Providing Everyday Users an Easy to Use and Adaptable Mobile Credit Score Application

Dragana Ruzic

BSc. Software Design 2019

Athlone Institute of Technology

Declaration

I hereby certify that the material, which is submitted in this thesis towards the award of BSc. Software Design, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above named award.

Future students may use the material contained in this thesis provided that the source is acknowledged in full.

Signed…….

Date…….

Abstract

Credit scores are becoming more and more important in our everyday lives. This study is set out to investigate the different types of credit rating applications and companies within the European and American economy. Companies are finding themselves in trouble with lenders and access to credit scores. A newly developed Score 9 system is in place to better the lives of individuals and set out to incorporate more categories into calculating the ratings.

A research method of different online and mobile applications were used to access the information gathered to calculate credit ratings.

The introduction of a mobile credit score application with accurate information based on European and American bank information and credit categories will allow users to effectively find out information required for acquiring loans and any other finances.

Acknowledgements

I would sincerely like to thank the following people for all of their help and support in the last seven months.

1. My supervisor, Mohammad Samar Ansari, for all the guidance and support.
2. My course colleagues who, through laughter and joy, helped and assisted me during this time.
3. My partner, Adam Timpson, for all his words of encouragement and constant pestering.
4. My family, my sisters and my parents, for their constant support and enthusiasm towards everything I make and design. Without them, none of this would have been possible.

Table of Contents

Declaration……..…2

Abstract……...3

Acknowledgements……..4

Table of Contents……..5

List of Figures……7

Chapter 1: Introduction……8

Chapter 2: Background Research……...10

2.1 Introduction……..10

2.2 Seminal work……...10

2.3 Current work……12

2.4 Problems in the area of credit scores……13

Chapter 3: System Design……15

3.1 Introduction……..15

3.2 Requirements……..15

3.3 Architecture……...16

3.4 Design……..18

3.5 Implementation……19

Chapter 4: Testing and Evaluation……22

4.1 Introduction……22

4.2 Testing……22

4.3 The Application……..23

Chapter 5: Conclusions……25

5.1 Introductions……...25

5.2 Recommendations…….26

5.3 Reflections……...27

References……28

Glossary……29

List of Abbreviations……..……31

List of Figures

Figure 1: Credit distribution by age

Figure 2: Architecture flow chart

Figure 3: Use case diagram for Credit Score Application

Figure 4: App Landing Page

Figure 5: App Register Page

Figure 6: App Main Menu

Figure 7: App About Page

Figure 8: App Tips Page

Figure 9: App Questions

Figure 10: App Help Page

Figure 11: App Answers – Saved

Figure 12: App Answers - Updated

Chapter 1: Introduction

Central databases all around the world keep records given by banks and other lenders about borrowers. These databases are operated by credit agencies like the Irish Credit Bureau in Ireland. The information kept in these databases is a brief history of an individual's re-payments. This information is used to generate credit rating or a credit score which indicates to banks and other lenders whether the individual's credit repayment is good or bad. The databases are only available to banks and lenders, not to the public. The issue with this is that as a person of the public, it is difficult to access a score as easily as that. There are waiting times and forms to be filled out. To beat this system, I have developed a mobile application that generates an estimated score based on a series of questions. I believe that my application will fulfil the needs of individuals who want quick knowledge of their scores before they go to a bank for a loan or any other type of credit needed. I will take you through a couple of brief points that users should know before they use the application.

Credit bureaus around the world that issue these scores all have different evaluation systems and each of them are based on different factors. From multiple sources online, especially the FICO score (American) [1], the primary factors used to calculate these scores comes from mostly your credit mix, applications for new credit, re-payments, current debts and in Ireland only – your job income. Although most places base the score off of those primary factors, each issues different scores. Unlike the European and American ratings, Canada use a scale of 0-9 to rate your credit. I have not chosen to do it this way as I believe a score from 0 to 810 will impact the individual more.

The breakdown for all these factors is not the same, even though they use the same information. More percent is given to the more important factors such as credit repayments rather than the 10% that is given to your job income.

This application will provide an easy way to access this information with also a bonus of how to improve if you have a poor score. I feel that this will help individuals increase their score and help them sort out priorities when it comes to credit.

Why is a credit rating important? You may ask. Well, when you apply for a loan like a mortgage or a credit card or even a simple bill-pay phone plan, your credit rating is

checked. By checking this score, lenders and banks will determine whether or not you are a risk to them, and how big of a risk. For example, somebody with three loans who pays the right amount each month or week is more likely to be given another loan compared to somebody who has three loans but doesn't pay the right amount or on time.

The aim of my application mostly is to help people who either don't know what a credit rating is and what it entails as well as helping people who would like to keep track of their bills, payments and their score. By doing so, a person will always be able to view their score and update/edit their details. So say for example if a person has paid off their mortgage it would affect their score greatly. This was the person is able to tick off their mortgage and their score increases or decreases depending on the other components. I will explain the idea and different applications throughout the next section.

Chapter 2: Background Research

2.1: Introduction

There are a couple of important factors to take in when researching. I go into detail about previous work done in this area, as well as new and upcoming work being done and the problem I think is most evident in the area of credit scores.

2.2: Seminal work

In relation to the seminal work in this area, there are many online resources for finding your credit score. Each one has the same usual question and rating scenario – which talk about your spending, income, credit application and so on. Most of them are across the sea so they haven't got the same requirements as Ireland would have. The most known credit rating application is FICO [2] scoring system. Based in America, it is most known around the world. FICO credit scores range from 300 to 850. FICO also do scores for companies as well as individuals. It is based on three national credit bureaus which are Experian, Equifax and TransUnion. It is different to the way Ireland calculates a score in that FICO's aim is for banks and credit grantors to check their clients. However you can still use the online tester [2] to check your score.

If we take into account age groups in reference to credit scores, they seem to improve the older you get according to the FICO system I mentioned earlier. The only thing that is odd is that between the ages of 30 and 39, 41% of scores are 621 or under. This seems to defeat the stereotype that as you get older your score gets better. According to ValuePenguin [10], 38% of people 30 or younger, have a score under 621. This is due to a CARD act where it makes it more difficult for 18 to 21 year to open credit card accounts. This leads to younger people having worse credit scores as there is no proof of history. If you take a look at the chart in *Figure 1* below, you will notice the change in the ages and the higher the score is for older generations.

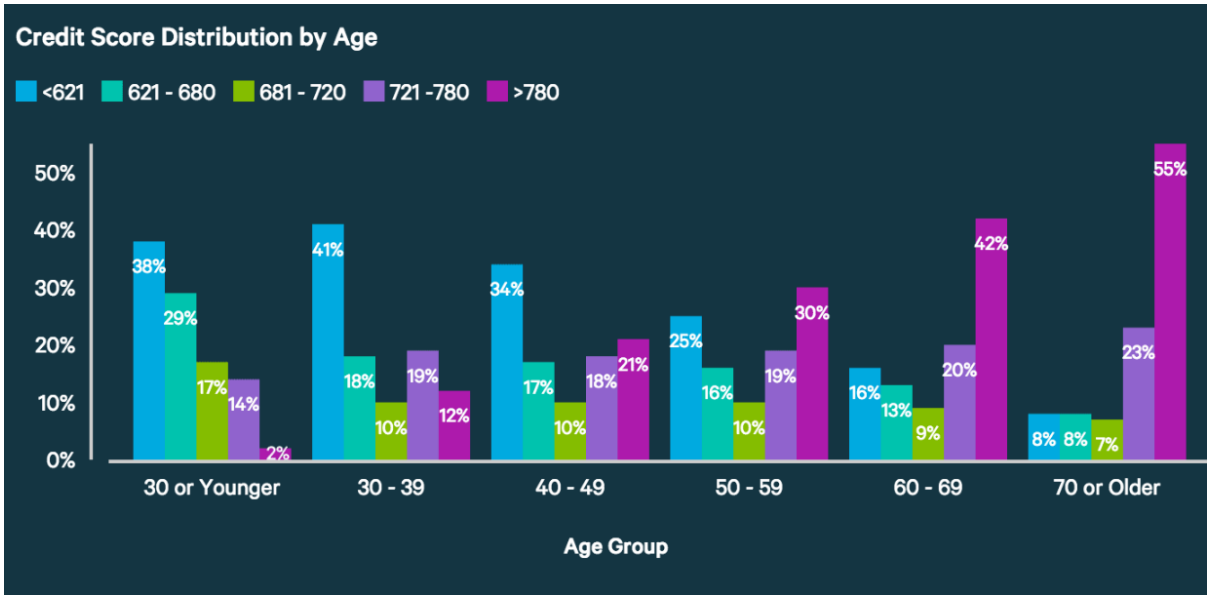


Figure 1: Credit Distribution by Age [10]

In Ireland, the Central Credit Register [3] is a form of credit scoring in which helps stop loans being given to individuals who can't or refuse to repay them. It was set up in 2017. To get a credit report it is free however you have to fill out an online form to request a rating. Since 2017, all loans must be recorded and stored in the database. All lenders that are giving out loans for €2,000 or more are obliged to enquire about their client and see their credit report. This register also includes business loans. Obtaining a credit rating from the CCR (Central Credit Register) is free however with these reports, one does not get a set score like most places where you would have a range of numbers for your score.

Before the CCR, the Irish Credit Bureau [3] held all the information included in your credit score. Before 2018, the cost of acquiring a credit report was €6, however it is now free. Like all other places, one can ring the ICB (Irish Credit Bureau) and request one or get an online application. Unlike the CCR, the ICB has a scoring system. There is no known reason as to why they no longer give scores. A high score is positive and a lower score is negative. Both registers operate side by side, however, the ICB will soon be removed.

By searching "credit score rating free" on Google, you can easily find a couple of questions and answer them without. The problem with this is that not all of them are accurate or accurate at all and most require an individual to enter their email and from

personal experience, it puts people off as it's more of an effort. Due to this, there isn't many online websites other than the FICO system that are of much help.

Many mobile applications have been put on the play store for easy access to gain a report. With over 100,000 downloads and the most popular rating on the play store, *Credit Score Report Check: Loan Credit Score.* [4] As well as calculating your score, this application lets you buy credit score, get a company credit report, apply for a loan, and more. It is India based so it's not the exact same as the Irish applications would be. Most of the applications on the play store are foreign to us in Ireland as the Irish banks and lenders are very reluctant to give out this information as easily as others would. Even though many of these mobile apps are deemed to be inaccurate, it is still giving people a chance to get an estimated rating and most of them have advice on how to grow a better rating or build it up. Similar to these credit score applications, there is one app that doesn't calculate your score but it gives advice and tips on the score. It has sections like credit card fraud, how to avoid credit card debt and common credit card mistakes. *Credit Score Advice & Tips* [5] is an application which gives you advice on numerous subjects relating to credit scores. I find this similar to my application because there is no point calculating a score if there's nothing to follow up and help you understand the result you have achieved. This is also great for first time credit card holders. Usually a person who has their first credit card has no idea about credit ratings or how they will affect their later on life.

2.3: Current work

There is one very important section that is being developed upon in the data analytics company, FICO. As I had mentioned in my previous paragraphs, FICO is a company focused on credit scoring services. It is the most popular one in America. According to National Mortgage News [7] lenders purchased more than 30 million FICO scores in the US. Although FICO have scores relating to many things more than credit like insurance, cyber risk and safe driving, the topic I am focusing on is how their credit for individuals is calculated, simply known as the FICO Score. The latest US version that is being worked on now is the FICO Score 9, which is the current and most predictive score at the moment. In relation to previous FICO scores, the Score 9 is a much more powerful and predictive result. New in Score 9 [8], there is an addition of a scorecard for consumers with high debt as well as further refined thin file treatments which

address lenders desire for more effective risk assessment for consumers with limited credit history which is a good idea as there could be many young people that won't be given any scores or credit because they have no previous history of how they well they pay off their debt. As of right now, there has been no more development on any other web-apps or mobile apps other than the FICO Score 9 which will suffice for another few years.

2.4: Problems in the area of credit scores

The problem in this area of credit ratings is that most are very hard to access and as you can tell, the ones that are accessible quickly aren't very accurate. My application has been carefully crafted to fit the needs of single people, individuals who have their first loans or who are worried about their credit rating, families and so on. This application contains everything from your job income to credit applications. The most important information is kept in this application and perfected to give you an estimated result of your score. Every application differs because each has unique points given to each section. The points I have assigned to each question and answer is gathered from the FICO system and the ICB. The real problem banks and other lenders have with credit scores is the astounding amount of errors and the human error in typing. With our very own FICO system we know of, Equifax credit bureau exposed the personal information of 143 million consumers [6]. This was a huge data breach and not many people are able to trust sources like these nowadays. Due to this, lenders are reluctant to give people loans because there could be major errors in the score that could make your 700 rating go down to a 300. This can be because of human error where somebody logging in information got a number wrong or a date. This is very serious as it could affect many people in different ways. I understand that a lot of people trust banks more than they would an online application or a mobile application especially of the GDPR rules that have come in last year, however banks aren't correct 100% of the time. What I aim for my app to do is allow people to log in their details and have a safe and secure environment to grab information from. As well as this, the application is free and even though a user must register, it's for their own safety. This is because a person may have one score one day and then pay off their auto loan for example and this would affect the rating which they will then be able to log back into the application and update their score. This way they will always be able to keep up to

date with their money and loans. With a bank, you will have to send emails and fill out forms just to get the same result the application will give you.

Taking in this information, I want consumers to understand that even though banks are reliable, the information can always be attacked and stolen. The app is logging the information on the app only and not on other servers and places where the user does not know of. Breach of personal information is dangerous and affects people very quickly.

In the next section I will go through all the architecture and design required to build this application and the implementation itself.

Chapter 3: System Design

3.1: Introduction

System design is where the requirements, implementation, and architecture models are defined. I am going to take you through some of the components that made the app and how this was all implemented.

3.2: Requirements

I had a number of requirements that I had chosen to put into this application. I will go into detail about each requirement. The first and main part I wanted to go into this app was the scoring system. Of course, I didn't want to just have a calculator on a screen and that was that. From looking at most applications on the play store, each has different menus and information about the application. For my application I had first decided on a login and register page. The reason I wanted to do so was for the user's benefit as well as the admins. I wanted to be able to save the user and their answers and I wanted the application easier on the user so that they were sure that their information wouldn't be leaked and it was being stored somewhere safe. After this, the main menu was next. The reason I had chosen a main menu is so that the application wasn't too plain. Also, there are many people, especially in Ireland, that aren't really aware of what counts into their score so I wanted to show people how it works. In my about section, I explained what the application was and then gave a brief overview of which score's mean which category so for example between 460 and 600 points is categorised as 'good'. I believe every application should have a 'Help' section which shows the user how to use the app. This defeats any mistakes that can happen when using the app and won't affect the calculations. The main part of the application like I had mentioned, the calculations, is where the score is gathered. There was a lot of ways I could have made this section. There was examples of just a question and buttons, questions and text areas for users to type in their answers but ultimately, I decided to give the users choices and allow them to click whichever applies to them. I did this because I thought about other applications that have text areas and I thought there would be a lot of human error along with this and it wouldn't have made the calculations as accurate. My choice of design was to ask users a question and have drop down boxes with typical or expected answers that would be relevant to the

question and the user. There are eight questions and seven of them are drop down boxes and the other one is a checkbox. For me personally it seemed more practical to have it shown this way. I also decided for the benefit of the user, to add a page with tips on how to make your score better. This can apply to any individual as there are multiple possibilities. This way, any user can just log in to see the tips if they just wanted to have a look and not do any calculations. Instead of just sending out a score I decided with the results to show the score on a gauge meter as well as some advice based on the level of the score the user has achieved. The user can then re-run the application in order to update their score which can result in it becoming better or worse.

I will now talk about the architecture of the system and explain in detail the flow of how the pages connect.

3.3: Architecture

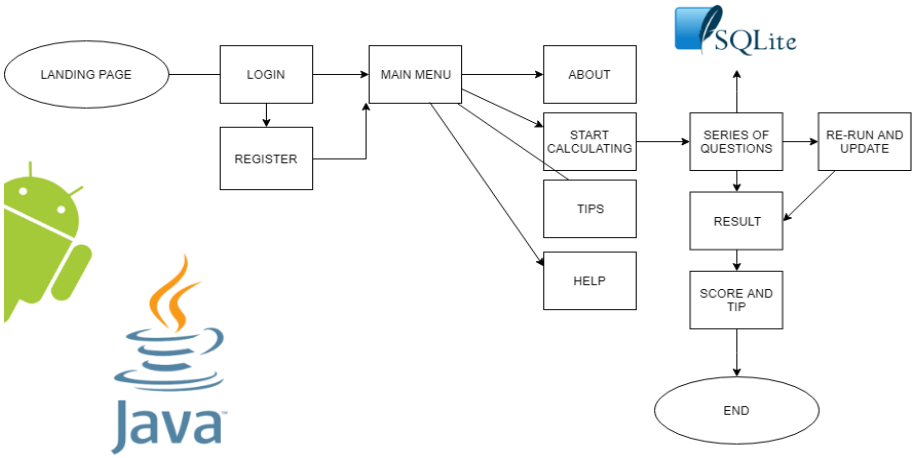


Figure 2: Architecture - Flow Chart

The system architecture of anything refers to the structural layout of how an application is designed. If you refer to *Figure 2* above, it is the layout of the Credit Score Application. The application is built on Android Studio, Java and SQLite. The app starts off with the landing page where the login and register buttons are. After logging in and registering, the app takes you to the main menu which has four different sections – about, start calculating, tips and help.

After you start calculating, you are given a series of eight questions and gathering a result from this which gives the advice and tips. After exiting the application, the user can re-run it and update their answers which could result in a better or worse credit score.

The use case for this application is typically anybody who is interested in acquiring their credit score. Usually, this could be anyone over the age of 18 that has a bank account. It's very difficult to calculate a credit score with no credit. The need for a credit score could arise from very simple things like wanting to buy a bill pay phone. If somebody with high debts and many loans wants to buy a bill pay phone, the chances of them being allowed is very low. This is because of the way their history has shown that they don't pay their debts on time or in full.

In *Figure 2* you can see a very simple use case diagram for the Credit Score Application. It shows the various processes in the application which are login, register, calculate score and update details. These are the only sections where the user can interact with the application hence why the other sections aren't included.

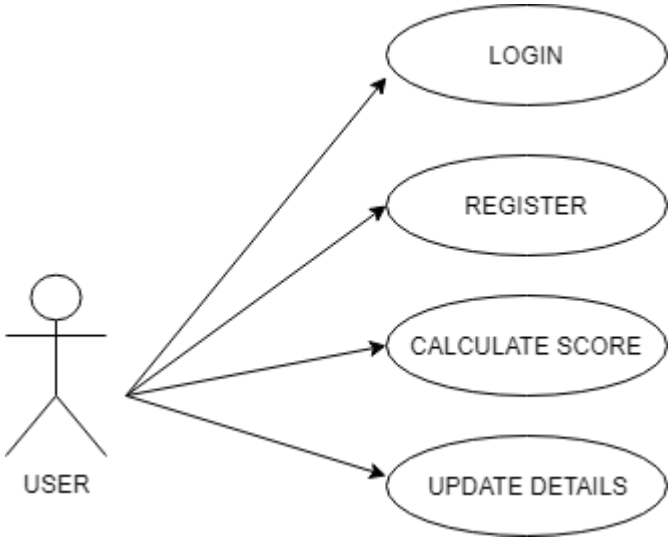


Figure 3: Use case diagram for Credit Score Application

The application is designed easy so that anyone will be able to use it. There is nothing complex about the way pages connect and how they relate to one another. I will now talk about the design of the application and mention how the behaviour of the pages works with the functionalities of the pages.

3.4: Design

The design of the application is an important part that needs to be mentioned. I am going to take you through the different behaviours of each page and their functionalities. I want to mention first the pages that don't have functionality to them and are just static pages. These are the about, help and tips page. These are pages that I felt were necessary to add into the application as they are important to help the user navigate through the app and make sure they know what they are doing. The about page is there so that the user can get some information on what the app is about and how the scores are made in terms of the category and the numbers. The tips page is there for users who don't necessarily want to go through the question but do however, want to see a few tips on how to better your score or improve on the one you already have – if they have done the calculations already. The help page is a simple few paragraphs that tell the user how to go through the calculations and questions and what their result means. Even though there is no functionality in these pages, they are important for any application that is made. The register page is there so that the user is able to register their account and then login. The behaviour of the register and login page is to simply have the user make their account so in the backend, there is ability to save the users information. The next page is the main part of the app which is the calculations. The design of this application is a very simple, standard design that is easily read and fun to use. I will next just mention a few important implementation parts that made this application to be what it is.

The login and the register pages are crucial to any application. It provides the user with knowledge and security that their data isn't out of breach and it is secure. The register page was designed so that the user will have their own personal identity that will distinguish them from other users. With their identity, it could then save their answers and update them. I designed this by asking the user to fill out a form. This form asked for the user's first name, last name, email address and password. This allows the log in to be very simple. The login allows the user to have a profile and update and save their answers. The login contains just input fields that ask for the email the user entered when they were registering and it must be the same as this is what it is saved in the database, and the password they also entered. By logging in, the user then moves onto the main menu.

The next part I want to mention is the calculations. This is the whole application itself. This is necessary to be able to calculate a score, and the user must supply the answers to the questions. The way this was developed was that there was a list of eight questions in the forms of checkboxes and drop down boxes. The user must select the choice that is most relevant to them.

The final part is the result part and I designed this so that the user is able to see their score and get advice and tips based on where it is in the categories. Once the user gets their results which is in the form of a gauge meter, it comes with a paragraph of advice based on what the score is. In the backend, there is an algorithm that calculates the details the user submitted and the result from this is what is given back to the user. This happens when the user clicks the submit button on the calculations page. The user can also re-enter the application by logging in and change their details. For example, if they have paid off two loans their score is bound to change so all they have to do is re-run the application and enter their changed details, submit their details and it is updated.

Next, I am going to talk about the different sections of the applications and how they work and what I used to develop them.

3.5: Implementation

To develop this mobile application, I am using Android Studio. Android Studio is a Google integrated development environment made for developing mobile applications. It is built on JetBrains IntelliJ IDEA software. The reason I chose Android Studio is due to the fact that it is highly recommended over all the other platforms. It is a free IDE which is useful as not everyone likes to or can afford to pay for programming platforms. The choice was between Android Studio and. I was considering using Visual Studio in order to develop and strengthen my Visual Studio C# skills, however, Android Studio seemed like it had more support when it came to developing the application. Android Studio has gradle build support which builds android packages and manages dependencies. It also has a visual editor and flexible build system. It uses XML and Java. I wanted to build my skills on my mobile app development and this is why I chose to make it on the Android platform. I think that compared to web applications, mobile applications are easier to access and it is easier to access the things you need to see.

For example, a lot of people now use online banking on their phones instead of on desktops. I am using my own Android phone to run the application. I'm going to take you through the implementation and what is used to implement each feature. There are certain pages that only used XML to be made. This includes the tips page, the about page, and the help page. These were built using XML in Android Studio. Whereas, the login, register, calculation and result pages used more than just XML. The login page used XML for the layout and the design. However, these each also used Java and the SQLite database to help cater to each page need. The main parts of the project all use Java, XML and SQLite. The main page with the login and register use XML for the design and Java for the logic behind logging in and registering. In order to save the password, username and email to the database, SQLite uses the local server to store these details and gives each user an assigned id which auto increments when a new user is made. When the user is logged in or registered the application takes you straight to the main menu instead of making the user log in again or exit the application. The Java logic behind letting the user log back in again is because the password and the email are both checked to make sure they are the correct details and this information is gathered from inside the database. The database used is SQLite because as this is a local server application, it's faster and more practical to use compared to SQL and having a database that is on the server. The database helper class was made so that the two tables (user and answers) could easily be created because without creating these, there wouldn't be any possibility to save the user or the answers.

To save the answers to the database, a cursor was used. A cursor is similar to a ResultSet in SQL in a way that it can be thought of as a pointer to a specific row in a query result. There is different ways of saving the details to make sure each part is memorised. So the database for example has methods to save the user id and then the answer that comes along with it. As for saving the users to the database, it needed more methods to make sure that all areas are covered. There is a list of users to start with and when a new user is registered, they are added to the list. By doing so, the application is able to run through the list of all the users and get one certain user which is necessary when you want to save and update their data. The id and the password of the user is all based on the email. If the user provides the correct email, the

database searches for that id and password connected to that email and this is how the log in works.

The result is the most important as not only does it use the three components, XML, Java, SQLite, but it also uses an outside library that is not developed by Android. This library I found on GitHub[9] and it was developed by a person called Pawel. It is basically a way of showing a meter with a number. After a lot of negotiating, the decision to use this type of meter was because a number on the screen would have been too plain however the gauge has a kick to it and even shows the number on the meter which is good visually as people are more drawn to pictures rather than words. The way this gauge was implemented was through custom attributes made by the author of the gauge and this was implemented through Java. It had a number of view attributes that were easy to add to the class. To actually make sure that the gauge is implemented properly, there had to have been a change made to the build.gradle file which is a copy and paste from GitHub. The only problem with this was that the attributes couldn't be applied in the XML, they could only work if the attributes were in the Java class. Since all of the scores were set on a scale from bad to excellent, there was numbers set to each score and this was added in the Java. The application is built on Android Studio, XML, SQLite and Java.

Next, I am going to discuss some of the testing involved in this application.

Chapter 4: Testing and Evaluation

4.1: Introduction

Testing is an important part of every system. Any system designed must make sure that there are little to no bugs in the code which will result in the application being flawless. Most of this application is implementation based, however, there were a couple of methods that could be testing through unit testing. I will mention a few of these methods, what they are for and how they were tested.

4.2: Testing

The first method was the entry classes. These were the user and the answer classes. Both of these classes had add, update and delete methods. These methods were all boolean, so they either returned true or false. If there was no user or answer added, updated or deleted, the error would be false. If there was no error, the method returned true and perform the required action.

In the input class, there was one method which was *saveAnswers*. What this class did, was make sure that the user has either submitted new answers or updated their existing answers. The way this was tested was it said if the users hadn't submitted any answers and there was no errors along with it then it returned true and added the user's answers. As well as this, if the user has existing answers and there was no errors, it would return true and update their answers. Anything other than this, that returned false, was an error.

The last and final testing that I had done was in the register class and the method called was *canAddNewUser*. This also returned a boolean, true or false. What this method does is that it makes sure that when you register a new user, it adds it to the database and they are able to log back into the application. How this was tested was that it checks for valid input, for example, the email must have a '@' – at sign – character and depending on what the user has inputted, whether it is correct or incorrect, it returns true or false.

Another testing method I had used was to test the static methods by printing anything to the console. For example if I wanted to make sure that the XML was displaying

properly or if there was any data not sending, I would print “Working” out to the console and if it showed, it meant that it was working and if it didn’t I knew there was an error.

Along with unit testing, I also had many people manually use the application and this way I could find the errors and see what they were struggling to use. For example, if the user couldn’t register even though there was error checking done or if the answers weren’t updating after the user re-runs the application.

4.3 The Application

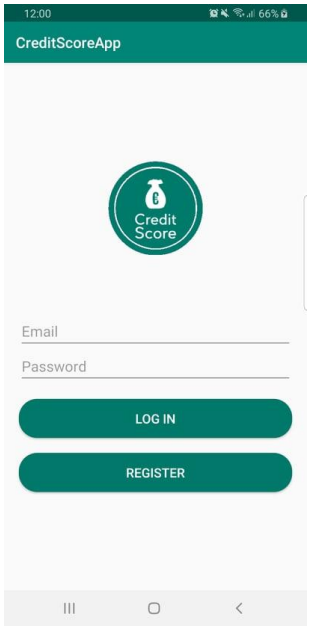


Figure 4: App Landing Page

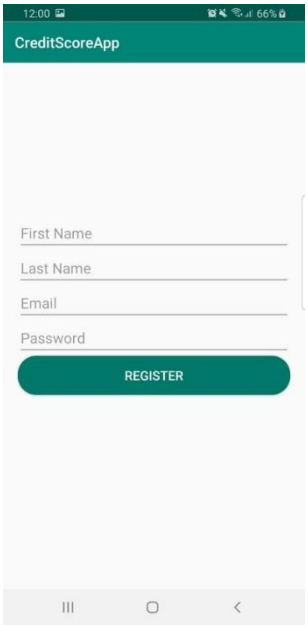


Figure 5: App Register Page

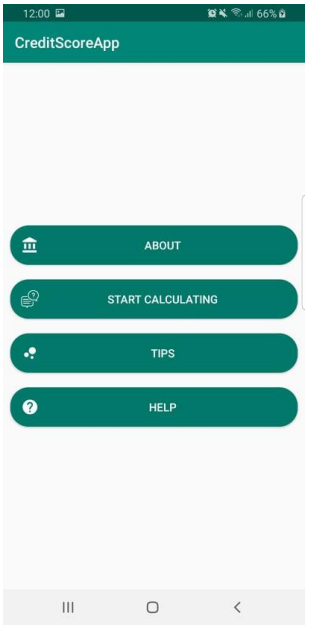


Figure 6: App Main Menu

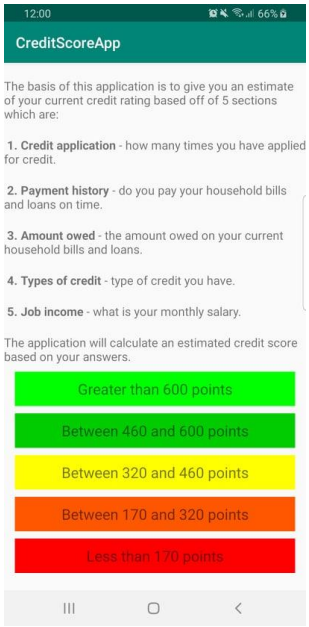
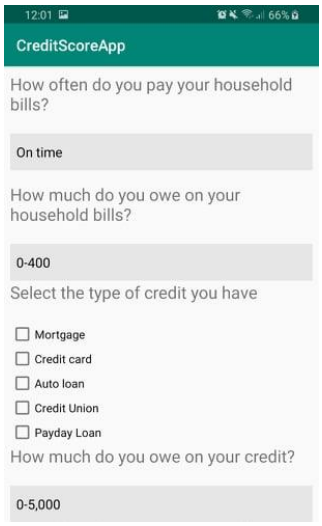
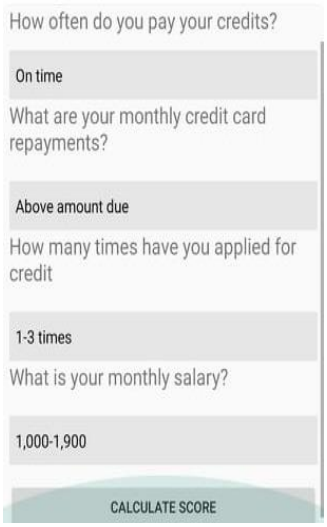


Figure 7: App About Page

Figure 8: App Tips Page

Figure 8.1: App Tips Page





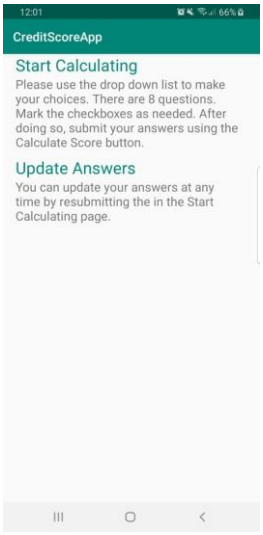
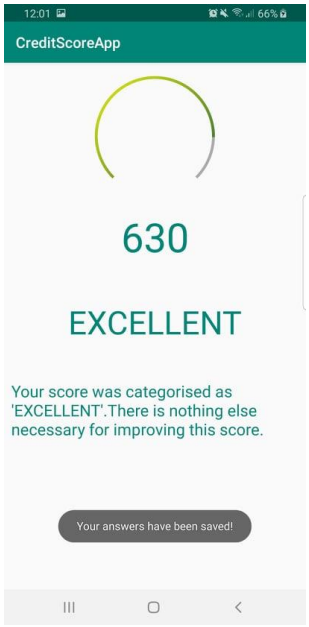


Figure 9: App Questions Page

Figure 9.1: App Questions Page

Figure 10: App Help Page



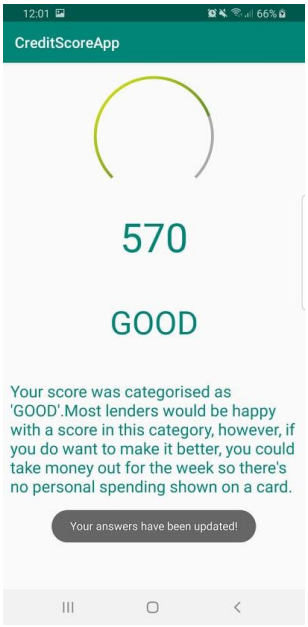


Figure 11: App Answers – Saved Page

Figure 12: App Answers – Updated Page

Chapter 5: Conclusions

5.1: Introduction

All in all, this application is designed for individuals who are either entering the world of work and want to understand more of how their use of money affects their credit, individuals who don't even know what a credit score is or anyone who is just interested in learning more about credit scores. I have designed a very simple and soluble solution to many credit problems. I believe that my application will help many people sort out their finances and learn to budget. I have developed an application that not only gives you your estimated credit score, but also supplies you with information on how to make it better or improve on it. In this thesis, I have talked about the application itself, how it was designed, which technologies it used and the need for it. I have talked about credit ratings in Ireland and where the idea arose from. I have provided structure to the system and explained in detail how the flow works. It is a very simple but very effective and important idea as it's used all over the world. Sometimes, even used wrong.

The conclusion that can be gathered from developing this application and spending time on it is that credit ratings are very important. They're not only important if you're buying a house but they're also important if you want to set up a bill pay phone plan. If lenders in a bank see that you spend a lot of money on unnecessary items (for example a lot of online shopping), they are very reluctant in giving you money as they aren't so sure whether you will be able to pay it back. The main point of this application is that people will understand how rates will affect them and what they should be mindful of.

The fact that this application has a user log in in it is good as they will always be able to revert back to it for reference. Most details will always stay the same, the formula for the credit score and the categories that are in it as they are all based on what is new in society and the economy. Like I had mentioned before, Irish credit scores are very difficult to find and crack so the only way to incorporate all the details is to research as much as possible to make sure all sections are covered.

Even though there are many applications like mine on the play store, on Google and banks of course, my application is there so that whenever and wherever a person is,

they can log in and check what they want whether this be the calculation or just to check the tips or see if anything has changed, the app comes free of charge unlike the banks that charge €6 or any type of fee and a form to fill out.

Now that the app is developed, banks could scrap their free and form and use the app. It has security and the database is ready so that all details will be saved. Even though banks do use a database that is secure, this defeats human error as the forms that are filled out are by individuals themselves and for example if a person writes the wrong number or the wrong type of loan it will affect it greatly.

5.2: Recommendations

In terms of recommendations, if somebody else was to develop another application there are a few things that I would recommend to have and not to have. The first would be that there should be a user profile. I think this is an important feature to have because it would seem like a more updated app. This is where there should be a side bar navigation that has options to update answers or delete profile for example. I feel like there should also be an option for the user to write in their own expenses. For example if there wasn't enough types of credit covered and they had another existing loan that wasn't in the list, they should be able to add this. In the database, this could have a certain amount of points set to it or let the user add their own and it would be estimated based on what they put. This is a tricky situation to have but maybe in the future there should be more types of credit or job salaries for example, to be added into the application.

One interesting point to note, I think there should be some sort of additional page for first time credit holders. This would set them on a good path to adulthood as they have a good start and understand what credit is, how it's calculated and how important it is. Even if a first time credit holder just wants to check what it's about, they can.

Key things to note when developing an application like this is to make sure you are one hundred percent sure on the categories that are involved. Even though the numbers are estimated, for example between 460 and 600, it is still better than the number being hundreds difference. You should also make sure Ireland haven't updated their details or the likes of the FICO system haven't updated their details because it is crucial that you are up to date on everything that is involved with credit

ratings. Just like human error when inputting the details wrong, there could be error because you haven't updated the details.

Another thing that would be interesting to implement, is a place for lenders on that application. The lenders could log in and have a profile, same as the users checking their scores, and they could be able to log into a bank API and the database, and check if a certain person is eligible for a loan. This takes time off of their hands for contacting banks and filling out forms, sending forms, receiving forms, and instead of all this, a lender could easily just check the app, type in the applicant's id or name and see their details for example they could check how well the person pays off their household bills or how many times they have applied for credit and then just based off of that, either allow them to borrow money or deny it.

5.3: Reflection

Developing this application taught me a lot about not only the technical parts but also about finances in Ireland. I learned a lot about what is being checked when you're applying for loans and how much of your information banks really keep. There are strict rules in place and if these aren't followed, it can be really difficult to do simple things like I mentioned before, buying a phone or applying to rent a house.

On the technical side of things, I learned a lot about Android Studio and developed a greater knowledge of the Java in Android Studio. I chose to develop this application in order to strengthen my Android skills. I believe I have achieved this as there are many things I know now that I didn't know how to implement before I started this application.

In conclusion, there should be more awareness in relation to credit ratings in Ireland. There should also be easy access to how this information is gathered to make the algorithm that calculates the score. I believe it should not be hidden in banks and not given out to the public. All people want is to make their lives better and in order to do this, banks and other financial institutions should allow their customers the right and access to this information.

References

[1] howstuffworks. 2002. *How Credit Scores Work*. [ONLINE] Available at: https://money.howstuffworks.com/personal-finance/debt-management/credit-score1.html [Accessed 12 March 2019]

[2] myFICO. 2001. *Free Credit Score Estimator: Get Your Estimated FICO Scores Range.* [ONLINE] Available at: https://www.myfico.com/fico-credit-score-estimator/estimator/ [Accessed 23 March 2019]

[3] Money Guide Ireland. 2018. *How to check your credit rating in Ireland.* [ONLINE] Available at: http://www.moneyguideireland.com/how-to-check-your-credit-rating-in-ireland.html [Accessed 23 March 2019]

[4] Delux App Zone. 2017. *Credit Score Report Check: Loan Credit Score.* (1.8) [MOBILE APPLICATION SOFTWARE]

[5] Alex Dabek. 2017. *Credit Score Advice & Tips*. (2.2) [MOBILE APPLICATION SOFTWARE]

[6] Personal Finance. 2017. *The real problem with credit reports is the astounding number of errors.* [ONLINE] Available at: https://www.cnbc.com/2017/09/27/the-real-problem-with-credit-reports-is-the-astounding-number-of-errors-equifax-commentary.html [Accessed 23 March 2019]

[7] National Mortgage News. 2014. *New FICO Scoring Model Coming This Summer.* [ONLINE] Available at: https://www.nationalmortgagenews.com/news/new-fico-scoring-model-coming-this-summer [Accessed 18 April 2019]

[8] FICO Score 9. 2019. *Some of What's New in FICO Score 9.* [ONILINE] Available at: https://www.fico.com/en/resource-download-file/4198 [Accessed 18 April 2019]

[9] Custom Gauge. 2018. [ONLINE] Available at: https://github.com/pkleczko/CustomGauge [Accessed 19 April 2019]

[10] Value Penguin. 2019. *Average Credit Score By Age.* [ONLINE] Available at: https://www.valuepenguin.com/average-credit-score [Accessed 23 April 2019]

Glossary

Credit Score: *number that reflects the likelihood of how well you pay your credit back*

Seminal work: *previous work already done in the research area*

System architecture: *formal representation, the structure and the behaviour of the system*

Use case: *what way the product or service could potentially be used and by who*

Static pages: *a page in which no user input is required*

Backend: *part of a system or application that is not directly accessed by the user*

Algorithm: *a process or set of rules to be used in calculations or formulae*

Android Studio: *an IDE used to develop Android applications*

JetBrains IntelliJ IDEA: *an IDE used to primarily develop Java applications*

Visual Studio: *an IDE used to develop computer applications*

Java: *programming language used to develop computer applications*

XML: *human and machine readable mark-up language*

SQLite: *relational database management system*

C#: *programming language used to develop computer applications*

ResultSet: *retrieve data from a set of row in a database*

Cursor: *retrieve data from a set of row in a database*

Build.gradle: *custom tool to build android packages*

GitHub: *web-hosting service for version control*

Java class: *templates used to create objects and define data types and methods*

Unit testing: *individual components of a piece of software are tested*

Boolean: *java object used to hold either true or false*

Bugs: *an error or flaw in a piece of code*

List of Abbreviations

FICO: *Fair, Isaac and Company*

CARD: *Credit Card Accountability Responsibility and Disclosure*

CCR: *Central Credit Register*

ICB: *Irish Credit Bureau*

GDPR: *General Data Protection Regulation*

IDE: *Integrated Development Environment*

XML: *eXtensible Mark-up Language*