page when they log out of the application
21	22/03/2018	The user will be able to login if they leave both fields blank after logging out	No	 Password Locker Options Help Search by name: Refresh Number Name Password Description New Update Delete Introduce all the passwords that you want to save in your locker User can login without a username or password ONLY after logging out
22	22/03/2018	Information is stored in the database correctly and is encrypted	Yes	 19 6 added BLOB BLOB added The information is stored as it should and any data that is meant to be encrypted is, this can be seen via the BLOB tag in its field instead of the data that should be there
23	22/03/2018	Information is stored in the database correctly and is encrypted	Yes	 19 6 added update BLOB BLOB added

24	22/03/2018	Information is deleted from the database as expected	Yes	 14 2 Sainsburies BLOB NULL Sainsnburys online password: NULL NULL NULL NULL NULL NULL
25	22/03/2018	User is unable to sign in using an incorrect password	Yes	 Username: admin Password: Invalid •••• Sign in User is shown an invalid message if the password is wrong
26	22/03/2018	User is unable to sign in with an incorrect username	Yes	 Username: Does not exist test Password: Invalid ••••• Sign in User is shown a Does not exist message if user is incorrect, aswell as the password being listed as invalid. This is so if the person trying to access doesn't know either they wont know if one or both are incorrect

27	22/03/2018	User can login to application provided they use the correct/valid user and pass combination	Yes	 User is sent to this form after they login
28	22/03/2018	Help button doesn't take the user anywhere or display any information	No	Help button doesn't work, altohugh you can select the button, it doesn't bring a new form or any information up for the user
29	22/03/2018	The user can open the form clicking the new account option in the options menu	Yes	

Forename	Surname	Date of Birth
Alvaro	Ruiz Arciniegas	31/03/1962
Steven	Ruiz Boyle	20/04/1988
Leandro	Ruiz Boyle	12/02/1991
Alvaro	Ruiz Arciniegas	31/03/1962
Steven	Ruiz Boyle	20/04/1988
Leandro	Ruiz Boyle	12/02/1991

New account details

Name:

Surname:

Date of Birth: Day 1 Month 1 Year 2018

Username:

Security questions

1) What is the name of your favourite plate of food?

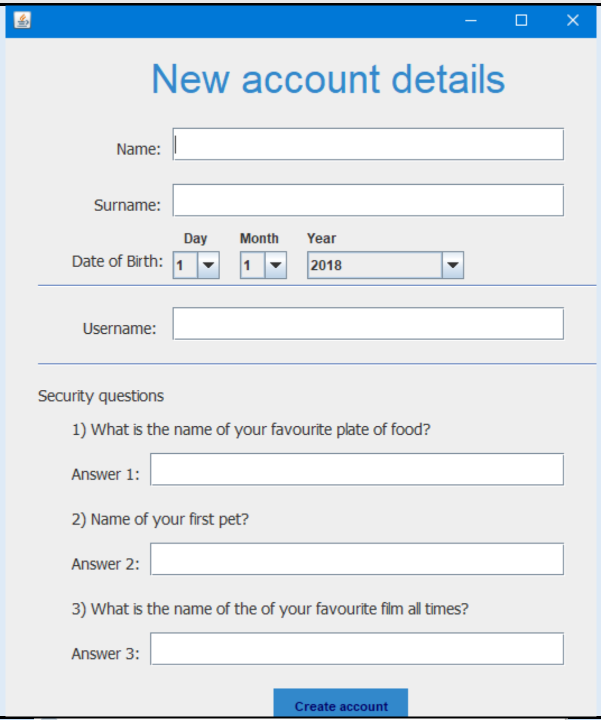
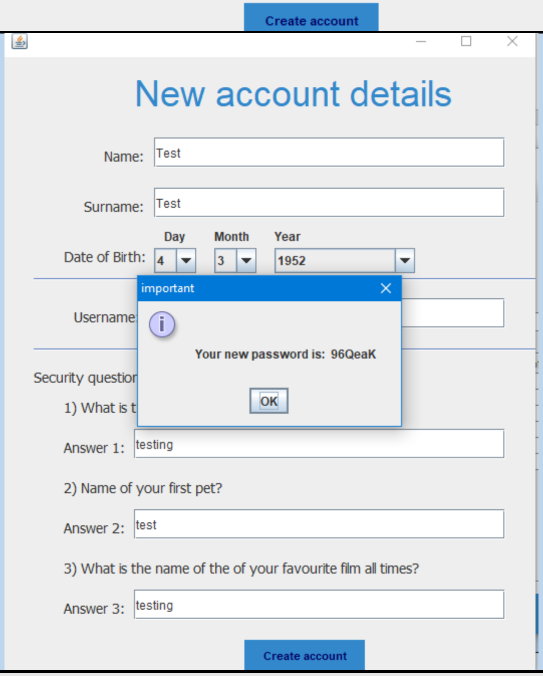
Answer 1:

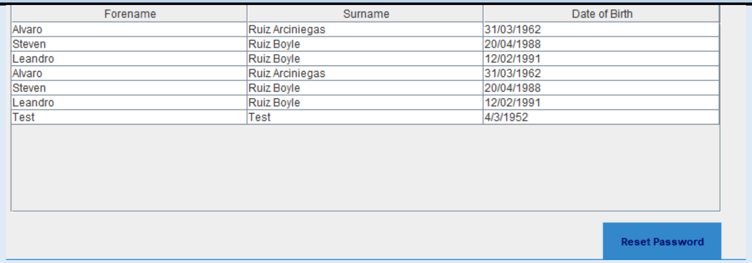
2) Name of your first pet?

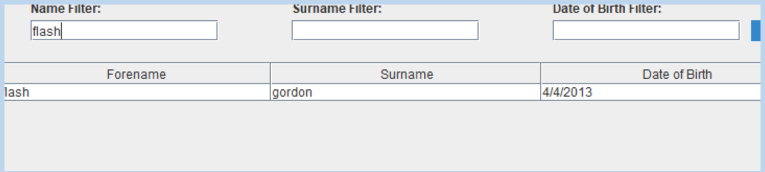
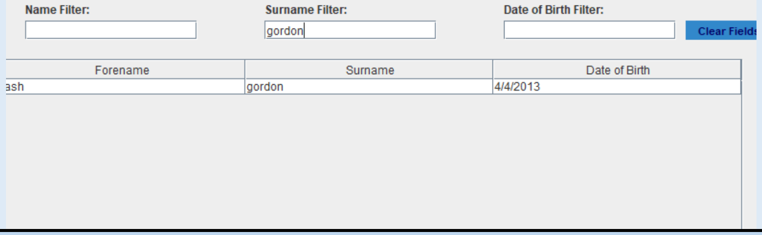
Answer 2:

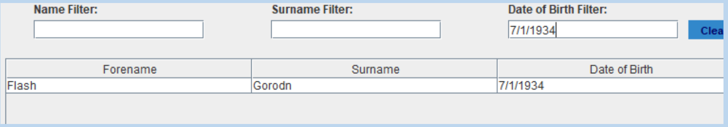
3) What is the name of the of your favourite film all times?

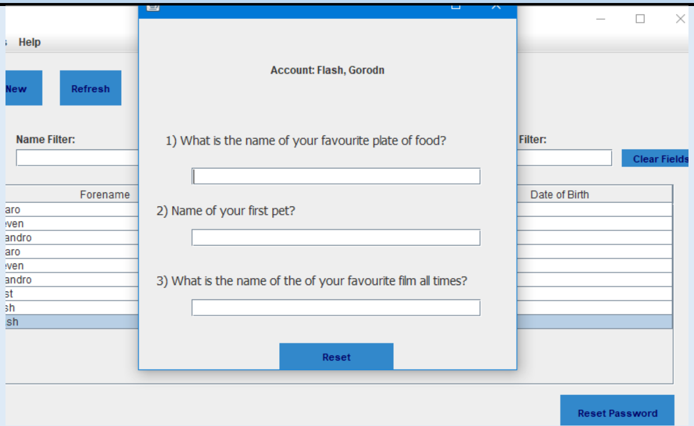
Answer 3:

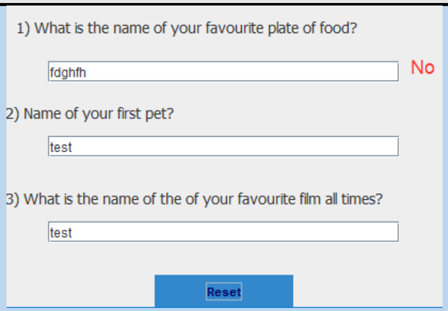
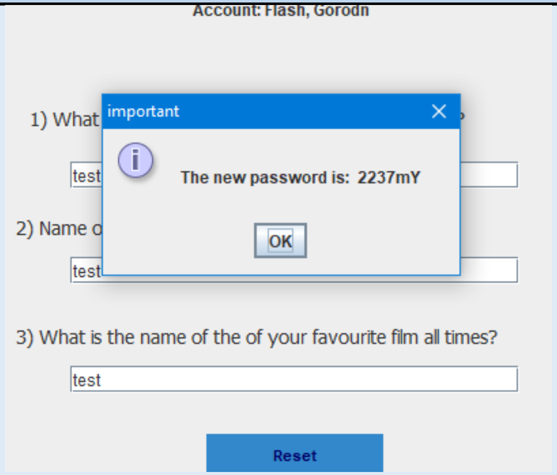
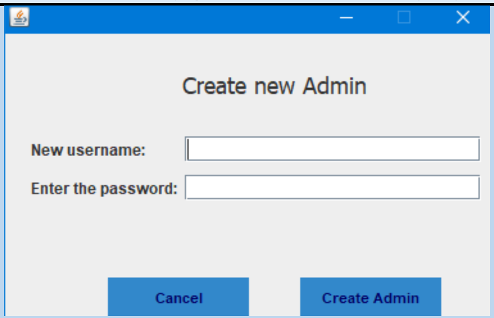
30	22/03/2018	The user can also open the form my clicking the button on the main form labelled “new”	Yes	 New account details Name: Surname: Date of Birth: Day 1 Month 1 Year 2018 Username: Security questions 1) What is the name of your favourite plate of food? Answer 1: 2) Name of your first pet? Answer 2: 3) What is the name of the of your favourite film all times? Answer 3: Create account
31	22/03/2018	User can fill in all the fields without any problem and create their account by clicking the create account button	Yes	 New account details Name: Test Surname: Test Date of Birth: Day 4 Month 3 Year 1952 Username important Your new password is: 96QeaK OK Security questior 1) What is t Answer 1: testing 2) Name of your first pet? Answer 2: test 3) What is the name of the of your favourite film all times? Answer 3: testing Create account

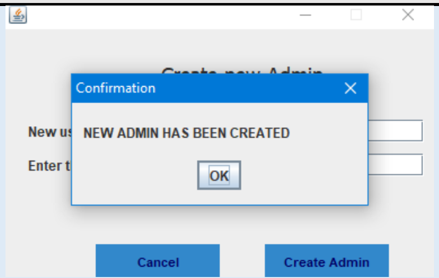
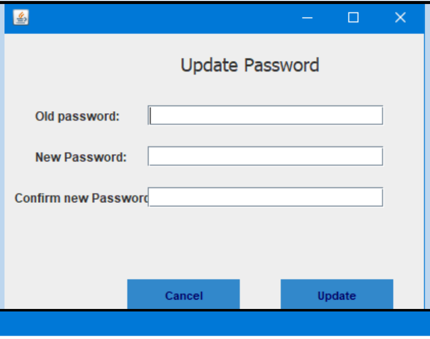
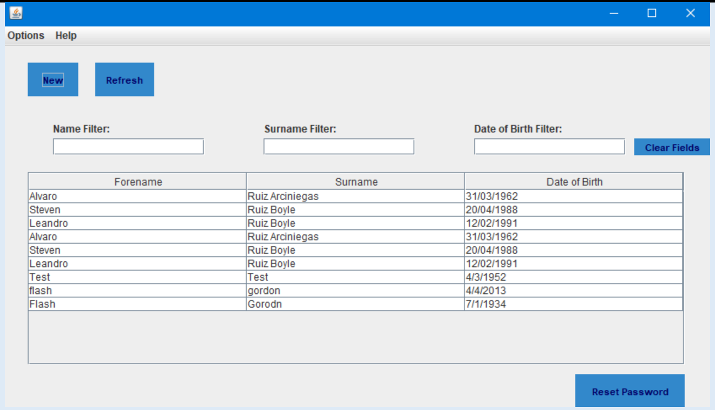
<table>
<tr><td></td><td></td><td></td><td></td><td>Once they have created their account a display box showing them their randomised password is displayed for them</td></tr>
<tr><td>32</td><td>22/03/2018</td><td>Pressing the refresh button displays the new information as intended</td><td>Yes</td><td>
<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Test</td><td>Test</td><td>4/3/1952</td></tr>
</table>

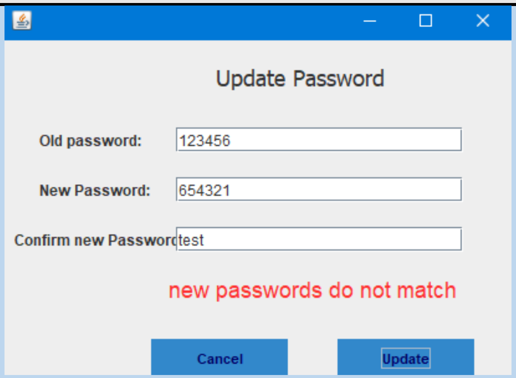
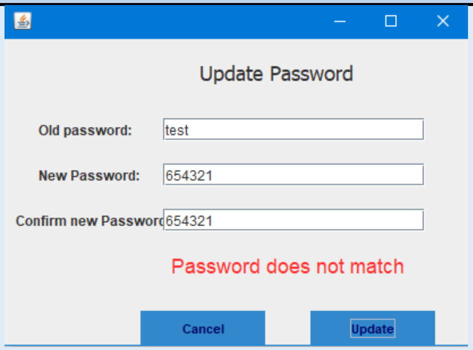
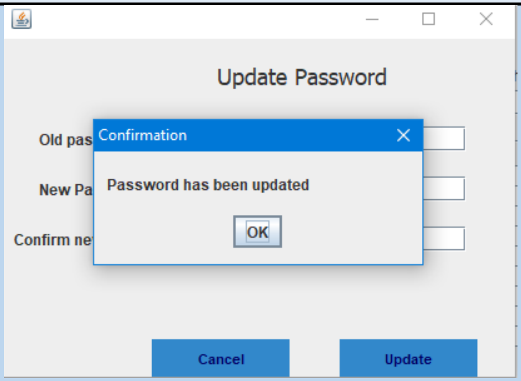
The new account is shown in the list at the bottom as it should be</td></tr>
<tr><td>33</td><td>22/03/2018</td><td>Typing into the forename box displays any records that match what the user typed</td><td>Yes</td><td>
Name Filter: flash
Surname Filter:
Date of Birth Filter:
<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>lash</td><td>gordon</td><td>4/4/2013</td></tr>
</table></td></tr>
<tr><td>34</td><td>22/03/2018</td><td>Typing into the surname box displays any records that match what the user typed</td><td>Yes</td><td>
Name Filter:
Surname Filter: gordon
Date of Birth Filter:

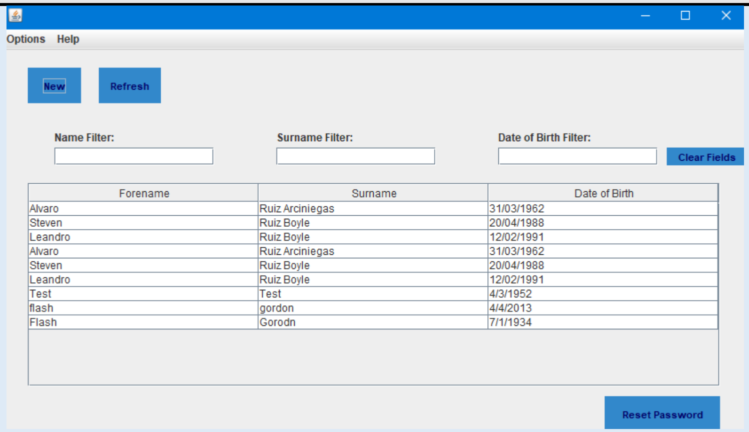
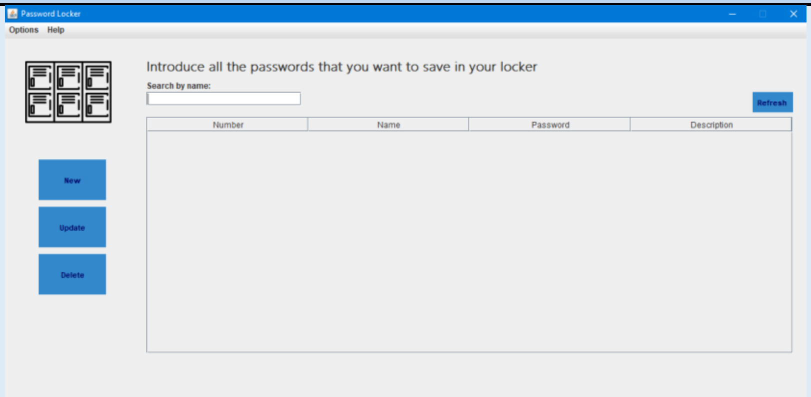
<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>ash</td><td>gordon</td><td>4/4/2013</td></tr>
</table></td></tr>
<tr><td>35</td><td>22/03/2018</td><td>Typing into the DOB box displays any records that match what the user typed</td><td>Yes</td><td>
Name Filter:
Surname Filter:
Date of Birth Filter: 7/1/1934

<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>Flash</td><td>Gorodn</td><td>7/1/1934</td></tr>
</table></td></tr>
<tr><td>36</td><td>22/03/2018</td><td>Pressing the reset password button opens the form, asking you to enter your security questions to reset the password</td><td>Yes</td><td>
Help
New
Refresh
Name Filter:
Filter:
Clear Fields
Forename
Date of Birth
aro
even
andro
aro
even
andro
st
sh
sh
Reset Password
Account: Flash, Gorodn
1) What is the name of your favourite plate of food?
2) Name of your first pet?
3) What is the name of the of your favourite film all times?
</td></tr>
</table>

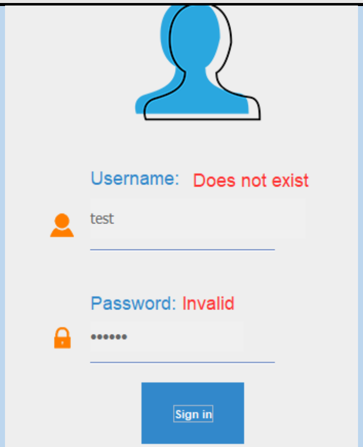
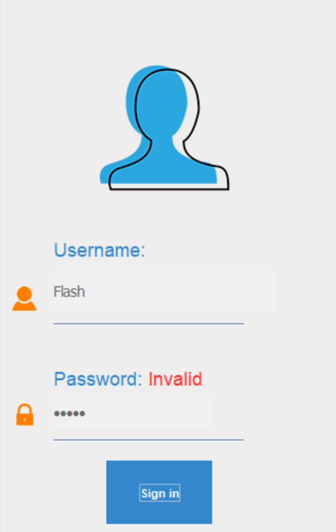
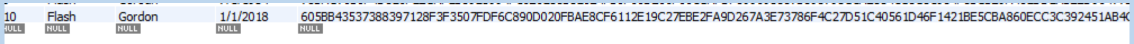
37	22/03/2018	User will be informed if the information in one of the fields is incorrect by a message in read next to the box	Yes	 1) What is the name of your favourite plate of food? fdghfh No 2) Name of your first pet? test 3) What is the name of the of your favourite film all times? test Reset A “No” message is displayed next to the incorrect field
38	22/03/2018	Once the user has entered the correct information they are shown a display box with the new password in for them to remember	Yes	 Account: Flash, Gorodn 1) What test 2) Name o test important The new password is: 2237mY OK 3) What is the name of the of your favourite film all times? test Reset New password is randomly generated then displayed to the user
39	22/03/2018	User can click the new admin account in the options menu to start creating a new admin account	Yes	 Create new Admin New username: Enter the password: Cancel Create Admin

<table>
<tr><td>40</td><td>22/03/2018</td><td>User can create an admin account once they have entered a user and password</td><td>Yes</td><td>
Create new Admin
New us
Enter t
Confirmation
NEW ADMIN HAS BEEN CREATED
OK
Cancel
Create Admin</td></tr>
<tr><td>41</td><td>22/03/2018</td><td>User can click the change admin password option, and this will display the update password form</td><td>Yes</td><td>
Update Password
Old password:
New Password:
Confirm new Passwor
Cancel
Update</td></tr>
<tr><td>42</td><td>22/03/2018</td><td>User can login to the newly created account assuming they use the correct login information</td><td>Yes</td><td>
Options Help
New
Refresh
Name Filter:
Surname Filter:
Date of Birth Filter:
Clear Fields<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Test</td><td>Test</td><td>4/3/1952</td></tr>
<tr><td>flash</td><td>gordon</td><td>4/4/2013</td></tr>
<tr><td>Flash</td><td>Gorodn</td><td>7/1/1934</td></tr>
</table>Reset Password
User is taken to this form when they login</td></tr>
</table>

43	22/03/2018	Having the confirm new password field not the same as the new password field displays an error message	Yes	 Update Password Old password: 123456 New Password: 654321 Confirm new Passwor test new passwords do not match ”New passwords do no match” is displayed so the user knows there is a mistake
44	22/03/2018	Typing in the old password incorrectly displays an error message	Yes	 Update Password Old password: test New Password: 654321 Confirm new Passwor 654321 Password does not match ”Password does not match” is displayed so the user knows there was a mistake
45	22/03/2018	User can update their password without any issue	Yes	 Update Password Old pas New Pa Confirm ne Confirmation Password has been updated A message box displays that the password has been updated

<table>
<tr><td>46</td><td>22/03/2018</td><td>The user can login to the form with the newly amended password</td><td>Yes</td><td>
Options
Help
New
Refresh
Name Filter:
Surname Filter:
Date of Birth Filter:
Clear Fields
Reset Password
<table>
<tr><th>Forename</th><th>Surname</th><th>Date of Birth</th></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Alvaro</td><td>Ruiz Arciniegas</td><td>31/03/1962</td></tr>
<tr><td>Steven</td><td>Ruiz Boyle</td><td>20/04/1988</td></tr>
<tr><td>Leandro</td><td>Ruiz Boyle</td><td>12/02/1991</td></tr>
<tr><td>Test</td><td>Test</td><td>4/3/1952</td></tr>
<tr><td>flash</td><td>gordon</td><td>4/4/2013</td></tr>
<tr><td>Flash</td><td>Gorodn</td><td>7/1/1934</td></tr>
</table>
assuming the user enters the correct information they will be taken to this form after they login with their new password</td></tr>
<tr><td>47</td><td>22/03/2018</td><td>The log out button doesn’t close the form and log the user out</td><td>No</td><td>Log out button does not work, user can click the button but nothing happens</td></tr>
<tr><td>48</td><td>22/03/2018</td><td>User can log into the application using their new account that was generated from the admin form</td><td>Yes</td><td>
Password Locker
Options
Help
Search by name:
Refresh
Number
Name
Password
Description
New
Update
Delete
Introduce all the passwords that you want to save in your locker

Assuming the user has the correct information the user can log into the account that was created for them</td></tr>
</table>

49	22/03/2018	User cannot login using an incorrect name even if their randomly generated password is correct	Yes	 Username: Does not exist test Password: Invalid •••••• An error message will inform the user that one or more of the fields is incorrect so they cant login
50	22/03/2018	User cannot login to the application if their randomly generated password is incorrect even if their username is correct	Yes	 Username: Flash Password: Invalid ••••• The user will be informed of an incorrect password by an "invalid" message in red next to the password label
51	22/03/2018	The newly created account in inserted into the database as it was intended	Yes	 10 Flash Gordon 1/1/2018 605BB43537388397128F3F3507FDF6C890D020FBAE8CF6112E19C27EBE2FA9D267A3E73786F4C27D51C40561D46F1421BE5CBA860ECC3C392451AB4 NULL NULL NULL NULL NULL As you can see the user is displayed here, however their password is encrypted and because of this si displayed as a lot of random characters
52	22/03/2018	Passwords are updated in the database as intended	Yes	Passwords are encrypted, thus meaning we cant have an image showing if they have changed. Althought logging in with new password does work.

Source Code Password Locker

userLogin.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.awt.event.KeyEvent;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;


public class UserLogin extends javax.swing.JFrame {
    boolean correctUser = false;
    boolean correctPassword = false;
    boolean login = false;

    int chosenAccount;          //para test

    public UserLogin() {
        initComponents();
        lbl_UserMissing.setVisible(false);
        lbl_InvalidPass.setVisible(false);
    }

    public static void main(String[] args){
       UserLogin ul = new UserLogin();
       ul.setLocationRelativeTo(null);
       ul.setVisible(true);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lbl_LoginImage = new javax.swing.JLabel();
        lbl_BlueUser = new javax.swing.JLabel();
        lbl_UserMissing = new javax.swing.JLabel();
        tb_Username = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        lbl_BluePass = new javax.swing.JLabel();
        lbl_InvalidPass = new javax.swing.JLabel();
        tb_Password = new javax.swing.JPasswordField();
        jLabel5 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        bt_SignIn = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setLocationByPlatform(true);
        setMinimumSize(new java.awt.Dimension(400, 600));
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        lbl_LoginImage.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/logo.png")));        //
NOI18N
        getContentPane().add(lbl_LoginImage,                                       new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, -1, -1));

        lbl_BlueUser.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_BlueUser.setForeground(new java.awt.Color(50, 136, 203));
        lbl_BlueUser.setText("Username:");
        getContentPane().add(lbl_BlueUser,                                         new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 290, 100, 20));

        lbl_UserMissing.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_UserMissing.setForeground(new java.awt.Color(255, 51, 51));
        lbl_UserMissing.setText("Does not exist");
        getContentPane().add(lbl_UserMissing,                                      new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 290, -1, -1));

        tb_Username.setBackground(new java.awt.Color(240, 240, 240));
        tb_Username.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        tb_Username.setForeground(new java.awt.Color(102, 102, 102));
        tb_Username.setText("Enter a username");
        tb_Username.setToolTipText("");
        tb_Username.setBorder(null);
        tb_Username.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                tb_UsernameFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                tb_UsernameFocusLost(evt);
            }
        });
        tb_Username.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tb_UsernameActionPerformed(evt);
            }
        });
        tb_Username.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                tb_UsernameKeyPressed(evt);
            }
        });
        getContentPane().add(tb_Username,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 320, 210, 40));

        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/user.png")));        //
NOI18N
        getContentPane().add(jLabel3,                                              new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 330, 30, 30));

        lbl_BluePass.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_BluePass.setForeground(new java.awt.Color(50, 136, 203));
        lbl_BluePass.setText("Password:");
        getContentPane().add(lbl_BluePass,                                         new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 410, -1, -1));

        lbl_InvalidPass.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_InvalidPass.setForeground(new java.awt.Color(255, 51, 51));
        lbl_InvalidPass.setText("Invalid");
        getContentPane().add(lbl_InvalidPass,                                      new
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 410, -1, -1));

        tb_Password.setBackground(new java.awt.Color(240, 240, 240));
        tb_Password.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        tb_Password.setForeground(new java.awt.Color(102, 102, 102));
        tb_Password.setText("......");
        tb_Password.setBorder(null);
        tb_Password.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                tb_PasswordFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                tb_PasswordFocusLost(evt);
            }
        });
        tb_Password.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                tb_PasswordKeyPressed(evt);
            }
        });
        getContentPane().add(tb_Password,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 440, 150, 30));

        jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/password.png"))); //
NOI18N
        getContentPane().add(jLabel5,                                            new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 440, -1, 30));
        getContentPane().add(jSeparator1,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 480, 180, 20));
        getContentPane().add(jSeparator2,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 370, 180, 20));

        bt_SignIn.setBackground(new java.awt.Color(50, 136, 203));
        bt_SignIn.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        bt_SignIn.setForeground(new java.awt.Color(255, 255, 255));
        bt_SignIn.setText("Sign in");
        bt_SignIn.setBorder(null);
        bt_SignIn.setBorderPainted(false);
        bt_SignIn.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        bt_SignIn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bt_SignInActionPerformed(evt);
            }
        });
        bt_SignIn.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                bt_SignInKeyPressed(evt);
            }
        });
        getContentPane().add(bt_SignIn,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 500, 100, 60));

        pack();
    }// </editor-fold>

    private void tb_UsernameFocusGained(java.awt.event.FocusEvent evt) {

        if(tb_Username.getText().equals("Enter a username")){
            tb_Username.setText("");
        }

    }

    private void tb_UsernameFocusLost(java.awt.event.FocusEvent evt) {

        if(tb_Username.getText().length() == 0){
            tb_Username.setText("Enter a username");
        }else{
            try {
                ConnectionJDBC gestor = new ConnectionJDBC();
                int count = gestor.checkUsername(tb_Username.getText());
                if (count == 0 ){
                    lbl_UserMissing.setVisible(true);
                    correctUser = false;
                }else{
                    lbl_UserMissing.setVisible(false);
                    correctUser = true;
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }

    }

    private void tb_UsernameKeyPressed(java.awt.event.KeyEvent evt) {

        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            tb_Password.requestFocus();
        }
    }

    private void tb_PasswordFocusGained(java.awt.event.FocusEvent evt) {

        if(tb_Password.getText().equals("......")){
            tb_Password.setText("");
        }

    }

    private void tb_PasswordFocusLost(java.awt.event.FocusEvent evt) {

        if(tb_Password.getText().length() == 0){
            tb_Password.setText("......");
        }else{
            try {
                ConnectionJDBC gestor = new ConnectionJDBC();
                int                          count                          =
gestor.checkCredentials(tb_Username.getText(),tb_Password.getText());
                if (count == 0 ){
                    lbl_InvalidPass.setVisible(true);
                    correctPassword = false;
                }else{
                    lbl_InvalidPass.setVisible(false);
                    correctPassword = true;
                    chosenAccount        =        gestor.loadID(tb_Username.getText(),
tb_Password.getText());
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }

    }

    private void tb_PasswordKeyPressed(java.awt.event.KeyEvent evt) {

        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            if(tb_Password.getText().length() == 0){
            tb_Password.setText("......");
        }else{
            try {
                ConnectionJDBC gestor = new ConnectionJDBC();
                int count =
gestor.checkCredentials(tb_Username.getText(),tb_Password.getText());
                if (count == 0 ){
                    lbl_InvalidPass.setVisible(true);
                    correctPassword = false;
                }else{
                    lbl_InvalidPass.setVisible(false);
                    correctPassword = true;
                    chosenAccount = gestor.loadID(tb_Username.getText(),
tb_Password.getText());
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        bt_SignIn.doClick();
        }
    }

    private void tb_UsernameActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void bt_SignInActionPerformed(java.awt.event.ActionEvent evt) {

        if(correctUser && correctPassword){
            try {
                PasswordManagerScreen pms;
                pms = new PasswordManagerScreen(chosenAccount);
                pms.setLocationRelativeTo(null);
                pms.setVisible(true);
                tb_Username.setText("");
                tb_Password.setText("");
            } catch (NoSuchAlgorithmException | InvalidKeyException |
NoSuchPaddingException | IllegalBlockSizeException | IOException |
BadPaddingException | SQLException | InvalidKeySpecException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    private void bt_SignInKeyPressed(java.awt.event.KeyEvent evt) {
        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            if(correctUser && correctPassword){
            try {
                PasswordManagerScreen pms = new
PasswordManagerScreen(chosenAccount);
                pms.setLocationRelativeTo(null);
                pms.setVisible(true);
                tb_Username.setText("");
                tb_Password.setText("");
            } catch (SQLException | NoSuchAlgorithmException | InvalidKeyException |
NoSuchPaddingException | IllegalBlockSizeException | IOException |
BadPaddingException | InvalidKeySpecException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        }
    }
```

```
    // Variables declaration - do not modify
    private javax.swing.JButton bt_SignIn;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JLabel lbl_BluePass;
    private javax.swing.JLabel lbl_BlueUser;
    private javax.swing.JLabel lbl_InvalidPass;
    private javax.swing.JLabel lbl_LoginImage;
    private javax.swing.JLabel lbl_UserMissing;
    private javax.swing.JPasswordField tb_Password;
    private javax.swing.JTextField tb_Username;
    // End of variables declaration
}
```

PasswordManagerScreen.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.swing.JOptionPane;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import models.Password;
import security.Decrypting;



public class PasswordManagerScreen extends javax.swing.JFrame {

    ArrayList<Password> passwordList = new ArrayList<>();
    ArrayList<Password> tempList = new ArrayList<>();
    Password chosenPassword;
    int selection = -1;
    final int ID;


    public PasswordManagerScreen(int id) throws SQLException,
NoSuchAlgorithmException, InvalidKeyException, NoSuchPaddingException,
IllegalBlockSizeException, IOException, BadPaddingException,
InvalidKeySpecException {
        initComponents();
        ID = id;
        loadData();
        loadTable();
        setTextChangeListener();
    }
    private void loadData() throws SQLException, InvalidKeyException,
IllegalBlockSizeException, IllegalBlockSizeException, BadPaddingException,
IOException, NoSuchAlgorithmException, InvalidKeySpecException,
NoSuchPaddingException{
        ConnectionJDBC gestor = new ConnectionJDBC();
        //cargar todos los passwords bajo el id
        passwordList = gestor.loadPasswords(ID);
        for (Password p : passwordList){
            p.setPass(Decrypting.runDecryption(p.getPid()));
        }
        tempList = passwordList;     //We want to set it equal on the constructor only.

     }

    private void loadTable(){
         String matris[][] = new String[passwordList.size()][4];
         for(int i = 0;i < passwordList.size();i++){
             matris[i][0]= Integer.toString(i+1);
             matris[i][1]=passwordList.get(i).getPassName();
             matris[i][2]= new String (passwordList.get(i).getPass());
             matris[i][3]=passwordList.get(i).getPassDescription();
         }
         table_Password.setModel(new javax.swing.table.DefaultTableModel(matris,new
String[]{"Number",
             "Name", "Password","Description"}));
     }

    public void searchPassword(){
        String hint = txt_SearchPass.getText();
        tempList = new ArrayList<>();

        for(Password pass:passwordList){
            if(pass.getPassName().toLowerCase().contains(hint.toLowerCase())){
               tempList.add(pass);
            }
        }
        if(tempList.size()>0){
        String matris[][] = new String[tempList.size()][4];
            for(int i = 0;i<=tempList.size()-1;i++){
                matris[i][0]= Integer.toString(i+1);
                matris[i][1]=tempList.get(i).getPassName();
                matris[i][2]= new String(tempList.get(i).getPass());
                matris[i][3]=tempList.get(i).getPassDescription();
                table_Password.setModel(new
javax.swing.table.DefaultTableModel(matris,new String[]{"Number",
                "Name", "Password","Description"}));
            }
        }else{
            String matris[][] = new String[1][4];
                matris[0][0]= Integer.toString(0);
                matris[0][1]= "no result";
                matris[0][2]= "no result";
                matris[0][3]= "no result";
                table_Password.setModel(new
javax.swing.table.DefaultTableModel(matris,new String[]{"Number",
                "Name", "Password","Description"}));
        }
    }

    public void setTextChangeListener(){



        txt_SearchPass.getDocument().addDocumentListener(new DocumentListener() {
```

```
        @Override
        public void changedUpdate(DocumentEvent e) {
            searchPassword();

        }
        @Override
        public void removeUpdate(DocumentEvent e) {
            searchPassword();

        }
        @Override
        public void insertUpdate(DocumentEvent e) {
          searchPassword();

        }
    });
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        btn_Refresh = new javax.swing.JButton();
        bt_UpdatePass = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        table_Password = new javax.swing.JTable();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        bt_DeletePass = new javax.swing.JButton();
        txt_SearchPass = new javax.swing.JTextField();
        bt_NewPass = new javax.swing.JButton();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        item_NewPass = new javax.swing.JMenuItem();
        btn_UpdatePass = new javax.swing.JMenuItem();
        item_DeletePass = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        jMenuItem4 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Password Locker");
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        setMinimumSize(new java.awt.Dimension(1200, 600));
        setResizable(false);
        addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                formFocusGained(evt);
            }
        });
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setText("Search by name:");
        getContentPane().add(jLabel1,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 60, 180, 20));

        btn_Refresh.setBackground(new java.awt.Color(50, 136, 203));
        btn_Refresh.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_Refresh.setForeground(new java.awt.Color(0, 0, 102));
        btn_Refresh.setText("Refresh ");
        btn_Refresh.setBorder(null);
        btn_Refresh.setBorderPainted(false);
        btn_Refresh.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_Refresh.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_RefreshActionPerformed(evt);
            }
        });
        getContentPane().add(btn_Refresh,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(1110, 80, 60, 30));

        bt_UpdatePass.setBackground(new java.awt.Color(50, 136, 203));
        bt_UpdatePass.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        bt_UpdatePass.setForeground(new java.awt.Color(0, 0, 102));
        bt_UpdatePass.setText("Update");
        bt_UpdatePass.setBorder(null);
        bt_UpdatePass.setBorderPainted(false);
        bt_UpdatePass.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        bt_UpdatePass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bt_UpdatePassActionPerformed(evt);
            }
        });
        getContentPane().add(bt_UpdatePass,                                  new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 250, 100, 60));

        table_Password.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(table_Password);

        getContentPane().add(jScrollPane1,                                   new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 117, 960, 350));

        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/main.png")));        //
NOI18N
        getContentPane().add(jLabel2,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, -1, 100));

        jLabel3.setFont(new java.awt.Font("Microsoft YaHei UI Light", 1, 18)); // NOI18N
        jLabel3.setText("Introduce all the passwords that you want to save in your
locker");
        getContentPane().add(jLabel3,                                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 30, 940, -1));

        bt_DeletePass.setBackground(new java.awt.Color(50, 136, 203));
        bt_DeletePass.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        bt_DeletePass.setForeground(new java.awt.Color(0, 0, 102));
        bt_DeletePass.setText("Delete");
        bt_DeletePass.setBorder(null);
        bt_DeletePass.setBorderPainted(false);
        bt_DeletePass.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        bt_DeletePass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bt_DeletePassActionPerformed(evt);
            }
        });
        getContentPane().add(bt_DeletePass,                                  new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 320, 100, 60));
        getContentPane().add(txt_SearchPass,                                 new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 80, 230, -1));

        bt_NewPass.setBackground(new java.awt.Color(50, 136, 203));
        bt_NewPass.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        bt_NewPass.setForeground(new java.awt.Color(0, 0, 102));
        bt_NewPass.setText("New");
        bt_NewPass.setBorder(null);
        bt_NewPass.setBorderPainted(false);
        bt_NewPass.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        bt_NewPass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bt_NewPassActionPerformed(evt);
            }
        });
        getContentPane().add(bt_NewPass,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 180, 100, 60));

        jMenu1.setText("Options");

        item_NewPass.setText("New Password");
        item_NewPass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                item_NewPassActionPerformed(evt);
            }
        });
        jMenu1.add(item_NewPass);

        btn_UpdatePass.setText("Update Password");
        btn_UpdatePass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_UpdatePassActionPerformed(evt);
            }
        });
        jMenu1.add(btn_UpdatePass);

        item_DeletePass.setText("Delete Password");
        item_DeletePass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                item_DeletePassActionPerformed(evt);
            }
        });
        jMenu1.add(item_DeletePass);
        jMenu1.add(jSeparator1);

        jMenuItem4.setText("Log Out");
        jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem4ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem4);

        jMenuBar1.add(jMenu1);

        setJMenuBar(jMenuBar1);

        pack();
    }// </editor-fold>

    private void btn_RefreshActionPerformed(java.awt.event.ActionEvent evt) {
       try {
            loadData();
            loadTable();
            selection = -1;
        } catch (SQLException | InvalidKeyException | IllegalBlockSizeException |
BadPaddingException | IOException | NoSuchAlgorithmException |
InvalidKeySpecException | NoSuchPaddingException ex) {

Logger.getLogger(PasswordManagerScreen.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

    private void bt_UpdatePassActionPerformed(java.awt.event.ActionEvent evt) {
       selection = table_Password.getSelectedRow();
       if(selection != -1){
       selection = table_Password.getSelectedRow();
       UpdatePassword up = new UpdatePassword(passwordList.get(selection));
       up.setLocationRelativeTo(null);
       up.setVisible(true);
       }
    }

    private void bt_DeletePassActionPerformed(java.awt.event.ActionEvent evt) {
        selection = table_Password.getSelectedRow();
        if(selection != -1){
        ConnectionJDBC gestor = new ConnectionJDBC();
        int s = tempList.get(table_Password.getSelectedRow()).getPid();
        gestor.deletePassword(s);
        passwordList.remove(table_Password.getSelectedRow());
        JOptionPane.showMessageDialog(null,"Password
deleted","Confirmation",JOptionPane.PLAIN_MESSAGE);
        loadTable(); //refresh screen
        }
    }

    private void formFocusGained(java.awt.event.FocusEvent evt) {
        try {
            loadData();
            loadTable();
        } catch (SQLException | InvalidKeyException | IllegalBlockSizeException |
BadPaddingException | IOException | NoSuchAlgorithmException |
InvalidKeySpecException | NoSuchPaddingException ex) {

Logger.getLogger(PasswordManagerScreen.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

    private void bt_NewPassActionPerformed(java.awt.event.ActionEvent evt) {
       NewPassword n = new NewPassword(passwordList, ID);
       n.setLocationRelativeTo(null);
       n.setVisible(true);
    }

    private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    private void item_NewPassActionPerformed(java.awt.event.ActionEvent evt) {
        bt_NewPass.doClick();
    }

    private void btn_UpdatePassActionPerformed(java.awt.event.ActionEvent evt) {
        bt_UpdatePass.doClick();
    }

    private void item_DeletePassActionPerformed(java.awt.event.ActionEvent evt) {
        bt_DeletePass.doClick();
    }


    // Variables declaration - do not modify
    private javax.swing.JButton bt_DeletePass;
    private javax.swing.JButton bt_NewPass;
    private javax.swing.JButton bt_UpdatePass;
    private javax.swing.JButton btn_Refresh;
    private javax.swing.JMenuItem btn_UpdatePass;
    private javax.swing.JMenuItem item_DeletePass;
    private javax.swing.JMenuItem item_NewPass;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JPopupMenu.Separator jSeparator1;
    private javax.swing.JTable table_Password;
    private javax.swing.JTextField txt_SearchPass;
    // End of variables declaration
}
```

UpdatePassword.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.swing.JOptionPane;
import models.Password;


public class UpdatePassword extends javax.swing.JFrame {
    Password selectedPassword;

    public UpdatePassword(Password p ) {
        initComponents();
        selectedPassword = p;
        txt_UpdateName.setText(selectedPassword.getPassName());
        txt_UpdatePass.setText(new String(selectedPassword.getPass()));
        txt_UpdateDescription.setText(selectedPassword.getPassDescription());
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txt_UpdateName = new javax.swing.JTextField();
        txt_UpdatePass = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        txt_UpdateDescription = new javax.swing.JTextArea();
        btn_UpdateCancel = new javax.swing.JButton();
        btn_UpdateAccept = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setMinimumSize(new java.awt.Dimension(430, 300));
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setText("Description:");
        getContentPane().add(jLabel1,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 150, -1, 20));

        jLabel2.setText("Name:");
        getContentPane().add(jLabel2,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 70, -1, 20));
        getContentPane().add(txt_UpdateName,                             new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 70, 231, 20));
        getContentPane().add(txt_UpdatePass,                             new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 231, 20));

        jLabel3.setText("Password:");
        getContentPane().add(jLabel3,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 110, -1, 20));

        txt_UpdateDescription.setColumns(20);
        txt_UpdateDescription.setRows(5);
        jScrollPane1.setViewportView(txt_UpdateDescription);

        getContentPane().add(jScrollPane1,                               new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 230, -1));

        btn_UpdateCancel.setBackground(new java.awt.Color(50, 136, 203));
        btn_UpdateCancel.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_UpdateCancel.setForeground(new java.awt.Color(0, 0, 102));
        btn_UpdateCancel.setText("Cancel");
        btn_UpdateCancel.setBorder(null);
        btn_UpdateCancel.setBorderPainted(false);
        btn_UpdateCancel.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_UpdateCancel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_UpdateCancelActionPerformed(evt);
            }
        });
        getContentPane().add(btn_UpdateCancel,                           new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 260, 110, 30));

        btn_UpdateAccept.setBackground(new java.awt.Color(50, 136, 203));
        btn_UpdateAccept.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_UpdateAccept.setForeground(new java.awt.Color(0, 0, 102));
        btn_UpdateAccept.setText("Update");
        btn_UpdateAccept.setBorder(null);
        btn_UpdateAccept.setBorderPainted(false);
        btn_UpdateAccept.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_UpdateAccept.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_UpdateAcceptActionPerformed(evt);
            }
        });
        getContentPane().add(btn_UpdateAccept,                           new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 260, 110, 30));

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jLabel4.setText("Update Password");
        getContentPane().add(jLabel4,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 20, 200, 30));

        pack();
    }// </editor-fold>

    private void btn_UpdateCancelActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    private void btn_UpdateAcceptActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            ConnectionJDBC gestor = new ConnectionJDBC();
            Password pu = new Password();
            pu.setPassName(txt_UpdateName.getText());
            pu.setPass(txt_UpdatePass.getText().getBytes());
            pu.setPassDescription(txt_UpdateDescription.getText());
            gestor.updatePassword(pu,selectedPassword.getPid());
            JOptionPane.showMessageDialog(null,"Password
updated","Confirmation",JOptionPane.PLAIN_MESSAGE);
            this.dispose();
        } catch (IOException | NoSuchAlgorithmException | InvalidKeySpecException |
IllegalBlockSizeException | NoSuchPaddingException | InvalidKeyException |
BadPaddingException ex) {
            Logger.getLogger(UpdatePassword.class.getName()).log(Level.SEVERE, null,
ex);
        }

    }



    // Variables declaration - do not modify
    private javax.swing.JButton btn_UpdateAccept;
    private javax.swing.JButton btn_UpdateCancel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txt_UpdateDescription;
    private javax.swing.JTextField txt_UpdateName;
    private javax.swing.JTextField txt_UpdatePass;
    // End of variables declaration
}
```

NewPassword.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.swing.JOptionPane;
import models.Password;



public class NewPassword extends javax.swing.JFrame {
    ArrayList<Password>  passwordList  =  new  ArrayList<>();  //to  check  it  doesn't
already exist
    final int ID;
    byte[] secretKey;

    public NewPassword(ArrayList<Password> list, int id) {
        initComponents();
        passwordList = list;
        ID = id;
    }


    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txt_NewName = new javax.swing.JTextField();
        txt_NewPass = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        txt_NewDescription = new javax.swing.JTextArea();
        btn_NewCancel = new javax.swing.JButton();
        btn_NewAccept = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setText("Description:");
        getContentPane().add(jLabel1,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 140, -1, 20));

        jLabel2.setText("Write a related name:");
        getContentPane().add(jLabel2,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, -1, 20));
        getContentPane().add(txt_NewName,                                      new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 80, 231, 20));
        getContentPane().add(txt_NewPass,                                      new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 231, 20));

        jLabel3.setText("Enter the password:");
        getContentPane().add(jLabel3,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 110, -1, 20));

        txt_NewDescription.setColumns(20);
        txt_NewDescription.setRows(5);
        jScrollPane1.setViewportView(txt_NewDescription);

        getContentPane().add(jScrollPane1,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 230, -1));

        btn_NewCancel.setBackground(new java.awt.Color(50, 136, 203));
        btn_NewCancel.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_NewCancel.setForeground(new java.awt.Color(0, 0, 102));
        btn_NewCancel.setText("Cancel");
        btn_NewCancel.setBorder(null);
        btn_NewCancel.setBorderPainted(false);
        btn_NewCancel.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_NewCancel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_NewCancelActionPerformed(evt);
            }
        });
        getContentPane().add(btn_NewCancel,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 260, 110, 30));

        btn_NewAccept.setBackground(new java.awt.Color(50, 136, 203));
        btn_NewAccept.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_NewAccept.setForeground(new java.awt.Color(0, 0, 102));
        btn_NewAccept.setText("Add Password");
        btn_NewAccept.setBorder(null);
        btn_NewAccept.setBorderPainted(false);
        btn_NewAccept.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_NewAccept.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_NewAcceptActionPerformed(evt);
            }
        });
        getContentPane().add(btn_NewAccept,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 260, 110, 30));

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jLabel4.setText("New password");
        getContentPane().add(jLabel4,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 30, 130, 20));

        pack();
    }// </editor-fold>

    private void btn_NewCancelActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    private void btn_NewAcceptActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            ConnectionJDBC gestor = new ConnectionJDBC();
            gestor.insertPassword(new
Password(txt_NewName.getText(),txt_NewPass.getText().getBytes(),txt_NewDescript
ion.getText()), ID);
            JOptionPane.showMessageDialog(null,"Password
inserted","Confirmation",JOptionPane.PLAIN_MESSAGE);
            this.dispose();
        } catch  (IOException | NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException    |    IllegalBlockSizeException    |    BadPaddingException    |
InvalidKeySpecException ex) {
            Logger.getLogger(NewPassword.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
    // Variables declaration - do not modify
    private javax.swing.JButton btn_NewAccept;
    private javax.swing.JButton btn_NewCancel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txt_NewDescription;
    private javax.swing.JTextField txt_NewName;
    private javax.swing.JTextField txt_NewPass;
    // End of variables declaration
}
```

ConnectionJDBC.java

```
package databaseConection;

import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.swing.JOptionPane;
import models.Estructura;
import models.Password;
import security.Hashing;
import security.Encrypting;


public class ConnectionJDBC {

    Connection conn;

private void conectar() throws SQLException{
        String url = "jdbc:mysql://localhost:3306/passwordlocker";
        String user = "root";
        String pass = "";
        conn = DriverManager.getConnection(url, user, pass);
    }

private void desconectar() throws SQLException{
        if (conn != null){ conn.close();}
    }

public Estructura loadPassword(int id) throws SQLException{
    conectar();
    byte[] s = null;
    byte[] secret = null;
    String query = "SELECT pass, secret FROM t_passwords \n" +
                "WHERE idp = '"+id+"'";
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        s = rs.getBytes("pass");
        secret = rs.getBytes("secret");

    }
    Estructura e = new Estructura(s,secret);

    desconectar();
    return e;
}

public ArrayList<Password> loadPasswords(int id) throws SQLException{
    conectar();
    ArrayList<Password> temporalList = new ArrayList<>();

    String query = "SELECT * FROM t_passwords\n" +
                "WHERE ida_fk = '"+id+"'";

    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        int pid = rs.getInt("idp");
        int aid = rs.getInt("ida_fk");
        String name = rs.getString("passname");
        byte[] pass = rs.getBytes("pass");
        byte[] secret = rs.getBytes("secret");
        String desc = rs.getString("description");

        Password p = new Password(pid,aid,name,pass,secret,desc);
        temporalList.add(p);
    }
    desconectar();
    return temporalList;
}

public int checkUsername(String userInput) throws SQLException{
    conectar();
    int numberReturned = 0;
    String query = "SELECT count(username) AS total FROM t_accounts \n" +
                "WHERE username = '"+Hashing.getHash(userInput.getBytes())+"'";

    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        numberReturned = rs.getInt("total");
    }
    desconectar();
    return numberReturned;
}

public int checkCredentials(String user, String pass) throws SQLException{
    conectar();
    int numberReturned = 0;
    String query = "SELECT count(username) AS total FROM t_accounts \n" +
                "WHERE    username    =    '"+Hashing.getHash(user.getBytes())+"'    AND
masterkey = '"+Hashing.getHash(pass.getBytes())+"'";

    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);
    while(rs.next()){
        numberReturned = rs.getInt("total");
    }
    desconectar();
    return numberReturned;
}

public int loadID(String user, String pass) throws SQLException{
    conectar();
    int numberReturned = 0;
    String query = "SELECT ida FROM t_accounts \n" +
                "WHERE username = '"+Hashing.getHash(user.getBytes())+"' AND
masterkey = '"+Hashing.getHash(pass.getBytes())+"'";
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        numberReturned = rs.getInt("ida");
    }
    desconectar();
    return numberReturned;
}

public void insertPassword(Password p, int id) throws IOException,
NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
IllegalBlockSizeException, BadPaddingException, InvalidKeySpecException{
   try{
            Estructura temp = Encrypting.runEncrypt(p.getPass());
            p.setPass(temp.getPass());
            p.setSecretkey(temp.getSecret());



            conectar();
            String insert = "INSERT INTO t_passwords (ida_fk,
passname,pass,secret,description) VALUES(?,?,?,?,?)";
            PreparedStatement pst = conn.prepareStatement(insert);
            pst.setInt(1,id);
            pst.setString(2, p.getPassName());
            pst.setBytes(3, p.getPass());
            pst.setBytes(4, p.getSecretkey());
            pst.setString(5, p.getPassDescription());
            pst.execute();
            desconectar();
        }catch(SQLException e){
            JOptionPane.showMessageDialog(null,"Error","No pasa
nada",JOptionPane.ERROR);
        }
}

public void updatePassword(Password p, int pid) throws IOException,
NoSuchAlgorithmException, InvalidKeySpecException, IllegalBlockSizeException,
NoSuchPaddingException, InvalidKeyException, BadPaddingException{
     try{
            Estructura temp = Encrypting.runEncrypt(p.getPass());
            p.setPass(temp.getPass());
            p.setSecretkey(temp.getSecret());

            String update = ("UPDATE t_passwords SET passname = ? , pass = ? , secret =
? , description = ? WHERE idp= ?");
            conectar();
            PreparedStatement pst = conn.prepareStatement(update);
            pst.setString(1, p.getPassName());
            pst.setBytes(2,p.getPass());
            pst.setBytes(3,p.getSecretkey());
            pst.setString(4,p.getPassDescription());
            pst.setInt(5, pid);
            pst.executeUpdate();
            desconectar();
         }catch(SQLException e){
           System.out.println(e.getMessage());
        }
}

public void deletePassword(int pid){
    try{

        conectar();
        String sql = "DELETE FROM t_passwords WHERE idp = ? ";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, pid);
        pst.executeUpdate();
        desconectar();
    }catch(SQLException e){
          System.out.println(e.getMessage());
    }
}

}
```

Account.java

```
package models;

public class Account {
    private String username;
    private String password;
    private String question1;
    private String quesiotn2;
    private String question3;

    public Account(){

    }

    public Account(String username, String password) {
        this.username = username;
        this.password = password;
    }

    public Account(String username, String password, String question1, String
quesiotn2, String question3) {
        this.username = username;
        this.password = password;
        this.question1 = question1;
        this.quesiotn2 = quesiotn2;
        this.question3 = question3;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getQuestion1() {
        return question1;
    }

    public void setQuestion1(String question1) {
        this.question1 = question1;
    }

    public String getQuesiotn2() {
        return quesiotn2;
    }

    public void setQuesiotn2(String quesiotn2) {
        this.quesiotn2 = quesiotn2;
    }

    public String getQuestion3() {
        return question3;
    }

    public void setQuestion3(String question3) {
        this.question3 = question3;
    }

    @Override
    public String toString() {
        return "Account{" + "username=" + username + ", password=" + password + ", question1=" + question1 + ", quesiotn2=" + quesiotn2 + ", question3=" + question3 + '}';
    }
}
```

Estructura.java

```
package models;


public class Estructura {
    private byte[] pass;
    private byte[] secret;

    public Estructura(byte[] word, byte[] secret) {
        this.pass = word;
        this.secret = secret;
    }

    public byte[] getPass() {
        return pass;
    }

    public void setPass(byte[] word) {
        this.pass = word;
    }

    public byte[] getSecret() {
        return secret;
    }

    public void setSecret(byte[] secret) {
        this.secret = secret;
    }
}
```

Password.java

```
package models;

public class Password {
    private int pid;
    private int aid;
    private String passName;
    private byte[] pass;
    private byte[] secretkey;
    private String passDescription;

    public Password() {
    }

    public Password(String passName, byte[] passKey,byte[] key, String passDescription) {
        this.passName = passName;
        this.pass = passKey;
        this.secretkey = key;
        this.passDescription = passDescription;
    }

    public Password( String passName, byte[] passKey, String passDescription) {
        this.passName = passName;
        this.pass = passKey;
        this.passDescription = passDescription;
    }

    public Password(int pid, int aid, String passName, byte[] pass, byte[] secretkey, String passDescription) {
        this.pid = pid;
        this.aid = aid;
        this.passName = passName;
        this.pass = pass;
        this.secretkey = secretkey;
        this.passDescription = passDescription;
    }



    public int getPid() {
        return pid;
    }

    public void setPid(int pid) {
        this.pid = pid;
    }

    public int getAid() {
        return aid;
    }

    public void setAid(int aid) {
        this.aid = aid;
    }

    public String getPassName() {
        return passName;
    }

    public void setPassName(String passName) {
        this.passName = passName;
    }

    public byte[] getPass() {
        return pass;
    }

    public void setPass(byte[] pass) {
        this.pass = pass;
    }

    public byte[] getSecretkey() {
        return secretkey;
    }

    public void setSecretkey(byte[] secretkey) {
        this.secretkey = secretkey;
    }



    public String getPassDescription() {
        return passDescription;
    }

    public void setPassDescription(String passDescription) {
        this.passDescription = passDescription;
    }

    @Override
    public String toString() {
        return "Password{" + "pid=" + pid + ", aid=" + aid + ", passName=" + passName + ",
pass=" + pass + ", secretkey=" + secretkey + ", passDescription=" + passDescription +
'}';
    }
}
```

Decrypting.java

```
package security;

import databaseConection.ConnectionJDBC;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;
import java.security.spec.PKCS8EncodedKeySpec;
import java.sql.SQLException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import models.Estructura;


public class Decrypting {

        static KeyPairGenerator keygen;
        static KeyPair kp;
        static PrivateKey privatek;
        static int selectedRecord;
        static Cipher cipher;

    public static byte[] runDecryption(int number) throws
NoSuchAlgorithmException, InvalidKeySpecException, IOException,
NoSuchPaddingException, InvalidKeyException, IllegalBlockSizeException,
BadPaddingException, SQLException {
        selectedRecord = number;
        //instanciate objects
        keygen = KeyPairGenerator.getInstance("RSA");
        kp = keygen.generateKeyPair();
        privatek = kp.getPrivate();
        cipher = Cipher.getInstance("RSA");

        //obtain data from database
        ConnectionJDBC c = new ConnectionJDBC();
        Estructura vehiculo;
        vehiculo = c.loadPassword(selectedRecord);

        //creamos el keyfactory para descrifrar la clave guardada
        KeyFactory kf = KeyFactory.getInstance("RSA");
        KeySpec ks = new PKCS8EncodedKeySpec(vehiculo.getSecret());
        PrivateKey keyFromBytes = kf.generatePrivate(ks);
        privatek = keyFromBytes;

        //decrypt message
        byte[] devolucion = vehiculo.getPass();
        cipher.init(Cipher.DECRYPT_MODE, privatek);
        byte[] desencriptar = cipher.doFinal(devolucion);
        return desencriptar;
            }
}
```

Encrypting.java

```
package security;

import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.InvalidKeySpecException;
import java.sql.SQLException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import models.Estructura;

public class Encrypting {
      static KeyPairGenerator keygen;
       static KeyPair kp;
       static PublicKey puKey;
       static PrivateKey  pKey;
       static Cipher cipher;


    public static Estructura runEncrypt(byte[] mensaje) throws IOException, NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException, IllegalBlockSizeException, BadPaddingException, SQLException, InvalidKeySpecException {
        // Generamos el par de claves.
        // Generamos el par de claves.
                keygen = KeyPairGenerator.getInstance("RSA");
                kp = keygen.generateKeyPair();
                puKey = kp.getPublic();
                pKey = kp.getPrivate();
                //invocamos al metodo de guardar la clave

                //creamos el cipher
                cipher = Cipher.getInstance("RSA");

        cipher.init(Cipher.ENCRYPT_MODE, puKey);
                byte[] encriptado = cipher.doFinal(mensaje);

                //guardamos el mensaje en el objeto
        byte[] publicKeyBytes = pKey.getEncoded();
        Estructura vehicle = new Estructura(encriptado,publicKeyBytes);
        return vehicle;
        }
}
```

Hashing.java

```
package security;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import javax.xml.bind.DatatypeConverter;

public class Hashing {

    public static String getHash(byte[] input){
        String hash = "";
        try{
            MessageDigest md = MessageDigest.getInstance("Sha-512");
            md.update(input);
            byte[] output = md.digest();
            hash = DatatypeConverter.printHexBinary(output);

        }catch(NoSuchAlgorithmException e){
            System.out.println("Hashing went wrong");
        }
        return hash;
    }
}
```

Source Code Password Locker Administrador

UserLogin.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.awt.event.KeyEvent;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;


public class UserLogin extends javax.swing.JFrame {
    boolean correctUser = false;
    boolean correctPassword = false;
    boolean login = false;

    int chosenAccount;          //para test

    public UserLogin() {
        initComponents();
        lbl_UserMissing.setVisible(false);
        lbl_InvalidPass.setVisible(false);

    }

    public static void main(String[] args) {
        UserLogin ul = new UserLogin();
        ul.setLocationRelativeTo(null);
        ul.setVisible(true);
    }


    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lbl_LoginImage = new javax.swing.JLabel();
        lbl_BlueUser = new javax.swing.JLabel();
        lbl_UserMissing = new javax.swing.JLabel();
        tb_Username = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        lbl_BluePass = new javax.swing.JLabel();
        lbl_InvalidPass = new javax.swing.JLabel();
        tb_Password = new javax.swing.JPasswordField();
        jLabel5 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        bt_SignIn = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setLocationByPlatform(true);
        setMinimumSize(new java.awt.Dimension(400, 600));
        setPreferredSize(new java.awt.Dimension(400, 600));
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        lbl_LoginImage.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/logo.png"))); //
NOI18N
        getContentPane().add(lbl_LoginImage, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 70, 160, 170));

        lbl_BlueUser.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_BlueUser.setForeground(new java.awt.Color(50, 136, 203));
        lbl_BlueUser.setText("Username:");
        getContentPane().add(lbl_BlueUser, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 290, 100, 20));

        lbl_UserMissing.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_UserMissing.setForeground(new java.awt.Color(255, 51, 51));
        lbl_UserMissing.setText("Does not exist");
        getContentPane().add(lbl_UserMissing, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 290, -1, -1));

        tb_Username.setBackground(new java.awt.Color(240, 240, 240));
        tb_Username.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        tb_Username.setForeground(new java.awt.Color(102, 102, 102));
        tb_Username.setText("Enter a username");
        tb_Username.setToolTipText("");
        tb_Username.setBorder(null);
        tb_Username.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                tb_UsernameFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                tb_UsernameFocusLost(evt);
            }
        });
        tb_Username.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tb_UsernameActionPerformed(evt);
            }
        });
        tb_Username.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                tb_UsernameKeyPressed(evt);
            }
        });
        getContentPane().add(tb_Username, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 320, 210, 40));

        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/user.png"))); //
NOI18N
        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 330, 30, 30));

        lbl_BluePass.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_BluePass.setForeground(new java.awt.Color(50, 136, 203));
        lbl_BluePass.setText("Password:");
        getContentPane().add(lbl_BluePass, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 410, -1, -1));

        lbl_InvalidPass.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_InvalidPass.setForeground(new java.awt.Color(255, 51, 51));
        lbl_InvalidPass.setText("Invalid");
        getContentPane().add(lbl_InvalidPass, new
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 410, -1, -1));

        tb_Password.setBackground(new java.awt.Color(240, 240, 240));
        tb_Password.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        tb_Password.setForeground(new java.awt.Color(102, 102, 102));
        tb_Password.setText("......");
        tb_Password.setBorder(null);
        tb_Password.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                tb_PasswordFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                tb_PasswordFocusLost(evt);
            }
        });
        tb_Password.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                tb_PasswordKeyPressed(evt);
            }
        });
        getContentPane().add(tb_Password, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 440, 150, 30));

        jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/frames/images/password.png"))); //
NOI18N
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 440, -1, 30));
        getContentPane().add(jSeparator1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 480, 180, 20));
        getContentPane().add(jSeparator2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 370, 180, 20));

        bt_SignIn.setBackground(new java.awt.Color(50, 136, 203));
        bt_SignIn.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        bt_SignIn.setForeground(new java.awt.Color(255, 255, 255));
        bt_SignIn.setText("Sign in");
        bt_SignIn.setBorder(null);
        bt_SignIn.setBorderPainted(false);
        bt_SignIn.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        bt_SignIn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bt_SignInActionPerformed(evt);
            }
        });
        bt_SignIn.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                bt_SignInKeyPressed(evt);
            }
        });
        getContentPane().add(bt_SignIn, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 500, 100, 60));

        pack();
    }// </editor-fold>

    private void tb_UsernameFocusGained(java.awt.event.FocusEvent evt) {

        if(tb_Username.getText().equals("Enter a username")){
            tb_Username.setText("");
        }

    }

    private void tb_UsernameFocusLost(java.awt.event.FocusEvent evt) {

        if(tb_Username.getText().length() == 0){
            tb_Username.setText("Enter a username");
        }else{
            try {
                ConnectionJDBC gestor = new ConnectionJDBC();
                int count = gestor.checkUsername(tb_Username.getText());
                if (count == 0 ){
                    lbl_UserMissing.setVisible(true);
                    correctUser = false;
                }else{
                    lbl_UserMissing.setVisible(false);
                    correctUser = true;
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }

    }

    private void tb_UsernameKeyPressed(java.awt.event.KeyEvent evt) {

        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            tb_Password.requestFocus();
        }
    }

    private void tb_PasswordFocusGained(java.awt.event.FocusEvent evt) {

        if(tb_Password.getText().equals("......")){
            tb_Password.setText("");
        }

    }

    private void tb_PasswordFocusLost(java.awt.event.FocusEvent evt) {

        if(tb_Password.getText().length() == 0){
            tb_Password.setText("......");
        }else{
            try {
                ConnectionJDBC gestor = new ConnectionJDBC();
                int count =
gestor.checkCredentials(tb_Username.getText(),tb_Password.getText());
                if (count == 0 ){
                    lbl_InvalidPass.setVisible(true);
                    correctPassword = false;
                }else{
                    lbl_InvalidPass.setVisible(false);
                    correctPassword = true;
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }

    }

    private void tb_PasswordKeyPressed(java.awt.event.KeyEvent evt) {

        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            if(tb_Password.getText().length() == 0){
            tb_Password.setText("......");
        }else{
            try {
```

```
                ConnectionJDBC gestor = new ConnectionJDBC();
                int count =
gestor.checkCredentials(tb_Username.getText(),tb_Password.getText());
                if (count == 0 ){
                    lbl_InvalidPass.setVisible(true);
                    correctPassword = false;
                }else{
                    lbl_InvalidPass.setVisible(false);
                    correctPassword = true;
                }
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        bt_SignIn.doClick();
        }
    }

    private void tb_UsernameActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void bt_SignInActionPerformed(java.awt.event.ActionEvent evt) {

        if(correctUser && correctPassword){
            try {
                ConnectionJDBC c = new ConnectionJDBC();
                chosenAccount = c.loadChosenAccount(tb_Username.getText());
                AdminPasswordManager pms = new
AdminPasswordManager(chosenAccount);
                pms.setLocationRelativeTo(null);
                pms.setVisible(true);
                tb_Username.setText("");
                tb_Password.setText("");
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    private void bt_SignInKeyPressed(java.awt.event.KeyEvent evt) {
        if(evt.getKeyCode() == KeyEvent.VK_ENTER){
            if(correctUser && correctPassword){
            try {
                ConnectionJDBC c = new ConnectionJDBC();
                chosenAccount = c.loadChosenAccount(tb_Username.getText());
                AdminPasswordManager pms = new
AdminPasswordManager(chosenAccount);
                pms.setLocationRelativeTo(null);
                pms.setVisible(true);
                tb_Username.setText("");
                tb_Password.setText("");
            } catch (SQLException ex) {
                Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        }
    }



    // Variables declaration - do not modify
    private javax.swing.JButton bt_SignIn;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JLabel lbl_BluePass;
    private javax.swing.JLabel lbl_BlueUser;
    private javax.swing.JLabel lbl_InvalidPass;
    private javax.swing.JLabel lbl_LoginImage;
    private javax.swing.JLabel lbl_UserMissing;
    private javax.swing.JPasswordField tb_Password;
    private javax.swing.JTextField tb_Username;
    // End of variables declaration
}
```

AdminPasswordManager.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import models.Account;
import models.Admin;


public class AdminPasswordManager extends javax.swing.JFrame {
    ArrayList<Account> accountsList = new ArrayList<>();
    ArrayList<Account> tempList = new ArrayList<>();
    ArrayList<Admin> adminList = new ArrayList<>();
    int adminID;
    int selection = -1;


    public AdminPasswordManager(int id) throws SQLException {
        initComponents();
        adminID = id;
        loadData();
        loadTable();
        setTextListeners();

    }



    private void loadData() throws SQLException{
        ConnectionJDBC c = new ConnectionJDBC();
        accountsList = c.loadAccounts();
        tempList = accountsList;
        adminList = c.loadAdmins();
     }

    private void loadTable(){
         String matris[][] = new String[accountsList.size()][3];
         for(int i = 0;i < accountsList.size();i++){
             matris[i][0]= accountsList.get(i).getForename();
             matris[i][1]=accountsList.get(i).getSurname();
             matris[i][2]=accountsList.get(i).getDob();
         }
         table_Accounts.setModel(new  javax.swing.table.DefaultTableModel(matris,new
String[]{
             "Forename", "Surname","Date of Birth"}));
     }


    public void searchAccount(){
        String namehint = txt_SearchAccountName.getText();
        String surhint = txt_SearchAccountSurname.getText();
        String dobhint = txt_SearchAccountDOB.getText();

        tempList = new ArrayList<>();

        for(Account acc:accountsList){


if(acc.getForename().toLowerCase().contains(namehint.toLowerCase())||txt_SearchA
ccountName.getText().length()==0){

if(acc.getSurname().toLowerCase().contains(surhint.toLowerCase())||txt_SearchAcco
untSurname.getText().length()==0){

if(acc.getDob().toLowerCase().contains(dobhint.toLowerCase())||txt_SearchAccountD
OB.getText().length()==0){
                        tempList.add(acc);
                    }
                }
            }
        }

        if(tempList.size()>0){
        String matris[][] = new String[tempList.size()][3];
            for(int i = 0;i<=tempList.size()-1;i++){

                matris[i][0]= tempList.get(i).getForename();
                matris[i][1]=tempList.get(i).getSurname();
                matris[i][2]=tempList.get(i).getDob();
                table_Accounts.setModel(new
javax.swing.table.DefaultTableModel(matris,new String[]{
             "Forename", "Surname","Date of Birth"}));
            }
        }else{
            String matris[][] = new String[1][3];

                matris[0][0]= "no result";
                matris[0][1]= "no result";
                matris[0][2]= "no result";
                table_Accounts.setModel(new
javax.swing.table.DefaultTableModel(matris,new String[]{
             "Forename", "Surname","Date of Birth"}));

        }
    }

    public void setTextListeners(){
        setForenameTextChangeListener();
        setSurnameTextChangeListener();
        setDOBTextChangeListener();
    }

     public void setForenameTextChangeListener(){
        txt_SearchAccountName.getDocument().addDocumentListener(new
DocumentListener() {
        @Override
        public void changedUpdate(DocumentEvent e) {
            searchAccount();
        }
        @Override
        public void removeUpdate(DocumentEvent e) {
            searchAccount();

        }
        @Override
        public void insertUpdate(DocumentEvent e) {
          searchAccount();

        }
    });
    }

     public void setSurnameTextChangeListener(){
        txt_SearchAccountSurname.getDocument().addDocumentListener(new
DocumentListener() {
        @Override
        public void changedUpdate(DocumentEvent e) {
            searchAccount();
        }
        @Override
        public void removeUpdate(DocumentEvent e) {
        searchAccount();
        }
        @Override
        public void insertUpdate(DocumentEvent e) {
        searchAccount();
        }
    });
    }

     public void setDOBTextChangeListener(){
        txt_SearchAccountDOB.getDocument().addDocumentListener(new
DocumentListener() {
        @Override
        public void changedUpdate(DocumentEvent e) {
        searchAccount();
        }
        @Override
        public void removeUpdate(DocumentEvent e) {
        searchAccount();
        }
        @Override
        public void insertUpdate(DocumentEvent e) {
        searchAccount();
        }
    });
    }




    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        table_Accounts = new javax.swing.JTable();
        txt_SearchAccountDOB = new javax.swing.JTextField();
        txt_SearchAccountName = new javax.swing.JTextField();
        txt_SearchAccountSurname = new javax.swing.JTextField();
        btn_NewAccount = new javax.swing.JButton();
        lbl_SearchAccount = new javax.swing.JLabel();
```

```
        lbl_SearchAccount1 = new javax.swing.JLabel();
        lbl_SearchAccount2 = new javax.swing.JLabel();
        btn_ResetPassword = new javax.swing.JButton();
        btn_ClearFields = new javax.swing.JButton();
        btn_Refresh = new javax.swing.JButton();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        Item_newAccount = new javax.swing.JMenuItem();
        item_ResetPass = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        item_NewAdmin = new javax.swing.JMenuItem();
        item_changeAdminPass = new javax.swing.JMenuItem();
        jSeparator2 = new javax.swing.JPopupMenu.Separator();
        jMenuItem6 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        table_Accounts.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(table_Accounts);

        getContentPane().add(jScrollPane1,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 150, 780, 229));

        txt_SearchAccountDOB.addInputMethodListener(new
java.awt.event.InputMethodListener() {
            public void caretPositionChanged(java.awt.event.InputMethodEvent evt) {
            }
            public void inputMethodTextChanged(java.awt.event.InputMethodEvent evt) {
                txt_SearchAccountDOBInputMethodTextChanged(evt);
            }
        });
        getContentPane().add(txt_SearchAccountDOB,                                  new
org.netbeans.lib.awtextra.AbsoluteConstraints(560, 110, 180, -1));

        txt_SearchAccountName.addInputMethodListener(new
java.awt.event.InputMethodListener() {
            public void caretPositionChanged(java.awt.event.InputMethodEvent evt) {
            }
            public void inputMethodTextChanged(java.awt.event.InputMethodEvent evt) {
                txt_SearchAccountNameInputMethodTextChanged(evt);
            }
        });
        getContentPane().add(txt_SearchAccountName,                                 new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 110, 180, -1));

        txt_SearchAccountSurname.addInputMethodListener(new
java.awt.event.InputMethodListener() {
            public void caretPositionChanged(java.awt.event.InputMethodEvent evt) {
            }
            public void inputMethodTextChanged(java.awt.event.InputMethodEvent evt) {
                txt_SearchAccountSurnameInputMethodTextChanged(evt);
            }
        });
        getContentPane().add(txt_SearchAccountSurname,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(310, 110, 180, -1));

        btn_NewAccount.setBackground(new java.awt.Color(50, 136, 203));
        btn_NewAccount.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_NewAccount.setForeground(new java.awt.Color(0, 0, 102));
        btn_NewAccount.setText("New");
        btn_NewAccount.setBorder(null);
        btn_NewAccount.setBorderPainted(false);
        btn_NewAccount.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_NewAccount.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_NewAccountActionPerformed(evt);
            }
        });
        getContentPane().add(btn_NewAccount,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, 60, 40));

        lbl_SearchAccount.setText("Date of Birth Filter:");
        getContentPane().add(lbl_SearchAccount,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(560, 90, 200, -1));

        lbl_SearchAccount1.setText("Name Filter:");
        getContentPane().add(lbl_SearchAccount1,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 90, 190, -1));

        lbl_SearchAccount2.setText("Surname Filter:");
        getContentPane().add(lbl_SearchAccount2,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(310, 90, 170, -1));

        btn_ResetPassword.setBackground(new java.awt.Color(50, 136, 203));
        btn_ResetPassword.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_ResetPassword.setForeground(new java.awt.Color(0, 0, 102));
        btn_ResetPassword.setText("Reset Password");
        btn_ResetPassword.setBorder(null);
        btn_ResetPassword.setBorderPainted(false);
        btn_ResetPassword.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_ResetPassword.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_ResetPasswordActionPerformed(evt);
            }
        });
        getContentPane().add(btn_ResetPassword,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(680, 390, 130, 40));

        btn_ClearFields.setBackground(new java.awt.Color(50, 136, 203));
        btn_ClearFields.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_ClearFields.setForeground(new java.awt.Color(0, 0, 102));
        btn_ClearFields.setText("Clear Fields");
        btn_ClearFields.setBorder(null);
        btn_ClearFields.setBorderPainted(false);
        btn_ClearFields.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_ClearFields.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_ClearFieldsActionPerformed(evt);
            }
        });
        getContentPane().add(btn_ClearFields,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(750, 110, 90, 20));

        btn_Refresh.setBackground(new java.awt.Color(50, 136, 203));
        btn_Refresh.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_Refresh.setForeground(new java.awt.Color(0, 0, 102));
        btn_Refresh.setText("Refresh");
        btn_Refresh.setBorder(null);
        btn_Refresh.setBorderPainted(false);
        btn_Refresh.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_Refresh.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_RefreshActionPerformed(evt);
            }
        });
        getContentPane().add(btn_Refresh,                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 20, 70, 40));

        jMenu1.setText("Options");

        Item_newAccount.setText("New Account");
        Item_newAccount.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Item_newAccountActionPerformed(evt);
            }
        });
        jMenu1.add(Item_newAccount);

        item_ResetPass.setText("Reset Account Password");
        item_ResetPass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                item_ResetPassActionPerformed(evt);
            }
        });
        jMenu1.add(item_ResetPass);
        jMenu1.add(jSeparator1);

        item_NewAdmin.setText("New Admin Account");
        item_NewAdmin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                item_NewAdminActionPerformed(evt);
            }
        });
        jMenu1.add(item_NewAdmin);

        item_changeAdminPass.setText("Change Admin Password");
        item_changeAdminPass.addActionListener(new   java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                item_changeAdminPassActionPerformed(evt);
            }
        });
        jMenu1.add(item_changeAdminPass);
        jMenu1.add(jSeparator2);

        jMenuItem6.setText("Log out");
        jMenu1.add(jMenuItem6);

        jMenuBar1.add(jMenu1);

        setJMenuBar(jMenuBar1);

        pack();
    }// </editor-fold>

    private                                                            void
txt_SearchAccountDOBInputMethodTextChanged(java.awt.event.InputMethodEvent
evt) {
    }

    private void
txt_SearchAccountNameInputMethodTextChanged(java.awt.event.InputMethodEvent
evt) {

    }

    private void
txt_SearchAccountSurnameInputMethodTextChanged(java.awt.event.InputMethodEv
ent evt) {

    }

    private void btn_NewAccountActionPerformed(java.awt.event.ActionEvent evt) {
       NewAccountScreen nas = new NewAccountScreen(accountsList,adminID);
       nas.setLocationRelativeTo(null);
       nas.setVisible(true);
    }

    private void btn_ResetPasswordActionPerformed(java.awt.event.ActionEvent evt) {

        selection = table_Accounts.getSelectedRow();
       if(selection != -1){
           try {
               selection = table_Accounts.getSelectedRow();
               int chosenID = accountsList.get(selection).getId();
               ResetPassword rp = new ResetPassword(chosenID);
               rp.setLocationRelativeTo(null);
               rp.setVisible(true);
           } catch (SQLException ex) {

Logger.getLogger(AdminPasswordManager.class.getName()).log(Level.SEVERE, null,
ex);
           }
       }
    }

    private void item_NewAdminActionPerformed(java.awt.event.ActionEvent evt) {
        NewAdminAccount naa = new NewAdminAccount(adminID, adminList);
        naa.setLocationRelativeTo(null);
        naa.setVisible(true);
    }

    private void item_changeAdminPassActionPerformed(java.awt.event.ActionEvent
evt) {
        UpdateAdminPass uap = new UpdateAdminPass(adminID,adminList);
        uap.setLocationRelativeTo(null);
        uap.setVisible(true);

    }

    private void btn_ClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
       txt_SearchAccountName.setText("");
       txt_SearchAccountSurname.setText("");
       txt_SearchAccountDOB.setText("");
    }

    private void btn_RefreshActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            loadData();
            loadTable();
        } catch (SQLException ex) {

Logger.getLogger(AdminPasswordManager.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

    private void Item_newAccountActionPerformed(java.awt.event.ActionEvent evt) {
        btn_NewAccount.doClick();
    }

    private void item_ResetPassActionPerformed(java.awt.event.ActionEvent evt) {
        btn_ResetPassword.doClick();
    }


    // Variables declaration - do not modify
    private javax.swing.JMenuItem Item_newAccount;
    private javax.swing.JButton btn_ClearFields;
    private javax.swing.JButton btn_NewAccount;
    private javax.swing.JButton btn_Refresh;
    private javax.swing.JButton btn_ResetPassword;
    private javax.swing.JMenuItem item_NewAdmin;
    private javax.swing.JMenuItem item_ResetPass;
    private javax.swing.JMenuItem item_changeAdminPass;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem6;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JPopupMenu.Separator jSeparator1;
    private javax.swing.JPopupMenu.Separator jSeparator2;
    private javax.swing.JLabel lbl_SearchAccount;
    private javax.swing.JLabel lbl_SearchAccount1;
    private javax.swing.JLabel lbl_SearchAccount2;
    private javax.swing.JTable table_Accounts;
    private javax.swing.JTextField txt_SearchAccountDOB;
    private javax.swing.JTextField txt_SearchAccountName;
    private javax.swing.JTextField txt_SearchAccountSurname;
    // End of variables declaration
}
```

ConnectionJDBC.java

```
package databaseConection;

import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import models.Account;
import models.Admin;
import security.Hashing;


public class ConnectionJDBC {
    Date date;
    SimpleDateFormat format = new SimpleDateFormat("yyyy/mm/dd");
    Connection conn;

    private void conectar() throws SQLException{
        String url = "jdbc:mysql://localhost:3306/passwordLocker";
        String user = "root";
        String pass = "";
        conn = DriverManager.getConnection(url, user, pass);
    }

    private void desconectar() throws SQLException{
        if (conn != null){ conn.close();}
    }

public boolean checkUsername(){

    return false;

}

public int checkUsername(String userInput) throws SQLException{
    conectar();
    int numberReturned = 0;
    String query = "SELECT count(userAdmin) AS total FROM t_admins \n" +
                "WHERE userAdmin = '"+Hashing.getHash(userInput.getBytes())+"'";

    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        numberReturned = rs.getInt("total");
    }
    desconectar();
    return numberReturned;
}

public int checkCredentials(String user, String pass) throws SQLException{
    conectar();
    int numberReturned = 0;
    String query = "SELECT count(userAdmin) AS total FROM t_admins \n" +
                "WHERE userAdmin = '"+Hashing.getHash(user.getBytes())+"' AND
passwordAdmin = '"+Hashing.getHash(pass.getBytes())+"'";
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);
    while(rs.next()){
        numberReturned = rs.getInt("total");
    }
    desconectar();
    return numberReturned;
}

public Account loadAccountSelected(int input) throws SQLException{
    conectar();
    Account tempAccount =  null;

    String query = "SELECT * FROM t_accounts\n" +
                "WHERE ida = '"+input+"'";

    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        int id = rs.getInt("ida");
        String name = rs.getString("forename");
        String sur = rs.getString("surname");
        String dob = rs.getString("dob");
        String user = rs.getString("username");
        String mas = rs.getString("masterkey");
        String a1 = rs.getString("a1");
        String a2 = rs.getString("a2");
        String a3 = rs.getString("a3");

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-mm-dd");
        Date entry = null;


        tempAccount = new Account(id,name,sur,dob,user,mas,a1,a2,a3);

    }
    desconectar();
    return tempAccount;
}

public ArrayList<Account> loadAccounts() throws SQLException{
    conectar();
    ArrayList<Account> temporalList = new ArrayList<>();

    String query = "SELECT * FROM t_accounts";
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(query);

    while(rs.next()){
        int id = rs.getInt("ida");
        String name = rs.getString("forename");
        String sur = rs.getString("surname");
        String dob = rs.getString("dob");
        String user = rs.getString("username");
        String mas = rs.getString("masterkey");
        String a1 = rs.getString("a1");
        String a2 = rs.getString("a2");
        String a3 = rs.getString("a3");

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-mm-dd");
        Date entry = null;


        Account a = new Account(id,name,sur,dob,user,mas,a1,a2,a3);
        temporalList.add(a);
    }
    desconectar();
    return temporalList;
}

public void insertAccount(Account a) throws ParseException{
   try{
            //hashing
            String huser = Hashing.getHash(a.getUsername().getBytes());
            String hpassword = Hashing.getHash(a.getPassword().getBytes());

            conectar();
            String insert = "INSERT INTO t_accounts
(forename,surname,dob,username,masterkey,a1,a2,a3,responsible)
VALUES(?,?,?,?,?,?,?,?,?)";

            PreparedStatement pst = conn.prepareStatement(insert);
            pst.setString(1,a.getForename());
            pst.setString(2, a.getSurname());
            pst.setString(3, a.getDob());
            pst.setString(4, huser);
```

```
      pst.setString(5, hpassword);
      pst.setString(6, a.getQuestion1());
      pst.setString(7, a.getQuestion2());
      pst.setString(8, a.getQuestion3());
      pst.setInt(9, a.getResponsible());

      pst.execute();
      desconectar();

    }catch(SQLException e){
      JOptionPane.showMessageDialog(null,"Error","No pasa
nada",JOptionPane.ERROR_MESSAGE);
      System.out.println(e.getMessage());
    }
}

public void updateAccount(Account selectedAccount) {
    try{
      String convertedPass =
Hashing.getHash(selectedAccount.getPassword().getBytes());
      String update = ("UPDATE t_accounts SET masterkey = '"+convertedPass+"'" "
          + "WHERE ida= "+selectedAccount.getId());
      conectar();
      Statement st = conn.createStatement();
      st.executeUpdate(update);
      desconectar();
     }catch(SQLException e){
      System.out.println(e.getMessage());
    }
  }

public void insertAdmin(Admin admin) {
  try{
    conectar();

    String u = Hashing.getHash(admin.getUserAdmin().getBytes());
    String p = Hashing.getHash(admin.getAdminPass().getBytes());

    String insert = "INSERT INTO t_admins (userAdmin, passwordAdmin)
VALUES(?,?)";
    PreparedStatement pst = conn.prepareStatement(insert);
    pst.setString(1, u);
    pst.setString(2, p);
    pst.execute();
    desconectar();
  }catch(SQLException e){
    JOptionPane.showMessageDialog(null,"Error","No pasa
nada",JOptionPane.ERROR);
  }
}

public ArrayList<Admin> loadAdmins() throws SQLException {

conectar();
ArrayList<Admin> temporalList = new ArrayList<>();

String query = "SELECT * FROM t_admins";

Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(query);

while(rs.next()){
  int i = rs.getInt("idadmin");
  String n = rs.getString("userAdmin");
  String p = rs.getString("passwordAdmin");

  Admin a = new Admin(i,n,p);
  temporalList.add(a);
}
desconectar();
return temporalList;
}

public int loadChosenAccount(String user) throws SQLException {
  conectar();
  String condition = "";
  condition = Hashing.getHash(user.getBytes());
  String query = "SELECT idadmin FROM t_admins WHERE userAdmin LIKE
'"+condition+"'";
  Statement st = conn.createStatement();
  ResultSet rs = st.executeQuery(query);
  int i = 0;
  while(rs.next()){
    i = rs.getInt("idadmin");
  }
  desconectar();
  return i;
}

public void updateAdminPass(String newPass, int id) {

    String newPassHashed = Hashing.getHash(newPass.getBytes());

    try{
      String update = "UPDATE t_admins SET passwordAdmin =
'"+newPassHashed+"'"
              + "WHERE idadmin= "+id;
      conectar();
      Statement st = conn.createStatement();
      st.executeUpdate(update);
      desconectar();
     }catch(SQLException e){
      System.out.println(e.getMessage());
    }
  }
}
```

NewAccountScreen.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.sql.Date;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import models.Account;


public class NewAccountScreen extends javax.swing.JFrame {
  ArrayList<Account> accountList =  new ArrayList<>();
  int ID;
  Random random;
```

```
public NewAccountScreen(ArrayList<Account> list, int id) {
    initComponents();
    accountList = list;
    this.ID = id;
}

public boolean checkName(){
    boolean exists = false;
    for(Account a : accountList){
       if(a.getForename().equals(txt_NewAccountName.getText())){
           exists = true;
       }
    }
return exists;
}   //combine with surname
public boolean checkSurname(){
    boolean exists = false;
    for(Account a : accountList){
       if(a.getSurname().equals(txt_NewAccountSurname.getText())){
           exists = true;
       }
    }
return exists;
}  //combine with name
public boolean checkUsername(){
    boolean exists = false;
    for(Account a : accountList){
       if(a.getUsername().equals(txt_NewAccountUsername.getText())){
           exists = true;
       }
    }
return exists;
}
public boolean checkAnswers(){
    boolean empty = false;
    if(txt_NewAccountA1.getText().length() <= 0){
        empty = true;
    }
    if(txt_NewAccountA2.getText().length() <= 0){
        empty = true;
    }
    if(txt_NewAccountA3.getText().length() <= 0){
        empty = true;
    }
    return empty;
}
public String randomPassword(){
    String code = "";

    final String upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    final String lower = upper.toLowerCase();
    final String digits = "0123456789";

    String[] optionList = {upper,lower,digits};



    random = new Random();
    for(int i = 0; i<=5;i++){
        int stringNum = random.nextInt(optionList.length);
        int randomLetter = random.nextInt(optionList[stringNum].length());
        //System.out.println(randomLetter);
        code += optionList[stringNum].charAt(randomLetter);
        System.out.println(code);
    }
```

```
        return code;
    }
    public boolean passwordCheck(String code){
        boolean exists = false;
        for(Account a : accountList){
           if(a.getPassword().equals(code)){
               exists = true;
           }
        }
        return exists;
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        txt_NewAccountName = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        txt_NewAccountSurname = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jLabel4 = new javax.swing.JLabel();
        txt_NewAccountUsername = new javax.swing.JTextField();
        jSeparator2 = new javax.swing.JSeparator();
        jLabel5 = new javax.swing.JLabel();
        txt_NewAccountA1 = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        txt_NewAccountA2 = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        txt_NewAccountA3 = new javax.swing.JTextField();
        jLabel12 = new javax.swing.JLabel();
        btn_CreateAccount = new javax.swing.JButton();
        lbl_BlueUser = new javax.swing.JLabel();
        cb_year = new javax.swing.JComboBox<>();
        cb_day = new javax.swing.JComboBox<>();
        cb_month = new javax.swing.JComboBox<>();
        jLabel13 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel1.setText("1) What is the name of your favourite plate of food?");
        getContentPane().add(jLabel1,                                         new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 340, 460, -1));
        getContentPane().add(txt_NewAccountName,                              new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 80, 350, 30));

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel2.setText("Surname:");
        getContentPane().add(jLabel2,                                         new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 140, -1, -1));
        getContentPane().add(txt_NewAccountSurname,                           new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 130, 350, 30));

        jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel3.setText("Date of Birth:");
        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 190, -1, -1));
        getContentPane().add(jSeparator1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 220, 500, 10));

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel4.setText("Username:");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 250, -1, -1));
        getContentPane().add(txt_NewAccountUsername, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 240, 350, 30));
        getContentPane().add(jSeparator2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 290, 500, 10));

        jLabel5.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 320, -1, -1));
        getContentPane().add(txt_NewAccountA1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 370, 370, 30));

        jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel6.setText("Name:");
        getContentPane().add(jLabel6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 90, -1, -1));

        jLabel7.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel7.setText("Answer 1:");
        getContentPane().add(jLabel7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 380, 120, -1));

        jLabel8.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel8.setText("Security questions");
        getContentPane().add(jLabel8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 310, 120, -1));

        jLabel9.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel9.setText("2) Name of your first pet?");
        getContentPane().add(jLabel9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 420, 460, -1));
        getContentPane().add(txt_NewAccountA2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 450, 370, 30));

        jLabel10.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel10.setText("Answer 2:");
        getContentPane().add(jLabel10, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 460, 120, -1));

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel11.setText("3) What is the name of the of your favourite film all times?");
        getContentPane().add(jLabel11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 500, 460, -1));
        getContentPane().add(txt_NewAccountA3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 530, 370, 30));

        jLabel12.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel12.setText("Answer 3:");
        getContentPane().add(jLabel12, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 540, 120, -1));

        btn_CreateAccount.setBackground(new java.awt.Color(50, 136, 203));
        btn_CreateAccount.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_CreateAccount.setForeground(new java.awt.Color(0, 0, 102));
        btn_CreateAccount.setText("Create account");
        btn_CreateAccount.setBorder(null);
```

```
        btn_CreateAccount.setBorderPainted(false);
        btn_CreateAccount.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_CreateAccount.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_CreateAccountActionPerformed(evt);
            }
        });
        getContentPane().add(btn_CreateAccount, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 580, 120, 30));

        lbl_BlueUser.setFont(new java.awt.Font("Decker", 0, 36)); // NOI18N
        lbl_BlueUser.setForeground(new java.awt.Color(50, 136, 203));
        lbl_BlueUser.setText("New account details");
        getContentPane().add(lbl_BlueUser, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 10, 380, 50));

        cb_year.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"2018", "2017", "2016", "2015", "2014", "2013", "2012", "2011", "2010", "2009", "2008",
"2007", "2006", "2005", "2004", "2003", "2002", "2001", "2000", "1999", "1998",
"1997", "1996", "1995", "1994", "1993", "1992", "1991", "1990", "1989", "1988", "1987",
"1986", "1985", "1984", "1983", "1982", "1981", "1980", "1979", "1978", "1977", "1976",
"1975", "1974", "1973", "1972", "1971", "1970", "1969", "1968", "1967", "1966", "1965",
"1964", "1963", "1962", "1961", "1960", "1959", "1958", "1957", "1956", "1955", "1954",
"1953", "1952", "1951", "1950", "1949", "1948", "1947", "1946", "1945", "1944", "1943",
"1942", "1941", "1940", "1939", "1938", "1937", "1936", "1935", "1934", "1933", "1932",
"1931", "1930", "1929", "1928", "1927", "1926", "1925", "1924", "1923", "1922", "1921",
"1920", "1919", "1918", "1917", "1916", "1915", "1914", "1913", "1912", "1911", "1910",
"1909", "1908", "1907", "1906", "1905", "1904", "1903", "1902", "1901", "1900" }));
        getContentPane().add(cb_year, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 190, 140, -1));

        cb_day.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "1",
"2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" }));
        getContentPane().add(cb_day, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 190, -1, -1));

        cb_month.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12" }));
        getContentPane().add(cb_month, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 190, -1, -1));

        jLabel13.setText("Year");
        getContentPane().add(jLabel13, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 170, -1, -1));

        jLabel14.setText("Day");
        getContentPane().add(jLabel14, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 170, -1, -1));

        jLabel15.setText("Month");
        getContentPane().add(jLabel15, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 170, -1, -1));

        pack();
    }// </editor-fold>

    private void btn_CreateAccountActionPerformed(java.awt.event.ActionEvent evt) {
        boolean ok = true;
        String newPassword = "";

        if(checkName() && checkSurname()){
            ok = false;
        }
        if(checkUsername()){
            ok = false;
        }
        if (checkAnswers()){
            ok = false;
        }
        if(ok){
            try {
                //Generate random password
                do{
                    newPassword = randomPassword();
                }while(passwordCheck(newPassword));

                //insert new Account
                Account a =  new Account();
                a.setForename(txt_NewAccountName.getText());
                a.setSurname(txt_NewAccountSurname.getText());

                Date d;
                int year = Integer.parseInt(cb_year.getSelectedItem().toString());
                int month = Integer.parseInt(cb_month.getSelectedItem().toString());
                int day = Integer.parseInt(cb_day.getSelectedItem().toString());
                String date = day+"/"+month+"/"+year;
                a.setDob(date);
                a.setUsername(txt_NewAccountUsername.getText());
                a.setPassword(newPassword);
                a.setQuestion1(txt_NewAccountA1.getText());
                a.setQuesiotn2(txt_NewAccountA2.getText());
                a.setQuestion3(txt_NewAccountA3.getText());
                a.setResponsible(ID);

                ConnectionJDBC c =  new ConnectionJDBC();
                c.insertAccount(a);
                JOptionPane.showMessageDialog(null, "\n\nYour new password is:
"+newPassword+"\n\n", "important",JOptionPane.INFORMATION_MESSAGE);
                this.dispose();
            } catch (ParseException ex) {
                Logger.getLogger(NewAccountScreen.class.getName()).log(Level.SEVERE,
null, ex);
            }
        }else{
            JOptionPane.showMessageDialog(null,                              "Not
entered","Notice",JOptionPane.INFORMATION_MESSAGE);
        }


    }

    // Variables declaration - do not modify
    private javax.swing.JButton btn_CreateAccount;
    private javax.swing.JComboBox<String> cb_day;
    private javax.swing.JComboBox<String> cb_month;
    private javax.swing.JComboBox<String> cb_year;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JLabel lbl_BlueUser;
    private javax.swing.JTextField txt_NewAccountA1;
    private javax.swing.JTextField txt_NewAccountA2;
    private javax.swing.JTextField txt_NewAccountA3;
    private javax.swing.JTextField txt_NewAccountName;
    private javax.swing.JTextField txt_NewAccountSurname;
    private javax.swing.JTextField txt_NewAccountUsername;
    // End of variables declaration
}
```

NewAdminAccount.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import models.Admin;
import security.Hashing;

public class NewAdminAccount extends javax.swing.JFrame {
    ArrayList<Admin> adminList = new ArrayList<>(); //to check it doesn't already exist
    final int adminID;

    public NewAdminAccount(int id,ArrayList<Admin> list) {
        initComponents();
        adminID = id;
        adminList = list;

        lbl_UserExists.setVisible(false);
    }

    public boolean checkUsername(){
        boolean exists = false;
        String value = Hashing.getHash(txt_NewAdUser.getText().getBytes());
        for(Admin a : adminList){
           if(a.getUserAdmin().equals(value)){
               exists = true;
           }
        }
    return exists;
    }

    public boolean passwordOK(){
        boolean ok = false;
        if(txt_NewAdPass.getText().length() > 4){
            ok = true;

        }else{
            lbl_UserExists.setText("Password too short");
            lbl_UserExists.setVisible(true);

        }
        return ok;
    }
    public boolean userOK(){
         boolean ok = false;
        if(txt_NewAdUser.getText().length() > 0){
            ok = true;

        }else{
            lbl_UserExists.setText("Need to enter a username");
            lbl_UserExists.setVisible(true);
        }
        return ok;
    }


    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel2 = new javax.swing.JLabel();
        txt_NewAdUser = new javax.swing.JTextField();
        txt_NewAdPass = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        btn_NewAdminCancel = new javax.swing.JButton();
        btn_NewCreateAdmin = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        lbl_UserExists = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setText("New username:");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, -1, 20));

        txt_NewAdUser.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_NewAdUserFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                txt_NewAdUserFocusLost(evt);
            }
        });
        getContentPane().add(txt_NewAdUser, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 80, 231, 20));
        getContentPane().add(txt_NewAdPass, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 231, 20));

        jLabel3.setText("Enter the password:");
        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 110, -1, 20));

        btn_NewAdminCancel.setBackground(new java.awt.Color(50, 136, 203));
        btn_NewAdminCancel.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_NewAdminCancel.setForeground(new java.awt.Color(0, 0, 102));
        btn_NewAdminCancel.setText("Cancel");
        btn_NewAdminCancel.setBorder(null);
        btn_NewAdminCancel.setBorderPainted(false);
        btn_NewAdminCancel.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_NewAdminCancel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_NewAdminCancelActionPerformed(evt);
            }
        });
        getContentPane().add(btn_NewAdminCancel, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 190, 110, 30));

        btn_NewCreateAdmin.setBackground(new java.awt.Color(50, 136, 203));
        btn_NewCreateAdmin.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_NewCreateAdmin.setForeground(new java.awt.Color(0, 0, 102));
        btn_NewCreateAdmin.setText("Create Admin");
        btn_NewCreateAdmin.setBorder(null);
        btn_NewCreateAdmin.setBorderPainted(false);
        btn_NewCreateAdmin.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_NewCreateAdmin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_NewCreateAdminActionPerformed(evt);
            }
        });
        getContentPane().add(btn_NewCreateAdmin, new
org.netbeans.lib.awtextra.AbsoluteConstraints(230, 190, 110, 30));

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setText("Create new Admin");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 30, 340, 20));

        lbl_UserExists.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_UserExists.setForeground(new java.awt.Color(255, 51, 51));
        lbl_UserExists.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lbl_UserExists.setText("User already exists");
        getContentPane().add(lbl_UserExists, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 140, 260, 20));

        pack();
    }// </editor-fold>

    private void btn_NewAdminCancelActionPerformed(java.awt.event.ActionEvent
evt) {
        this.dispose();
    }

    private void btn_NewCreateAdminActionPerformed(java.awt.event.ActionEvent
evt) {

        if(passwordOK()&&userOK()&&(!checkUsername())){
        ConnectionJDBC gestor = new ConnectionJDBC();
        gestor.insertAdmin(new
Admin(txt_NewAdUser.getText(),txt_NewAdPass.getText()));
        JOptionPane.showMessageDialog(null,"NEW ADMIN HAS BEEN
CREATED","Confirmation",JOptionPane.PLAIN_MESSAGE);
        this.dispose();
        }

    }

    private void txt_NewAdUserFocusLost(java.awt.event.FocusEvent evt) {
      if(checkUsername()){
         lbl_UserExists.setVisible(true);
         lbl_UserExists.setText("user already exists");
      }
    }

    private void txt_NewAdUserFocusGained(java.awt.event.FocusEvent evt) {
        lbl_UserExists.setVisible(false);
    }
    // Variables declaration - do not modify
    private javax.swing.JButton btn_NewAdminCancel;
    private javax.swing.JButton btn_NewCreateAdmin;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel lbl_UserExists;
    private javax.swing.JTextField txt_NewAdPass;
    private javax.swing.JTextField txt_NewAdUser;
    // End of variables declaration
}
```

ResetPassword.java

```
package frames;

import databaseConection.ConnectionJDBC;
import java.sql.SQLException;
import java.util.Random;
import javax.swing.JOptionPane;
import models.Account;


public class ResetPassword extends javax.swing.JFrame {

    //Attributes
    int ID;
    Account selectedAccount;
    Random random;

    //constructor
    public ResetPassword(int chosenaccount) throws SQLException {
        initComponents();
        this.ID = chosenaccount;
        loadAccount(ID);

        lbl_nameSelected.setText("Account:            "+selectedAccount.getForename()+", 
"+selectedAccount.getSurname());
        lbl_no1.setVisible(false);
        lbl_no2.setVisible(false);
        lbl_no3.setVisible(false);

    }
    //loads the account using the attribute ID
    public void loadAccount(int id) throws SQLException{
        ConnectionJDBC c = new ConnectionJDBC();
        this.selectedAccount =  c.loadAccountSelected(id);
    }
    //Generates a random password
    public String randomPassword(){
        String code = "";

        final String upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        final String lower = upper.toLowerCase();
        final String digits = "0123456789";

        String[] optionList = {upper,lower,digits};

        random = new Random();
        for(int i = 0; i<=5;i++){
            int stringNum = random.nextInt(optionList.length);
            int randomLetter = random.nextInt(optionList[stringNum].length());
            //System.out.println(randomLetter);
            code += optionList[stringNum].charAt(randomLetter);

        }
        return code;
    }
    //This checks all questions but splits the flagging from the triple condition.
    public boolean checkAnswers(){
        boolean a1 = isQ1();
        boolean a2 = isQ2();
        boolean a3 = isQ3();
        return a1&&a2&&a3;
    }
    //Question 1 ok?
    public boolean isQ1(){
        boolean answer;
        answer = txt_answer1.getText().equals(selectedAccount.getQuestion1());
        if (!answer){
            lbl_no1.setVisible(true);
        }
        return answer;
    }
    //Question 2 ok?
    public boolean isQ2(){
        boolean answer;
        answer = txt_answer2.getText().equals(selectedAccount.getQuestion2());
        if (!answer){
            lbl_no2.setVisible(true);
        }
        return answer;
    }
    //Question 3 ok?
    public boolean isQ3(){
        boolean answer;
        answer = txt_answer3.getText().equals(selectedAccount.getQuestion3());
        if (!answer){
            lbl_no3.setVisible(true);
        }
        return answer;
    }



    //LISTENERS
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        txt_answer1 = new javax.swing.JTextField();
        txt_answer2 = new javax.swing.JTextField();
        txt_answer3 = new javax.swing.JTextField();
        btn_ResetMasterKey = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        lbl_nameSelected = new javax.swing.JLabel();
        lbl_no3 = new javax.swing.JLabel();
        lbl_no1 = new javax.swing.JLabel();
        lbl_no2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        txt_answer1.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_answer1FocusGained(evt);
            }
        });
        getContentPane().add(txt_answer1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 170, 330, -1));

        txt_answer2.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_answer2FocusGained(evt);
            }
        });
        getContentPane().add(txt_answer2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 240, 330, -1));

        txt_answer3.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_answer3FocusGained(evt);
            }
        });
        getContentPane().add(txt_answer3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 320, 330, -1));

        btn_ResetMasterKey.setBackground(new java.awt.Color(50, 136, 203));
        btn_ResetMasterKey.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_ResetMasterKey.setForeground(new java.awt.Color(0, 0, 102));
        btn_ResetMasterKey.setText("Reset");
        btn_ResetMasterKey.setBorder(null);
        btn_ResetMasterKey.setBorderPainted(false);
        btn_ResetMasterKey.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_ResetMasterKey.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_ResetMasterKeyActionPerformed(evt);
            }
        });
        getContentPane().add(btn_ResetMasterKey, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 370, 130, 30));

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel2.setText("1) What is the name of your favourite plate of food?");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 130, 330, -1));

        jLabel9.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel9.setText("2) Name of your first pet?");
        getContentPane().add(jLabel9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 210, 180, -1));

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel11.setText("3) What is the name of the of your favourite film all times?");
        getContentPane().add(jLabel11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 290, 400, -1));

        lbl_nameSelected.setText("nAME OF THE aCCOUNT");
        getContentPane().add(lbl_nameSelected, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 50, -1, -1));

        lbl_no3.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_no3.setForeground(new java.awt.Color(255, 51, 51));
        lbl_no3.setText("No");
        getContentPane().add(lbl_no3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 310, 30, 30));

        lbl_no1.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_no1.setForeground(new java.awt.Color(255, 51, 51));
        lbl_no1.setText("No");
        getContentPane().add(lbl_no1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 160, 30, 30));

        lbl_no2.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_no2.setForeground(new java.awt.Color(255, 51, 51));
        lbl_no2.setText("No");
        getContentPane().add(lbl_no2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 230, 30, 30));

        pack();
    }// </editor-fold>

    private void btn_ResetMasterKeyActionPerformed(java.awt.event.ActionEvent evt)
{
    if(checkAnswers()){
        //generate password
        String newPassword = randomPassword();

        selectedAccount.setPassword(newPassword);
        ConnectionJDBC c = new ConnectionJDBC(); //update database
        c.updateAccount(selectedAccount);
        JOptionPane.showMessageDialog(null, "\nThe new password is:
"+newPassword+"\n\n", "important",JOptionPane.INFORMATION_MESSAGE);
        this.dispose();
    }


    //LISTENERS
    }

    private void txt_answer1FocusGained(java.awt.event.FocusEvent evt) {
        lbl_no1.setVisible(false);
    }

    private void txt_answer2FocusGained(java.awt.event.FocusEvent evt) {
        lbl_no2.setVisible(false);
    }

    private void txt_answer3FocusGained(java.awt.event.FocusEvent evt) {
        lbl_no3.setVisible(false);
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btn_ResetMasterKey;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JLabel lbl_nameSelected;
    private javax.swing.JLabel lbl_no1;
    private javax.swing.JLabel lbl_no2;
    private javax.swing.JLabel lbl_no3;
    private javax.swing.JTextField txt_answer1;
    private javax.swing.JTextField txt_answer2;
    private javax.swing.JTextField txt_answer3;
    // End of variables declaration
}
```

UpdateAdminPass.java

```java
package frames;

import databaseConection.ConnectionJDBC;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import models.Admin;
import security.Hashing;


public class UpdateAdminPass extends javax.swing.JFrame {
    ArrayList<Admin> adminList = new ArrayList<>();
    int adminID;

    public UpdateAdminPass(int id,ArrayList<Admin> list) {
        initComponents();
        adminList = list;
        adminID = id;
    }

    public boolean checkOldPassword(){
        boolean ok = false;
        String comparison = txt_UpdateOldPass.getText();
        comparison = Hashing.getHash(comparison.getBytes());
        for(Admin a : adminList){
            if(adminID == a.getId()){
                if(a.getAdminPass().equals(comparison)){
                    ok = true;
                }else{
                    lbl_UpdateNote.setText("Password does not match");
                    lbl_UpdateNote.setVisible(true);
                }
            }
        }
        return ok;
    }

    public boolean newPasswordsMatch(){
        boolean ok = false;
        if(txt_UpdateConfirm.getText().length()>0){
            ok = txt_UpdateNewPass.getText().equals(txt_UpdateConfirm.getText());
            if(!ok){
                lbl_UpdateNote.setText("new passwords do not match");
                lbl_UpdateNote.setVisible(true);
            }
        }
        return ok;
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel2 = new javax.swing.JLabel();
        txt_UpdateOldPass = new javax.swing.JTextField();
        txt_UpdateConfirm = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        btn_UpdateCancelAdmin = new javax.swing.JButton();
        btn_UpdateAdminPass = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        lbl_UpdateNote = new javax.swing.JLabel();
        txt_UpdateNewPass = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setText("New Password:");
        getContentPane().add(jLabel2,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 110, -1, 20));

        txt_UpdateOldPass.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_UpdateOldPassFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                txt_UpdateOldPassFocusLost(evt);
            }
        });
        getContentPane().add(txt_UpdateOldPass,                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 70, 231, 20));
        getContentPane().add(txt_UpdateConfirm,                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 231, 20));

        jLabel3.setText("Confirm new Password:");
        getContentPane().add(jLabel3,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150, -1, 20));

        btn_UpdateCancelAdmin.setBackground(new java.awt.Color(50, 136, 203));
        btn_UpdateCancelAdmin.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_UpdateCancelAdmin.setForeground(new java.awt.Color(0, 0, 102));
        btn_UpdateCancelAdmin.setText("Cancel");
        btn_UpdateCancelAdmin.setBorder(null);
        btn_UpdateCancelAdmin.setBorderPainted(false);
        btn_UpdateCancelAdmin.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_UpdateCancelAdmin.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_UpdateCancelAdminActionPerformed(evt);
            }
        });
        getContentPane().add(btn_UpdateCancelAdmin,                      new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 240, 110, 30));

        btn_UpdateAdminPass.setBackground(new java.awt.Color(50, 136, 203));
        btn_UpdateAdminPass.setFont(new java.awt.Font("Decker", 1, 11)); // NOI18N
        btn_UpdateAdminPass.setForeground(new java.awt.Color(0, 0, 102));
        btn_UpdateAdminPass.setText("Update");
        btn_UpdateAdminPass.setBorder(null);
        btn_UpdateAdminPass.setBorderPainted(false);
        btn_UpdateAdminPass.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_UpdateAdminPass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_UpdateAdminPassActionPerformed(evt);
            }
        });
        getContentPane().add(btn_UpdateAdminPass,                        new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 240, 110, 30));

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setText("Update Password");
        getContentPane().add(jLabel4,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 20, 340, 20));
```

```
        lbl_UpdateNote.setFont(new java.awt.Font("Decker", 0, 18)); // NOI18N
        lbl_UpdateNote.setForeground(new java.awt.Color(255, 51, 51));
        lbl_UpdateNote.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lbl_UpdateNote.setText("User already exists");
        getContentPane().add(lbl_UpdateNote, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 190, 260, 20));

        txt_UpdateNewPass.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                txt_UpdateNewPassFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                txt_UpdateNewPassFocusLost(evt);
            }
        });
        getContentPane().add(txt_UpdateNewPass, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 231, 20));

        jLabel5.setText("Old password:");
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 70, -1, 20));

        pack();
    }// </editor-fold>

    private void btn_UpdateCancelAdminActionPerformed(java.awt.event.ActionEvent
evt) {
        this.dispose();
    }

    private void btn_UpdateAdminPassActionPerformed(java.awt.event.ActionEvent
evt) {

        if(newPasswordsMatch()&&checkOldPassword()){
            ConnectionJDBC gestor = new ConnectionJDBC();
            gestor.updateAdminPass(txt_UpdateConfirm.getText(),adminID);
            JOptionPane.showMessageDialog(null,"Password has been
updated","Confirmation",JOptionPane.PLAIN_MESSAGE);
            this.dispose();
        }

    }

    private void txt_UpdateNewPassFocusGained(java.awt.event.FocusEvent evt) {

    }

    private void txt_UpdateNewPassFocusLost(java.awt.event.FocusEvent evt) {

    }

    private void txt_UpdateOldPassFocusGained(java.awt.event.FocusEvent evt) {
        lbl_UpdateNote.setVisible(false);
    }

    private void txt_UpdateOldPassFocusLost(java.awt.event.FocusEvent evt) {
        checkOldPassword();
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btn_UpdateAdminPass;
    private javax.swing.JButton btn_UpdateCancelAdmin;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel lbl_UpdateNote;
    private javax.swing.JTextField txt_UpdateConfirm;
    private javax.swing.JTextField txt_UpdateNewPass;
    private javax.swing.JTextField txt_UpdateOldPass;
    // End of variables declaration
}
```

Account.java

```
package models;

public class Account {
    private int id;
    private String forename;
    private String surname;
    private String dob;
    private String username;
    private String password;
    private String question1;
    private String question2;
    private String question3;
    private int responsible;

    public Account(){

    }

    public Account(int id, String forename, String surname, String dob, String
username, String password, String question1, String quesiotn2, String question3) {
        this.id = id;
        this.forename = forename;
        this.surname = surname;
        this.dob = dob;
        this.username = username;
        this.password = password;
        this.question1 = question1;
        this.question2 = quesiotn2;
        this.question3 = question3;
    }

    public Account(int id, String forename, String surname, String dob, String
username, String password, String question1, String question2, String question3, int
responsible) {
        this.id = id;
        this.forename = forename;
        this.surname = surname;
        this.dob = dob;
        this.username = username;
        this.password = password;
        this.question1 = question1;
        this.question2 = question2;
        this.question3 = question3;
        this.responsible = responsible;
    }

    public int getResponsible() {
        return responsible;
    }
    public void setResponsible(int responsible) {
        this.responsible = responsible;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getQuestion1() {
        return question1;
    }

    public void setQuestion1(String question1) {
        this.question1 = question1;
    }

    public String getQuestion2() {
        return question2;
    }

    public void setQuesiotn2(String quesiotn2) {
        this.question2 = quesiotn2;
    }

    public String getQuestion3() {
        return question3;
    }

    public void setQuestion3(String question3) {
        this.question3 = question3;
    }

    public String getForename() {
        return forename;
    }

    public void setForename(String forename) {
        this.forename = forename;
    }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public String getDob() {
        return dob;
    }

    public void setDob(String dob) {
        this.dob = dob;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Override
    public String toString() {
        return "Account{" + "id=" + id + ", forename=" + forename + ", surname=" + surname + ", dob=" + dob + ", username=" + username + ", password=" + password + ", question1=" + question1 + ", quesiotn2=" + question2 + ", question3=" + question3 + '}';
    }
}
```

Admin.java

```
package models;

public class Admin {
    private int id;
    private String userAdmin;
    private String adminPass;

    public Admin() {
    }

    public Admin(String userAdmin, String adminPass) {
        this.userAdmin = userAdmin;
        this.adminPass = adminPass;
    }

    public Admin(int id, String userAdmin, String adminPass) {
        this.id = id;
        this.userAdmin = userAdmin;
        this.adminPass = adminPass;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUserAdmin() {
        return userAdmin;
    }

    public void setUserAdmin(String userAdmin) {
        this.userAdmin = userAdmin;
    }

    public String getAdminPass() {
        return adminPass;
    }

    public void setAdminPass(String adminPass) {
        this.adminPass = adminPass;
    }

    @Override
    public String toString() {
        return "Admin{" + "id=" + id + ", userAdmin=" + userAdmin + ", adminPass=" +
adminPass + '}';
    }

}
```

Hashing.java

```
package security;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import javax.xml.bind.DatatypeConverter;


public class Hashing {

    public static String getHash(byte[] input){
        String hash = "";
        try{
            MessageDigest md = MessageDigest.getInstance("Sha-512");
            md.update(input);
            byte[] output = md.digest();
            hash = DatatypeConverter.printHexBinary(output);

        }catch(NoSuchAlgorithmException e){
            System.out.println("Hashing went wrong");
        }
        return hash;
    }

}
```

SQL Database creation

PasswordLockerDatabaseCreation.sql

```
DROP DATABASE passwordLocker;
CREATE DATABASE IF NOT EXISTS passwordLocker;

USE passwordLocker;


  CREATE TABLE IF NOT EXISTS t_admins(
                idadmin INT AUTO_INCREMENT NOT NULL,
    userAdmin VARCHAR(200) DEFAULT NULL,
    passwordAdmin VARCHAR(200) DEFAULT NULL,
    PRIMARY KEY (idadmin)
);


CREATE TABLE IF NOT EXISTS t_accounts(
                ida INT(9) AUTO_INCREMENT NOT NULL,
    forename VARCHAR(50) DEFAULT NULL,
    surname VARCHAR(50) DEFAULT NULL,
                dob  VARCHAR(50) DEFAULT NULL,
    username VARCHAR(200) DEFAULT NULL,
    masterkey VARCHAR(200) DEFAULT NULL,
    a1 VARCHAR(50) DEFAULT NULL,
    a2 VARCHAR(50) DEFAULT NULL,
    a3 VARCHAR(50) DEFAULT NULL,
    responsible INT DEFAULT NULL,
    PRIMARY KEY (ida),

                CONSTRAINT fk_relation_account_admin FOREIGN KEY (responsible)
REFERENCES t_admins(idadmin)
    ON UPDATE CASCADE ON DELETE SET NULL
);


CREATE TABLE IF NOT EXISTS t_passwords(
                idp INT AUTO_INCREMENT NOT NULL,
    ida_fk INT DEFAULT NULL,
    passname VARCHAR(50) DEFAULT NULL,
    pass VARBINARY(300) DEFAULT NULL,
    secret VARBINARY(700) DEFAULT NULL,
    description TEXT DEFAULT NULL,
    PRIMARY KEY(idp),

    CONSTRAINT fk_relation_password_account FOREIGN KEY (ida_fk) REFERENCES
t_accounts(ida)
    ON UPDATE CASCADE ON DELETE SET NULL
    );



-- INSERTAR REGISTROS DE PRUEBA
INSERT INTO t_admins (userAdmin,passwordAdmin)
VALUES('C7AD44CBAD762A5DA0A452F9E854FDC1E0E7A52A38015F23F3EAB1D80
B931DD472634DFAC71CD34EBC35D16AB7FB8A90C81F975113D6C7538DC69DD8
DE9077EC','C7AD44CBAD762A5DA0A452F9E854FDC1E0E7A52A38015F23F3EAB1D
80B931DD472634DFAC71CD34EBC35D16AB7FB8A90C81F975113D6C7538DC69D
D8DE9077EC');
INSERT INTO t_admins (userAdmin,passwordAdmin)
VALUES('353BA90F8C0B3E0F355A3D6C960B7CAED5F2C1412992277C0669A04A
62E7DFD35FBA9F4631A7DC6D00FB44D93D305CC0B749C7501D9CE86F26148D0
5101B8324','353BA90F8C0B3E0F355A3D6C960B7CAED5F2C1412992277C0669A
04A62E7DFD35FBA9F4631A7DC6D00FB44D93D305CC0B749C7501D9CE86F2614
8D05101B8324');


INSERT INTO t_accounts
(forename,surname,dob,username,masterkey,a1,a2,a3,responsible)
VALUES('Alvaro','Ruiz
Arciniegas','31/03/1962','74C47DECC64FD921299567F5F6467860DC9179CE2E723
048C184FDF2FD6A32936470ECC3D639B6947E99F9C42735ED20552BE14FDA240
84AD79627195ACA3FB1','74C47DECC64FD921299567F5F6467860DC9179CE2E723
048C184FDF2FD6A32936470ECC3D639B6947E99F9C42735ED20552BE14FDA240
84AD79627195ACA3FB1','nose','tampocose','esasi',1);
```

```
INSERT INTO t_accounts
(forename,surname,dob,username,masterkey,a1,a2,a3,responsible)
VALUES('Steven','Ruiz
Boyle','20/04/1988','C7451E893211272F718A077226081539BC932BB90E010230B
19163FAFB257FB391DBC567FBC12C09DDA8822FE56767680C924BC22AC013219
07B5FA1D9DEC164','C7451E893211272F718A077226081539BC932BB90E010230B
19163FAFB257FB391DBC567FBC12C09DDA8822FE56767680C924BC22AC013219
07B5FA1D9DEC164','nose','tampocose','esasi',1);
INSERT INTO t_accounts
(forename,surname,dob,username,masterkey,a1,a2,a3,responsible)
VALUES('Leandro','Ruiz
Boyle','12/02/1991','02C6A396E9E75FA78DAC83D551732DA4BDE5585FB3801356
BA77CAD173036918D085D6FA4AADDCBCA6A9853C5FFD59EF8AD5CF095D5FFE0
2900C2724E4FBF2E0','02C6A396E9E75FA78DAC83D551732DA4BDE5585FB38013
56BA77CAD173036918D085D6FA4AADDCBCA6A9853C5FFD59EF8AD5CF095D5FF
E02900C2724E4FBF2E0','nose','tampocose','esasi',1);

INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(1,'Aldi','55555555','Aldi online passwords');
INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(1,'Mercadona','66666666','Mercadona online passwords');
INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(2,'Tesco','77777777','Tesco online passwords');
INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(2,'Asda','111111111','Asda online passwords');
INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(1,'Lidl','22222222','Lidl online passwords');
INSERT INTO t_passwords (ida_fk,passname,pass,description)
VALUES(2,'Sainsburies','52354354354','Sainsnburys online passwords');
```