Creditly

Mobile Credit Scoring and Recording

098420

Group 4B

An Information Systems Proposal Submitted to the Faculty of Information Technology in partial fulfilment of the requirement for the Bachelor of Business Information Technology at Strathmore University

Date of Submission: January 2021

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, this proposal contains no material previously published or written by another person except where due reference is made.

Student No: 098420

……..... [Signature]

……..... [Date].

Approval;

The proposal of 098420 is submitted for review and approval:

Faculty of Information Technology, Strathmore University

Abstract

Credit Scoring has in recent years become a very important issue especially with the growth of the money lending sector. The harsh economic times have dictated the need for more lending due to the very static disposable income people have against the ever-changing economic factors. Interest rates charged by banks and digital lenders are pretty high which has led more and more people from considering the sector when it comes to lending but they look to family and friends. Companies are also finding themselves in trouble with lenders with access to credit scores. A newly developed credit scoring 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 was 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. Methodology to be used in creating the system was prototyping as it will be possible to carry out initial tests and ensure at early stages all functional requirements are met. The prototypes will also allow for feedback to changes to be made in the system to improve the quality of life. This study is set to form a large step towards debt management in the country and credit score literacy.

Table of Contents

List of Figures

List of Abbreviations/Acronyms

API	Application Programming Interface
CBK	Central Bank of Kenya
CSE	Credit Scoring Engine
KNBS	Kenya National Bureau of Statistics
MFI	Microfinance Institution
OOAD	Object Oriented Analysis and Design
SDM	System Development Methodology

CHAPTER 1. Introduction

1.1. Background Information

Central databases all around the world keep records given by banks and other lenders about borrowers. These databases are operated by credit agencies, banks and money lending platforms. 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.

Over the last decade, lending has become bigger in the country with mobile-based lending platform getting the lion's share of the growth. The growth of mobile lending was accelerated by upshift, growth and investment in financial technology industry which brought forward mobile lending platforms such as Tala and Branch. The mobile-based lending platforms however come with the steepest interest rates in the country and can leave one worse than they were before acquiring the loan. This makes people look off to other sources of financial aid. Some studies tell us that many Kenyans are in financial despair due to the tough economic times that have been experienced in recent years and they are looking at friends, relatives and neighbourhood kiosks as financial barricades the economic downshift, E. Okoth (2019, April 07). The most recent household survey on financial access by the CBK, KNBS and FSD Kenya saw that majority of Kenyans can hardly save and are ill-prepared for economic shocks and will immediately to turn to friends and relatives when they meet any sort of financial crisis, CBK, KNBS and FSD Kenya (2019, April 03).

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), 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.

1.2. Problem Statement

A large population of Kenyans remains unbanked and only large institutions that offer lending facilities have the resources to be able to generate a credit score for someone. The scores that they generate are not even made public to the people with whom they relate to. It is the lack of this proper means of credit scoring for the common folk is what brings out cases much like the man who married off to settle sh.900 debt, Citizen (2014, February 28). We experience other cases in our day to day lives where two friends are bashing each other due to an unpaid debt that was given a while back. All this conflict comes from the fact that one of the parties is unable to meet the financial obligation of some money he received and the other party tired of waiting for payment previously promised to be made at an agreed point in time. This is what informed the need to have a system that individuals can easily access and can be that little bit sure about the decision to lend to another individual. The system is meant to increase the level of financial prudence a user has.

1.3. Research Objectives

1.3.1. General Objective

The problem highlighted in section 1.2 above gives us the need for this study. We want to mitigate the problems encountered when lending to or borrowing from one another by generating a reliable credit score that has the backing of the personal debt payment record in recent times accessible from a mobile phone. This should create financial prudence where people will be able to make better choices financial by being aware of their credit score levels

1.3.2. Specific Objectives

i. To investigate credit scoring methods currently in use in lending institutions and mobile lending platforms
ii. To analyse credit scoring algorithms that can be used to create a reliable credit score
iii. To develop a mobile credit scoring and recording system
iv. To develop the system

1.4. Justification of the Study

The debt crisis is not new to us as a nation and even to us on a personal level hence the importance of a system that can mitigate the same. It is aimed to start at the core of it all where systems that have already been put in place have been able to reach the root. By making an individual have the power to discern with truth the ability of a debt to be paid we are making steps towards the right

direction since this is what will inform lending in the following years when used correctly otherwise debt management will continue to itch us to the core.

1.5. Scope of the research

1.5.1. Scope

This project is set to create a mobile application that will competently create a credit score on an individual who is using the application. The credit score generated by the application will inform the user of their credit level and how to better it if need be

1.5.2. Assumptions

This project will assume that any one person who will be using the application is honest about selections he makes as credit scores are made based on selection of credit values that are given

Users will begin to build their knowledge on credit scoring through the application

1.5.3. Limitations

The application will only be usable on android smart devices but a segment of the population still does not use smartphones

The credit score generated is not accurate but an estimation based on selections the user makes

Manual entry will be done in order to get values needed for the credit score hence reliance on the honesty of the user to give a good credit score estimation

CHAPTER 2. Literature Review

2.1. Introduction

In this section, we will be differentiating credit scoring and credit scoring models. We will also be learning how credit scoring is being put into use and how it has been applied in the sectors where it is being used. The different credit scoring models used in different sectors understanding why a certain model was selected over another and the credit scorecard models.

2.2. Credit Scoring vs Credit Scoring Model

To create an effective credit scoring system one has to understand the difference between credit scoring and a Credit Scoring Model.

Credit Scoring is the use of statistical data models on performance data from previous loans acquired to predict a borrower's performance on loan to be acquired in the future by giving scores to the borrower known as credit scores as a means of evaluation of the borrower's ability to perform in the future loans

A credit scoring model is a tool used to make loan approval or rejection decisions. The tool uses a statistical model based on borrower's information on previous loans whether good or bad and uses this information to generate a credit score which will act as a probability of a borrower being a defaulter. Credit Scoring Models are what enable credit scoring to occur for it is in this model that scoring occurs and scores are generated Schreiner (2004).

2.2.1. Theory of Credit Scoring

When it comes to borrowing the lender is at a disadvantage as opposed to the borrower this is because beforehand the borrower already knows their risk level in terms of payment of the loans. Borrowers can even avoid disclosing some information to the lender if it would mean that they will be able to benefit from the loan these factors make lending generally risky. It is these factors which will then inform the need for there to be checks and balances that will allow the evening out of the playing field when it comes to lending. One of the ways this could be done is by increasing the interest rates so that a lender would be able to shield themselves from defaulters but this, in turn, would only draw in high-risk clients and push away "good clients" who would be having the fear of reimbursing the credits due to the bad rates. Another factor is that the high-risk client could adopt opportunistic tendencies that would affect their ability to reimburse the loan

This inability to control a borrower in terms of achieving the purpose and goals of the loan is what informed the need for credit scoring. Credit Scoring will give the lender the ability to generate a risk profile specific to each individual client and this will guide the lender in making rate specific to individuals when giving the loans hence alleviating this problem (Gool et al, 2009).

In order for the credit scoring system to work effectively, a good credit scoring model will have to be put in place. The model is what will dictate how the scoring will be done and different models will better suit different lender hence here we take a look at them

2.2.2. Credit Scoring Models

A Credit Scoring Model is a tool that will aid in loan approval and reject by statistically generating a credit score. Credit scoring models are the backbone of credit scoring as it is within the models that logic for the engine is held. To have a functional credit scoring system you will require a scoring model. This is point is visually aided by figure 1 below that shows the skeleton of a credit scoring system and the result it can achieve

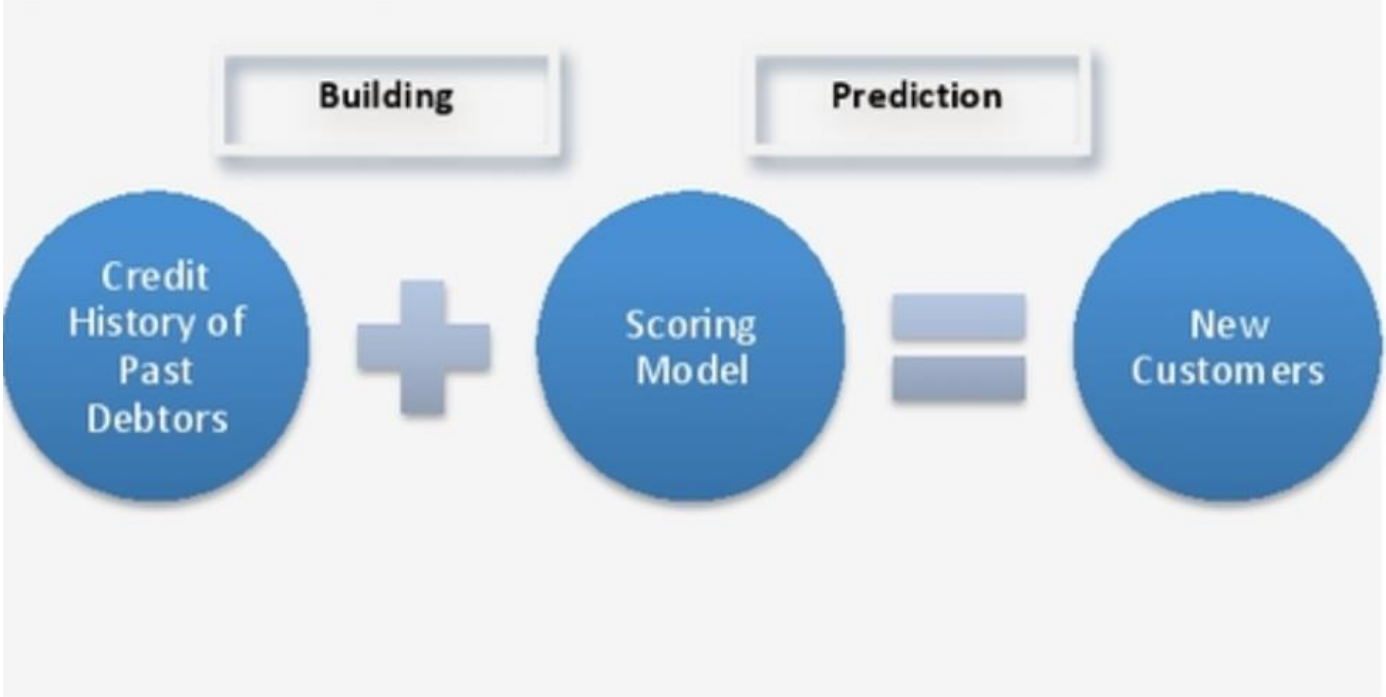


Figure 1. Credit Scoring System by Kinda, O., & Achonu, A. (2012).

From the illustration credit scoring system is built from taking a client's debt history as data this combined with a scoring model will give you a prediction on the probability of default and if prediction is good you will have attracted a new client.

There are several credit scoring models that we will look at, Abedi (2000).

2.2.2.1: *Linear probability and logit models*

This model takes a borrower's past data as its input and uses the data to generate the performance of the borrower in the past repayments. The performances generated are then used in forecasting the probability of the borrower getting other loans, Abedi (2000).

2.2.2.2: *Linear discriminant models*

This model classifies borrowers to different segments hence the name. It forms classes of high risk of defaulting to low risk of defaulting based on previously observed characteristics and then gives a score that distinguishes between good and bad loans, Abedi (2000).

2.2.2.3: *Risk adjusted*

This scoring model will assess the amount of risk the lender is getting themselves into by giving a loan and then goes forward to determine whether the lender are getting enough compensation for the risk they will be having, Abedi (2000).

2.2.2.4: *Option-pricing theory models*

In this model, a borrower's assets comes into play where mostly these are the securities they offer when borrowing. Due to the diminishing value of most assets the model checks the value of the assets, if it falls below the value of the debt still owed by the borrower, he/she is most likely going to default. The model uses the value of borrower's assets as a means to known the likeliness of the borrower being able to meet the obligation of the loan, Abedi (2000).

2.2.2.5: *Neural networks*

In these models, there is the use of machine learning algorithms that allow the models to learn a borrower's characteristics and their probability of default. The model is then supposed to determine the important and non-important characteristics to look at when predicting a client's default, (Baesens et al, 2003).

2.2.3. Credit Scorecards Models

These are widely used as credit risk analysis tools and therefore are very important and needed when generating credit scores. They hold different characteristics of the borrower (age, residential status, the current job) and gauge them as points and the total number of points will form the credit score. The concept of scorecards is shown using the scorecard in figure 2 below

Characteristic	Attribute	Scorecard Points
AGE	<22	100
AGE	22<=AGE<26	120
AGE	26<=AGE<29	185
AGE	29<=AGE<32	200
AGE	32<=AGE<37	210
AGE	37<=AGE<42	225
AGE	>=42	250
HOME	OWN	225
HOME	RENT	110
INCOME	<10000	120
INCOME	10000<=INCOME<17000	140
INCOME	17000<=INCOME<26000	160
INCOME	26000<=INCOME<35000	200
INCOME	35000<=INCOME<42000	225
INCOME	42000<=INCOME<58000	230
INCOME	>=58000	260

Let cutoff=600

So, a new customer applies for credit.....

AGE	35	210 points
INCOME	$38K	225 points
HOME	OWN	225 points
Total		660 points
Decision:		GRANT CREDIT

Figure 2 Credit Scorecard by Towards Data Science(2019)

From the illustration we see how characteristics every client will possess each with its own range assigned a value. A client will fall somewhere within the given ranges and points allocated within said range. Credit score is calculated by just adding the points and by having a cut-off point one can be considered to either have good or bad credit. Credit scorecards can be achieved by different ways;

2.2.3.1: *Expert System*

This type of scorecard model is based solely on judgement and experience of the one offering the loan. The loan officer identifies traits in a borrower and their relation to different types of clients, this informs their decision. It is also known as judgemental scoring because it relies on the judgement of experienced loan officers who will then generate your credit score. It is highly qualitative however, it lacks the predictive power that we need in creation of a credit scoring model.

2.2.3.2: *Statistical Tree*

In this scorecard a tree is made up of leaves, the leaves are in different segments with the segments going from the lowest risk all the way up to the highest risk. The scorecard works in a way that it links a client's past characteristics to some past unpaid dues. The scorecard then speculates that the client's future actions must be a reflection of actions done in the past and with this, the scorecard allows for the prediction of the probability to default. This has some predictive power. It is highly quantitative since it needs past client data to match with for it to generate a default probability

2.2.3.3: *Notation by Principle of Regression*

This uses mathematical formulas to establish the influence of the client's characteristics on the delinquency risk. This scorecard has the highest predictive power of the rest and reveals better than the other two scorecards the links between the risk and the client's characteristics. The scorecard shows whether by increasing or decreasing risk certain characteristic changes with all others held constant.

2.3. Major Application of Credit Scoring

2.3.1. Kenya's Finance Market

Finance is the major player when it comes to credit scoring. Banking, micro-lending facilities do the bulk of lending to people when it comes to the banked population. They use transaction information from a client's account be it earnings and payments to generate a credit score. This is the criteria used in giving loans to different clients. Credit scores are used to price loans specific to users, especially when risk-based pricing model is used.

2.3.2. FICO

The most known credit rating application is FICO 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.

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, 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 couple with the fact that they do not have any income.

This leads to younger people having worse credit scores as there is no proof of history. If you take a look at the chart below, you will notice the change in the ages and the higher the score is for older generations.

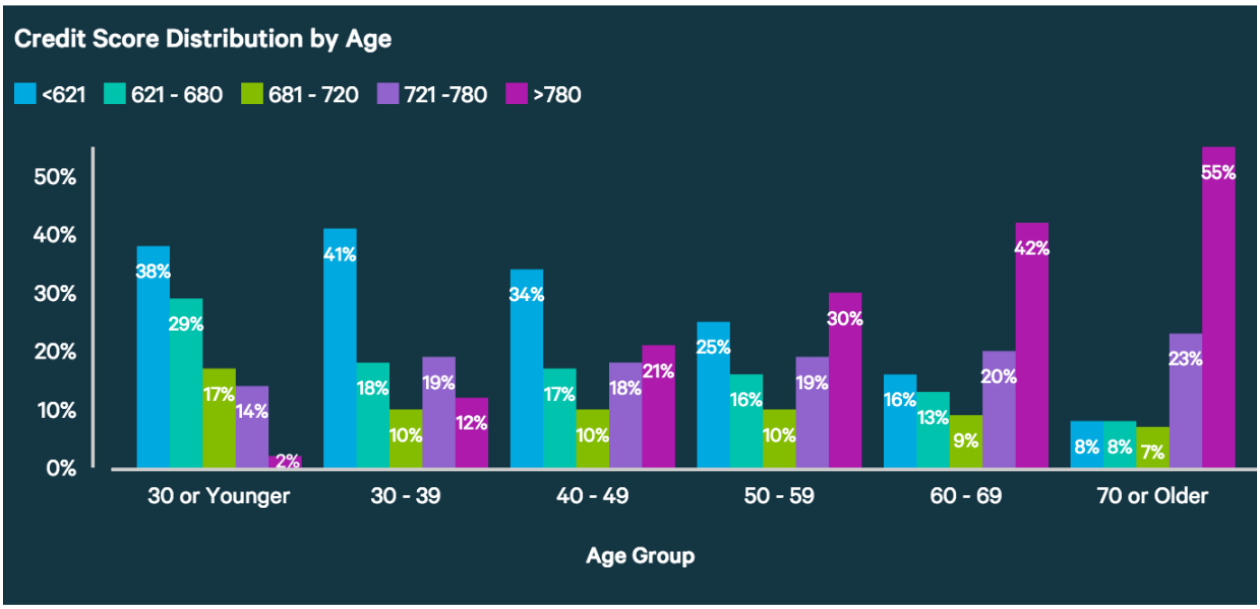


Figure 3. Credit Score Distribution by Age

2.3.3. *Weza Ventures*

Weza Ventures is a Kenyan based business venture who through their platform, The Weza Platform are able to carry out credit scoring for other businesses. Weza Ventures is able to do this by use of mobile and bio data from customers (Mpesa transactions and call logs) to generate a score for given customers. This has also enabled the organisation to have its own lending up known as Loanbee available in the Google playstore and also enables them to offer customizable micro lending applications to other businesses, Weza Ventures (2019).When credit scoring at Weza machine learning is also integrated to the sy stem. Figure 3 below shows architecture used by Weza

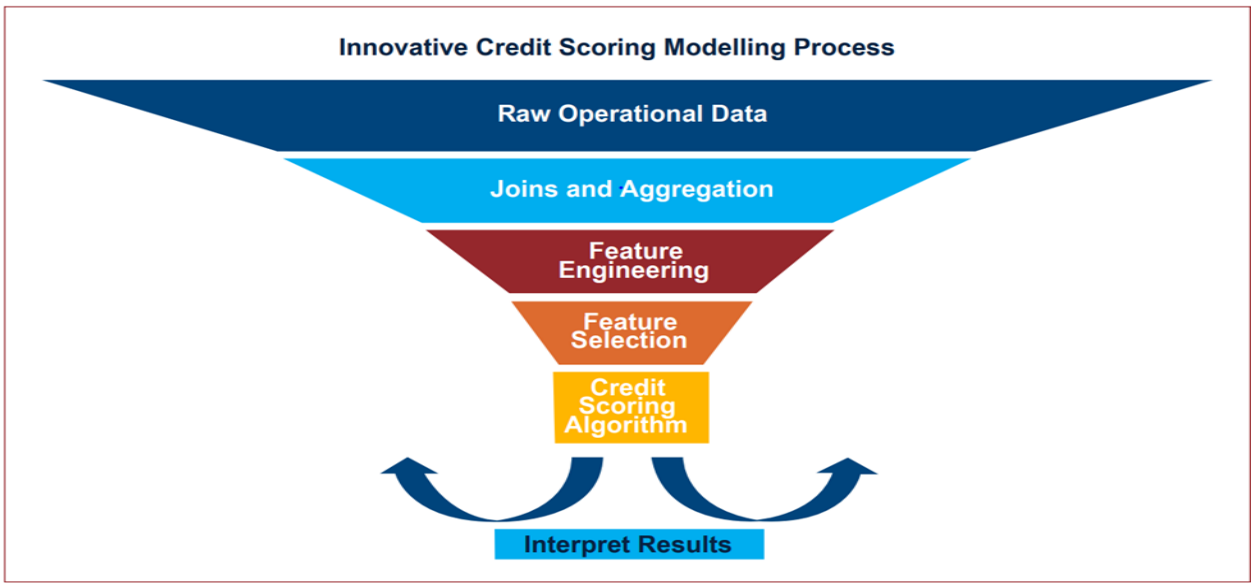


Figure 4. Innovative Credit Scoring Modelling Process by Akademia Baru, Penerbit. (2019).

Weza Ventures adopts such a model for its credit scoring. First they access raw data from client which are the Mpesa transactions. The data is then combined, joined and aggregated into input data. Engineer features, either manually using expert input or using automated approaches where the molested data goes down to the level as its input. Selecting the useful features from the input. The CSE now comes into play by using it's algorithm to generate the credit score after which results go for interpretation. Since Weza uses a machine learning CSE there is a feedback loop where the algorithm takes in the generated result and this enables it to learn from experience.

2.3.4. Big Data Scoring

Big Data Scoring is a cloud-based credit decision engine that works very much like Weza Ventures. They use credit scoring models that link a lender's internal data to a client's data which acts as the external data and generate a prediction on possible payment behaviour. This process is illustrated in figure 4 below

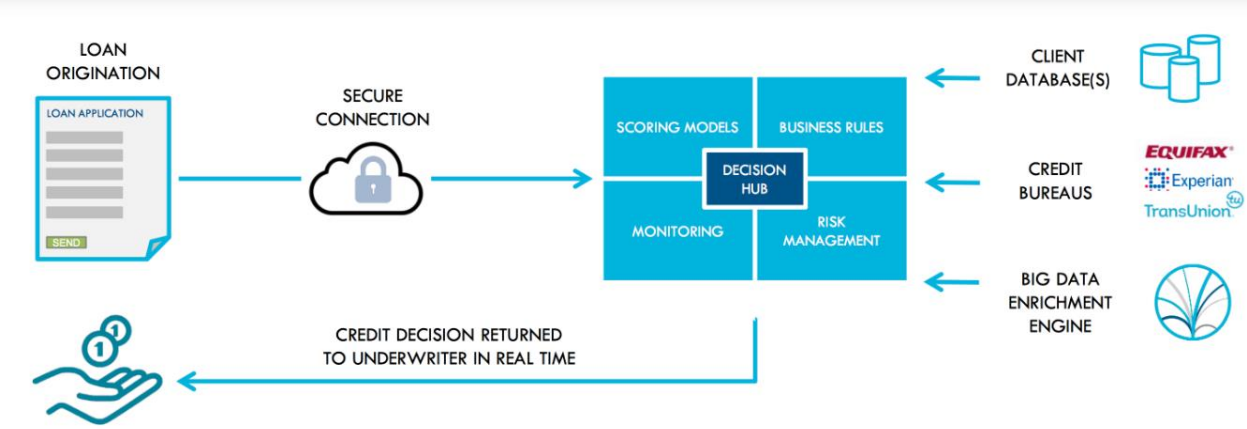


Figure 5. Scoring Model by Big Data Scoring (2019)

A decision hub holds the scoring model and internal business data. When loan request is made the client's history is retrieved from different external sources and this data is processed within the decision hub to make a decision, this credit decision that is made is the credit score

2.3.5. Plug&Score

Plug&Score is a credit scoring tool available on the online that enables one to access scorecard development software, access to a credit scoring system, it gives access to Loan Application Software, MFI credit scoring software and scorecard validation software. Plug&Score is meant to integrate with your already existing system by use of its APIs to quickly get you started and one can select aspect to suit need of their organisation. The business does not get access to the actual credit scoring logic but to APIs connecting to the logic. This is diagrammatically shown in figure 6 below

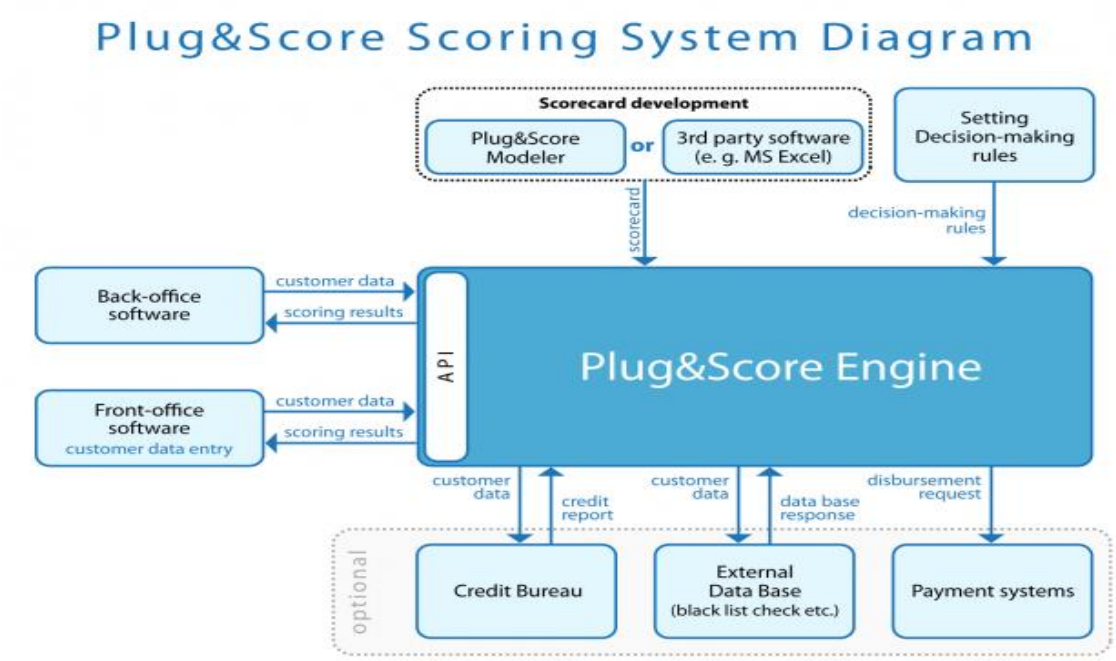


Figure 6. Plug&Score Scoring System Diagram by Plug&Score (2019)

After integration with your current system business rules have to be set, one is free to select ehat other components to integrate to their system to enhance their score and the Plug&Score engine is ready take customer data and will generate scores.

2.4. Limitations of Credit Scoring

The applications mentioned above is only limited to the banked section of the population whereas the vast majority of citizens are not banked. The ability to score is also restricted to business and for commercial purposes whereas we want to have the system available to people on demand. It, therefore, does not cover the subset of the population that does not use banks. It also does not cover small transactions between individuals.

2.5. Possible Development Technologies

By creating a smaller-scale version of the system used by banks we can be able to bring credit scoring to individuals who lend to each other day to day. This can also be embraced by small businesses like kiosks where retailer lend to consumers in goodwill but these olive branches being only extended to the people the traders are familiar with. This mini scale version will bring another aspect of financial management and control to the individuals

2.6. Conceptual framework

The figure below illustrates how the framework is supposed to work

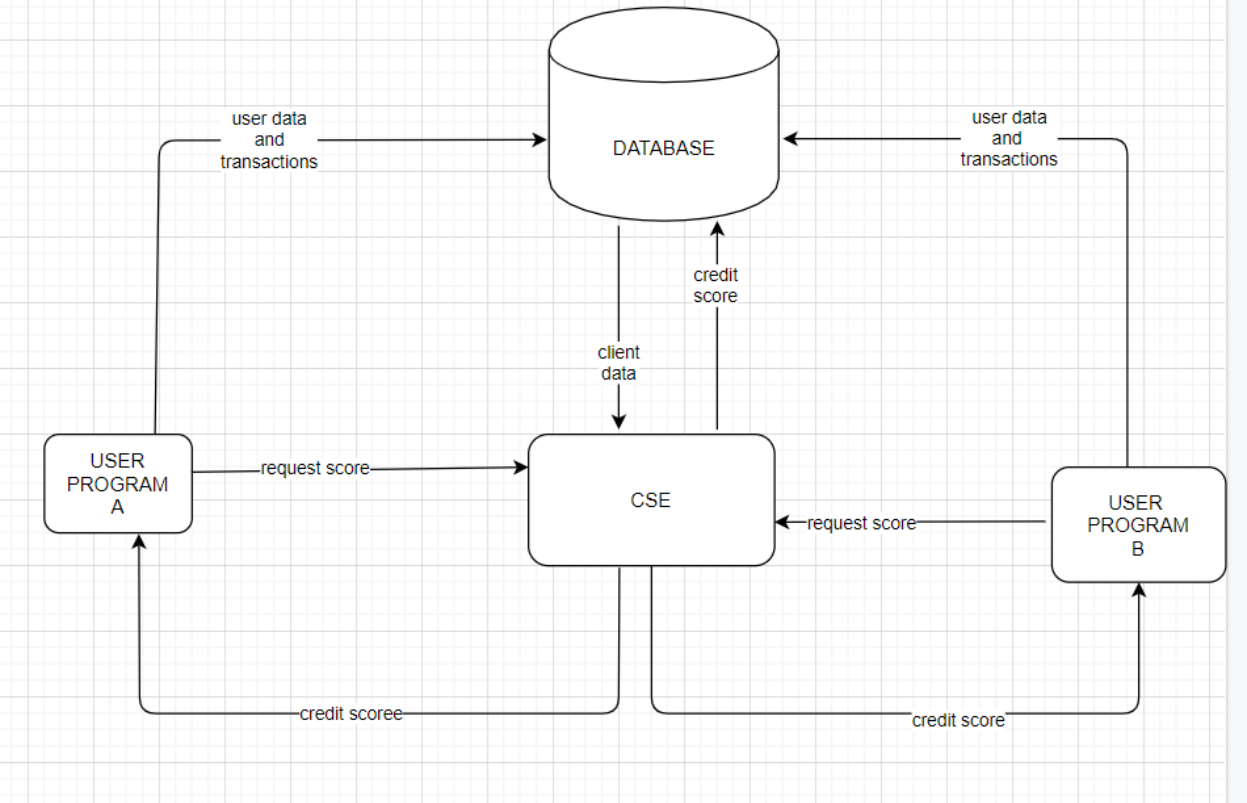


Figure 7. Conceptual Framework Diagram

The CSE takes client data from the database and generates a credit score which is then sent back to the database for storage and record keeping. This will allow a user to access their past credit scores. When a user requests for their credit score the CSE request for client data a process the most recent credit score which will be returned to the client program. When a client has the application for the first time they will have to create an account and this will implement a security feature about the system. Users will have to create an account and confirm via verification email this data is then sent to the database and is related to a specific user. When the a user of program

A wants to transact with the user of program B they can request for user B's credit score, to view their credit score approval will have to be made by user B only then will user A be able to view their credit score. They can then begin their transaction where the lender will have to initiate a transaction where the amount will be entered and be confirmed by both parties. When a debt is to be paid the loanee makes payment he makes approval request which is the accepted by the lender. This will automatically upload as a user's data and a new credit score is generated

CHAPTER 3. Research Methodology

3.1. Introduction

System Development Methodology refers to the approach used to solve a problem by either creating a new system to tackle the problem or by enhancing the already existing system to be capable of solving these problems. That being said, this chapter goes over the methodology that was selected to be used in creating the system based on various factors unique to the proposed system.

3.2. Approach to be used

OOAD was the chosen approach for the proposed system as it allowed breaking down of the development process into stages to accommodate the efficiency of the work put in place in the development stages and to allow correction or addition to the requirements even with uncertain requirements (Dba-oracle, 2017).

3.3. Methodology to be used

System Development Methodology was the blueprint used to create a structure, to plan and give me control of the process of developing an information system. Prototyping was the methodology chosen to be used in developing the system

3.3.1. Prototyping

This is an SDM that allows for the creation of a "mock form" of the application with the basic functionalities the system will have. This "mock form" of the application was the base on which more and more functionalities of the system were added to till all requirements of the system were reached. The aim the methodology helped achieve was creating a system that was be able to meet all of its functional and non-functional requirements. Prototyping was selected because we were able to introduce more of the functionalities of the system by using base versions of the system allowing me to be able to track capabilities of the system and requirements met over time. Below are the steps used in this methodology; (Avison & Fitzgerald, 2006)

3.3.1.1: *Requirement analysis*

This is the initial stage whereby the interface of the system is identified together with the customer's need for the functioning of the system.

3.3.1.2: *Quick design*

When the requirements are known, the preliminary design of the system is done while only containing the important aspects of the system to give the customers the feel of the system.

3.3.1.3: *Build a prototype*

A system is developed from the features in the quick design and given to the customer to have a feel.

3.3.1.4: *User evaluation*

The prototype is then evaluated by the users and the stakeholders to be able to see the system's functionality and point out the strength of the system to be maximized and the weakness to be eliminated or be corrected.

3.3.1.5: *Refining of the prototype*

If the user is not satisfied with the prototype, a new prototype is then developed with the changes made and then presented to the user. This is repeated until the user is contented with the system thus the coding and requirement analysis in prototyping as a cycle and is almost endless.

3.3.1.6: *Engineering product*

Once the system is accepted by the user the final system is examined thoroughly and then followed by routine maintenance to prevent large-scale failures and minimize downtime

This step by step process is visually represented in the figure below

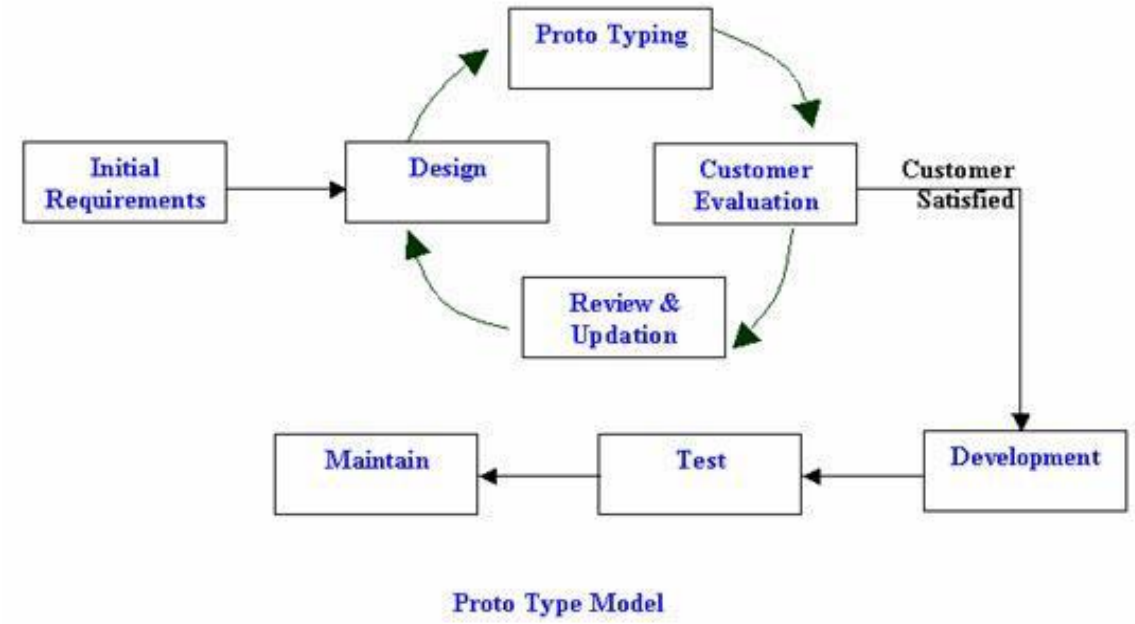


Figure 8 Prototype Model by Satoa (2012)

3.4. Justification of Methodology

Prototyping is an iterative methodology and hence allowed the system to undergo several iterations before generating the final system. Several iterations were needed in this proposed system since data sets to be captured needed fine-tuning before a full system was made or even fine-tuning values of the scorecard to generate a more reliable credit score, Crinnion, J. (1992).

3.5. Tools

3.5.1. Android Studio

Since the application is mobile-based, android studio was the tool used in creating the application

3.5.2. Firebase

This is the database used in the storage of any of the real time data like posts

3.5.3. Sqlite

This database was used for storage of user selections and answers needed to generate credit score

3.6. End to End Testing

End to end testing is the testing of the entire system from database interaction, network interaction and even system hardware interaction. This method of testing was used due to the methodology chosen in implemented in creating the system. Due to the prototypes that were generated changes made the system affected some aspects other than what was being changed and this method of testing came in handy as I was able to test the entire system to ensure changes on one side did not affect something else

CHAPTER 4. System Analysis and Design Description

4.1. Introduction

System analysis and 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.

4.2. Requirements Gathering

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 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.

4.3. System Requirements

4.3.1. Functional Requirements

These are the functions that the system has achieved and can be said to be fully functional

4.3.1.1: *Authentication*

This acts as the first point of defence in the application due to the sensitivity of the data that is contained in the application. This also acts as an assurance to users that they are protected

4.3.1.2: *Risk level Ratings*

Every user will be able to carry out credit scores in their account giving them their risk level in term of credit scores. A high-risk level means that you have a bad credit score hence in term of lending there's a high chance of non-payment and the vice versa

4.3.1.3: *Scorecard Building*

The system is able to generate a scorecard for the user which can be used to generate their credit score

4.3.1.4: *Credit Score Generation*

The system generates a credit score for a user based on values they have selected

4.3.2. Non-functional Requirements

These are the perspectives from which the system can be judged upon

4.3.2.1: *Reliability*

This refers to how credible the data generated by the system is. Credit Score would be the data generated and since the system can do this is highly reliable

4.3.2.2: *Accuracy*

The system gives scores that are accurate based on user input

4.3.2.3: *Usability*

This refers to the ease of use of the system. The system has been simplified for any everyday user to use it

4.3.2.4: *Performance*

The system is able to perform task quickly and effectively

4.4. System Architecture

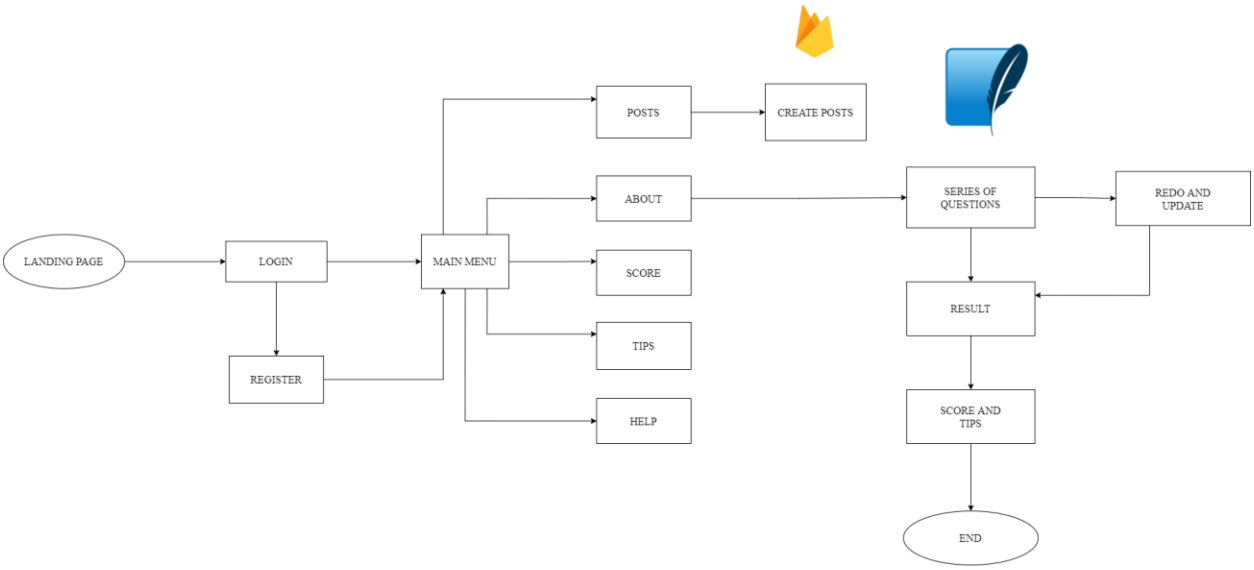


Figure 9. System Architecture

The system architecture of anything refers to the structural layout of how an application is designed. If you refer to Figure above, it is the layout of the Credit Score Application. The application is built on Android Studio, Java, Firebase and SQLite. The app starts off with the landing page where the login and register buttons are. After logging in and registering, user details are saved on both firebase and sqlite, the app takes you to the main menu which has five different sections – posts,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.

4.5. Analysis

4.5.1. Use Case

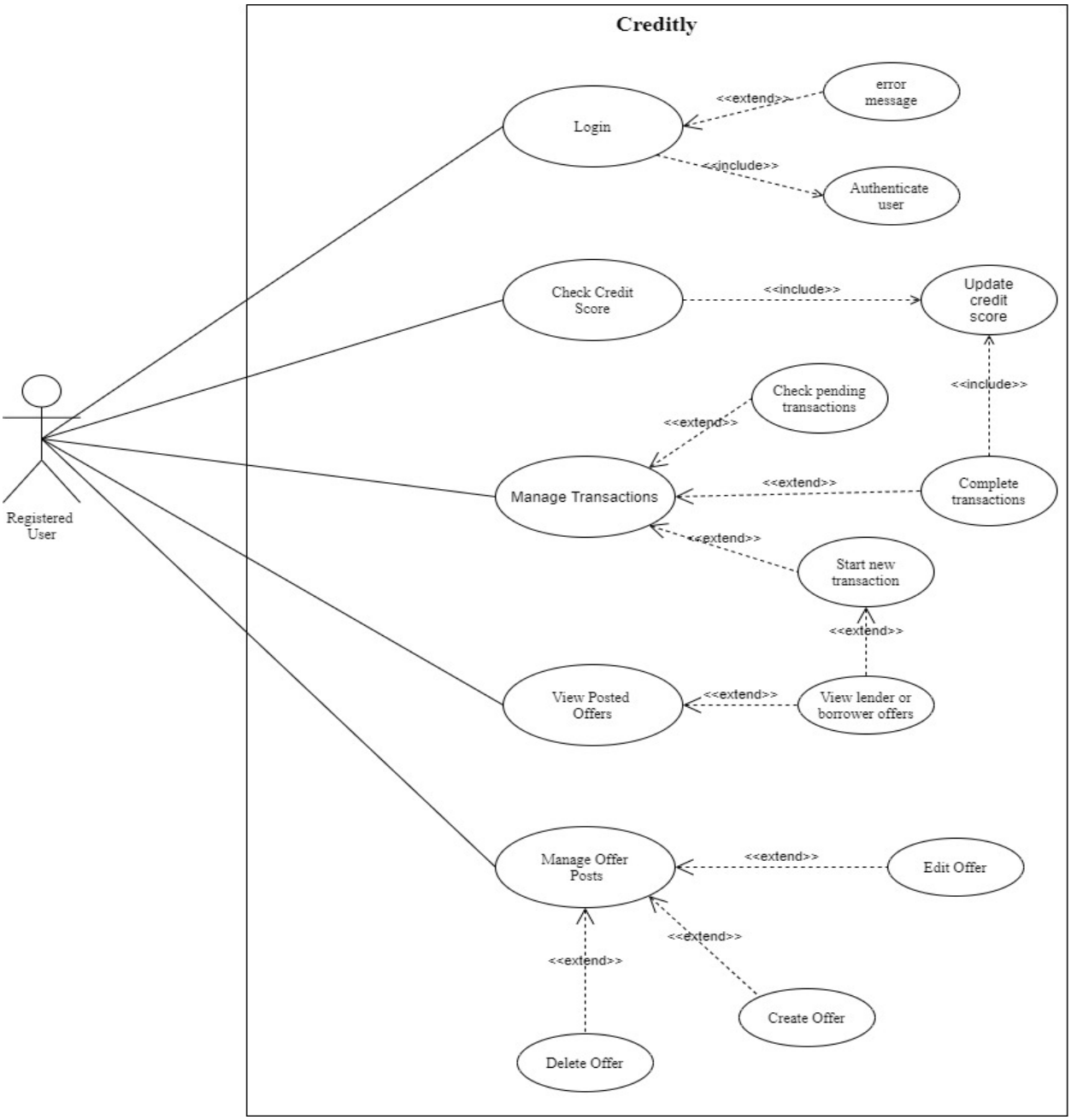


Figure 10. System Use Case

Primary Actor

A registered user will be the primary actor in the system. The actions of this actor will initiate all interactions with the system. The primary actor in this system can be either a lender or a borrower in the system through use of offers where one can post an offer to lend to someone or borrow to

someone. This will enable loan tendering within the application and this will also have an impact on the actor's credit score.

Use Cases

Login

In the use case diagram above, the primary actor will begin by logging into the system which will automatically validate the actor's credentials hence the include relationship. If the actor happens to have entered a wrong information in the login process the actor will be served with an error message to enter the correct password

Check Credit Score

When the primary actor checks their credit score, the Credit Score Engine automatically updates the actor's credit score and the same is displayed to the actor

Manage Transactions

Here the primary actor has the option to check their pending transactions, complete a transaction or start a new transaction. All these actions are dependent on which the actor selects hence they extend the use case. If the primary actor decides to complete a transaction this will automatically cause an update to the credit score of the user

View Posted Offers

Here the primary actor can view offers available from other users. He will have the option to select whether to view offers from lenders or to view offers from borrowers depending on his immediate needs. If the primary actor decides to take up an offer from those he is viewing a new transaction will be started.

Manage Offers

This is where the primary actor runs his offers whether to make a post to borrow or to lend to other users, to delete posted offers he made or edit offers he has posted

4.6. Designs

4.6.1. Class Diagram

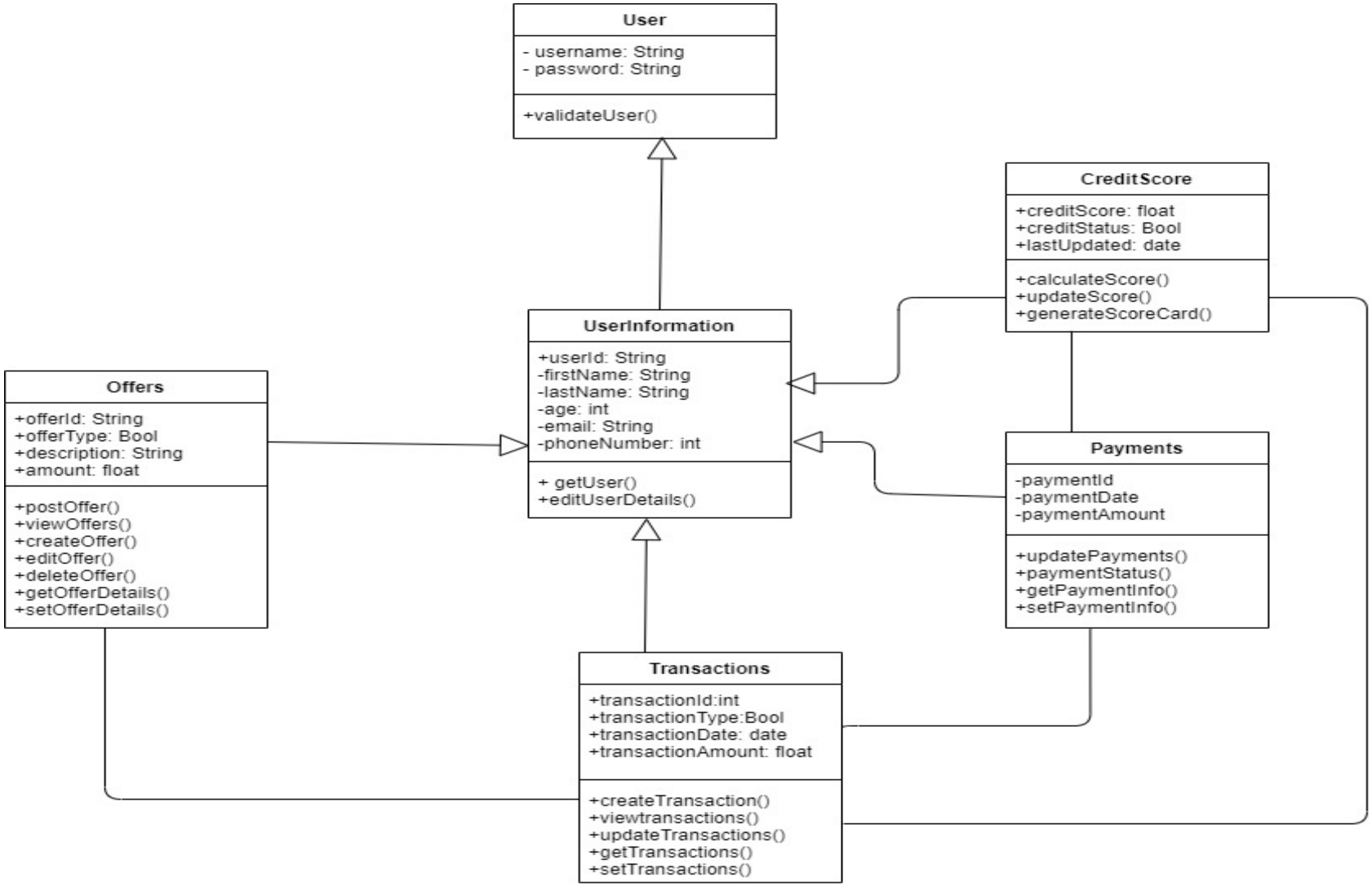


Figure 11 Class Diagram

4.6.2. Sequence diagram

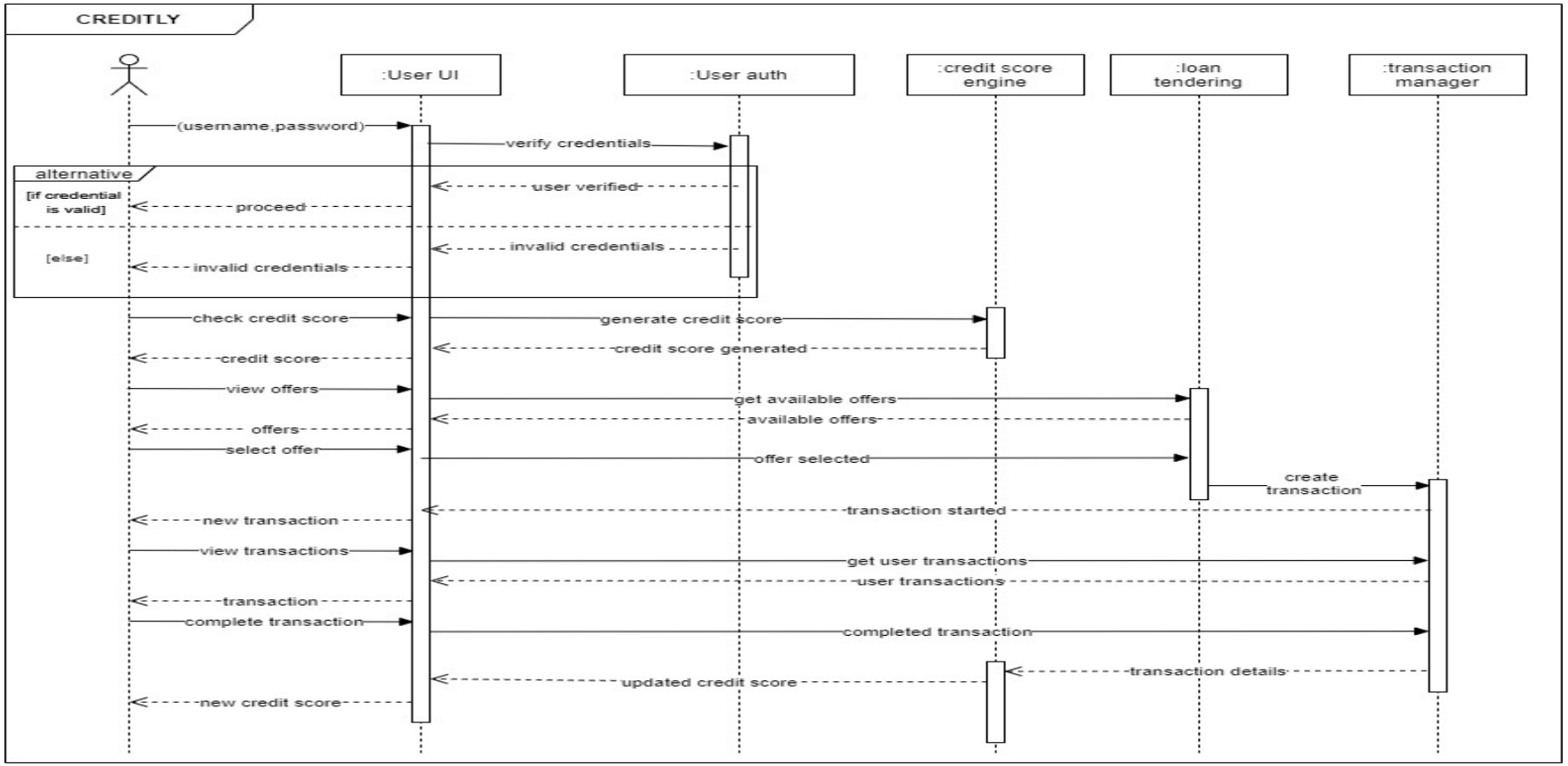


Figure 12. System Sequence Diagram

4.6.3. Entity Relationship Diagram

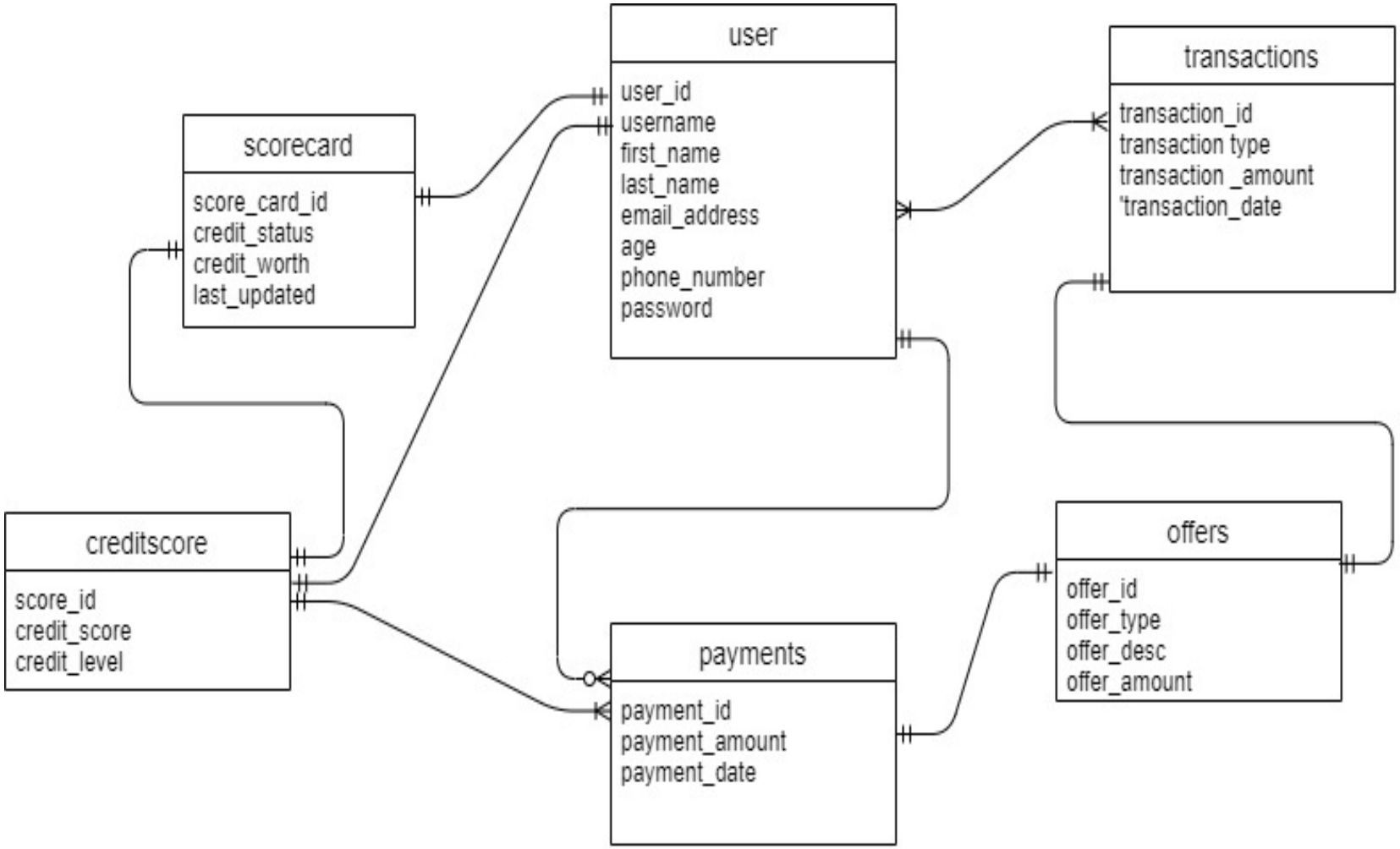


Figure 13 Entity Relationship Diagram

4.6.4. Database Schema

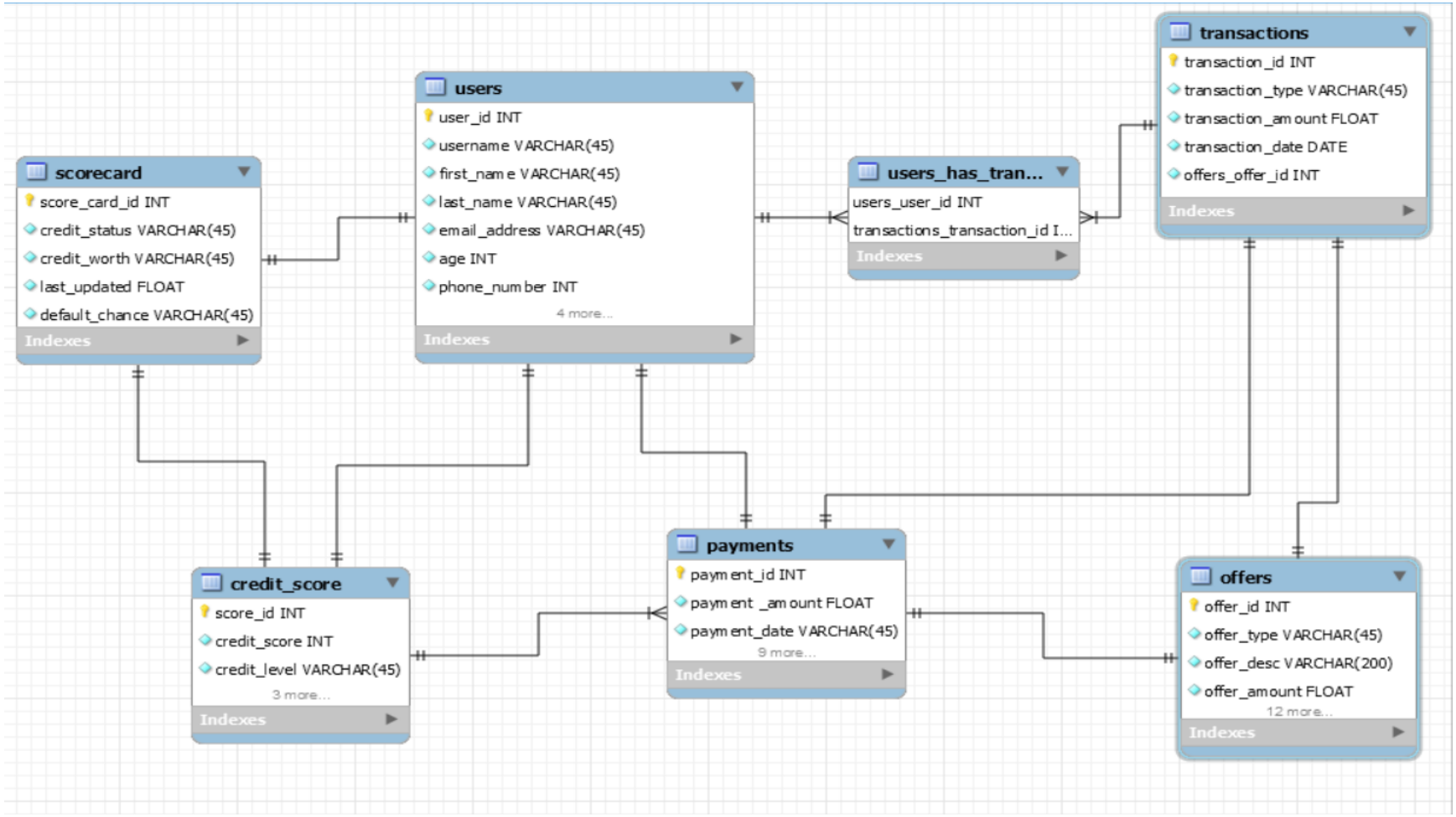


Figure 14. Database Schema

CHAPTER 5. System Implementation and Testing

5.1. 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.

5.2. 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 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.

5.3. 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.

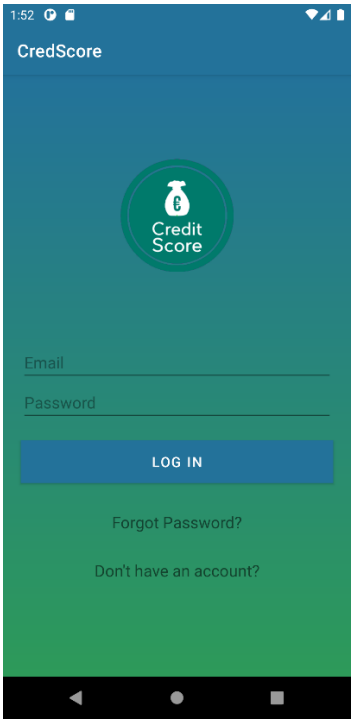
In the input class, there was one method which was *saveAnswers*. What this class did, was make sure that the user has either submitted new answers or updated their existing answers. The way this was tested was it said if the users hadn't submitted any answers and there was no errors along with it then 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.

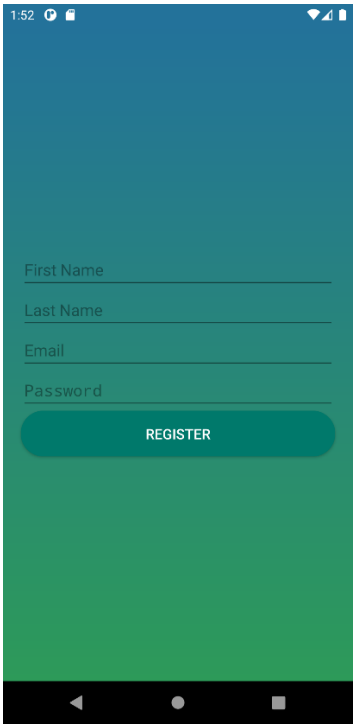
The last and final testing that I had done was in the register class and the method called was *canAddNewUser*. This also returned a boolean, true or false. What this method does is that it makes sure that when you register a new user, it adds it to the database and they are able to log back into the 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.

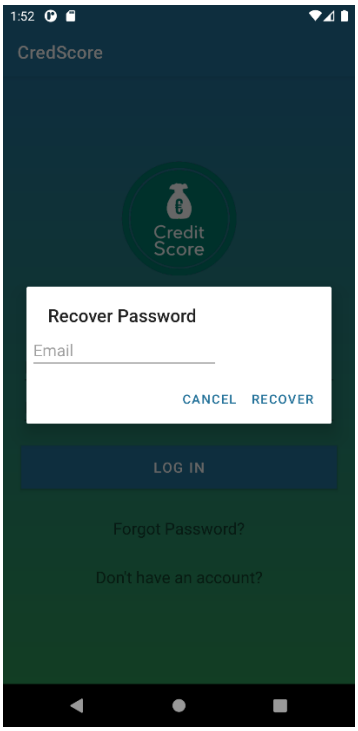
Another testing method I had used was to test the static methods by printing anything to the console. For example if I wanted to make sure that the XML was displaying properly or if there was any data not sending, I would print "Working" out to the console and if it showed, it meant that it was working and if it didn't I knew there was an error.

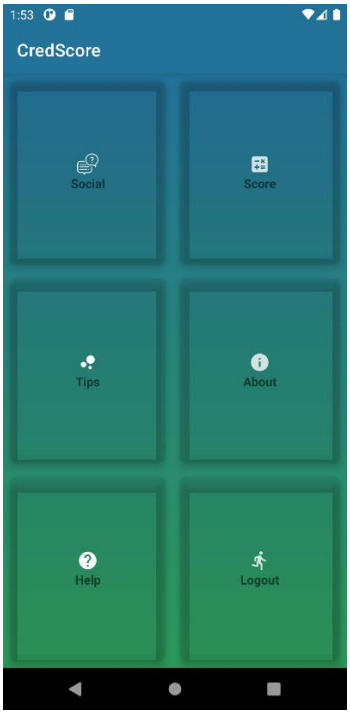
Along with unit testing, I also had many people manually use the application and this way I could find the errors and see what they were struggling to use. For example, if the user couldn't register even though there was error checking done or if the answers weren't updating after the user re-runs the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

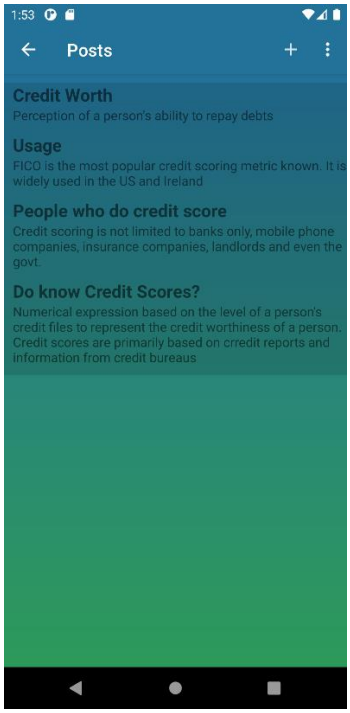
5.4. The Application

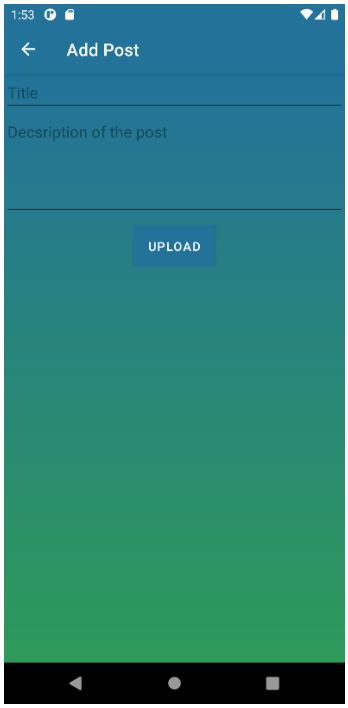


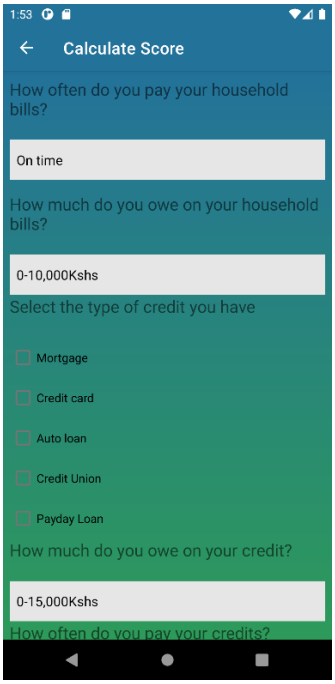


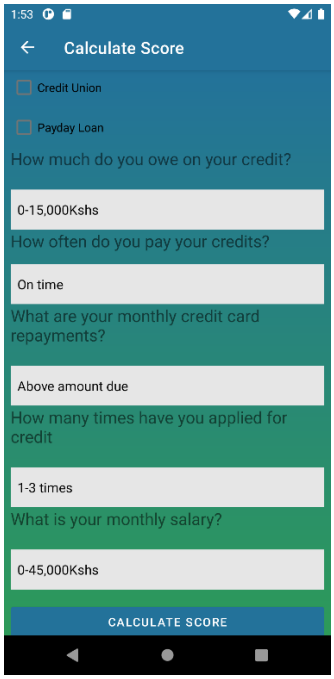


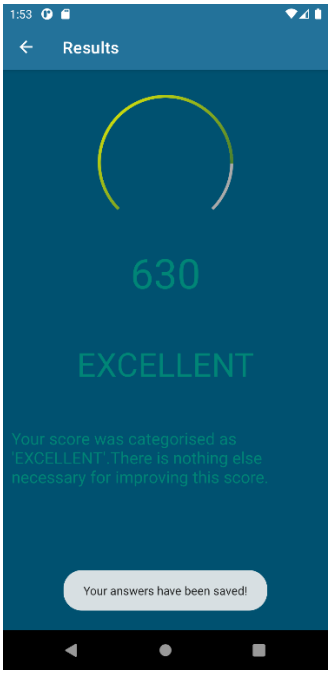


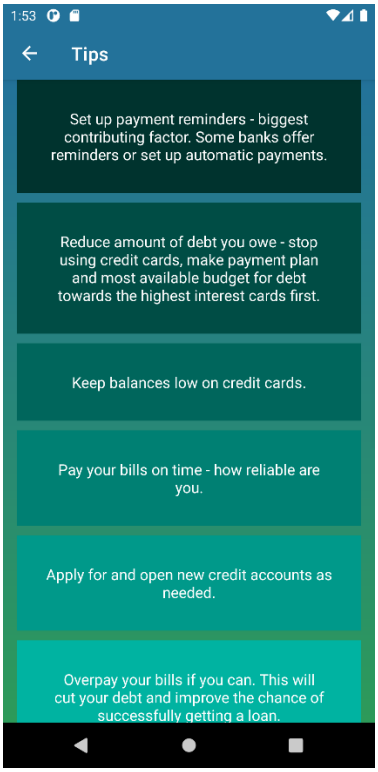


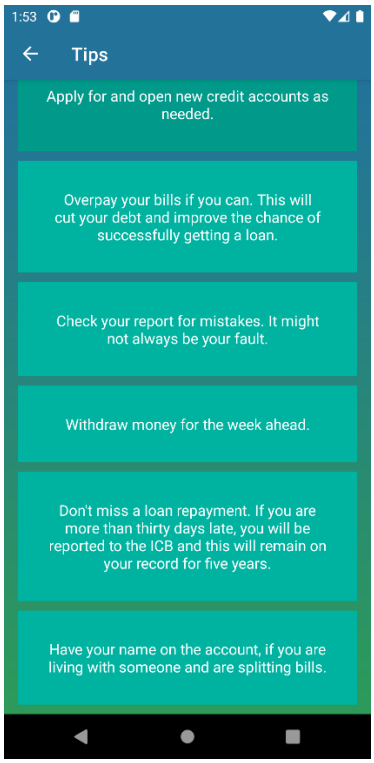


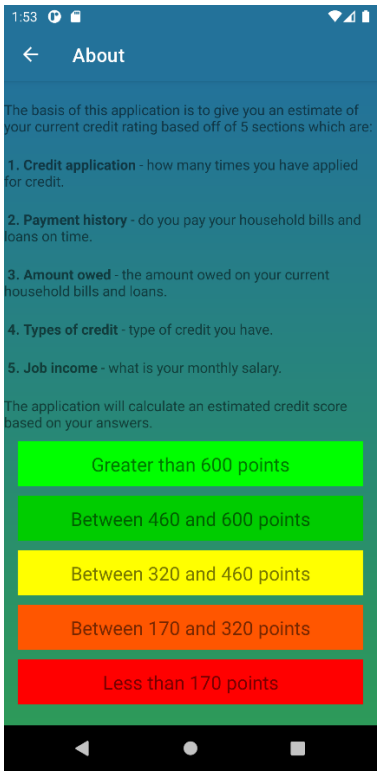

1:53
Calculate Score
How often do you pay your household bills?
On time
How much do you owe on your household bills?
0-10,000Kshs
Select the type of credit you have
Mortgage
Credit card
Auto loan
Credit Union
Payday Loan
How much do you owe on your credit?
0-15,000Kshs
How often do you pay your credits?

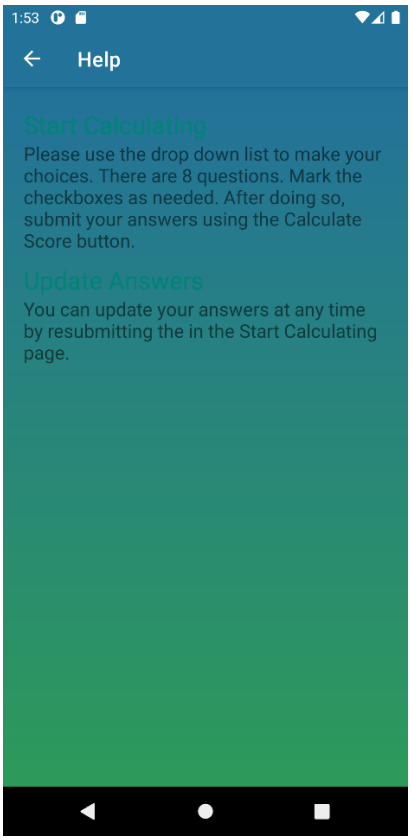

1:53
Calculate Score
Credit Union
Payday Loan
How much do you owe on your credit?
0-15,000Kshs
How often do you pay your credits?
On time
What are your monthly credit card repayments?
Above amount due
How many times have you applied for credit
1-3 times
What is your monthly salary?
0-45,000Kshs
CALCULATE SCORE


1:53
Results
630
EXCELLENT
Your score was categorised as 'EXCELLENT'.There is nothing else necessary for improving this score.
Your answers have been saved!


1:53
Tips
Set up payment reminders - biggest contributing factor. Some banks offer reminders or set up automatic payments.
Reduce amount of debt you owe - stop using credit cards, make payment plan and most available budget for debt towards the highest interest cards first.
Keep balances low on credit cards.
Pay your bills on time - how reliable are you.
Apply for and open new credit accounts as needed.
Overpay your bills if you can. This will cut your debt and improve the chance of successfully getting a loan.


1:53
Tips
Apply for and open new credit accounts as needed.
Overpay your bills if you can. This will cut your debt and improve the chance of successfully getting a loan.
Check your report for mistakes. It might not always be your fault.
Withdraw money for the week ahead.
Don't miss a loan repayment. If you are more than thirty days late, you will be reported to the ICB and this will remain on your record for five years.
Have your name on the account, if you are living with someone and are splitting bills.


1:53
About
The basis of this application is to give you an estimate of your current credit rating based off of 5 sections which are:
1. Credit application - how many times you have applied for credit.
2. Payment history - do you pay your household bills and loans on time.
3. Amount owed - the amount owed on your current household bills and loans.
4. Types of credit - type of credit you have.
5. Job income - what is your monthly salary.
The application will calculate an estimated credit score based on your answers.
Greater than 600 points
Between 460 and 600 points
Between 320 and 460 points
Between 170 and 320 points
Less than 170 points


1:53
Help
Start Calculating
Please use the drop down list to make your choices. There are 8 questions. Mark the checkboxes as needed. After doing so, submit your answers using the Calculate Score button.
Update Answers
You can update your answers at any time by resubmitting the in the Start Calculating page.

CHAPTER 6. Future works, Recommendations and Conclusion

6.1. Future works, recommendations and conclusion

Future works to this application or if someone is to build a similar application there are a few things that I would recommend to have and not to have. First thing I would recommend is having this system use machine learning where one would give the application read permissions in messages. With this permission allowed the application would access messages and use Mpesa messages maybe categorize them according to type of transactions by using terms used in Mpesa messages i.e. received, sent to, paid to… and extract the amounts used in these transaction. Using some machine learning algorithms one would molest the data achieved and this would be used to create a credit score.

Secondly 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. One should be aware how different levels of credit scores are categorized in the industry. 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 you are updated on industry level credit scores because it is crucial that you are up to date on everything that is involved with credit ratings to be able to give users an experience close to what is done in real life like by the banking sector. 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.

Developing this application taught me a lot about not only the technical parts but also about finances in general. 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.

I believe there should be more awareness in relation to credit ratings. 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

Abedi S. (2002), “Highway to Success”, Credit Management Journal (Electronic Edition). Source, https://leatherspinters.com/do you talk to your banker. Html

Akademia Baru, Penerbit. (2019). Credit Scoring Models: Techniques and Issues.

Avison, D., & Fitzgerald, S. (2006). Historical Perspective. In V. 2. IFIP Internation Federation for Information Processing. *The Past and Future of Information Systems*.

Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, J. B. And Vanthienen, (2003). Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring. The Journal of the Operational Research Society (Jun., 2003), pp. 627- 635

Big Data Scoring(2019). The Leaders in Big Data Credit Solutions retrieved from https://www.bigdatascoring.com/

Caire et al, (2006) A Handbook for Developing Credit Scoring Systems in Microfinance context USAID, February 2006

CBK, KNBS and FSD Kenya (2019, April 03). 2019 *FinAccess Household Survey* [PDF File]. Retrieved from https://www.centralbank.go.ke/

Citizen (2014, February 28). Man ‘Marries Off’ wife to settle sh.900 debt. *Citizen Digital,* Retrieved from https://citizentv.co.ke/

Crinnion, J. (1992). Evolutionary Systems Development: A Practical Guide to the Use of Prototyping within a Structured Systems Methodology.

E. Okoth (2019, April 07). Kenyans turn to friends, relatives as lenders shy off. *The Business Daily*, Retrieved from https://www.businessdailyafrica.com/

Kinda, O., & Achonu, A. (2012). Building A Credit Scoring Model For The Savings And Credit Mutual Of The Potou Zone (MECZOP)/Senegal. *Consilience,* (7), 17-31. Retrieved June 3, 2020, from www.jstor.org/stable/26167833

Plug&Score (2019). Practical Credit Scoring Tools retrieved from https://plug-n-score.com/

Maheshwari, S. (2002). A Comparative Analysis of Different types of Models in Software Development Life Cycle. *International Journal of Advanced Research in Computer Science and Software Engineering*.

Satoa, S. R. (2012). An Attendance Monitoring System Using Biometric Authentication. *International Journal of Advanced Research in Computer Science and Software*.

Schreiner, M. (2004) Benefits and Pitfalls of Statistical Credit Scoring for Microfinance, *Center for Social Development*, Washington University. St. Louis, USA

Towards Data Science(2019) Intro to Credit ScoreCard retrieved from https://towardsdatascience.com/

Weza Ventures (2019). Weza.io | New generation lending platforms retrieved from https://www.weza.io/

Yang Liu, (2001) New Issues in credit Scoring Application, *WorkReport No. 16*, institute fur Wirtschaftsinformatik, 2001

APPENDIX A: Time Schedule

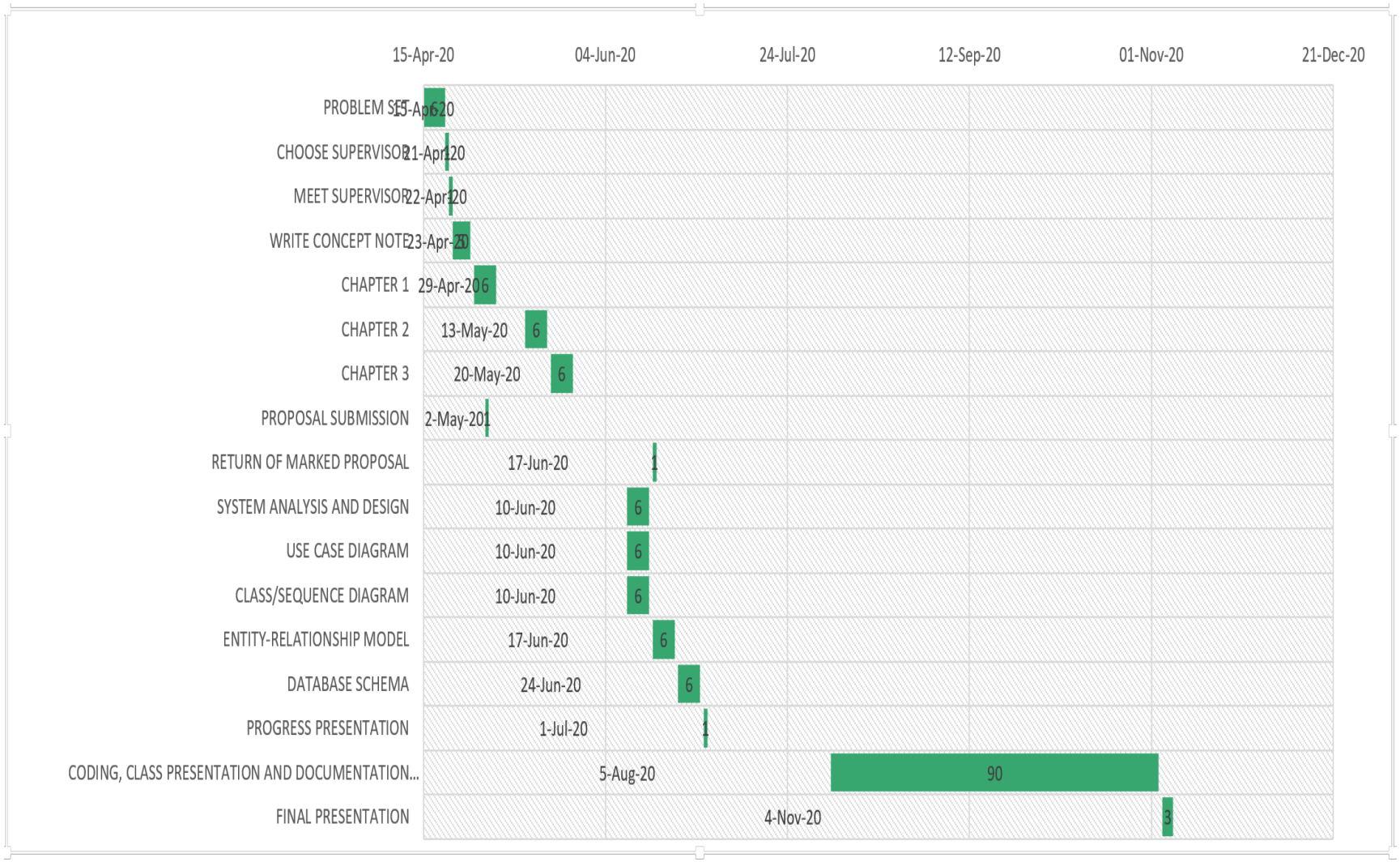


Figure 15. Gantt Chart