Fintech Peer-To-Peer Lending Mobile Application Development

By

OON WEI KIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)
BUSINESS INFORMATION SYSTEMS
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2022

REPORT STATUS DECLARATION FORM

Title: Fintech Peer-to-Peer Lending Mobile Application Development

Academic Session: May 2022

I ____________________OON WEI KIN___________________________
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

SuLoo

____
(Author's signature)

____
(Supervisor's signature)

Address:
A2-2 block A ,rumah pangsa taman
bandar baru selatan kampar perak dr
Kampar 31900 Perak

______Su Lee Seng__________
Supervisor's name

Date: _____07/09/2022___________

Date:__07/09/2022____________

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF Information and Communication Technology (FICT)

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___07/09/2022_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Oon Wei Kin***__________________________ (ID No: __***18ACB05279***_____) has completed this final year project/ dissertation/ thesis* entitled "________*Fintech Peer-to-Peer Mobile Application Development*__" under the supervision of ____Mr Su Lee Seng_________________ (Supervisor) from the Department of ____________ Digital Economy Technology______, Faculty/Institute* of ___ Information and Communication Technology______ , and _____________________ (Co-Supervisor)* from the Department of _________________________, Faculty/Institute* of ___________________________.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Fintech Peer-To-Peer Lending Mobile Application Development**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________

Name : ______Oon Wei Kin_______

Date : _______07/09/2022________

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr. Su Lee Seng who has given me this bright opportunity to engage in a Fintech peer-to-peer lending mobile application development project. A special thanks to him who provides simulation suggestions and encouragement, and helps me to coordinate my project, especially in report parts.

I would also like to acknowledge with much appreciation to my parents and friends for their constant motivation and encouragement.

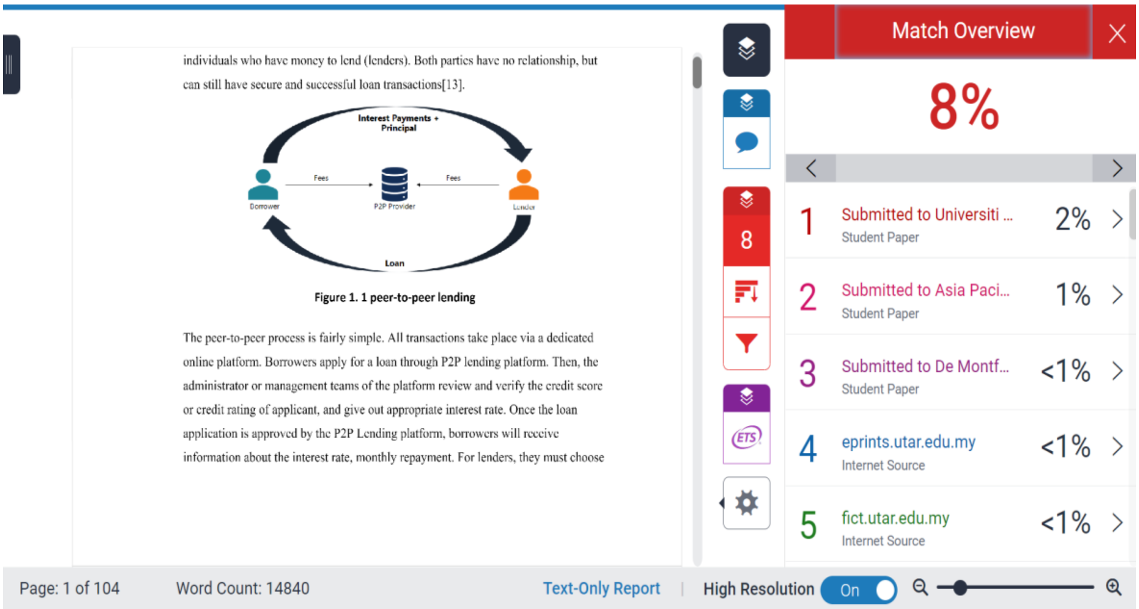
ABSTRACT

This project is a mobile application about peer-to-peer lending as final year project in UTAR for academic purpose. The project will provide students with knowledge and experience in developing mobile applications and designing user interfaces and databases. Mobile application is becoming increasingly common because of its portability and mobility to access everywhere and at any time. This is one of the main reasons that I decide to develop a peer-to-peer lending mobile application. The peer-to-peer loan applications offer both the borrower and the lender the opportunity to select the loan or investment they wish to opt for. Compared to traditional lending methods, peer-to-peer lending applications make the lending and borrowing process more practical. The purpose of this project is to make loan generation faster and provide low loan interests to users. Under the traditional lending method, borrowers had to come to the bank to request a loan. By using this application, the process of application lying for a loan will become faster and convenient. Borrowers can request a loan on the lending platform without going anywhere, just need to download this application. For investors, they can have a look on every request that posted by borrowers and choose the most suitable one to invest. As a result, lending works has taken great care to make our application process as easy and timely as possible. The methodology use for this project is Prototyping. The prototyping model is a systems development process in which a prototype is produced, tested, and changed as needed until an acceptable output is obtained from which the whole system or product may be developed. This paper presents some of the relevant literature reviews, research analysis, research findings and findings of this study.

TABLE OF CONTENTS

Table of Contents

PLAGIARISM CHECK RESULT



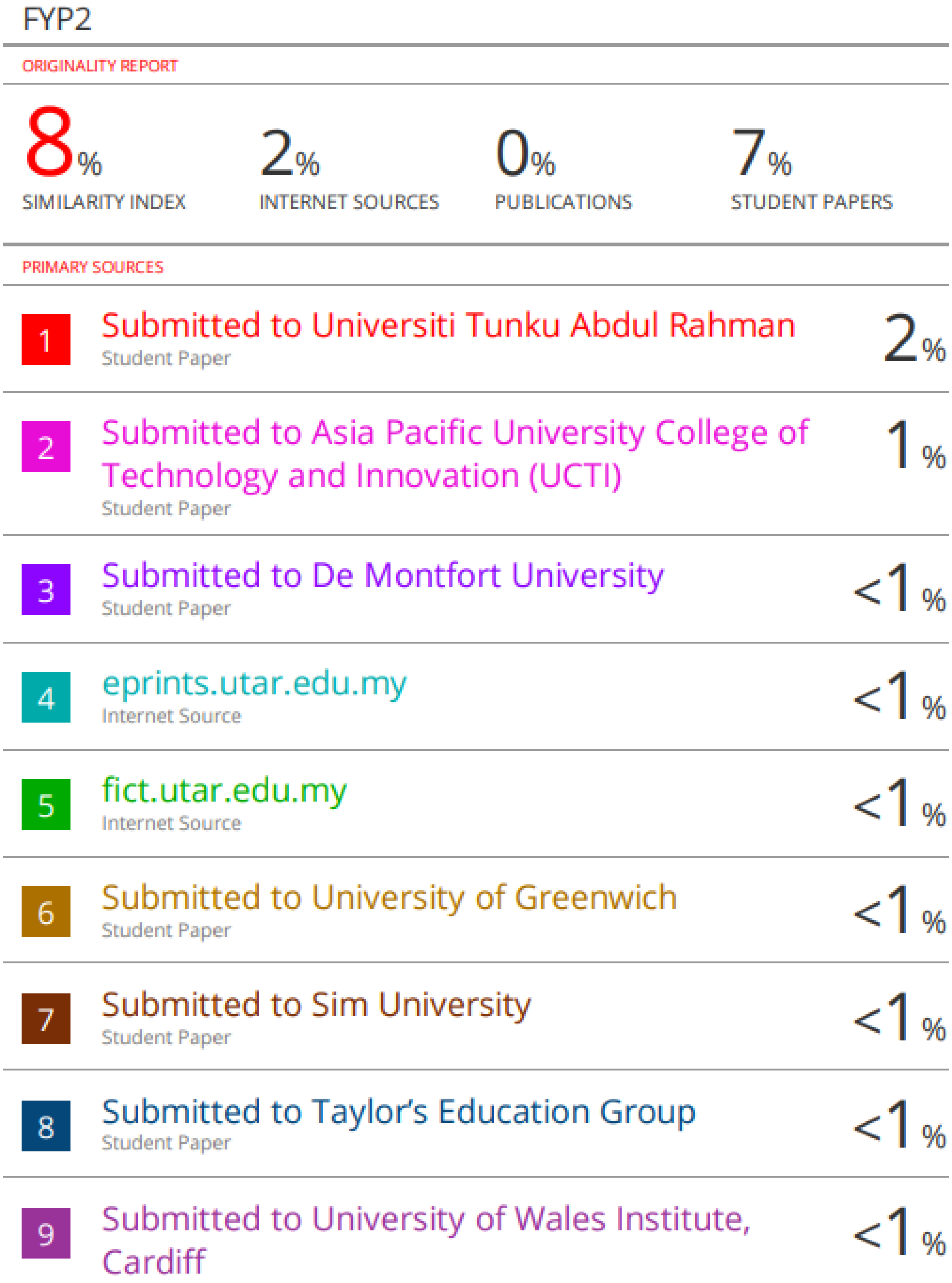
individuals who have money to lend (lenders). Both parties have no relationship, but can still have secure and successful loan transactions[13].

Figure 1. 1 peer-to-peer lending

The peer-to-peer process is fairly simple. All transactions take place via a dedicated online platform. Borrowers apply for a loan through P2P lending platform. Then, the administrator or management teams of the platform review and verify the credit score or credit rating of applicant, and give out appropriate interest rate. Once the loan application is approved by the P2P Lending platform, borrowers will receive information about the interest rate, monthly repayment. For lenders, they must choose

1	Submitted to Universiti ... Student Paper	2%
2	Submitted to Asia Paci... Student Paper	1%
3	Submitted to De Montf... Student Paper	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	fict.utar.edu.my Internet Source	<1%

Page: 1 of 104 Word Count: 14840



FYP2

ORIGINALITY REPORT

8%	2%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
3	Submitted to De Montfort University Student Paper	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	fict.utar.edu.my Internet Source	<1%
6	Submitted to University of Greenwich Student Paper	<1%
7	Submitted to Sim University Student Paper	<1%
8	Submitted to Taylor's Education Group Student Paper	<1%
9	Submitted to University of Wales Institute, Cardiff	<1%

CHECK LISTS

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student ID	18ACB05279
Student Name	Oon Wei Kin
Supervisor Name	Su Lee Seng

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgment
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)
Date: 17/09/2022

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

%	Percentage

LIST OF ABBREVIATIONS

Fintech	Financial Technology
P2P	Peer-to-peer
SME	Small to Medium-Sized Enterprise
ATM	Automated Teller Machine
US	United State
UK	United Kingdom
OTP	One-time password
App	Application

1 CHAPTER 1 Introduction

The word "FinTech," which is a condensed version of the phrase "financial technology," refers to businesses or representatives of businesses that integrate financial services with cutting-edge technology. It refers to a new generation of businesses that are revolutionizing the way consumers pay, transfer money, borrow, lend, and invest. FinTech refers to the innovative use of technology to better our lives, such as making banking more accessible and inexpensive. Fintech is present in a variety of financial services, including deposits, loans, and capital raising, insurance, investment management, and payments, clearing, and settlement. P2P lending is a result of fintech.

A peer-to-peer lending is a form of direct loan to individuals or businesses without the participation of a formal financial such as a bank as an intermediary[9]. P2P lending is typically carried out through online platforms that pair lenders with prospective borrowers. Individuals who want to borrow money (borrowers) communicate with individuals who have money to lend (lenders). Both parties have no relationship, but can still have secure and successful loan transactions[13].

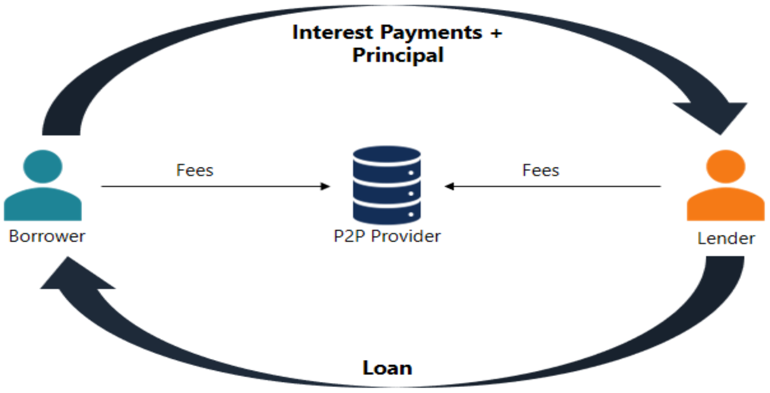
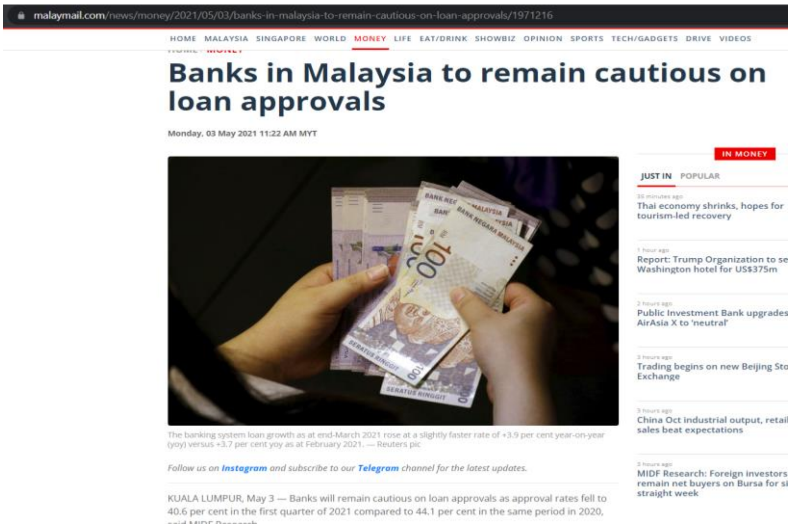


Figure 1. 1 peer-to-peer lending

The peer-to-peer process is fairly simple. All transactions take place via a dedicated online platform. Borrowers apply for a loan through P2P lending platform. Then, the administrator or management teams of the platform review and verify the credit score or credit rating of applicant, and give out appropriate interest rate. Once the loan application is approved by the P2P Lending platform, borrowers will receive information about the interest rate, monthly repayment. For lenders, they must choose the type of investment for the loan, such as the individual or business loan, and the amount, reimburse the investment. After they decided the investment preference, The P2P lending platform will use money from lenders to invest in accordance with their investment preferences. P2P will also push the lending application that correspond to their investment preference to lenders. The P2P lending platform acts as an intermediary in the loan deal between lenders and borrowers .

P2P lending is not offered by the traditional banks and a formal financial institution is not involved as an intermediary and it is easier to get a P2P loan compared to a traditional loan [10]. For borrowers, peer-to-peer lending provides a more accessible source of funding than traditional lending by traditional banks. For lenders, the P2P lending is an alternative means of investing[11]. The P2P lending also provides a choice for the borrowers to be able to obtain a loan from traditional banks or traditional loans.


malaymail.com/news/money/2021/05/03/banks-in-malaysia-to-remain-cautious-on-loan-approvals/1971216
HOME MALAYSIA SINGAPORE WORLD MONEY LIFE EAT/DRINK SHOWBIZ OPINION SPORTS TECH/GADGETS DRIVE VIDEOS
IN MONEY
JUST IN POPULAR

Banks in Malaysia to remain cautious on loan approvals

Monday, 03 May 2021 11:22 AM MYT

The banking system loan growth as at end-March 2021 rose at a slightly faster rate of +3.9 per cent year-on-year (yoy) versus +3.7 per cent yoy as at February 2021. — Reuters pic

Follow us on Instagram and subscribe to our Telegram channel for the latest updates.

KUALA LUMPUR, May 3 — Banks will remain cautious on loan approvals as approval rates fell to 40.6 per cent in the first quarter of 2021 compared to 44.1 per cent in the same period in 2020,

35 minutes ago
Thai economy shrinks, hopes for tourism-led recovery

1 hour ago
Report: Trump Organization to se Washington hotel for US$375m

2 hours ago
Public Investment Bank upgrades AirAsia X to 'neutral'

3 hours ago
Trading begins on new Beijing Sto Exchange

3 hours ago
China Oct industrial output, retail sales beat expectations

3 hours ago
MIDF Research: Foreign investors remain net buyers on Bursa for si straight week

Figure 1. 2 Loan approval cautions

Based on the newspaper article in figure 2, the bank in Malaysia have been remain cautions on loan approvals. The loan approval rates decreased to 40.6% in the first quarter of 2021 [3]. This may be due to a higher demand for loans than last year as banks may not be able to keep up with the pace of approvals. Hence, peer-to-peer lending will be an attractive alternative way for people to get loan. The borrowing interest rate is lower and the process is faster than traditional lending method. Therefore, a mobile peer-to-peer loan application that can be easily accessible allows us to borrow or make investments through a mobile phone.

1.1 Problem Statement and Motivation

Lengthy, cumbersome, unfriendly and expensive loan application process[5]

The preparation of a commercial loan application may also be a long and lengthy process. Even though Malaysia's Bank have provided loan application services via online, but Borrowers should always make their ways to the physical bank to provide the required information. For some situations, borrowers may need to go to physical bank for few times. For examples, missing of important information file like copy of identity card or the appointment for loan services was fully booked. All of these wastes a lot of time and the cost for transportations.

Another step that takes time is you must complete a loan application form for each lender or bank, but you will also be asked to provide a business plan, your loan account history, and your financial forecasts to demonstrate that your business is a viable potential client. Each application will require the applicant to submit many supporting documents which is tedious and cumbersome.

Commercial banks can take a long time to process the loan information and make a decision, particularly if your business is asking for a significant amount of money. Thus, the commercial bank needs time to process a loan request, especially if your business requires a lot of money.

Difficult to get loan approved due to strict eligibility requirements[8]

One of the reasons that banks take long time to process loan application is because banks are cautions about every loan they give. Banks are very strict toward the eligibility requirements of borrowers. Banker will refine their risk based on credit scores of borrowers. Their stringent lending requirements can make it particularly difficult for loans from start-ups and newer firms to be approved as they do not have the financial or business background to support their claim. If accepted, banks are likely to ask for a higher interest rates to counterbalance the additional risk. Hence, one might say that the bank loan is difficult to qualify unless borrowers have a significant history or high value security, such as real property. Many individuals and small businesses don't have credit history for their loans to be approved.

Lack of quality investment opportunities for investors, and difficult to find other ways of get loans for borrowers.

For commercial banks, any loan they make can be considered a bank investment. The bank can get financial return from borrower. Therefore, providing a loan was an investment by the lender or investor to the borrower. That was some sort of investment opportunity. I believed that our country has a lot of solid investors but there were too less of investment opportunities for our country. Most of the investors cannot find a qualify project or business to invest. Some may have no way of finding investments and there may have less platform that can allow them start their investments. For instance, large-scale funds are all about crypto these days, but there was a lack of a platform in our country that allows investors to invest. In the same case, peer-to-peer investor also faced the same problem, opportunities for investment are less numerous and well known. For borrowers, besides obtaining loans from banks, friends and family, there were fewer loan options for starting a business or operating a business. If those options have been turned down, they have no way of getting a loan. To conclude, there were fewer options for peer-to-peer investors to invest and for the borrowers to obtain a loan.

1.2 Project Objectives

Expedite the loan application process

Since this fintech peer-to-peer lending mobile application are usually entirely online and mobile based, this means that the application process is fast and convenient. That means users can apply for loan with a mobile phone without going anywhere. Save time for users because they may save time to make appointments with banks and transport time. This can be very convenient if you want to secure your money fast. This peer-to-peer lending mobile application will have a waiting list of investors to give loans to borrowers. When combined with an automated matchmaking process, it means that the turnaround time to complete a loan application will be quick. Hence, peer-to-peer lending can expedite the application process quickly with minimal documentation. It makes the loan application process easy by reducing the heavy documentation process carried out by traditional branks.

Eligibility criteria are often not that strict and hence, more eligible people can get loans

Peer-to-peer lending connect the borrowers to a network of investors. The eligibility criteria are often not that strict compared to traditional loan even though the factors that used to evaluate loan application are almost the same. That is because the banks will set a limit on the minimum qualifying criteria when there are a lot of peer-to-peer investors, the qualifying criteria were not the same for all investors. This project will have a prequalification test for borrowers to determine their credit scores. Credit scores of borrowers will be an important factor for investors to make investment decisions.

Provides an excellent opportunity to invest ethically and alternative method of loan application

Peer-to-peer lending mobile application brings diversity and a broad range of investors and borrowers on a unique platform to match the offer and demand. Interest rates are therefore more competitive and both sides may be very selective in what they seek. Peer-to-peer lending provides an excellent opportunity to invest ethically. Peer-to-peer lending typically offer investors a higher yield than other types of investments. For some borrowers, peer-to-peer lending provides a more accessible source of financing than traditional loans from financial institutions. For those who want get loan for business start-up and operating expenses, peer-to-peer lending is more suit compared with traditional loan. Peer-to-peer loans generally come with lower interest rates due to greater competition among investors and lower establishment fees and the process time taken is lesser than traditional loans. In conclusion, peer-to-peer lending is an alternative way for investors to get extra income and for borrowers to get quick loan with low interest rates.

1.3 Project Scope and Direction

The purpose of this project is to develop a fintech peer-to-peer lending mobile application which enables users to perform investment and borrowing activities through mobile devices. The main result of this project is the development of a Fintech peer-to-peer lending mobile application that aims to make the users experience more transparent, efficient and efficient.

Improve user experience

- Make the process of loan application and investment quick and convenient.

Improve efficiency on user manual operation

- Mobile application easy to get start for peer-to-peer lending.

Secure lending process

- mobile application that enables individuals and businesses to get loans from the crowd.

Besides, the fintech P2P lending mobile application will contain several user s modules such as borrower module, lender module, loan module, loan market module, community chat module, profile module, and administrator module.

<u>Borrower module</u>

- Borrowers able to register for a borrower account.
- Borrowers have to fill in all information needed for member registration of lending platform. After register as member, user have to wait for the management team of P2P lending platform to review and approve. In the event of approval, borrowers will be able to apply for a loan. If the application is rejected, borrowers will not be able to access the loan module.

Lender module

- Lenders able to register for a lender account.
- Lenders have to fill in all information needed for member registration of lending platform.
- Lenders have access to loan market module, profile module, and community chat module.

Loan module (For borrowers)

- Loan module open for borrowers only
- Borrowers can check the loan rates such as interest rate, monthly repayment, annual percentage rate.
- Borrowers able to track the loan application status. There are 6 status of loan application such as pending for review, pending for approval, verified and approved, processing of loan fund, completed and rejected.
- Borrowers can choose to apply loan from partner bank or get loan from lenders.

Loan market module (For lenders and borrowers)

- This module is open for both users, lenders and borrowers.
- Borrowers can post their loan application in loan market after reviewed by management teams of P2P lending mobile app.
- Lenders can offer low interest for the project (loan application) they wanted or lenders can have crowd funding for a big project.

Profile module

- Users (lenders and borrowers) able to access to profile module
- Users can check their personal information in profile
- User must verify their email address and phone number to continue using this mobile application.

Community chat module

- Administrator (management teams), lender and borrower able to communicate with this chat module.
- Administrator able to request for missing document or information from users.

Administrator module

- Admin able to view all user data, loan application and saving application.
- Admin able to approve user registration, loan application and saving application.
- Admin able to check blacklist to verify user registration and loan application. If user is blacklisted, admin can reject their member registration and loan application.
- Admin able to check the total of users and loans application.
- Admin able to update loan application status.

1.4 Contributions

The outcome of this project is a fintech peer-to-peer lending mobile application that can directly matching borrowers' lending needs with consenting lenders and legal contract obligations exist directly among borrowers and lenders. This is a free to use platform that make connection with investors and borrowers. For those who are looking for an alternative to traditional banks, peer-to-peer lending provides an excellent alternative which is worthwhile to explore. This application can fulfil for those who are looking for alternative finance for extra income. The significance of this application is that it will assess the solvency of borrowers in terms of parameters. The proposed features and solution to improve users experience on the lending process by automating the lending process. The evaluation of borrowers is at least partially automated against predefined rules of fast and transparent processing. This application will be free from legacy processes and technologies, enabling it to more efficiently integrate and evaluate borrowers and lenders.

1.5 Report Organization

The Chapter 1 of this report included the project introduction, problem statement and motivation, research objective, project scope and direction, contribution of the project. All the discussions mentioned in Chapter 1 will assist the developer to better understand the project and the user's needs.

Chapter 2 will provide a literature review of existing work and projects that are similar to the proposed project. It can give the developer a clear idea about the functionality and features of other similar projects or mobile application. The developer may have some preliminary idea based on the literature review. The strengths and weaknesses of the existing project will be discussed together with a comparative table of characteristics and functionalities.

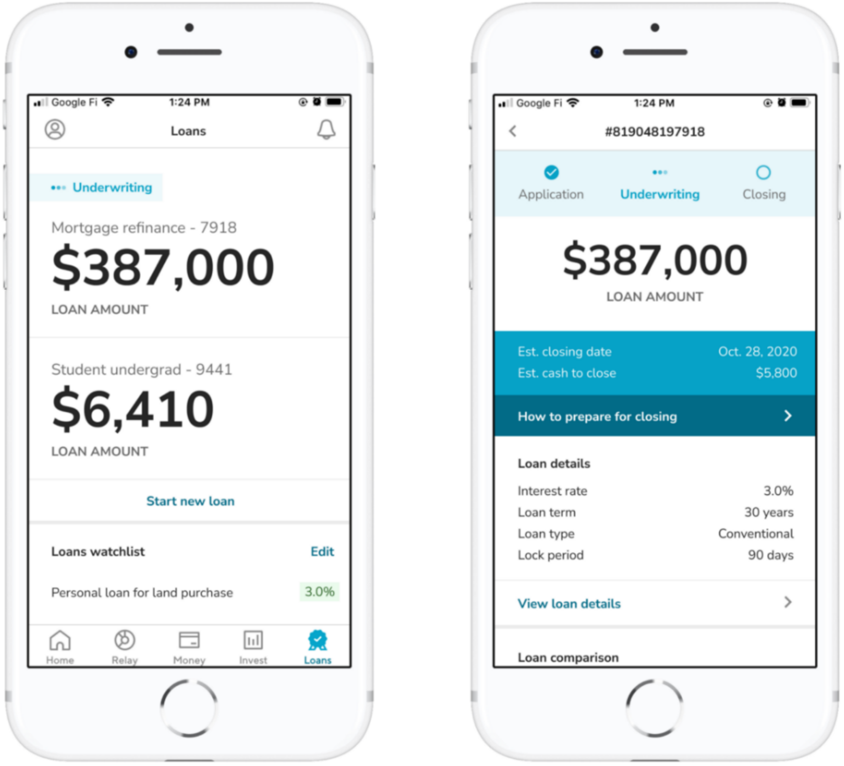
For Chapter 3, it will discuss about the system methodology and approach of the system. System Design Equation and the Overall Uses Case Diagram will be included in this Chapter. Chapter 4 will discuss about the system design of the proposed project such as the system block diagram and system flow description with screenshot.

In Chapter 5, system implementation will be discussed as well. The hardware and software setup will be defined, setting and configuration of the system will be discussed. The most important part in Chapter 5 is the system operation with screenshot. Chapter 6 will discuss about the system evaluation. An evaluation survey result will be shown and the testing setup of the proposed system. Chapter 7 will include the conclusion and recommendation of proposed system.

2 CHAPTER 2 Literature Reviews

In this chapter, the relevant literature on existing similar peer-to-peer lending mobile application will be reviewed. The purpose of the literature review for the existing similar mobile application is to identify its strengths and weaknesses. Four existing peer-to-peer lending mobile application which is Sofi, LendingCrowd, Funding Societies, Zopa, and LendingClub will be analysed, compared, and reviewed

2.1 Sofi mobile application



Mortgage refinance - 7918
$387,000
LOAN AMOUNT

Student undergrad - 9441
$6,410
LOAN AMOUNT

Personal loan for land purchase 3.0%

$387,000
LOAN AMOUNT

Est. closing date Oct. 28, 2020
Est. cash to close $5,800

Loan details
Interest rate 3.0%
Loan term 30 years
Loan type Conventional
Lock period 90 days

Figure 2. 1 Sofi Mobile Application Interface

Sofi (short for “social finance”) started as a venture focused on refinancing student loans and personal loans [4]. Sofi was founded by the students of Stanford business school student at Aug 2011. Initially, Sofi was created for graduate-funded loan model that used to establish connections between recent graduates and graduates from their community. In May 2012, Sofi introduced student loan refinancing, making it the first company to refinance both federal and private student loans. Now, Sofi have expanded their business by launching Sofi Money and Sofi Invest. Sofi offered personal loans, student loan, stock investment and online cash management. Sofi money is a well-known non-bank financial services provider that provides student loan refinancing services[12]. Sofi Invest is an all-in-one investment platform that allows users to trade stock, cryptocurrency and so on[1]. Sofi mobile application included all of these services; it was a powerful application.

Strengths

1. **variety of financial products** - Sofi is good for people who are seeking innovative new technologies in fund management. It has huge variety of

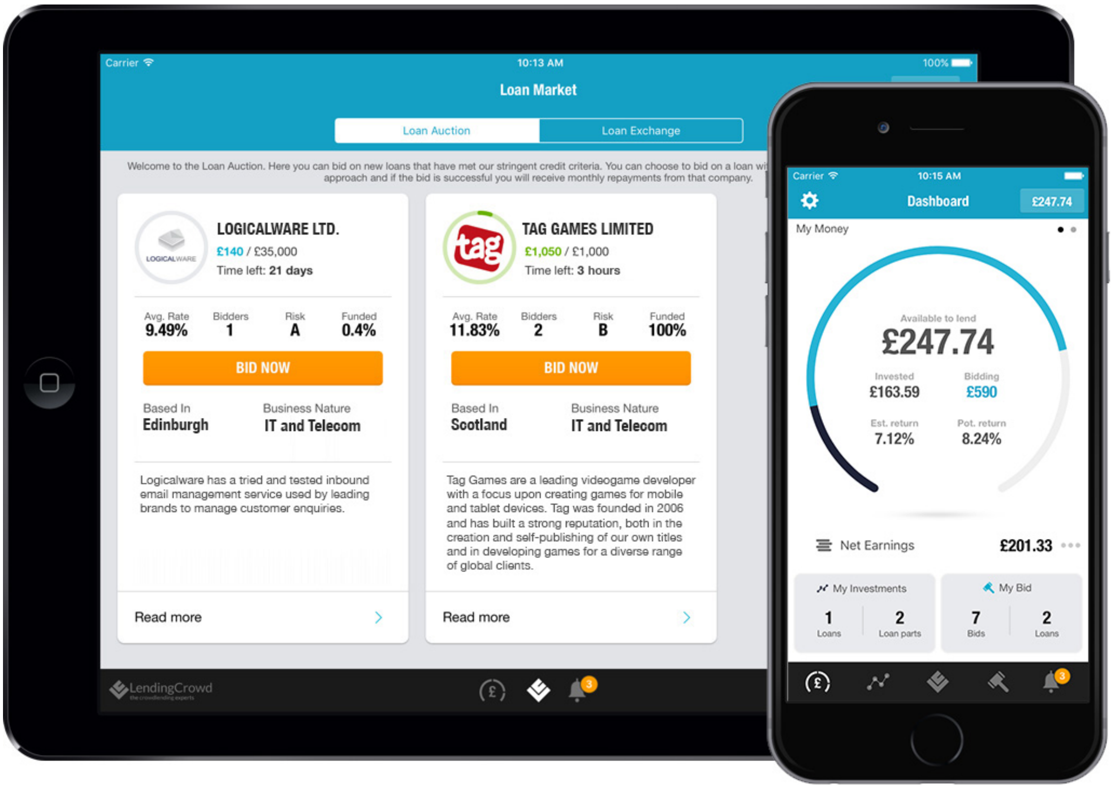
financial products such as loans, insurance, investments and others. Sofi have also invested heavily in the financial management technology development.

2. **Low loan fees-** Sofi give users the access to low-cost credit at no cost. Sofi do not have origination fee, prepayment penalty and late fees charges. It also offers competitive interest rates of a 0.25% discount for those who make automatic payments.
3. **Customer service-** Sofi provide 24/7 customer support service.
4. **More features-** The Sofi mobile application contained four main features such as Sofi Relay, Sofi Money, Sofi Invest and Sofi Loan. Sofi Relay allows users to tracks all of their money, monitor the credit score, expenditure breakdown, financial information and more with free of charge.
5. **ATM access supported -** Sofi provided more than 55000 free ATM access. It also provides a combination of control and savings features without monthly or overdraft charges and without a minimum opening balance.
6. **Expedite the application process**- Sofi loan application only required few documents such as state issued ID, utility billing statement and banking statement. Sofi has internal systems to automatically check pay information, so it was not necessary to print anything in PDF format and scramble tons of documents.
7. **More investment option -**Sofi Invest enables users to trade stock, cryptocurrency and others.
8. **Prequalification -** new users need to complete a prequalification test and provide the documents requested in order to sign up successful. For borrowers, the mobile application will analyze credit score of users and show in user profile.

Weakness

1. **High APR rate-**Sofi loan collected high annual percentage rate (APR) compared to other competitors. The financing process is slower compared to some other lenders.
2. **High eligibility requirements** - The minimum loan amount is $5000. High eligibility requirements compared to others. Individuals whose credit is more precarious may find it difficult to qualify.
3. **Restrict area -** Sofi was only available for U.S. residents. Not able to make international money transfers.
4. **No branches support -** Sofi Money have ATM access but have no branches.
5. **Limited trading record-** Sofi Invest has no tax-loss harvesting and there was limited trade record because it was launched since 2017 only.

2.2 LendingCrowd mobile application


Carrier 10:13 AM 100%

Loan Market

Loan Auction | Loan Exchange

Welcome to the Loan Auction. Here you can bid on new loans that have met our stringent credit criteria. You can choose to bid on a loan wi
approach and if the bid is successful you will receive monthly repayments from that company.

LOGICALWARE LTD.
£140 / £35,000
Time left: 21 days

Avg. Rate	Bidders	Risk	Funded
9.49%	1	A	0.4%

BID NOW

Based In: Edinburgh
Business Nature: IT and Telecom

Logicalware has a tried and tested inbound email management service used by leading brands to manage customer enquiries.

Read more

TAG GAMES LIMITED
£1,050 / £1,000
Time left: 3 hours

Avg. Rate	Bidders	Risk	Funded
11.83%	2	B	100%

BID NOW

Based In: Scotland
Business Nature: IT and Telecom

Tag Games are a leading videogame developer with a focus upon creating games for mobile and tablet devices. Tag was founded in 2006 and has built a strong reputation, both in the creation and self-publishing of our own titles and in developing games for a diverse range of global clients.

Read more

Carrier 10:15 AM

Dashboard £247.74

My Money

Available to lend: £247.74
Invested: £163.59
Bidding: £590
Est. return: 7.12%
Pot. return: 8.24%

Net Earnings: £201.33

My Investments: 1 Loans, 2 Loan parts
My Bid: 7 Bids, 2 Loans

Figure 2. 2 LendingCrowd Mobile Application Interface

LendingCrowd was a lending platform that leading in financial technology and headquartered in Scotland. LendingCrowd was launched on 1 October 2014 to finance the ambitions of companies throughout the United Kingdom. LendingCrowd online platform is completed by their highly experienced credit team, who oversees every loan request [6].

Strengths

1. **Prequalification –** new users need to complete a prequalification test and provide the documents requested in order to sign up successful. For borrowers, the mobile application will analyze credit score of users and show in user profile.
2. **Have better credit spread graph** that show difference between the return of two different debt securities having the same maturity but different credit ratings. Users can make a good decision by using the credit spread graph to compare different investment project.
3. **Notification when loan part sells.** Users get notification immediately when loan part sells. This is useful for users for the arrangement and decision making of investment.
4. **Best user experience and features** – able to view and transfer funds to all users' account. Users able to monitor their investment well with a mobile phone in anywhere and anytime.
5. **Easy to navigate** – easy to use and allows users to invest in UK Small and Mid-size Enterprise (SME).

6. **Fingerprint ID login –** users able to login their account via fingerprint.

Weaknesses

1. **Limited sort option –** loan market screen did not show the availability of some loan part. Users forced to click on the company that shown in loan market and the application required users to log in again every time.
2. **Fingerprint ID login not function well –** after enable fingerprint ID, users still need to type in password when using fingerprint to login every time.
3. **Lack of some important features compared with LendingCrowd Website –** mobile application version cannot view and manage ISAs
4. **restriction for UK resident only.**
5. **The mobile application available for IOS phone only currently.**

2.3 Funding Societies mobile application

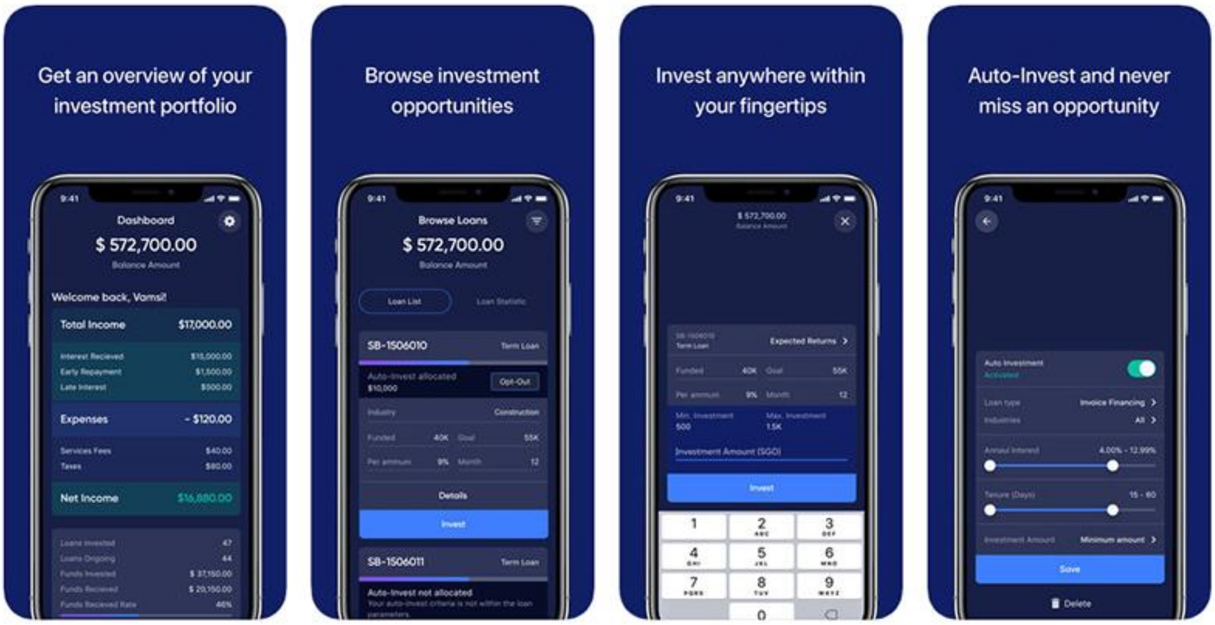


Figure 2. 3 Funding Societies Mobile Application Interface

Funding Societies is a loan investment platform for Small and Mid-size Enterprise (SME) in Singapore. Funding Societies launched in 2015 and aim to elevate Southeast Asian Companies by Creating Financial Opportunities for All. Funding Societies provided the short-term fixed income option for individuals and institutions. There are three main financing option provided by Funding Societies, which is micro financing, team financing and invoice financing. Micro financing for the smaller amount loan or individual's loan without collateral. Team Financing was funding for business expansion and large amount loan while Invoice Financing for customers to convert their invoices into cash for their account payables, account receivables, and purchases of facilities.

Strengths

1. **Prequalification** – new users need to complete a prequalification test and provide the documents requested in order to sign up successful. For borrowers, the mobile application will analyze credit score of users and show in user profile.
2. **User friendly interface** – clear statement of the income expenses and balance about users' investment.
3. **Neat categorization of portfolio** – Users can have a better look on investment option and easy for users to make quicker decisions.
4. **Low investment start amounts** – minimum investment start amounts was $20 for Singapore region, RM100 for Malaysia and Rp100,000 for Indonesia

5. **Auto invest bot** – Users able to set up auto invest or planned funding and allow the system to automatically roll out funds for users according to your investment preferences.
6. **Factsheet Available –** factsheet contain information about the businesses or the loan. It also shows a company heath like revenue ratio, net profit margins.
7. **Easy to deposit** – fast process and have a lot of deposit option such as, internet banking service, credit/debit card.
8. **Procedures in Place in event of Default –** Funding societies will recover as best as they can the repayment money that the borrower did not pay on time. Funding societies will start to negotiate with the borrowers or businesses that did not pay repayment or interest on time by reducing the interest charge or prolonging the period of repayment.
9. **Available for Malaysia, Singapore and Indonesia.**

Weaknesses

1. **Fingerprint login response slow –** slow detection of fingerprint and sometime not able to login with fingerprint.
2. **Slow withdrawal process and repayment –** First repayment will be in 2 months and no instant lumpsum. Takes more than two weeks for the withdrawal process.
3. **Inaccurate default rate –** the default rate display in the mobile application is only 1.37% but the actual default rate was higher than 1.37%
4. **Limited option for specification filter and auto investing** – Users cannot filter for the specific of borrowers, the mobile application cannot view borrowers' performance history such as loan application history. For auto invest, there was fewer option for users to set up.
5. **Not powerful filter –** less option for users to filter investments to find out the best choice.
6. **Time delay for manual investment** – There will be a delay for manual investment made by users when subscribe on an investment. When the manual investment was made, the application shown a success subscribed message, but after refreshed the application, the subscription gone. The reason for this problem was the auto bot invest. The auto bot investment was faster than manual investment.
7. **Restrict area -** Available for Malaysia, Singapore and Indonesia only.

2.4 Zopa mobile application

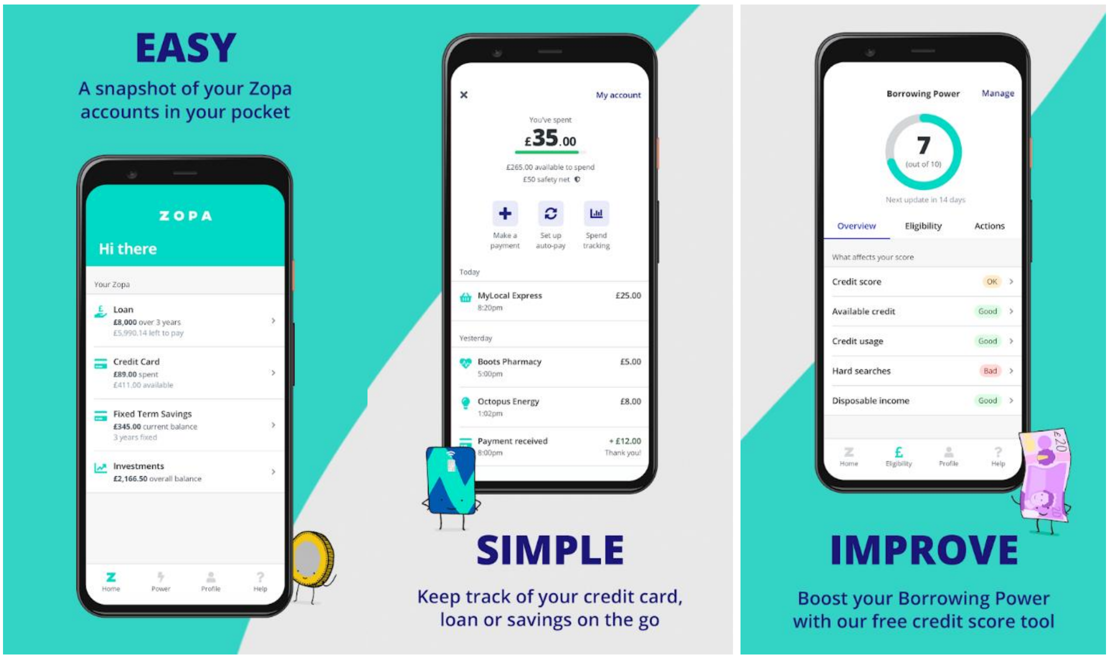


Figure 2. 4 Zopa Mobile Application Interface

Zopa was launched in 2005 as a peer-to-peer lending company that give people simple access, better values of loans and investments. Zopa have expanded their business by launching Zopa bank in June 2020. Zopa peer-to-peer lending service and Zopa bank work together to make things more equitable, easier and better for our clients.

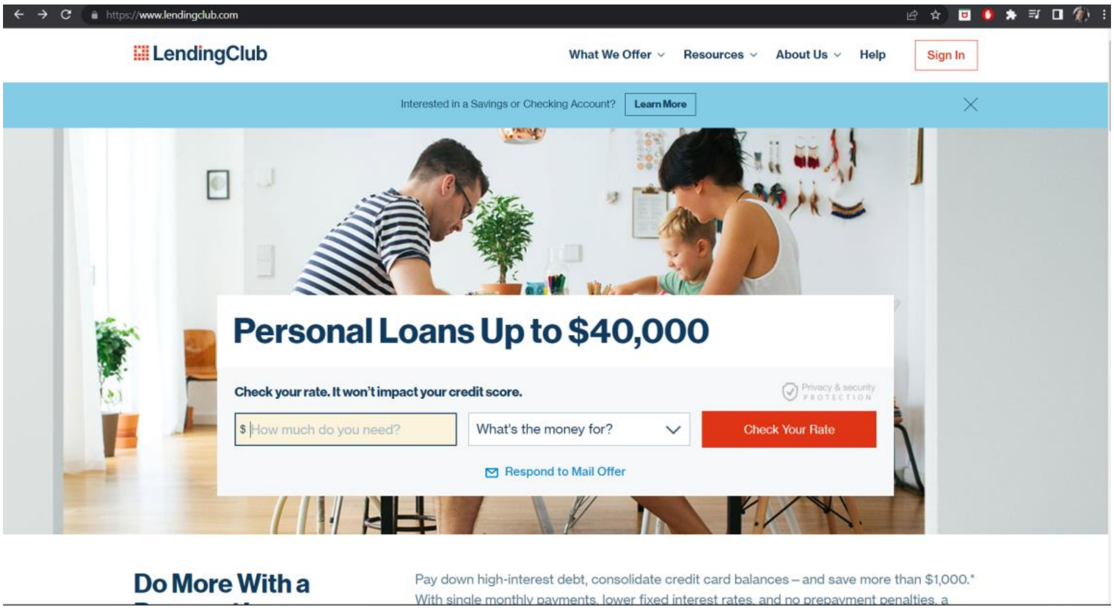
Strengths

1. **Prequalification** – new users need to complete a prequalification test and provide the documents requested in order to sign up successful. For borrowers, the mobile application will analyze credit score of users and show in user profile.
2. **More features** – Zopa mobile application contains four main features which is loan, credit card, fixed term savings and investments.
3. **Record track of credit card, loan and savings** – Users can keep track of all these records by Zopa mobile application. It has complete statements for every transaction that made by users.
4. **Credit score tool** – free credit score tool to analyze borrowers' borrowing power. Investors and borrowers can have a better understanding on their credit score in order to make best decisions.
5. **Fingerprint login** – Users able to check their account balance instantly and securely with fingerprint login.
6. **User experience** – simple mobile application contains basic features. Easy for users to monitor their account. Simple UI design.

Weaknesses

1. **Login issue –** White screen after user login to the mobile application. Users cannot perform any action like check account balance, make investment when having this login issue.
2. **Chat bot –** Zopa mobile application did not have chat bot. Users can only get help from their website customer services or phone call and email.
3. **Limited features on mobile application –** Users can perform basic action via Zopa mobile application. The features contained was lesser than website version, such as no ISA in mobile application.
4. **Restrict area –** available for US resident only.

2.5 LendingClub



Interested in a Savings or Checking Account?

Personal Loans Up to $40,000

Check your rate. It won't impact your credit score.

Do More With a

Pay down high-interest debt, consolidate credit card balances – and save more than $1,000.*
With single monthly payments, lower fixed interest rates, and no prepayment penalties, a

Figure 2. 5 LendingClub Website Interface

LendingClub is a peer-to-peer lending club that launched in 2007. This lending platform aim to help Americans to meet their life goals. LendingClub was the only large-scale financial technology market bank. LendingClub delivered a wide variety of financial products and services through a technology-enabled platform which is aimed at helping borrowers and lenders pay less when they borrow and earn more when they save. LendingClub offer personal loans, business loans, auto refinancing, and patient solutions to users. Personal loans provided by LendingClub have a limit of USD 40,000 and the business loan limit was USD 500,000. Auto refinancing of LendingClub refers to the to settle the balance of an existing automobile loan by requesting a new loan with lower interest rate. The patient's solution is similar to the medical card, but the medical card was like prepaid and the patient's solution is somehow post-paid. Patient Solution is a loan, payment plan for patients with affordable treatments.

Strengths

1. **Wide range of loan** -LendingClub has offered a wide range of loans such as personal loan, business loan, car refinancing loan as well as patient solution.

2. **Quick check interest rate**- It also provided different interest rates with different lending goals. For example, borrowers can borrow a low-interest vacation loan when compared to a home purchase loan. In addition, borrowers can also verify the lending rate of various loan objectives by completing the basic loan information.

3. **Credit scores base interest rate calculation-** LendingClub will verify the borrower's credit score prior to calculating the borrower's final interest rate.

4. **Basic P2P lending resource-** LendingClub has created a resource page for its members (lender and borrower) to familiarize themselves with the P2P credit granting process. In resource page, LendingClub provided borrower resource,

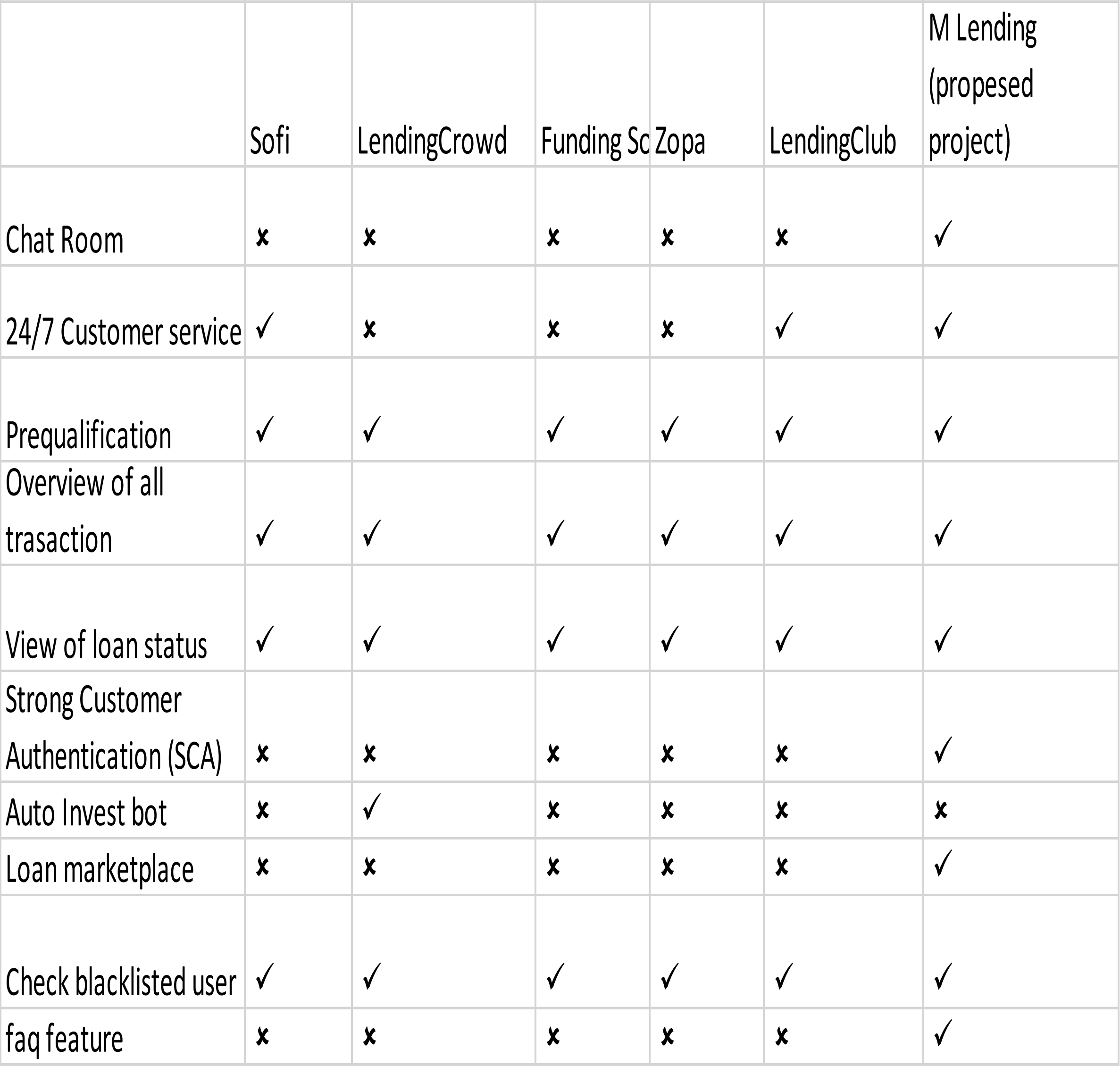
bank resource and investor resource for their members can have quick understanding about the basic knowledge of personal loan, fixed term saving or investment performance and strategy.

5. Co-borrowers accepted- LendingClub accepted co-borrower for loan application to have a lower interest rate. It is more secure for a loan application with a co-borrower.

Weaknesses

1. **No community chat or live chat-** LendingClub does not provide live chat or community chat services to enable all members to communicate effectively.
2. **High interest rate for credit scores base interest rate calculation -**For the calculation of the interest rate with credit score, the borrower with a low credit rating will get a higher interest rate than the conventional bank.
3. **Limited loan terms and an origination fee** -LendingClub have Limited loan terms and an origination fee, they charged an origination fee and have only two loan repayment term which is 36 months and 60 months. The origination fee will deduct directly from the loan funds.
4. **Restrict area -**Available for United State (US) resident only.

2.6 Comparison Table



	Sofi	LendingCrowd	Funding Sc	Zopa	LendingClub	M Lending (propesed project)
Chat Room	✘	✘	✘	✘	✘	✓
24/7 Customer service	✓	✘	✘	✘	✓	✓
Prequalification	✓	✓	✓	✓	✓	✓
Overview of all trasaction	✓	✓	✓	✓	✓	✓
View of loan status	✓	✓	✓	✓	✓	✓
Strong Customer Authentication (SCA)	✘	✘	✘	✘	✘	✓
Auto Invest bot	✘	✓	✘	✘	✘	✘
Loan marketplace	✘	✘	✘	✘	✘	✓
Check blacklisted user	✓	✓	✓	✓	✓	✓
faq feature	✘	✘	✘	✘	✘	✓

Table 2. 1 Table of Comparison

3 CHAPTER 3 Proposed Method/Approach

3.1 Methodology

The methodology applied in this project will be prototype model. Prototype model is a system development method in which a prototype is constructed, tested and then re-worked as needed until an acceptable result is achieved. The reasons to use this model is because it was flexible in design and it can actively engage users during the development phase. In other hand, it can help us to understand the system better. The integration requirements are very well understood and the deployment channels are determined at an early stage.

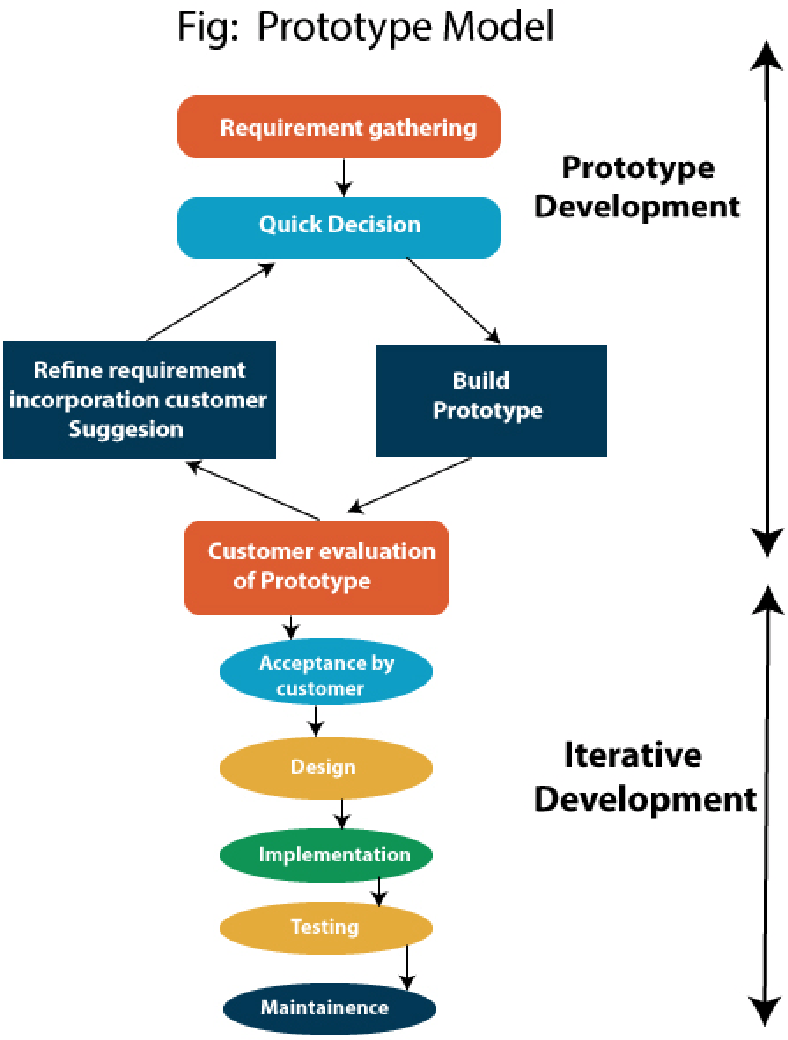


Figure 3. 1 Prototype Model

Prototype development

1. **Requirement gathering**

A prototyping model starts with requirement analysis. At this stage, I collect data from existing product like LendingClub, Sofi to gather the requirements.

2. **Quick design**

The second phase is a quick design. This stage will create a simple system design but not complete design. I have done research of similar product in Chapter 2 and make reference on the P2P lending mobile application design based on my research.

3. **Build a Prototype**

In this phase, a real prototype is designed on the basis of the information collected from the quick design. I build my financial peer-to-peer lending mobile application by using Android studio IDE and firebase as online database to store the lending data.

4. **User evaluation of Prototype and Refine requirement incorporation user suggestion**

During this phase, the proposed system will be presented to the customer for an initial assessment. I will collect the feedback from my friends, colleague and my supervisor about the testing of prototype. Then, make improvement on the prototype until the user is satisfied with the outcome, a final system will be developed based on the approved final prototype.

5. **Implementation and Testing**

In this last phrase, if the final P2P lending mobile application is built on the basis of the final prototype, it is thoroughly tested and deployed for production. Regular maintenance will be carried out on the final mobile application, so it will help minimize downtime and prevent large-scale failures

3.2 System Design Diagram/Equation

Following is the format of the scientific formula for calculating the monthly repayment loan amount:

$$\boldsymbol{A = P\frac{r(1+r)^n}{(1+r)^n - 1}}$$

where

- A = monthly repayment amount
- P = initial Principal (loan amount)
- r = interest rate per month
- n = total number of months

3.3 Use Cases Diagram

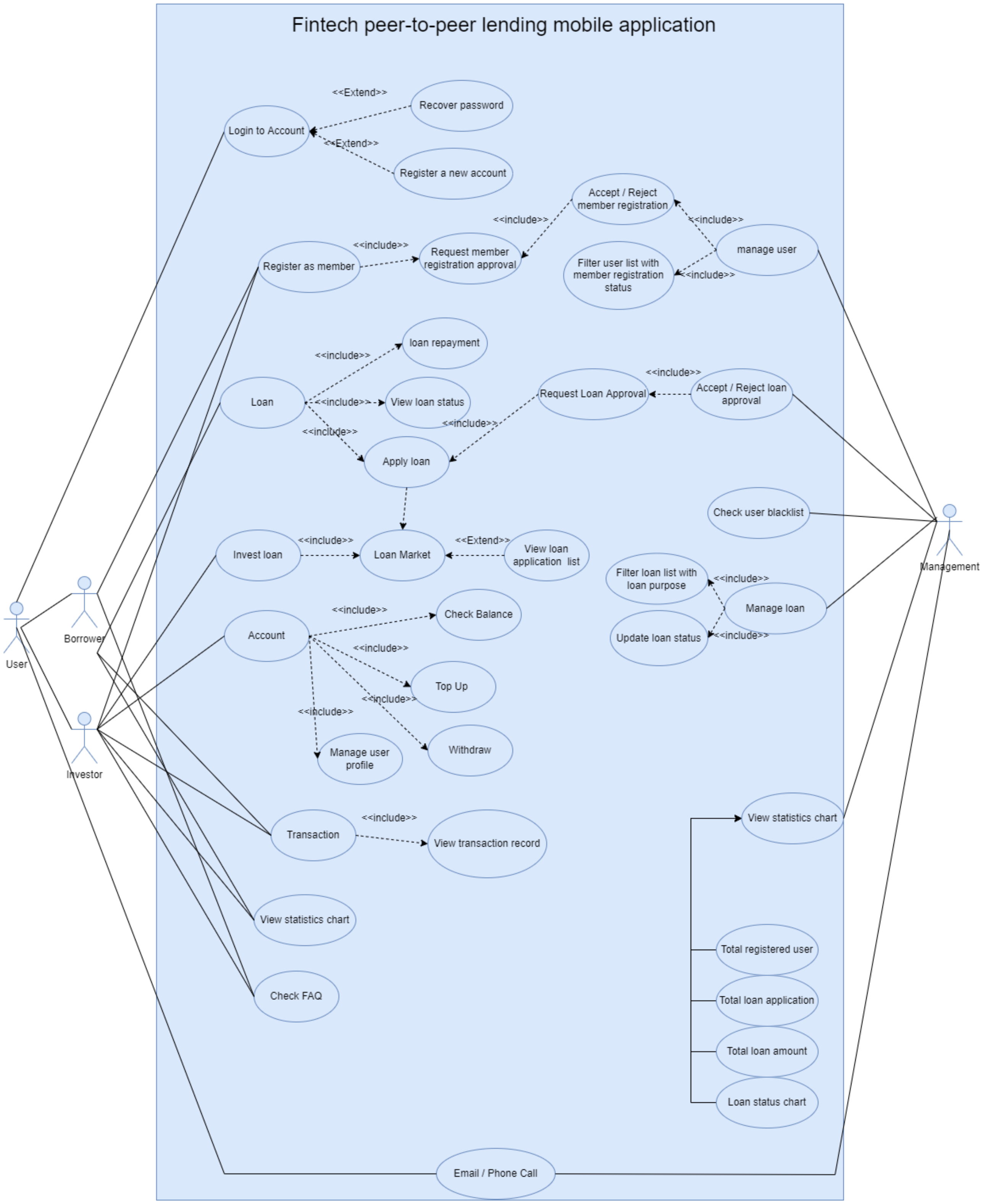


Figure 3. 2 Use Case Diagram

3.3.1 Login and Register Module

Use Case ID	1
Use Case Name	Login to account
Brief Description	This use case describes how to handle the process of login to the users' account
Actor	User
Trigger	Go to login page of the mobile application
Name flow of events	1. User needs to enter username and password. 2. The system will check the input validation. 3. The system will check wheather the username exists in the system or not. 4. The system will check the username and password correct or nto. 5 . The system will load all the user's details and direct access to the main page.
Sub flows	Not applicable
Alternate flows	1a. User needs to press "Forgot Password" to reset his/her password. 3a. User was not registered an account yet, he/she need to press "Register account" to register an account.

Table 3.3. 1 Login

Use Case ID	2
Use Case Name	Recover password
Brief Description	This use case describes how we handle the process when user forgot about his/she password
Actor	User , system
Trigger	User want to reset his/her password
Name flow of events	1. User needs to insert his/her username, email address, and new password. 2. The system will check the wheather username, email address exists in the system or not. 3. User needs to insert twice of tnew password. 4. The system will save and update the new password to Firebase database.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 2 Reset Password

Use Case ID	3
Use Case Name	Register a new account
Brief Description	Handle the process of new account registration
Actor	User, System
Trigger	User want to register an account
Name flow of events	1. User register as a M Lending user. 2. The user needs to insert personal details like full name, email address, password and also account type (borrower, investor). 3. User require to fill in all personal details, if not the system will focus on the column and alert user. 4. The system will check whether the user is exists or not. 5. The system will save and update all the user data to firebase. 6. The system will direct user to login page.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 3 Register

3.3.2 Member Registration Module

Use Case ID	4
Use Case Name	Register as member
Brief Description	Handle the process of member registration
Actor	Borrower, Investor, System
Trigger	User want to get access to loan and invest module
Name flow of events	1. User needs to insert personal details and contact person details. 2. User needs to provide all the details to register as a member. 3. It will take 2-3 working days for management team to approve / reject for the membership registration. 4. If approved, user can grant full user access to the system.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 4 Member Registration

3.3.3 Loan Module

Use Case ID	5
Use Case Name	Apply loan
Brief Description	Handle the process of loan application
Actor	Borrower, System
Trigger	User want to make a loan application
Name flow of events	1. User needs to insert the loan amount and select loan purpose. 2. User need to click "check rate" button to check the loan rate and details such as interest rate, monthly repayment, loan fees, apr rate and total repayment. 3. User can select the loan repayment month. 4. If user is not satisfied with the loan rate given by system, user can click "not satisfy" button to apply loan in loan market. 5. If user want to get the loan, the system will prompt password validation. 6. If password is verified, the system will update the loan details in Firebase. 7. It will take about 2-3 working days for management team to make decision on the loan application. 6. The system will direct user to view loan details page.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 5 Apply Loan

Use Case ID	6
Use Case Name	View loan status
Brief Description	Handle the process of letting user to view loan application status
Actor	Borrower, system
Trigger	User want to know about his/her loan status
Name flow of events	1. User clicks on "Loan" button. 2. The system will redirect user to loan page. 3. The system will display all user's loan status in a list in loan page. 4. User can click the list to view more loan status details. 5. The system will update the loan status on time after the management team updated the status.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 6 View Loan Status

Use Case ID	7
Use Case Name	Loan repayment
Brief Description	Handle the process of letting user to pay their loan repayment
Actor	Borrower, system
Trigger	User want to pay his/her loan repayment
Name flow of events	1. User select the "completed" status loan in the view loan status list. 2. The system will display a "repayment" button. 3. User clicks the "repayment" button, the system will redirect user to repayment list page. 4. The system will display all the monthly repayment amount. 5. User clicks the "pay" button to pay the loan repayment monthly. 6. The system will update the loan repayment status and save in firebase.
Sub flows	Not applicable
Alternate flows	5a. User key in wrong password, the system will verify user's password before performing the payment process. 5b. Insufficient message prompt, user do not have enough balance to pay the repayment.

Table 3.3. 7 Loan Repayment

3.3.4 Investment Module

Use Case ID	8
Use Case Name	Invest loan
Brief Description	Handle the process of loan investment
Actor	Investor, System
Trigger	User want to make an investment on loan
Name flow of events	1. User needs to clicks "invest" button. 2. The system will redirect user to investment page. 3. The system will display a list of all available loan application. 4. User can filter the loan list by loan purpose. 5. User can view his/her invested loan. 6. User can contact the borrower by clicking the "contact" button.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 8 Invest Loan

3.3.5 Account Module

Use Case ID	9
Use Case Name	User Profile
Brief Description	Handle the process of letting user to view account details
Actor	User, system
Trigger	User want to view user profile
Name flow of events	1. User needs to clicks "profile" button; the system will redirect to profile page. 2. User can change profile picture, view account balance, verify email address and phone number in profile page. 3. System will upload and save the profile picture in firebase. 4. User able to view the customer service details like email address and contact number
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 9 User Profile

Use Case ID	10
Use Case Name	Top up
Brief Description	Handle the process of top up account balance
Actor	User, system
Trigger	User want to top up account balance
Name flow of events	1. User needs to clicks "top up" button, the system will redirect user to top up page. 2. User can insert the amount to top up. 3. If user confirm to top up, system will prompt a password verification to user's password. 4. The system will save the latest user's account balance to firebase.
Sub flows	Not applicable
Alternate flows	1a. User have to link his/her bank card before they perform the top up process.

Table 3.3. 10 Top Up

Use Case ID	11
Use Case Name	Withdraw
Brief Description	Handle the process of withdraw account balance
Actor	User, system
Trigger	User want to top up account balance
Name flow of events	1. User needs to clicks "withdraw" button, the system will redirect user withdraw page. 2. User can insert the amount to withdraw. 3. If user confirm to withdraw, system will prompt a password verification to user's password. 4. The system will check whether the balance is sufficient to withdraw or not; If not, an "insufficient balance message" will prompt. 4. The system will save the latest user's account balance to firebase.
Sub flows	Not applicable
Alternate flows	1a. User have to link his/her bank card before they perform the withdraw process.

Table 3.3. 11 Withdraw

3.3.6 Transaction Module

Use Case ID	12
Use Case Name	Transaction
Brief Description	Handle the process of letting user to view transaction records
Actor	User, system
Trigger	User want to view transaction records
Name flow of events	1. User needs to clicks "transaction" button. 2. The system will redirect user to transaction page. 3. The system will display all the transaction records in a list. 4. Borrower can view transaction records such as repayment, loan fund, top up and also withdraw amount. 5. Investor can view transaction records such as investment, receive of repayment, top up and withdraw amount.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 12 Transaction

3.3.7 Manage User Module

Use Case ID	13
Use Case Name	Manage user
Brief Description	Handle the process of user account management
Actor	Management, system
Trigger	Management wants to manage all user account
Name flow of events	1. Management click "user list" button. 2. The system will redirect management to user list page. 3. The system will display all the user details in a list. 4. Management can approve / reject the member registration of user. 5 The system will update and save the user data in firebase.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 13 Manage User

3.3.8 Manage Loan Module

Use Case ID	15
Use Case Name	Manage loan
Brief Description	Handle the process of loan application management
Actor	Management, system
Trigger	Management wants to manage all loan application
Name flow of events	1. Management click "loan list" button. 2. The system will redirect management to loan list page. 3. The system will display all the loan application details in a list view. 4. Management can check the blacklist to decide whether to approve or reject the loan application of user. 5 The system will update and save the user data in firebase.
Sub flows	Not applicable
Alternate flows	Not applicable

Table 3.3. 14 Manage Loan

4 Chapter 4 System Design Diagram

4.1 System Block Diagram

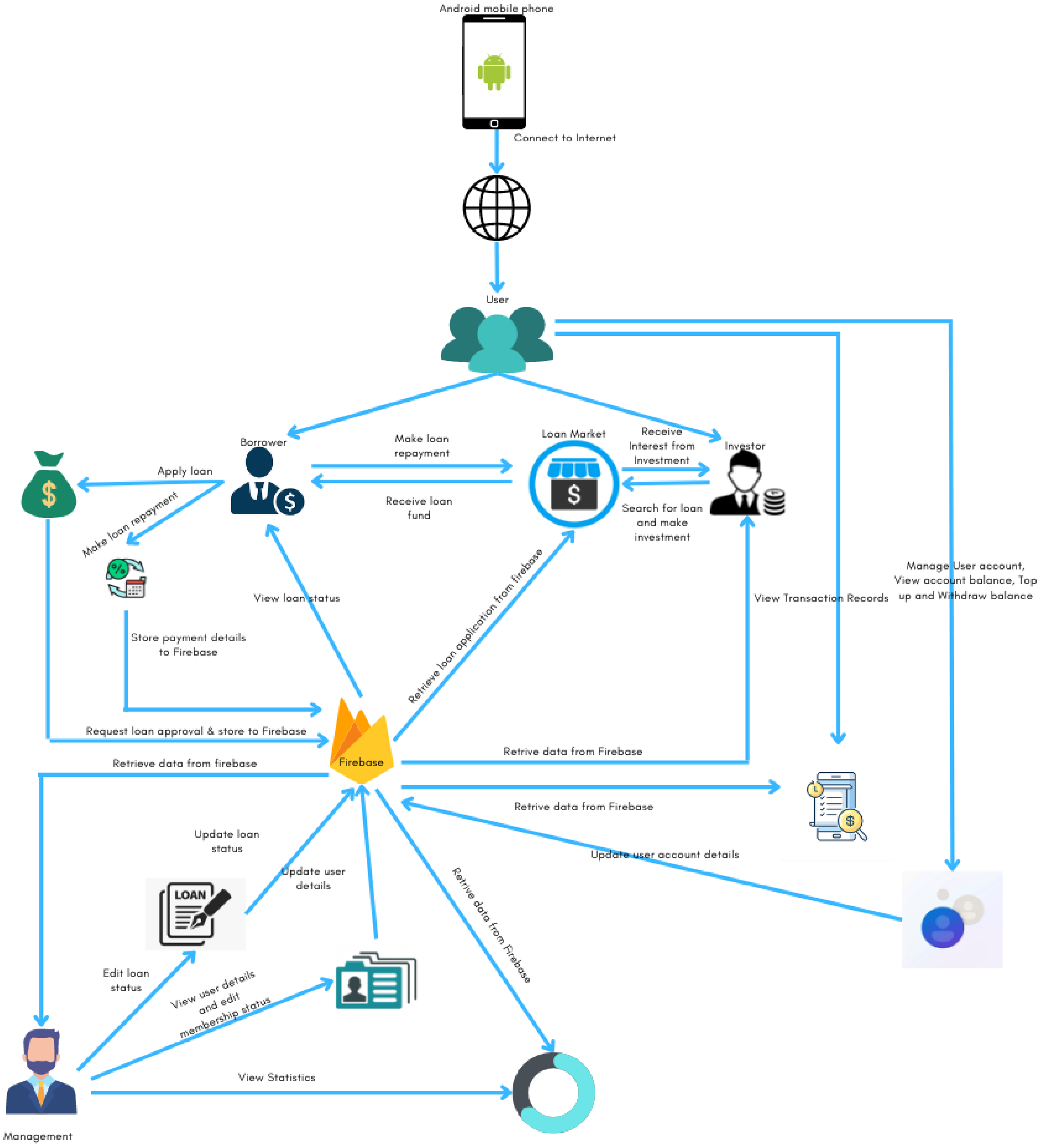
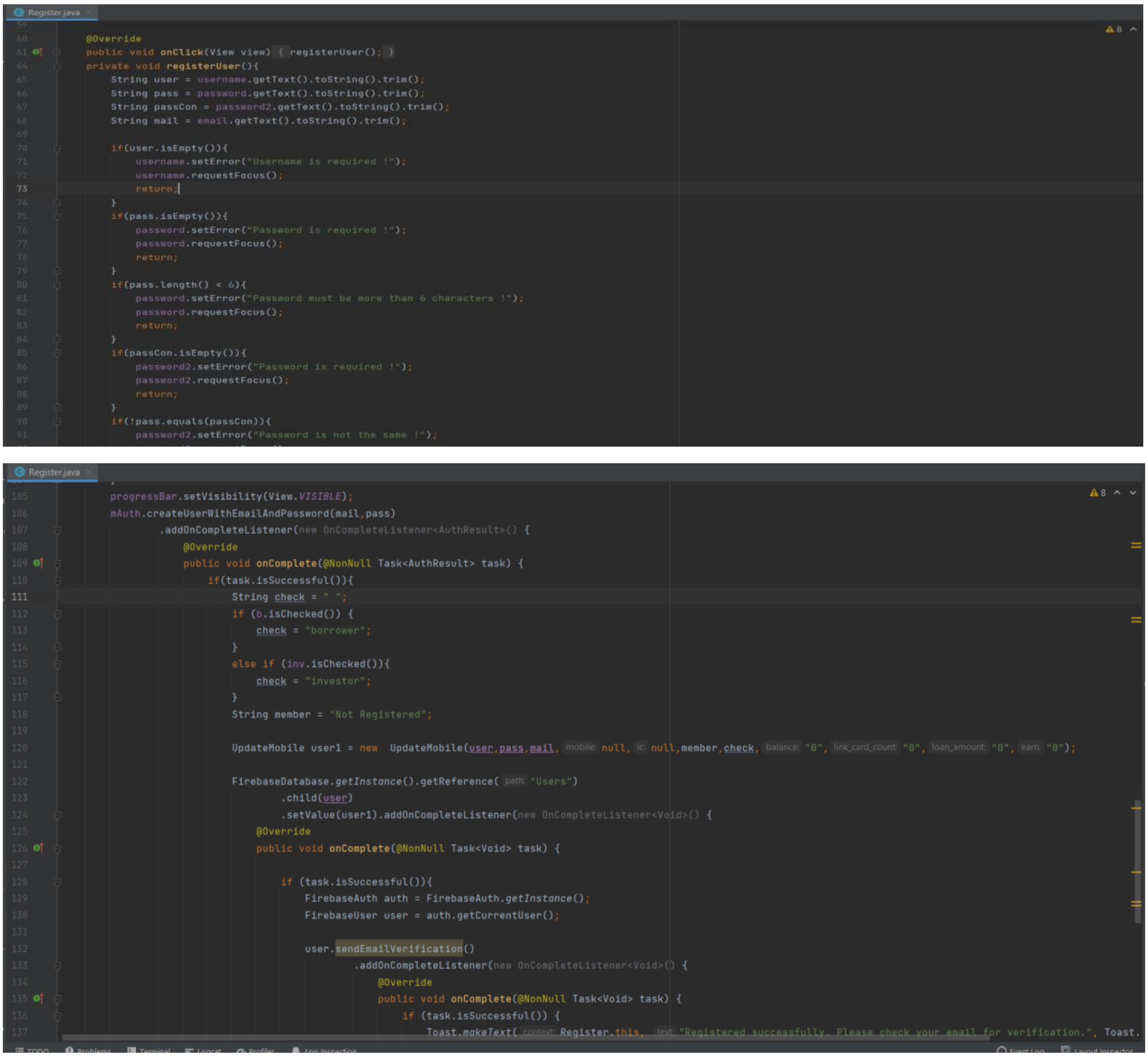

Android mobile phone
Connect to Internet
User
Borrower
Make loan repayment
Loan Market
Receive Interest from Investment
Investor
Apply loan
Receive loan fund
Search for loan and make investment
Make loan repayment
View loan status
Retrieve loan application from firebase
Manage User account, View account balance, Top up and Withdraw balance
View Transaction Records
Store payment details to Firebase
Request loan approval & store to Firebase
Firebase
Retrive data from Firebase
Retrieve data from firebase
Retrive data from Firebase
Update loan status
Update user details
Update user account details
LOAN
Retrive data from Firebase
Edit loan status
View user details and edit membership status
View Statistics
Management

Figure 4.1. 1 System Block Diagram

4.2 System Flow Description

To utilise the M Lending mobile application, the user must first connect to the Internet using his or her Android phone. The user must register as a user of M Lending mobile application, the user data can be found in Firebase Authentication, Else, the user is not able to login to the application.

4.2.1 Register.java



Register.java

```
        @Override
        public void onClick(View view) { registerUser(); }
        private void registerUser(){
            String user = username.getText().toString().trim();
            String pass = password.getText().toString().trim();
            String passCon = password2.getText().toString().trim();
            String mail = email.getText().toString().trim();

            if(user.isEmpty()){
                username.setError("Username is required !");
                username.requestFocus();
                return;
            }
            if(pass.isEmpty()){
                password.setError("Password is required !");
                password.requestFocus();
                return;
            }
            if(pass.length() < 6){
                password.setError("Password must be more than 6 characters !");
                password.requestFocus();
                return;
            }
            if(passCon.isEmpty()){
                password2.setError("Password is required !");
                password2.requestFocus();
                return;
            }
            if(!pass.equals(passCon)){
                password2.setError("Password is not the same !");
```

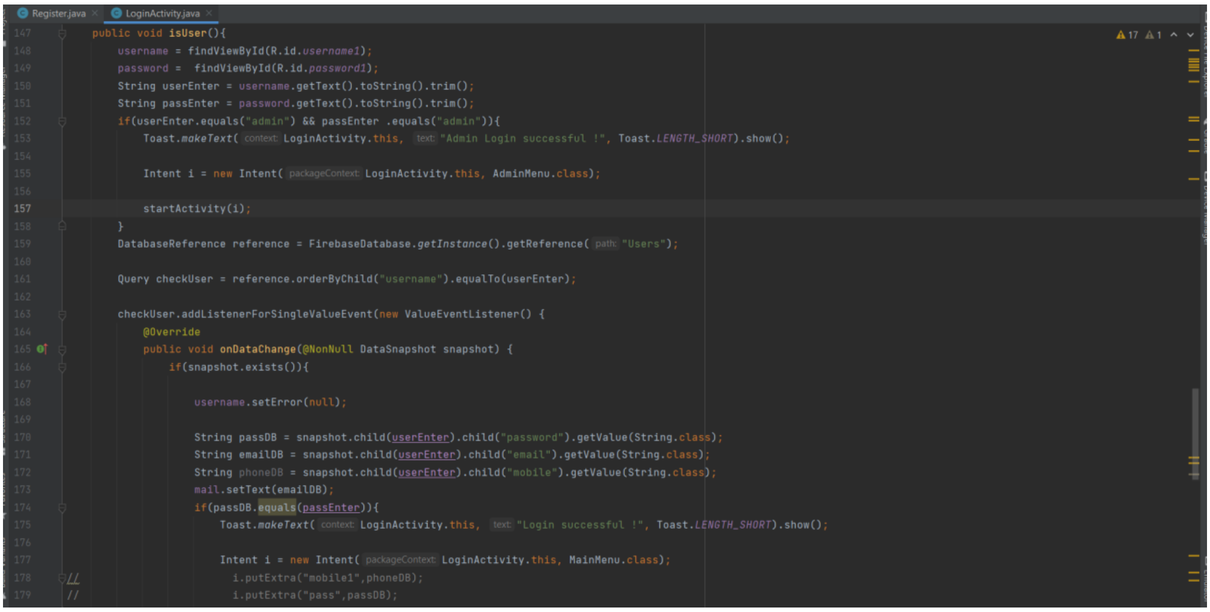
Register.java

```
            progressBar.setVisibility(View.VISIBLE);
            mAuth.createUserWithEmailAndPassword(mail,pass)
                    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult> task) {
                            if(task.isSuccessful()){
                                String check = " ";
                                if (b.isChecked()) {
                                    check = "borrower";
                                }
                                else if (inv.isChecked()){
                                    check = "investor";
                                }
                                String member = "Not Registered";

                                UpdateMobile user1 = new  UpdateMobile(user,pass,mail, mobile: null, ic: null,member,check, balance: "0", link_card_count: "0", loan_amount: "0", earn: "0");

                                FirebaseDatabase.getInstance().getReference( path: "Users")
                                        .child(user)
                                        .setValue(user1).addOnCompleteListener(new OnCompleteListener<Void>() {
                                    @Override
                                    public void onComplete(@NonNull Task<Void> task) {

                                        if (task.isSuccessful()){
                                            FirebaseAuth auth = FirebaseAuth.getInstance();
                                            FirebaseUser user = auth.getCurrentUser();

                                            user.sendEmailVerification()
                                                    .addOnCompleteListener(new OnCompleteListener<Void>() {
                                                        @Override
                                                        public void onComplete(@NonNull Task<Void> task) {
                                                            if (task.isSuccessful()) {
                                                                Toast.makeText( context: Register.this,  text: "Registered successfully. Please check your email for verification.", Toast.
```

Figure 4.2. 1 Register.java

Firebase is first initialized in order to connect Firebase authentication to the project for the register function. User have to create or register an account before perform to login activity. User required to fill in all fields of simple user details such as username, email address and password. User have to select the account type whether it is borrower account or investor account. The registerUser() method will be called to check the input validation after user click "register" button to register an account. If input is incorrect, it will clear the incorrect input column and focus it. After registered success, the user account details will be store in Firebase Realtime Database.

4.2.2 Login.java

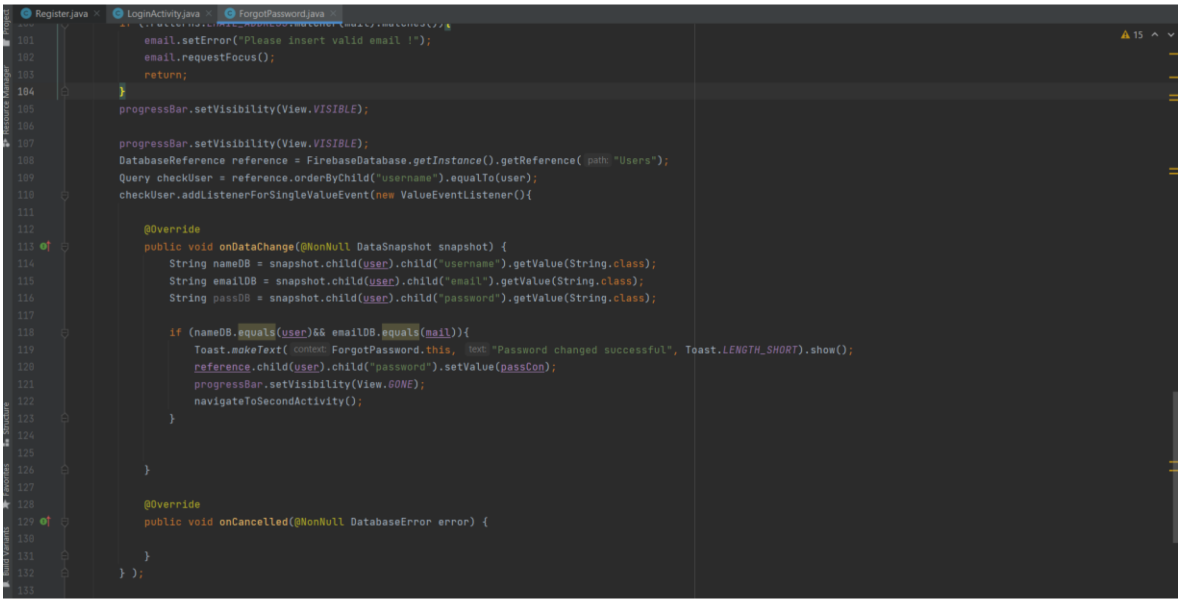


```
    public void isUser(){
        username = findViewById(R.id.username1);
        password =  findViewById(R.id.password1);
        String userEnter = username.getText().toString().trim();
        String passEnter = password.getText().toString().trim();
        if(userEnter.equals("admin") && passEnter .equals("admin")){
            Toast.makeText( context: LoginActivity.this,  text: "Admin Login successful !", Toast.LENGTH_SHORT).show();

            Intent i = new Intent( packageContext: LoginActivity.this, AdminMenu.class);

            startActivity(i);
        }
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");

        Query checkUser = reference.orderByChild("username").equalTo(userEnter);

        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()){

                    username.setError(null);

                    String passDB = snapshot.child(userEnter).child("password").getValue(String.class);
                    String emailDB = snapshot.child(userEnter).child("email").getValue(String.class);
                    String phoneDB = snapshot.child(userEnter).child("mobile").getValue(String.class);
                    mail.setText(emailDB);
                    if(passDB.equals(passEnter)){
                        Toast.makeText( context: LoginActivity.this,  text: "Login successful !", Toast.LENGTH_SHORT).show();

                        Intent i = new Intent( packageContext: LoginActivity.this, MainMenu.class);
//                        i.putExtra("mobile1",phoneDB);
//                        i.putExtra("pass",passDB);
```

Figure 4.2. 2 Login.java

If the user registered successful or have been registered before, user can enter username and password to log in to M Lending mobile application. The isUser() method will called to check the entered username and password match with Firebase records or not. If records not found, means user was not register yet. If data was found in Firebase, the method will check the account type of user such as borrower account, investor account or admin account. If it was a borrower account and Investor account, the user will be redirected to the MainMenu class. For admin account, it will redirect user to AdminMenu class.

4.2.3 ForgotPassword.java

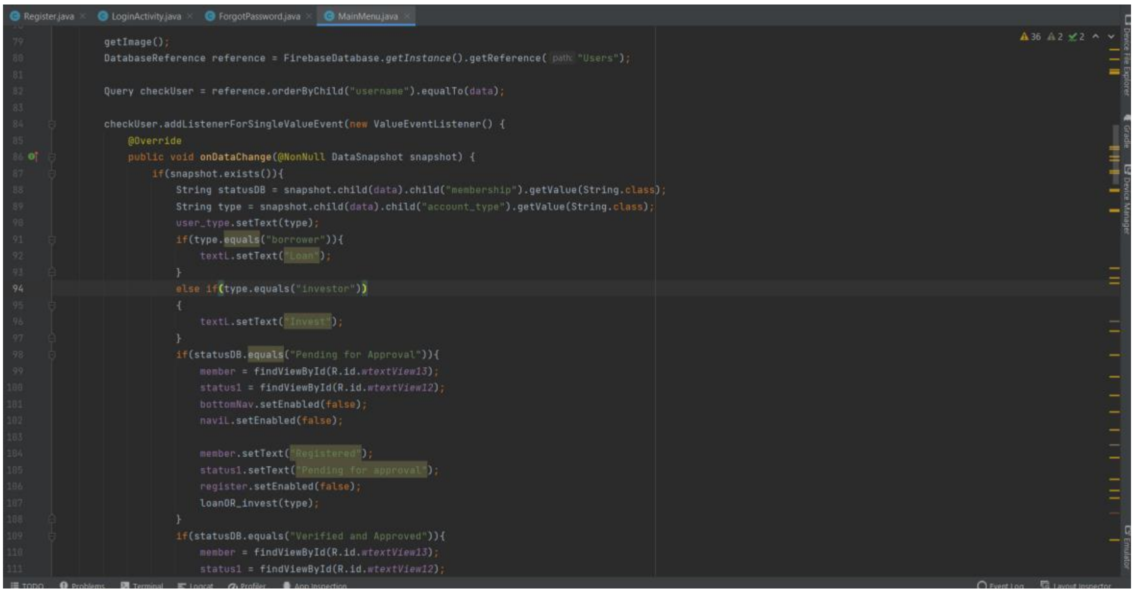


```java
        if (!Patterns.EMAIL_ADDRESS.matcher(mail).matches()){
            email.setError("Please insert valid email !");
            email.requestFocus();
            return;
        }
        progressBar.setVisibility(View.VISIBLE);

        progressBar.setVisibility(View.VISIBLE);
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");
        Query checkUser = reference.orderByChild("username").equalTo(user);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener(){

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String nameDB = snapshot.child(user).child("username").getValue(String.class);
                String emailDB = snapshot.child(user).child("email").getValue(String.class);
                String passDB = snapshot.child(user).child("password").getValue(String.class);

                if (nameDB.equals(user)&& emailDB.equals(mail)){
                    Toast.makeText(ForgotPassword.this, "Password changed successful", Toast.LENGTH_SHORT).show();
                    reference.child(user).child("password").setValue(passCon);
                    progressBar.setVisibility(View.GONE);
                    navigateToSecondActivity();
                }


            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        } );
```

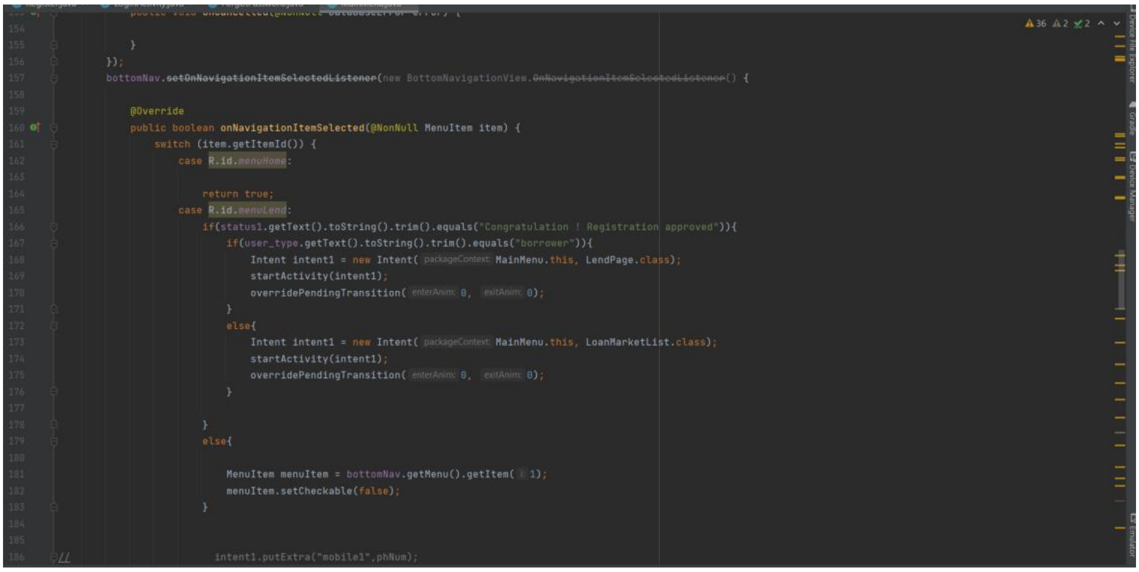
Figure 4.2. 3 ForgotPassword.java

The ForgotPassword class will handle the process of checking the pattern of entered username and email address is valid by comparing with the data stored in Firebase. If valid, the new password entered will be updated and store in Firebase.

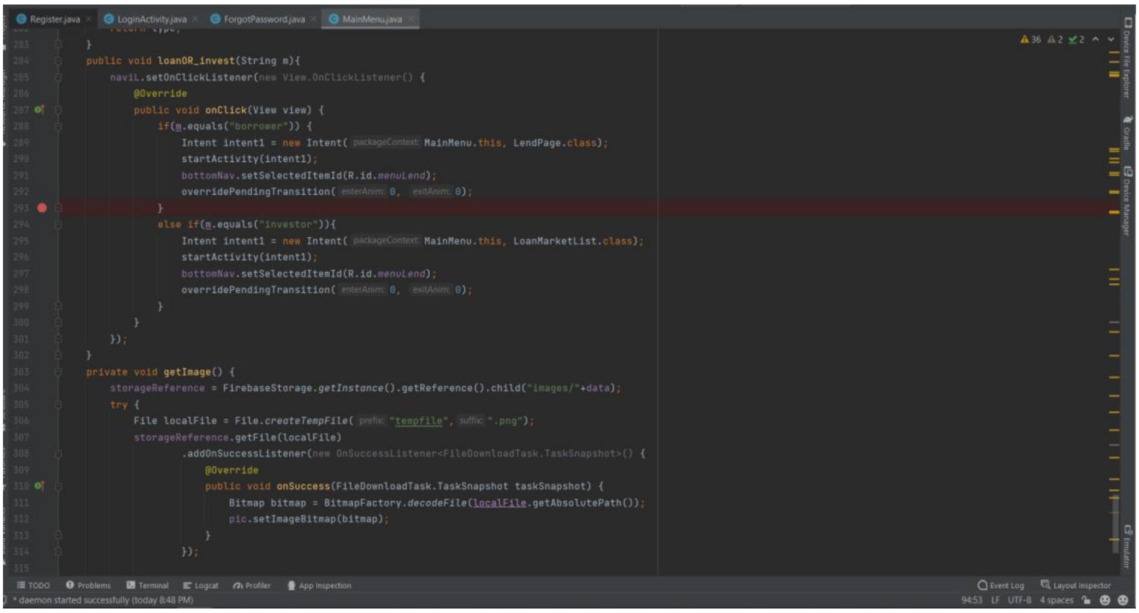
4.2.4 MainMenu.java



```java
        getImage();
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");

        Query checkUser = reference.orderByChild("username").equalTo(data);

        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()){
                    String statusDB = snapshot.child(data).child("membership").getValue(String.class);
                    String type = snapshot.child(data).child("account_type").getValue(String.class);
                    user_type.setText(type);
                    if(type.equals("borrower")){
                        textL.setText("Loan");
                    }
                    else if(type.equals("investor"))
                    {
                        textL.setText("Invest");
                    }
                    if(statusDB.equals("Pending for Approval")){
                        member = findViewById(R.id.wtextView13);
                        status1 = findViewById(R.id.wtextView12);
                        bottomNav.setEnabled(false);
                        naviL.setEnabled(false);

                        member.setText("Registered");
                        status1.setText("Pending for approval");
                        register.setEnabled(false);
                        loanOR_invest(type);
                    }
                    if(statusDB.equals("Verified and Approved")){
                        member = findViewById(R.id.wtextView13);
                        status1 = findViewById(R.id.wtextView12);
```



```java

            }
        });
        bottomNav.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.menuHome:

                        return true;
                    case R.id.menuLend:
                        if(status1.getText().toString().trim().equals("Congratulation ! Registration approved")){
                            if(user_type.getText().toString().trim().equals("borrower")){
                                Intent intent1 = new Intent( packageContext: MainMenu.this, LendPage.class);
                                startActivity(intent1);
                                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                            }
                            else{
                                Intent intent1 = new Intent( packageContext: MainMenu.this, LoanMarketList.class);
                                startActivity(intent1);
                                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                            }

                        }
                        else{

                            MenuItem menuItem = bottomNav.getMenu().getItem( i: 1);
                            menuItem.setCheckable(false);
                        }


//                       intent1.putExtra("mobile1",phNum);
```



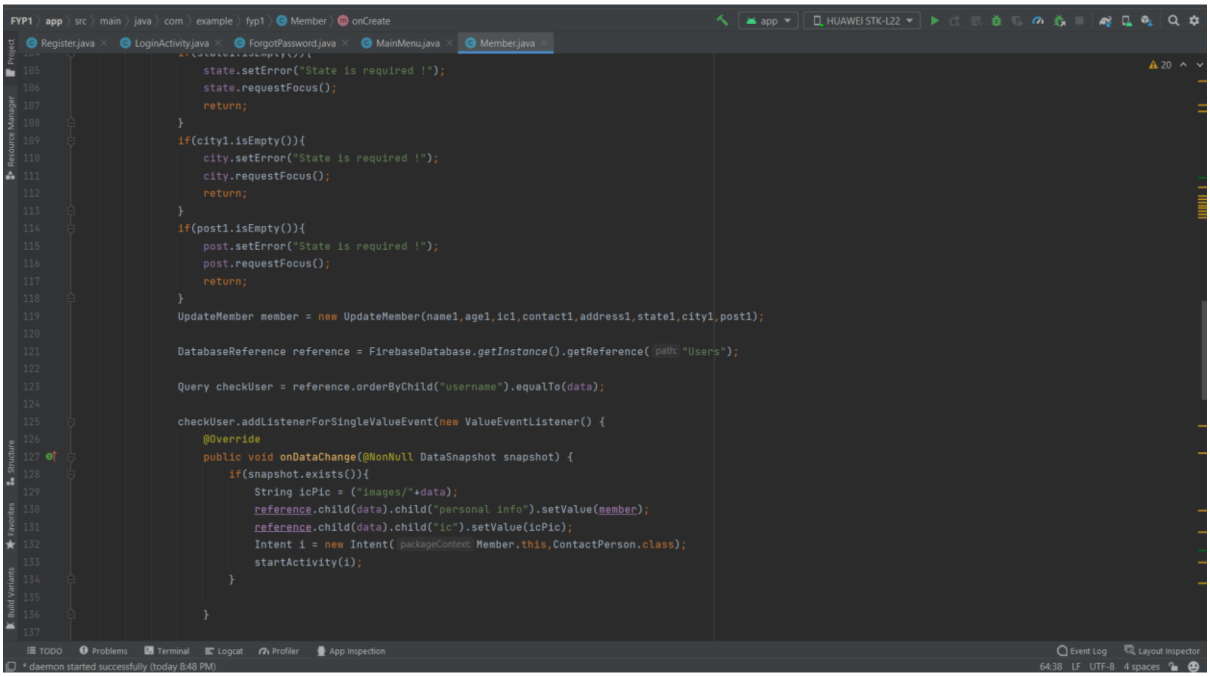
```java
    }
    public void loanOR_invest(String m){
        naviL.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(m.equals("borrower")) {
                    Intent intent1 = new Intent( packageContext: MainMenu.this, LendPage.class);
                    startActivity(intent1);
                    bottomNav.setSelectedItemId(R.id.menuLend);
                    overridePendingTransition( enterAnim: 0, exitAnim: 0);
                }
                else if(m.equals("investor")){
                    Intent intent1 = new Intent( packageContext: MainMenu.this, LoanMarketList.class);
                    startActivity(intent1);
                    bottomNav.setSelectedItemId(R.id.menuLend);
                    overridePendingTransition( enterAnim: 0, exitAnim: 0);
                }
            }
        });
    }
    private void getImage() {
        storageReference = FirebaseStorage.getInstance().getReference().child("images/"+data);
        try {
            File localFile = File.createTempFile( prefix: "tempfile", suffix: ".png");
            storageReference.getFile(localFile)
                    .addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
                            Bitmap bitmap = BitmapFactory.decodeFile(localFile.getAbsolutePath());
                            pic.setImageBitmap(bitmap);
                        }
                    });

```

* daemon started successfully (today 8:48 PM)

Figure 4.2. 4 MainMenu.java

In this class, it contained the bottom navigation bar of home, loan, transaction and profile. It also contained four buttons of loan, profile, transaction and chart. All these buttons and navigation bar will navigate user to related pages. At the bottom of the main menu page, it will display user’s membership status. This class will check whether the user is a registered member or not to give user access to loan page. Else, not registered user can only access to main menu page and profile page. The loanOR_Invest() will called to check user account type while getImage() will called to retrieve user uploaded profile picture and display in main menu page. If no profile picture is uploaded, the default profile picture will be displayed.

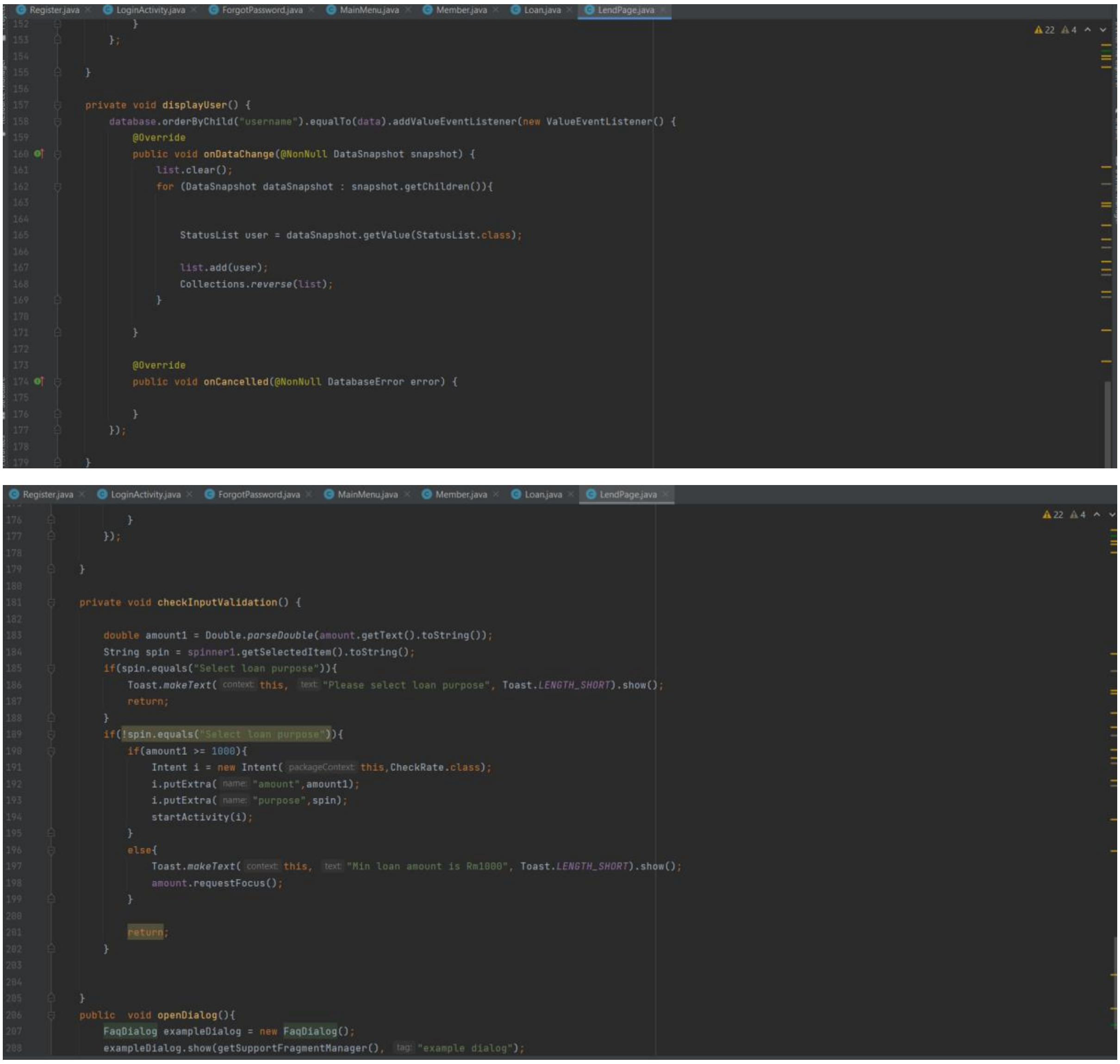
Member.java



```java
            state.setError("State is required !");
            state.requestFocus();
            return;
        }
        if(city1.isEmpty()){
            city.setError("State is required !");
            city.requestFocus();
            return;
        }
        if(post1.isEmpty()){
            post.setError("State is required !");
            post.requestFocus();
            return;
        }
        UpdateMember member = new UpdateMember(name1,age1,ic1,contact1,address1,state1,city1,post1);

        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");

        Query checkUser = reference.orderByChild("username").equalTo(data);

        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()){
                    String icPic = ("images/"+data);
                    reference.child(data).child("personal info").setValue(member);
                    reference.child(data).child("ic").setValue(icPic);
                    Intent i = new Intent(Member.this,ContactPerson.class);
                    startActivity(i);
                }

            }
```

This class is for membership registration. After the user register an account successful, user still need to provide a detailed personal detail and a contact person detail. This class will check the input validation and store all the data in Firebase.

4.2.5 LendPage.java



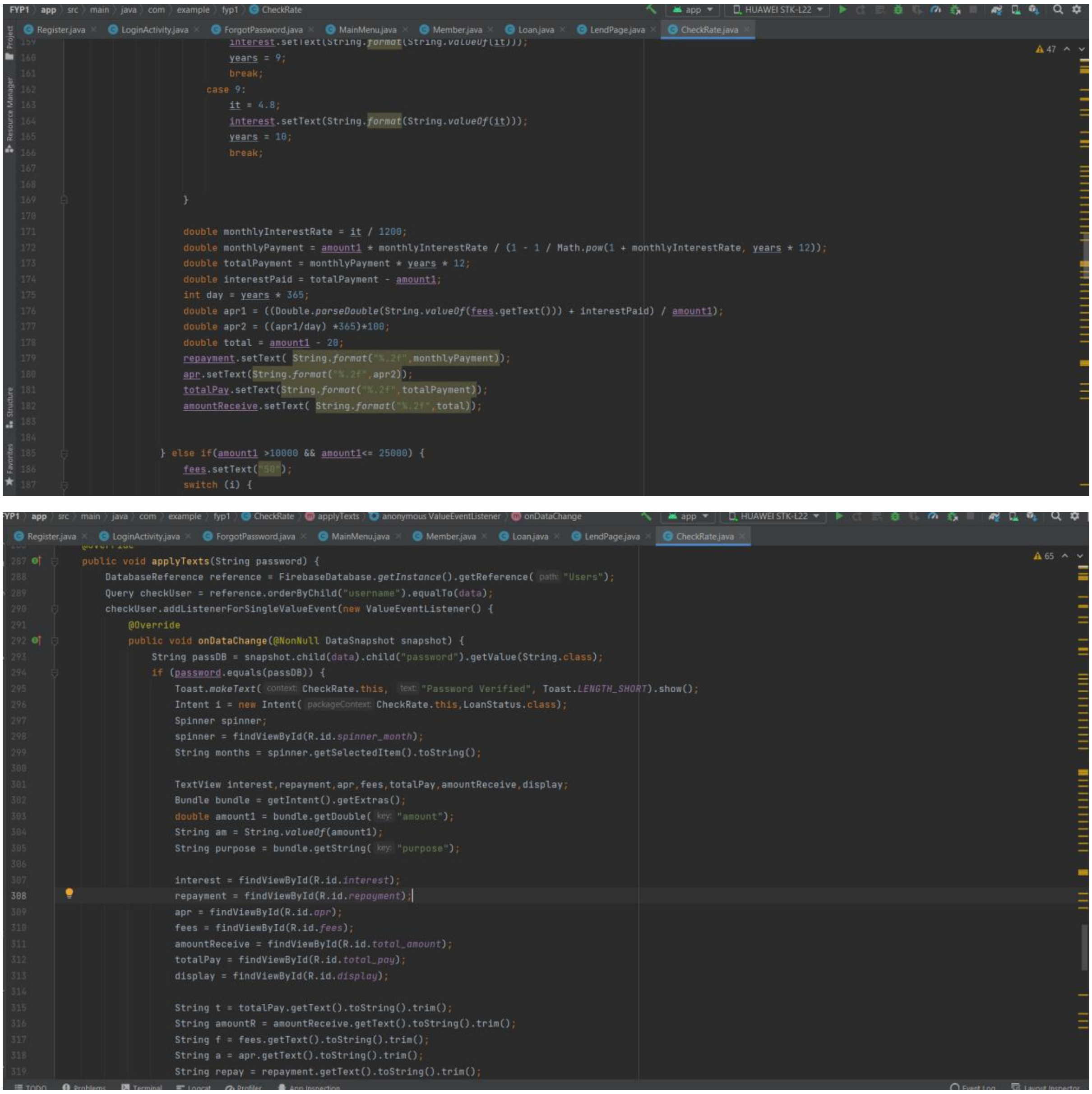
```java
            }
        };

    }

    private void displayUser() {
        database.orderByChild("username").equalTo(data).addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){


                    StatusList user = dataSnapshot.getValue(StatusList.class);

                    list.add(user);
                    Collections.reverse(list);
                }

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });

    }
```

```java
            }
        });

    }

    private void checkInputValidation() {

        double amount1 = Double.parseDouble(amount.getText().toString());
        String spin = spinner1.getSelectedItem().toString();
        if(spin.equals("Select loan purpose")){
            Toast.makeText( context: this,  text: "Please select loan purpose", Toast.LENGTH_SHORT).show();
            return;
        }
        if(!spin.equals("Select loan purpose")){
            if(amount1 >= 1000){
                Intent i = new Intent( packageContext: this,CheckRate.class);
                i.putExtra( name: "amount",amount1);
                i.putExtra( name: "purpose",spin);
                startActivity(i);
            }
            else{
                Toast.makeText( context: this,  text: "Min loan amount is Rm1000", Toast.LENGTH_SHORT).show();
                amount.requestFocus();
            }

            return;
        }


    }
    public  void openDialog(){
        FaqDialog exampleDialog = new FaqDialog();
        exampleDialog.show(getSupportFragmentManager(),  tag: "example dialog");
```

Figure 4.2. 5 LendPage.java

This class is the loan application page, It contain two sub modules which is apply loan and display loan status. displayUser() will retrieve the loan details which username is equal to current user and display in the list view. checkInputValidation() will called after user inserted the details to apply a loan. This method will check the input validation before redirect to next page.

4.2.6 CheckRate.java



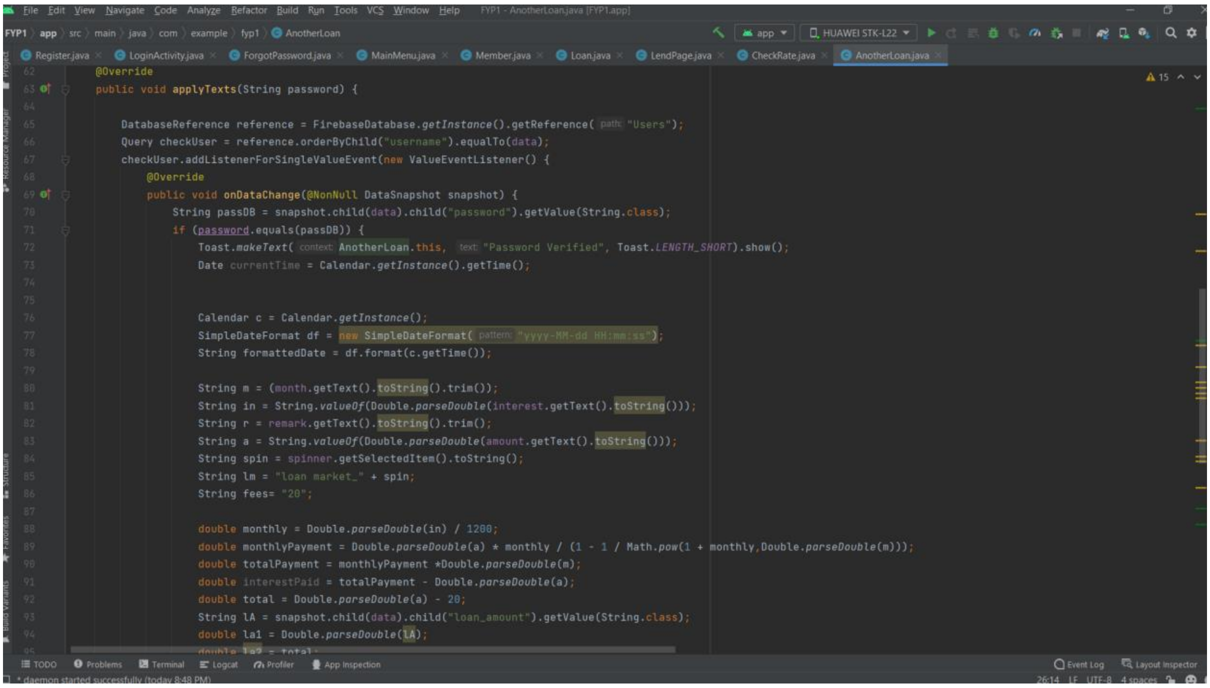
```
                        interest.setText(String.format(String.valueOf(it)));
                        years = 9;
                        break;
                    case 9:
                        it = 4.8;
                        interest.setText(String.format(String.valueOf(it)));
                        years = 10;
                        break;


                }

                double monthlyInterestRate = it / 1200;
                double monthlyPayment = amount1 * monthlyInterestRate / (1 - 1 / Math.pow(1 + monthlyInterestRate, years * 12));
                double totalPayment = monthlyPayment * years * 12;
                double interestPaid = totalPayment - amount1;
                int day = years * 365;
                double apr1 = ((Double.parseDouble(String.valueOf(fees.getText())) + interestPaid) / amount1);
                double apr2 = ((apr1/day) *365)*100;
                double total = amount1 - 20;
                repayment.setText( String.format("%.2f",monthlyPayment));
                apr.setText(String.format("%.2f",apr2));
                totalPay.setText(String.format("%.2f",totalPayment));
                amountReceive.setText( String.format("%.2f",total));


            } else if(amount1 >10000 && amount1<= 25000) {
                fees.setText("50");
                switch (i) {
```

```
    @Override
    public void applyTexts(String password) {
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
        Query checkUser = reference.orderByChild("username").equalTo(data);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String passDB = snapshot.child(data).child("password").getValue(String.class);
                if (password.equals(passDB)) {
                    Toast.makeText( context: CheckRate.this,  text: "Password Verified", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent( packageContext: CheckRate.this,LoanStatus.class);
                    Spinner spinner;
                    spinner = findViewById(R.id.spinner_month);
                    String months = spinner.getSelectedItem().toString();

                    TextView interest,repayment,apr,fees,totalPay,amountReceive,display;
                    Bundle bundle = getIntent().getExtras();
                    double amount1 = bundle.getDouble( key: "amount");
                    String am = String.valueOf(amount1);
                    String purpose = bundle.getString( key: "purpose");

                    interest = findViewById(R.id.interest);
                    repayment = findViewById(R.id.repayment);
                    apr = findViewById(R.id.apr);
                    fees = findViewById(R.id.fees);
                    amountReceive = findViewById(R.id.total_amount);
                    totalPay = findViewById(R.id.total_pay);
                    display = findViewById(R.id.display);

                    String t = totalPay.getText().toString().trim();
                    String amountR = amountReceive.getText().toString().trim();
                    String f = fees.getText().toString().trim();
                    String a = apr.getText().toString().trim();
                    String repay = repayment.getText().toString().trim();
```

Figure 4.2. 6 CheckRate.java

CheckRate class will be called after user inserted loan application details in LendPage.java. This class contained a loan calculator that calculate the interest rate, annual percentage rate, monthly repayment, total repayment. If user confirm to get the loan with the rate given, the openDialog() will be called and pop up a password verification prompt to verify user’s password in order to apply the loan. applyTexts() method will be called after user inserted the password and click “ok”. This method will check user entered password is valid or not by comparing with the actual password that stored in Firebase. If valid, the loan application will be processed and save in Firebase. The requested loan application will send to management of M Lending for further processing. If user does not satisfy with the given interest rate and repayment, user can apply loan through loan market, this will navigate user to AnotherLoan.java class.

4.2.7 AnotherLoan.java

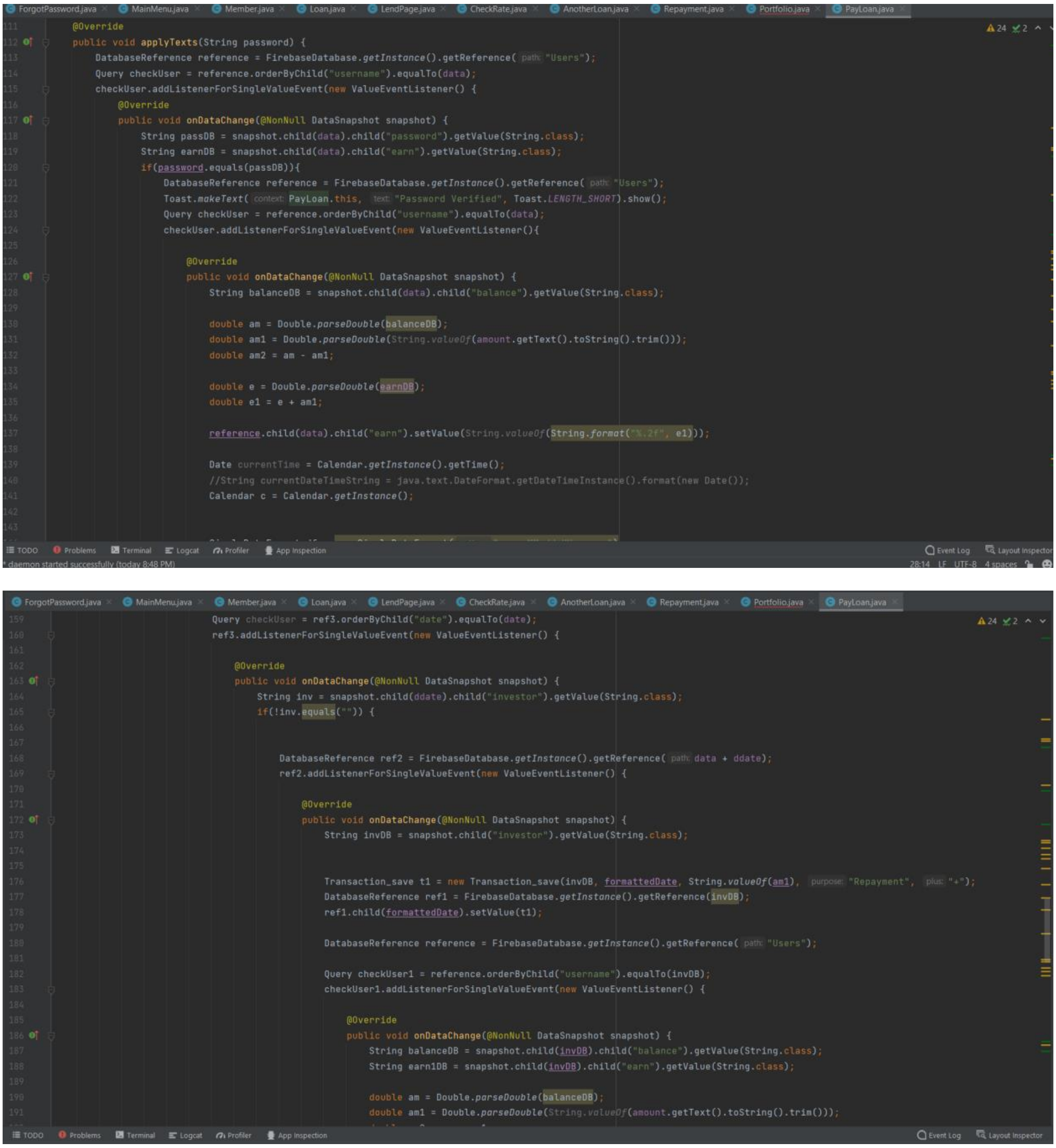


```
    @Override
    public void applyTexts(String password) {

        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");
        Query checkUser = reference.orderByChild("username").equalTo(data);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String passDB = snapshot.child(data).child("password").getValue(String.class);
                if (password.equals(passDB)) {
                    Toast.makeText(AnotherLoan.this, "Password Verified", Toast.LENGTH_SHORT).show();
                    Date currentTime = Calendar.getInstance().getTime();


                    Calendar c = Calendar.getInstance();
                    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                    String formattedDate = df.format(c.getTime());

                    String m = (month.getText().toString().trim());
                    String in = String.valueOf(Double.parseDouble(interest.getText().toString()));
                    String r = remark.getText().toString().trim();
                    String a = String.valueOf(Double.parseDouble(amount.getText().toString()));
                    String spin = spinner.getSelectedItem().toString();
                    String lm = "loan market_" + spin;
                    String fees= "20";

                    double monthly = Double.parseDouble(in) / 1200;
                    double monthlyPayment = Double.parseDouble(a) * monthly / (1 - 1 / Math.pow(1 + monthly,Double.parseDouble(m)));
                    double totalPayment = monthlyPayment *Double.parseDouble(m);
                    double interestPaid = totalPayment - Double.parseDouble(a);
                    double total = Double.parseDouble(a) - 20;
                    String lA = snapshot.child(data).child("loan_amount").getValue(String.class);
                    double la1 = Double.parseDouble(lA);
                    double la2 = total;
```

Figure 4.2. 7 AnotherLoan.java

This class is to handle the process of user to apply loan through loan market. User require to fill in loan application details and applyTexts() method will called to check user password validation. The loan application will send to management to check. If approved, the loan application will be posted in loan market for investor to invest.

4.2.8 PayLoan.java



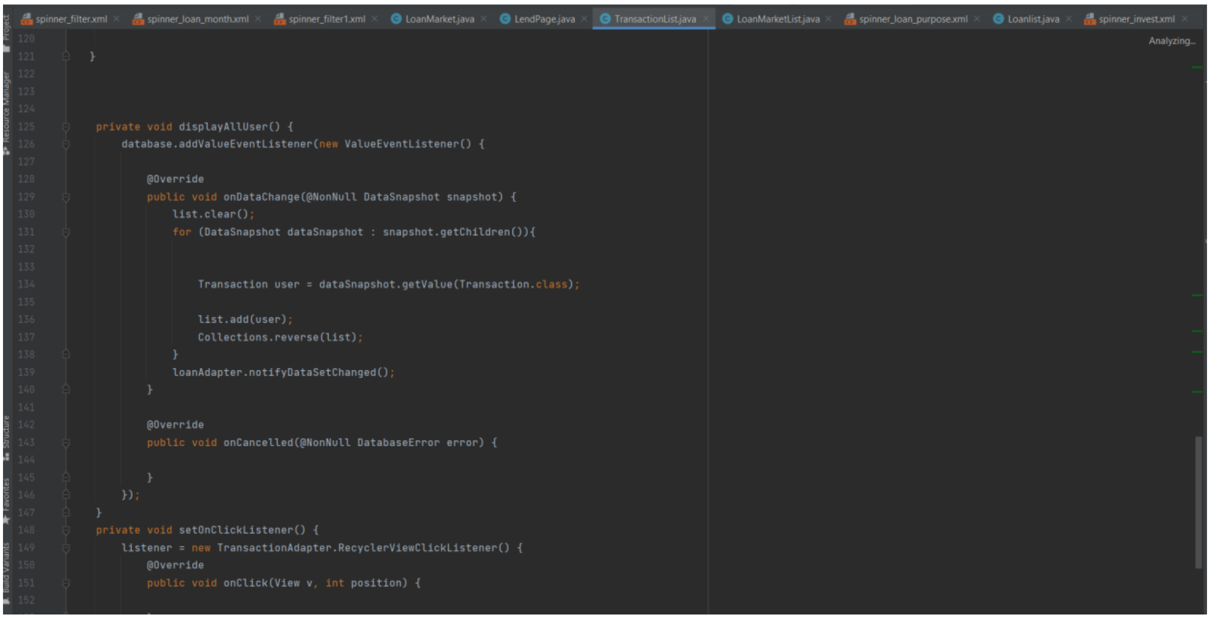
```java
    @Override
    public void applyTexts(String password) {
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
        Query checkUser = reference.orderByChild("username").equalTo(data);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String passDB = snapshot.child(data).child("password").getValue(String.class);
                String earnDB = snapshot.child(data).child("earn").getValue(String.class);
                if(password.equals(passDB)){
                    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
                    Toast.makeText( context: PayLoan.this,  text: "Password Verified", Toast.LENGTH_SHORT).show();
                    Query checkUser = reference.orderByChild("username").equalTo(data);
                    checkUser.addListenerForSingleValueEvent(new ValueEventListener(){

                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            String balanceDB = snapshot.child(data).child("balance").getValue(String.class);

                            double am = Double.parseDouble(balanceDB);
                            double am1 = Double.parseDouble(String.valueOf(amount.getText().toString().trim()));
                            double am2 = am - am1;

                            double e = Double.parseDouble(earnDB);
                            double e1 = e + am1;

                            reference.child(data).child("earn").setValue(String.valueOf(String.format("%.2f", e1)));

                            Date currentTime = Calendar.getInstance().getTime();
                            //String currentDateTimeString = java.text.DateFormat.getDateTimeInstance().format(new Date());
                            Calendar c = Calendar.getInstance();
```

```java
                                    Query checkUser = ref3.orderByChild("date").equalTo(date);
                                    ref3.addListenerForSingleValueEvent(new ValueEventListener() {

                                        @Override
                                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                                            String inv = snapshot.child(ddate).child("investor").getValue(String.class);
                                            if(!inv.equals("")) {


                                                DatabaseReference ref2 = FirebaseDatabase.getInstance().getReference( path: data + ddate);
                                                ref2.addListenerForSingleValueEvent(new ValueEventListener() {

                                                    @Override
                                                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                                                        String invDB = snapshot.child("investor").getValue(String.class);


                                                        Transaction_save t1 = new Transaction_save(invDB, formattedDate, String.valueOf(am1),  purpose: "Repayment",  plus: "+");
                                                        DatabaseReference ref1 = FirebaseDatabase.getInstance().getReference(invDB);
                                                        ref1.child(formattedDate).setValue(t1);

                                                        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");

                                                        Query checkUser1 = reference.orderByChild("username").equalTo(invDB);
                                                        checkUser1.addListenerForSingleValueEvent(new ValueEventListener() {

                                                            @Override
                                                            public void onDataChange(@NonNull DataSnapshot snapshot) {
                                                                String balanceDB = snapshot.child(invDB).child("balance").getValue(String.class);
                                                                String earn1DB = snapshot.child(invDB).child("earn").getValue(String.class);

                                                                double am = Double.parseDouble(balanceDB);
                                                                double am1 = Double.parseDouble(String.valueOf(amount.getText().toString().trim()));
```

Figure 4.2. 8 PayLoan.java

This class is used for loan repayment, it will display all the monthly repayment of related loan. The repayment details will be retrieved from Firebase. applyTexts() will called to verify user password and check user account balance. This method will check whether user account balance is sufficient or not to pay the monthly repayment.

4.2.9 TransactionList.java

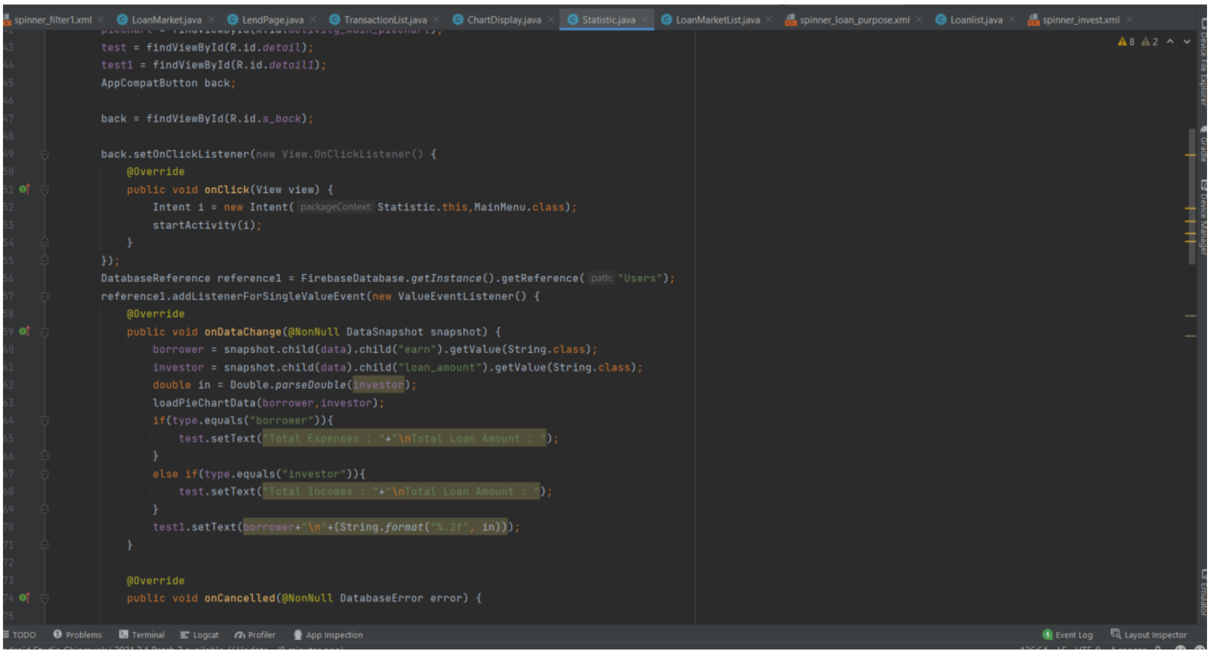


```java
    }



    private void displayAllUser() {
        database.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){


                    Transaction user = dataSnapshot.getValue(Transaction.class);

                    list.add(user);
                    Collections.reverse(list);
                }
                loanAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }
    private void setOnClickListener() {
        listener = new TransactionAdapter.RecyclerViewClickListener() {
            @Override
            public void onClick(View v, int position) {
```

Figure 4.2. 9 TransactionList.java

This class contains displayAllUser() method that will be called to retrieve all transaction records of current user. The transaction records including loan fund, repayment, investment, top up and withdraw of account balance.

4.2.10 Statistics.java

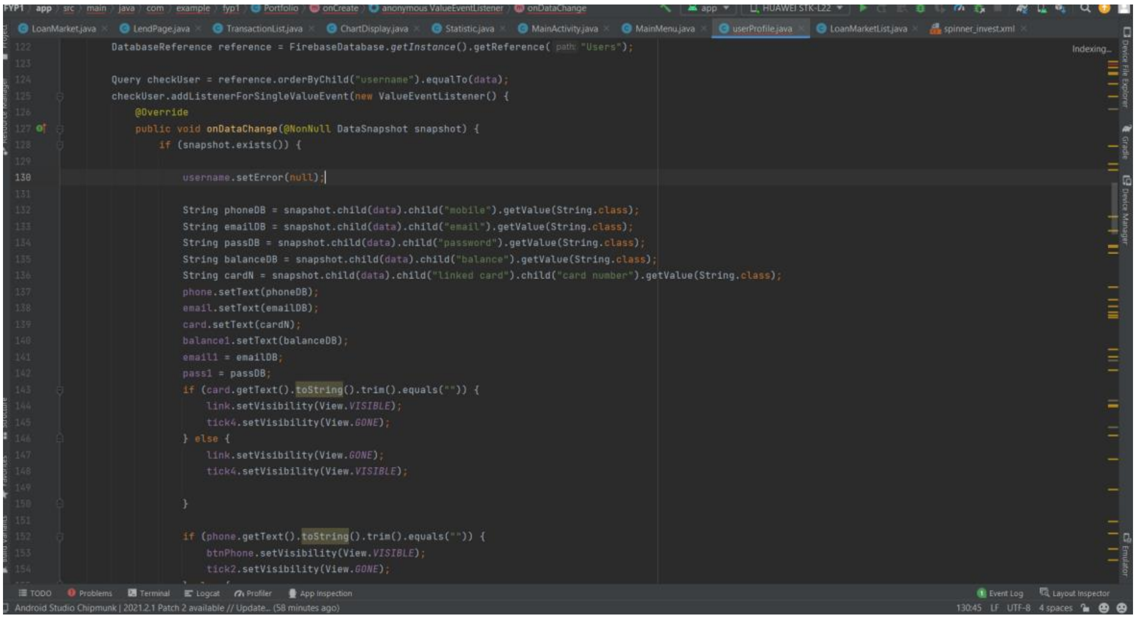


```java
        test = findViewById(R.id.detail);
        test1 = findViewById(R.id.detail1);
        AppCompatButton back;

        back = findViewById(R.id.s_back);

        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: Statistic.this,MainMenu.class);
                startActivity(i);
            }
        });
        DatabaseReference reference1 = FirebaseDatabase.getInstance().getReference( path: "Users");
        reference1.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                borrower = snapshot.child(data).child("earn").getValue(String.class);
                investor = snapshot.child(data).child("loan_amount").getValue(String.class);
                double in = Double.parseDouble(investor);
                loadPieChartData(borrower,investor);
                if(type.equals("borrower")){
                    test.setText("Total Expenses : "+"\nTotal Loan Amount : ");
                }
                else if(type.equals("investor")){
                    test.setText("Total Incomes : "+"\nTotal Loan Amount : ");
                }
                test1.setText(borrower+"\n"+(String.format("%.2f", in)));
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
```

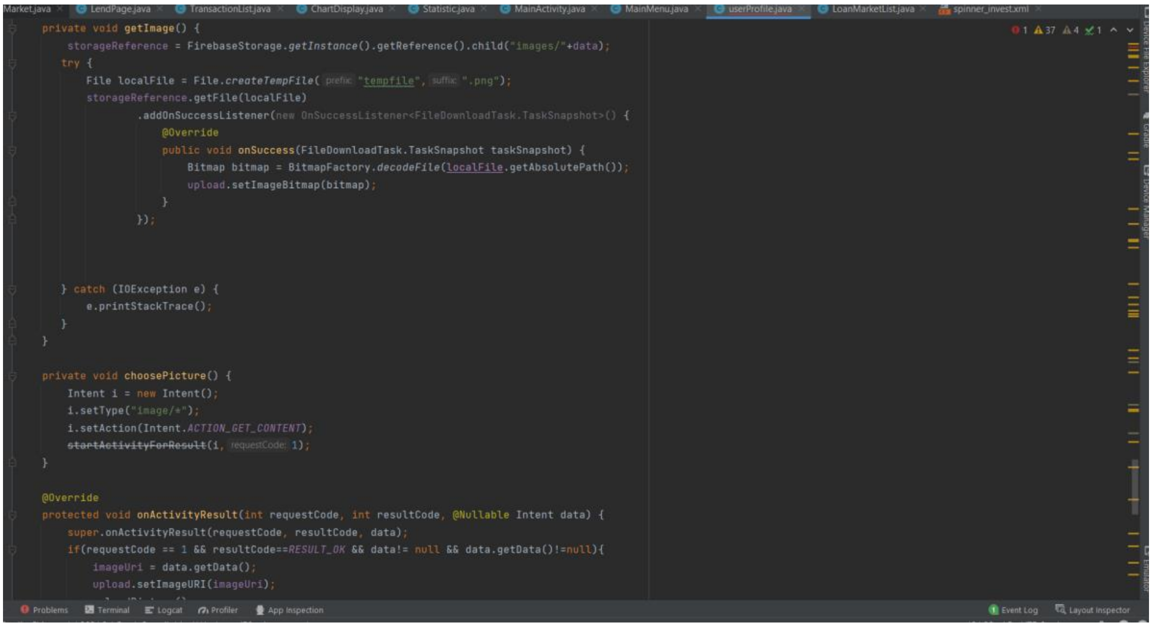
Figure 4.2. 10 Statistics.java

The statistics class retrieve all current user applied loan funds and expenses like loan repayment. This class will get all these data and display it in chart.

4.2.11 UserProfile.java



```
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");

        Query checkUser = reference.orderByChild("username").equalTo(data);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {

                    username.setError(null);

                    String phoneDB = snapshot.child(data).child("mobile").getValue(String.class);
                    String emailDB = snapshot.child(data).child("email").getValue(String.class);
                    String passDB = snapshot.child(data).child("password").getValue(String.class);
                    String balanceDB = snapshot.child(data).child("balance").getValue(String.class);
                    String cardN = snapshot.child(data).child("linked card").child("card number").getValue(String.class);
                    phone.setText(phoneDB);
                    email.setText(emailDB);
                    card.setText(cardN);
                    balance1.setText(balanceDB);
                    email1 = emailDB;
                    pass1 = passDB;
                    if (card.getText().toString().trim().equals("")) {
                        link.setVisibility(View.VISIBLE);
                        tick4.setVisibility(View.GONE);
                    } else {
                        link.setVisibility(View.GONE);
                        tick4.setVisibility(View.VISIBLE);

                    }

                    if (phone.getText().toString().trim().equals("")) {
                        btnPhone.setVisibility(View.VISIBLE);
                        tick2.setVisibility(View.GONE);
```

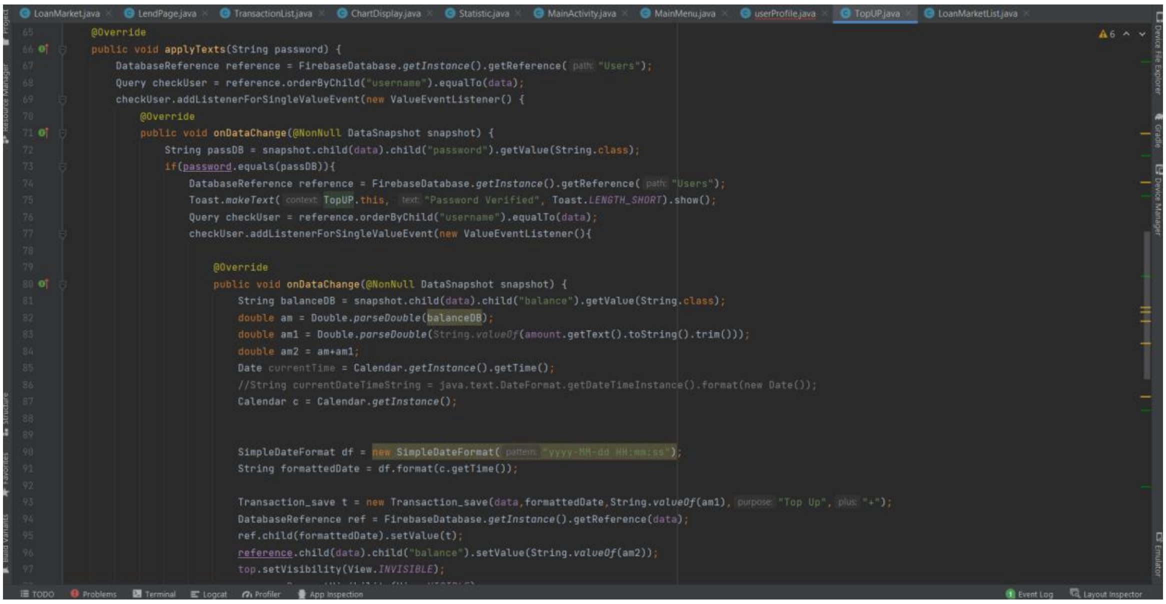


```
    private void getImage() {
        storageReference = FirebaseStorage.getInstance().getReference().child("images/"+data);
        try {
            File localFile = File.createTempFile( prefix: "tempfile", suffix: ".png");
            storageReference.getFile(localFile)
                    .addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
                            Bitmap bitmap = BitmapFactory.decodeFile(localFile.getAbsolutePath());
                            upload.setImageBitmap(bitmap);
                        }
                    });



        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void choosePicture() {
        Intent i = new Intent();
        i.setType("image/*");
        i.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(i, requestCode: 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode == 1 && resultCode==RESULT_OK && data!= null && data.getData()!=null){
            imageUri = data.getData();
            upload.setImageURI(imageUri);
```

Figure 4.2. 11 UserProfile.java

The userProfile class contains 3 sub modules which are display user details, account balance and upload profile picture. The system will retrieve current user's username, email address, phone number, account balance and display in this page. choosePicture() method will handle the process for user to choose image from their mobile device storage or cloud storage. The getImage() method will called to retrieve uploaded profile picture image from Firebase storage and display it in the ImageView of profile picture. If there is no image uploaded, the profile picture will be set to default profile image.

4.2.12 TopUp.java

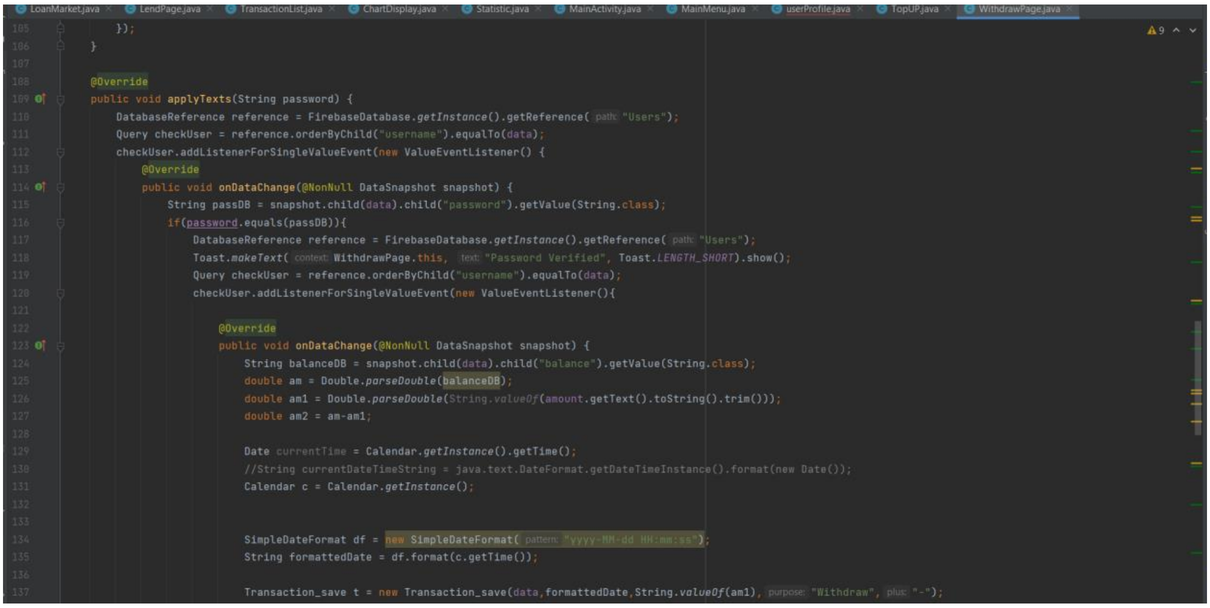


```java
@Override
public void applyTexts(String password) {
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
    Query checkUser = reference.orderByChild("username").equalTo(data);
    checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String passDB = snapshot.child(data).child("password").getValue(String.class);
            if(password.equals(passDB)){
                DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
                Toast.makeText( context: TopUP.this,  text: "Password Verified", Toast.LENGTH_SHORT).show();
                Query checkUser = reference.orderByChild("username").equalTo(data);
                checkUser.addListenerForSingleValueEvent(new ValueEventListener(){

                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        String balanceDB = snapshot.child(data).child("balance").getValue(String.class);
                        double am = Double.parseDouble(balanceDB);
                        double am1 = Double.parseDouble(String.valueOf(amount.getText().toString().trim()));
                        double am2 = am+am1;
                        Date currentTime = Calendar.getInstance().getTime();
                        //String currentDateTimeString = java.text.DateFormat.getDateTimeInstance().format(new Date());
                        Calendar c = Calendar.getInstance();


                        SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
                        String formattedDate = df.format(c.getTime());

                        Transaction_save t = new Transaction_save(data,formattedDate,String.valueOf(am1), purpose: "Top Up", plus: "+");
                        DatabaseReference ref = FirebaseDatabase.getInstance().getReference(data);
                        ref.child(formattedDate).setValue(t);
                        reference.child(data).child("balance").setValue(String.valueOf(am2));
                        top.setVisibility(View.INVISIBLE);
```

Figure 4.2. 12 TopUp.java

The system will be using the applyTexts() method to pop out a password verification prompt for user to insert their account password for further verification to confirm the top up request. After user entered the password, the system will retrieve user account balance from Firebase and add it with the top up amount. The latest account balance will be updated and save to Firebase. The top up record will be updated in user transaction records.

4.2.13 WithdrawPage.java

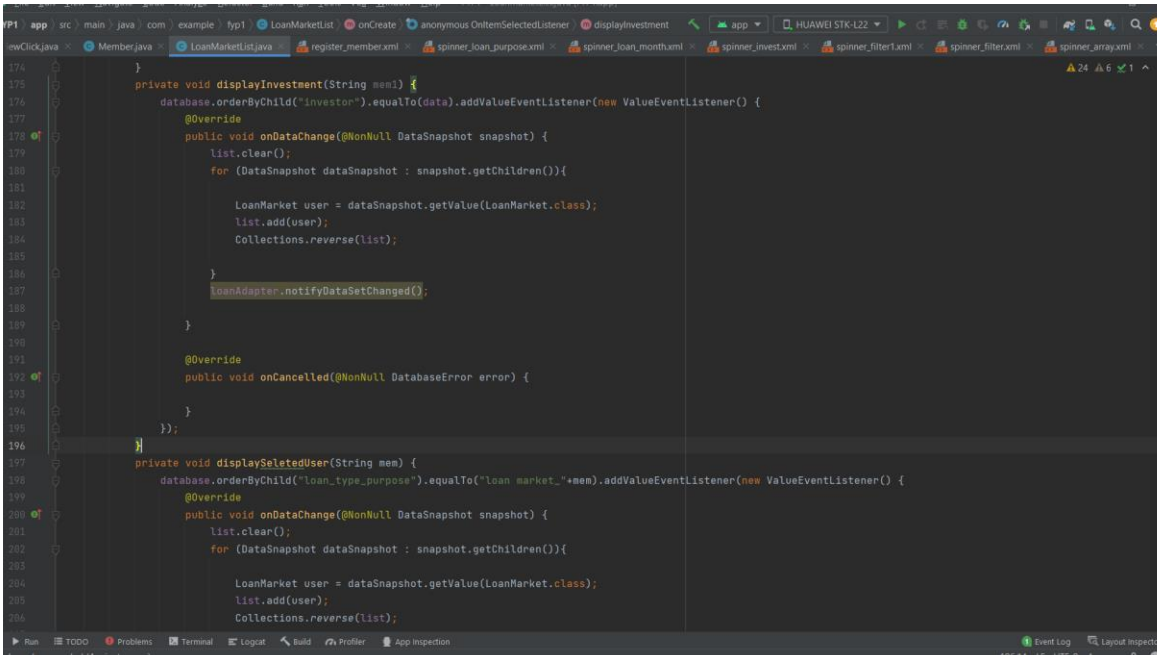


```java
        });
    }

    @Override
    public void applyTexts(String password) {
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
        Query checkUser = reference.orderByChild("username").equalTo(data);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String passDB = snapshot.child(data).child("password").getValue(String.class);
                if(password.equals(passDB)){
                    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
                    Toast.makeText( context: WithdrawPage.this,  text: "Password Verified", Toast.LENGTH_SHORT).show();
                    Query checkUser = reference.orderByChild("username").equalTo(data);
                    checkUser.addListenerForSingleValueEvent(new ValueEventListener(){

                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            String balanceDB = snapshot.child(data).child("balance").getValue(String.class);
                            double am = Double.parseDouble(balanceDB);
                            double am1 = Double.parseDouble(String.valueOf(amount.getText().toString().trim()));
                            double am2 = am-am1;

                            Date currentTime = Calendar.getInstance().getTime();
                            //String currentDateTimeString = java.text.DateFormat.getDateTimeInstance().format(new Date());
                            Calendar c = Calendar.getInstance();


                            SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
                            String formattedDate = df.format(c.getTime());

                            Transaction_save t = new Transaction_save(data,formattedDate,String.valueOf(am1), purpose: "Withdraw", plus: "-");
```

Figure 4.2. 13 WithdrawPage.java

In this class, user can withdraw loan funds from account balance. The system uses the applyTests() to verify user's password and check user's account balance whether is sufficient to withdraw or not. The system will get the withdraw amount inserted by user compare with user's account balance amount. If withdraw amount is greater than account balance amount, system will prompt "insufficient balance" message. Else if withdraw successful, the latest account balance will be updated and save in Firebase.

4.2.14 LoanMarketList.java

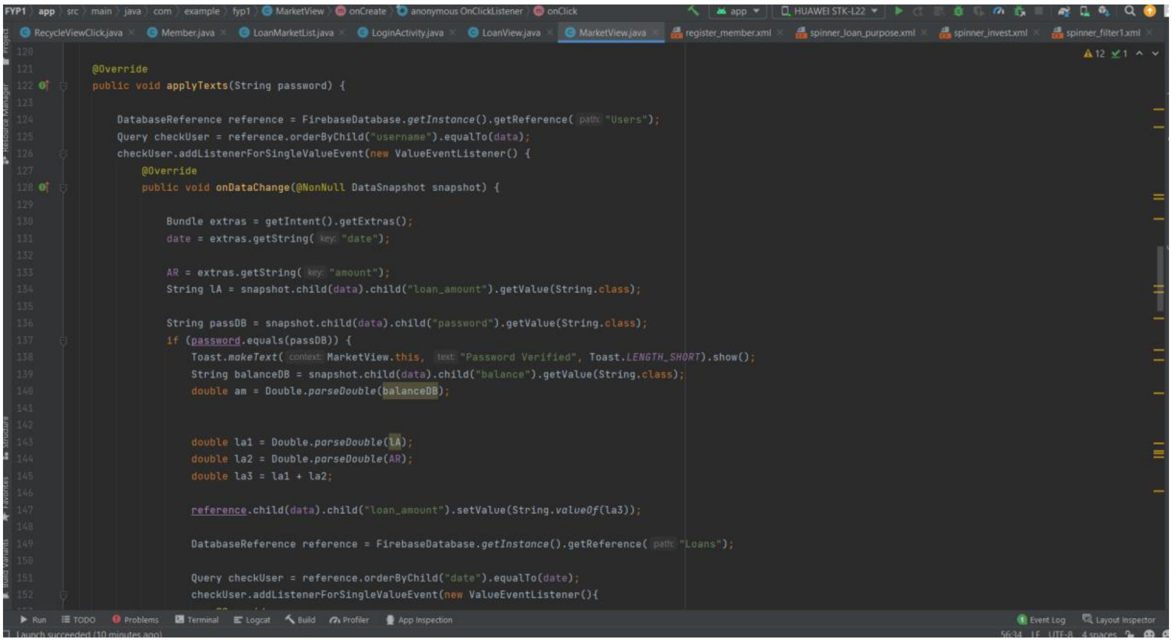


```java
    }
    private void displayInvestment(String mem1) {
        database.orderByChild("investor").equalTo(data).addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){

                    LoanMarket user = dataSnapshot.getValue(LoanMarket.class);
                    list.add(user);
                    Collections.reverse(list);

                }
                loanAdapter.notifyDataSetChanged();

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }
    private void displaySeletedUser(String mem) {
        database.orderByChild("loan_type_purpose").equalTo("loan market_"+mem).addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){

                    LoanMarket user = dataSnapshot.getValue(LoanMarket.class);
                    list.add(user);
                    Collections.reverse(list);
```

Figure 4.2. 14 LoanMarketList.java

This class is only available for investor to select suitable loan application for investment purpose. The loan that user apply in AnotherLoan.java class will display in loan market list. The system will retrieve all these loans from Firebase and filter it by using displaySelectedUser(). This method is to sort the loan application with loan purpose. The displayInvestment() method is called to display invested loan application of current user.

4.2.15 MarketView.java

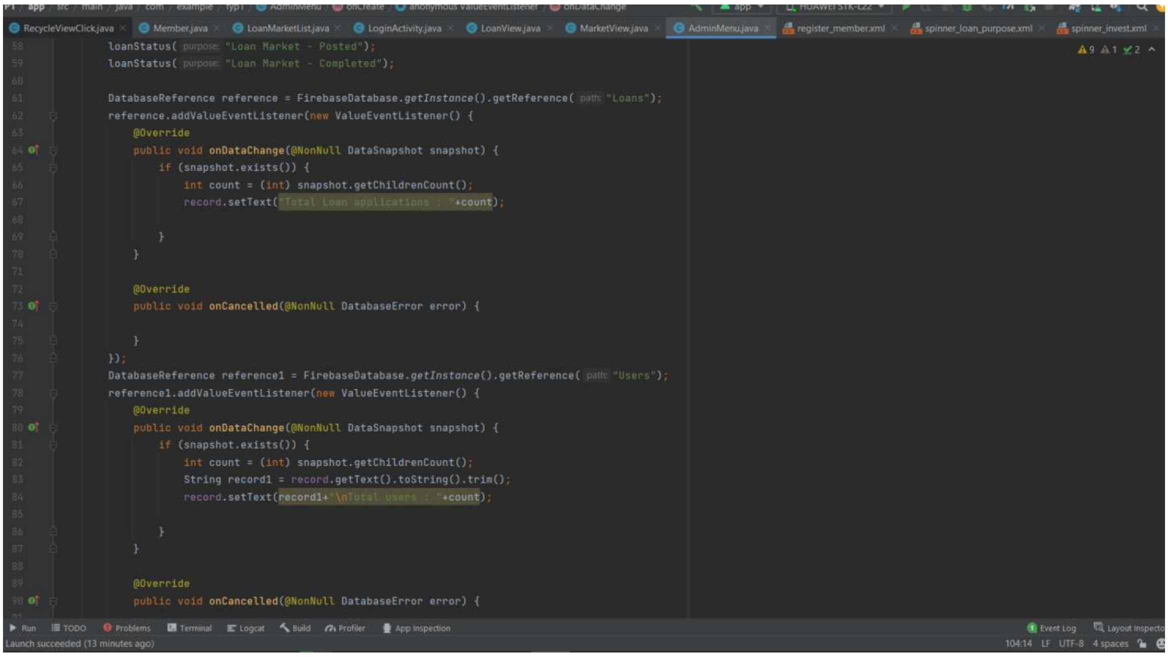


```java
@Override
public void applyTexts(String password) {

    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
    Query checkUser = reference.orderByChild("username").equalTo(data);
    checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

            Bundle extras = getIntent().getExtras();
            date = extras.getString( key: "date");

            AR = extras.getString( key: "amount");
            String lA = snapshot.child(data).child("loan_amount").getValue(String.class);

            String passDB = snapshot.child(data).child("password").getValue(String.class);
            if (password.equals(passDB)) {
                Toast.makeText( context: MarketView.this,  text: "Password Verified", Toast.LENGTH_SHORT).show();
                String balanceDB = snapshot.child(data).child("balance").getValue(String.class);
                double am = Double.parseDouble(balanceDB);


                double la1 = Double.parseDouble(lA);
                double la2 = Double.parseDouble(AR);
                double la3 = la1 + la2;

                reference.child(data).child("loan_amount").setValue(String.valueOf(la3));

                DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Loans");

                Query checkUser = reference.orderByChild("date").equalTo(date);
                checkUser.addListenerForSingleValueEvent(new ValueEventListener(){
```

Figure 4.2. 15 MarketView.java

The system will redirect user to this class after user have selected the investment loan application. This class will display the details of the loan application. User can contact applicant by clicking the “contact” button to get more information about the loan application. The applyTexts() method will be called if user confirm to invest on this loan, it will verify user password and check user account balance.

4.2.16 AdminMenu.java

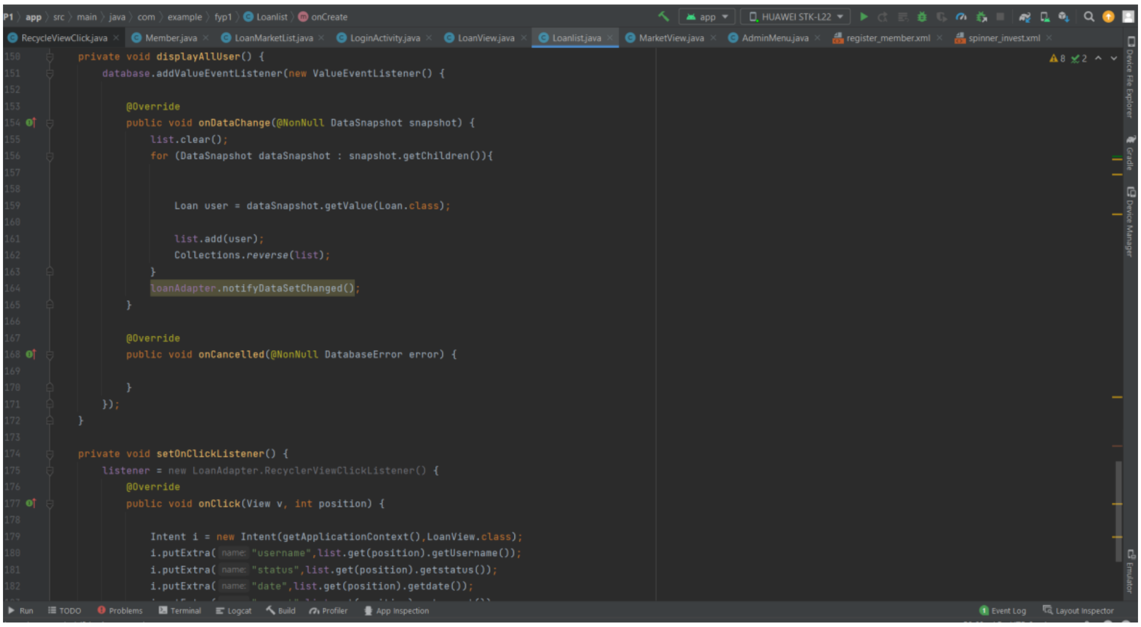


```
        loanStatus( purpose: "Loan Market - Posted");
        loanStatus( purpose: "Loan Market - Completed");

        DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Loans");
        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    int count = (int) snapshot.getChildrenCount();
                    record.setText("Total Loan applications : "+count);

                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
        DatabaseReference reference1 = FirebaseDatabase.getInstance().getReference( path: "Users");
        reference1.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    int count = (int) snapshot.getChildrenCount();
                    String record1 = record.getText().toString().trim();
                    record.setText(record1+"\nTotal users : "+count);

                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
```

Figure 4.2. 16 AdminMenu.java

This class is available for admin, the management of M Lending mobile application only. The administrator uses this module to manage to handle and manage M lending. This admin account contained four main module which are loan list, user list, chart and blacklist. Loan list will retrieve all the loan application data from Firebase and display it in a list while user list will retrieve all user data. The system will display the total loan application, total user and total loan amount information.

4.2.17 Loanlist.java

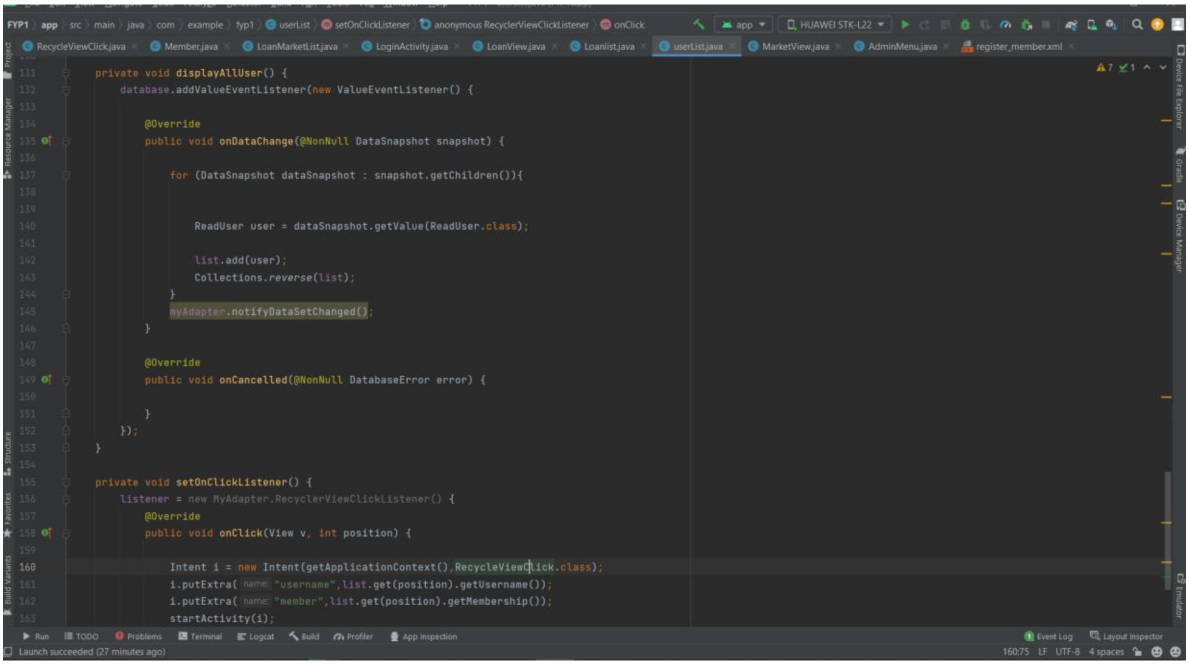


```java
    private void displayAllUser() {
        database.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){


                    Loan user = dataSnapshot.getValue(Loan.class);

                    list.add(user);
                    Collections.reverse(list);
                }
                loanAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }

    private void setOnClickListener() {
        listener = new LoanAdapter.RecyclerViewClickListener() {
            @Override
            public void onClick(View v, int position) {

                Intent i = new Intent(getApplicationContext(),LoanView.class);
                i.putExtra( name: "username",list.get(position).getUsername());
                i.putExtra( name: "status",list.get(position).getstatus());
                i.putExtra( name: "date",list.get(position).getdate());
```

Figure 4.2. 17 Loanlist.java

Loan list class retrieved all loan data from Firebase and display it in a list for administrator to have better management on loan applications. The administrator can update, approve or reject the loan by using this class. The setOnClickListener() method will display more detailed loan detail in LoanView class.

4.2.18 userList.java



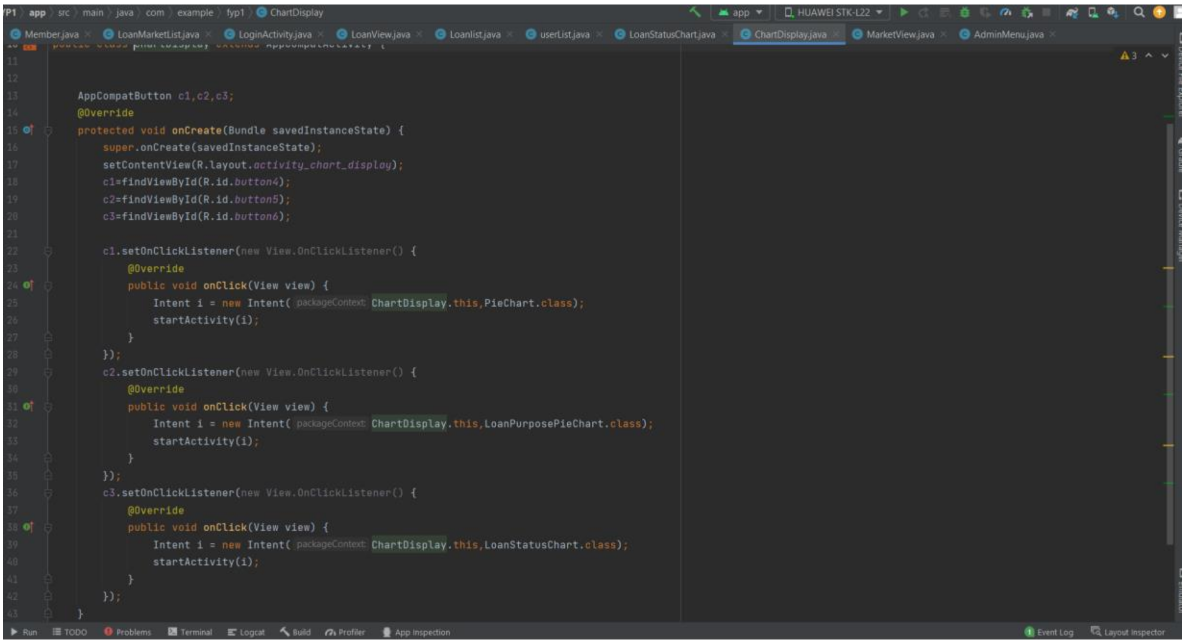
```
    private void displayAllUser() {
        database.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                for (DataSnapshot dataSnapshot : snapshot.getChildren()){


                    ReadUser user = dataSnapshot.getValue(ReadUser.class);

                    list.add(user);
                    Collections.reverse(list);
                }
                myAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }

    private void setOnClickListener() {
        listener = new MyAdapter.RecyclerViewClickListener() {
            @Override
            public void onClick(View v, int position) {

                Intent i = new Intent(getApplicationContext(),RecycleViewClick.class);
                i.putExtra( name: "username",list.get(position).getUsername());
                i.putExtra( name: "member",list.get(position).getMembership());
                startActivity(i);
```

Launch succeeded (27 minutes ago)

Figure 4.2. 18 userList.java

This is a class that similar with Loanlist.java but this class will retrieve all user information from Firebase. Administrator can manage user by using this class. Administrator able to approve or reject for the membership registration request. All these changes will update and save in Firebase.

4.2.19 ChartDisplay.java

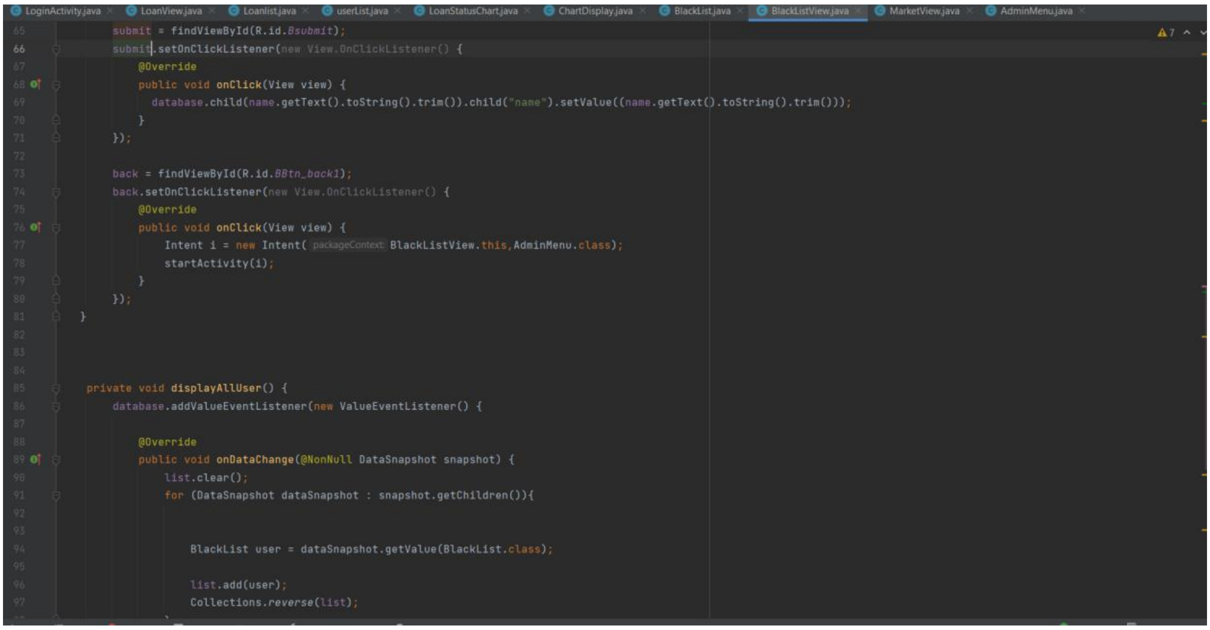


```
    AppCompatButton c1,c2,c3;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_chart_display);
        c1=findViewById(R.id.button4);
        c2=findViewById(R.id.button5);
        c3=findViewById(R.id.button6);

        c1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: ChartDisplay.this,PieChart.class);
                startActivity(i);
            }
        });
        c2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: ChartDisplay.this,LoanPurposePieChart.class);
                startActivity(i);
            }
        });
        c3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: ChartDisplay.this,LoanStatusChart.class);
                startActivity(i);
            }
        });
    }
```

Figure 4.2. 19 ChartDisplay.java

There are three chart display in this Chart Display class which are the chart of investor and borrower, chart of loan purpose and chart of loan status. Chart of investor and borrower show the amount of total investor and borrower, Chart of loan purpose will categories the loan application by loan purpose. While chart of loan status displays all the loan status by categories them with status. For example, it will display the total of complete status of loan application.

4.2.20 BlackListView.java



```java
        submit = findViewById(R.id.Bsubmit);
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
              database.child(name.getText().toString().trim()).child("name").setValue((name.getText().toString().trim()));
            }
        });

        back = findViewById(R.id.BBtn_back1);
        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: BlackListView.this,AdminMenu.class);
                startActivity(i);
            }
        });
    }



    private void displayAllUser() {
        database.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                list.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){


                    BlackList user = dataSnapshot.getValue(BlackList.class);

                    list.add(user);
                    Collections.reverse(list);
```

Figure 4.2. 20 BlackListView.java

This class retrieve all the blacklisted user and list it out in a list view. The administrator can insert new user to the blacklist. The system will update and save the records in Firebase. The blacklist will be checked for the review and approval of loans.

5 CHAPTER 5 Implementation and Testing

5.1 Hardware Setup

The hardware involved in this project is computer and android mobile device. A computer issued for the development of financial technology P2P lending mobile app and firebase access. An android mobile device is used for testing and deploying this P2P lending mobile application.

Description	Specifications
Model	Asus Vivobook A512
Processor	Intel Core i5-8265U
Operating System	Windows 10 Home Single Language
Graphic	NVIDIA GeForce MX250
Memory	8GB
Storage	512GB SSD

Table 5.1. 1 Specification of Laptop

5.2 Software Setup

Software Requirement for Android Studio

Description	Specifications
Operating System	Android 8.0 and above
Processor	2nd generation Intel Core or higher, or an AMD CPU with support for a Windows Hypervisor, and the X86 64 CPU architecture
Display Resolution	1280 x 800 and above
Memory	6GB
Storage	128GB

Table 5.1. 2 Requirement of Android Studio

Description	Specifications
Operating System	Android 8.0 and above
Screen resolution	1080 x 2340 and above
Memory	4GB and above
Storage	2GB and above

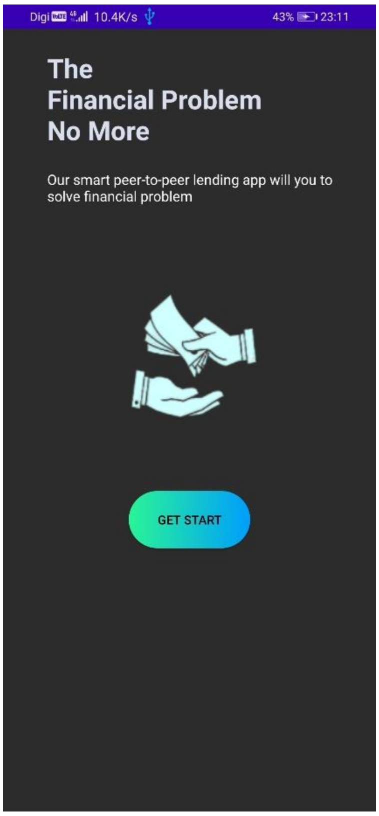
Table 5.1. 3 Requirement for the system

5.3 Setting and Configuration

Once development is complete, the developer may simply click the "Run" button to launch the app on an Android device or emulator. In order to conduct this project, I modified the device settings on my Android phone, a Huawei Y9s. To connect an Android device to Android Studio, the device's developer mode must be enabled and USB debugging enabled. This Android smartphone features a 1080 x 2340 resolution screen and an Android 11.0 CPU, all of which are suitable for the suggested application. Firebase Realtime Database and Firebase Cloud Storage are the databases that are utilized to store all of the user, loan application, investment, transaction, and profile data. Users' email addresses and passwords are stored using Firebase Authentication. Email addresses were also utilized for account verification.

5.4 System Operation (with Screenshot)

The Fintech peer-to-peer lending mobile application is named “M Lending” .

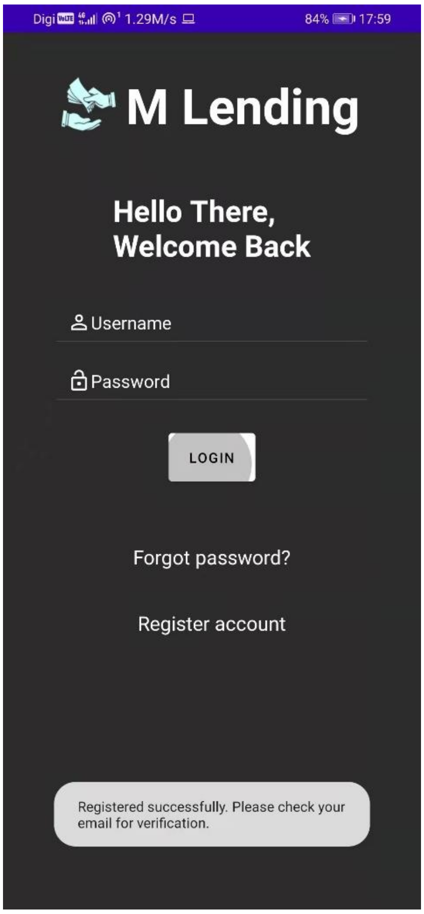


The
Financial Problem
No More

Our smart peer-to-peer lending app will you to solve financial problem

Figure 5.4. 1 Onboarding screen

When user launch the fintech peer-to-peer lending mobile application, it will display the onboarding screen to welcome user. The title of ‘The financial problem no more’ is the aim of the application. This P2P lending mobile application aim to help user to solve financial problem. If user decided to start to use this application, user can press get start and it will navigate to the login page.



Digi 1.29M/s 84% 17:59

M Lending

Hello There,
Welcome Back

Username

Password

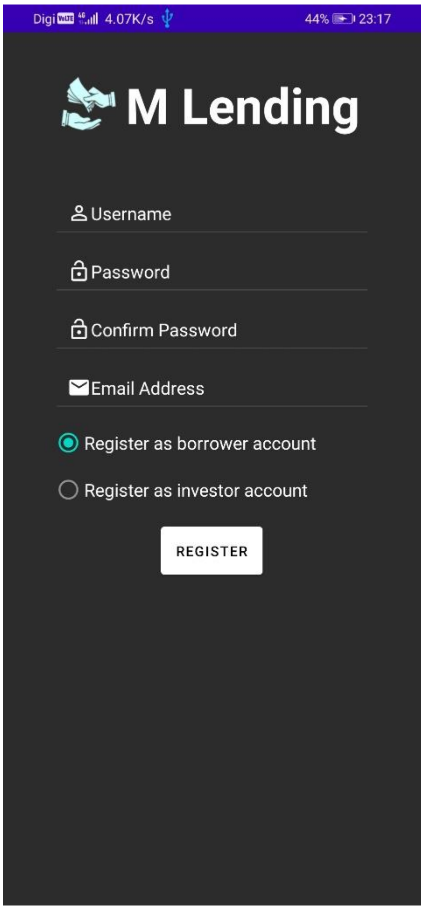
LOGIN

Forgot password?

Register account

Registered successfully. Please check your email for verification.

Figure 5.4. 2 Login Page



Digi 4.07K/s 44% 23:17

M Lending

Username

Password

Confirm Password

Email Address

● Register as borrower account

○ Register as investor account

REGISTER

Figure 5.4. 3 Register Page

After the onbroading screen, it will display the login screen. User login with username and password that registered by user at the register page. If user is new to this application, user can click ‘Register account. User have to key in username, password, confirm pasword, and email to register an account. Besides, there are three type of user account such as administrator account, borrower account and investor account. The user can only register for borrower and investor account as the administrator account is only available for the platform management user. After register successful, the application will direct user to login page and toast ‘Registered successfully. Please check your email for verification’ message. The “Forgot Password?” will redirect user to forgot password page (refer to Figure5.4.4).

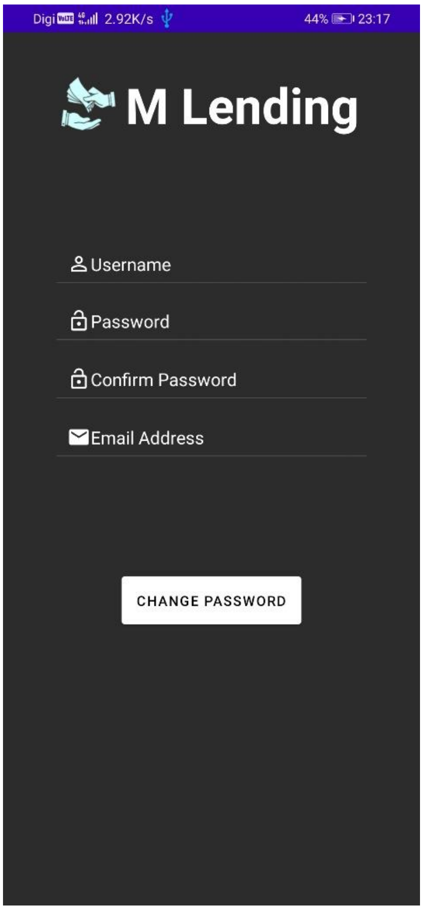


Figure 5.4. 4 Reset Password Page

The user is required to enter the username and email address in order to reset the account password. The system will check whether the username and email address are valid or not. User have to insert the new password twice and the two passwords entered must be the same. Once the system checked, the latest password will be update and save in Firebase for current user, the user can use the new password to login to his/her account immediately.

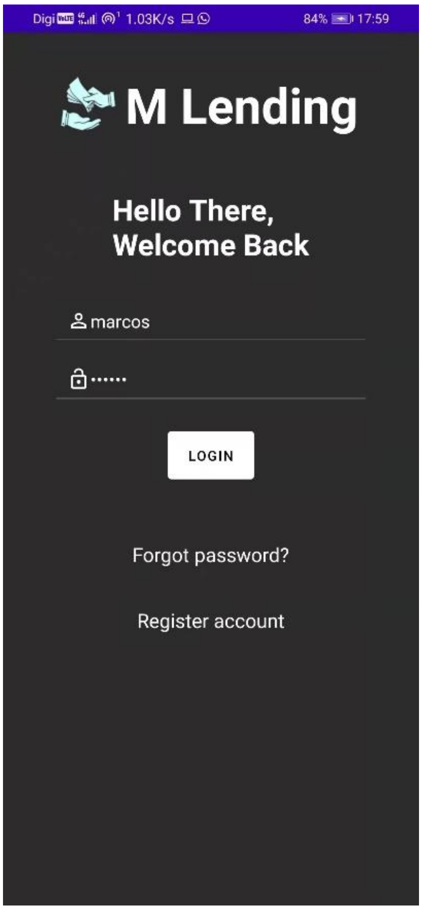


Figure 5.4. 5 Login Page 1

In order to start to use M Lending mobile application, user required to enter his/her username and password. The system will validate the entered username and password with the records stored in Firebase. If valid, "Login successful" message will be prompt and system will redirect user to Main Menu page (refer to Figure 5.4.6).

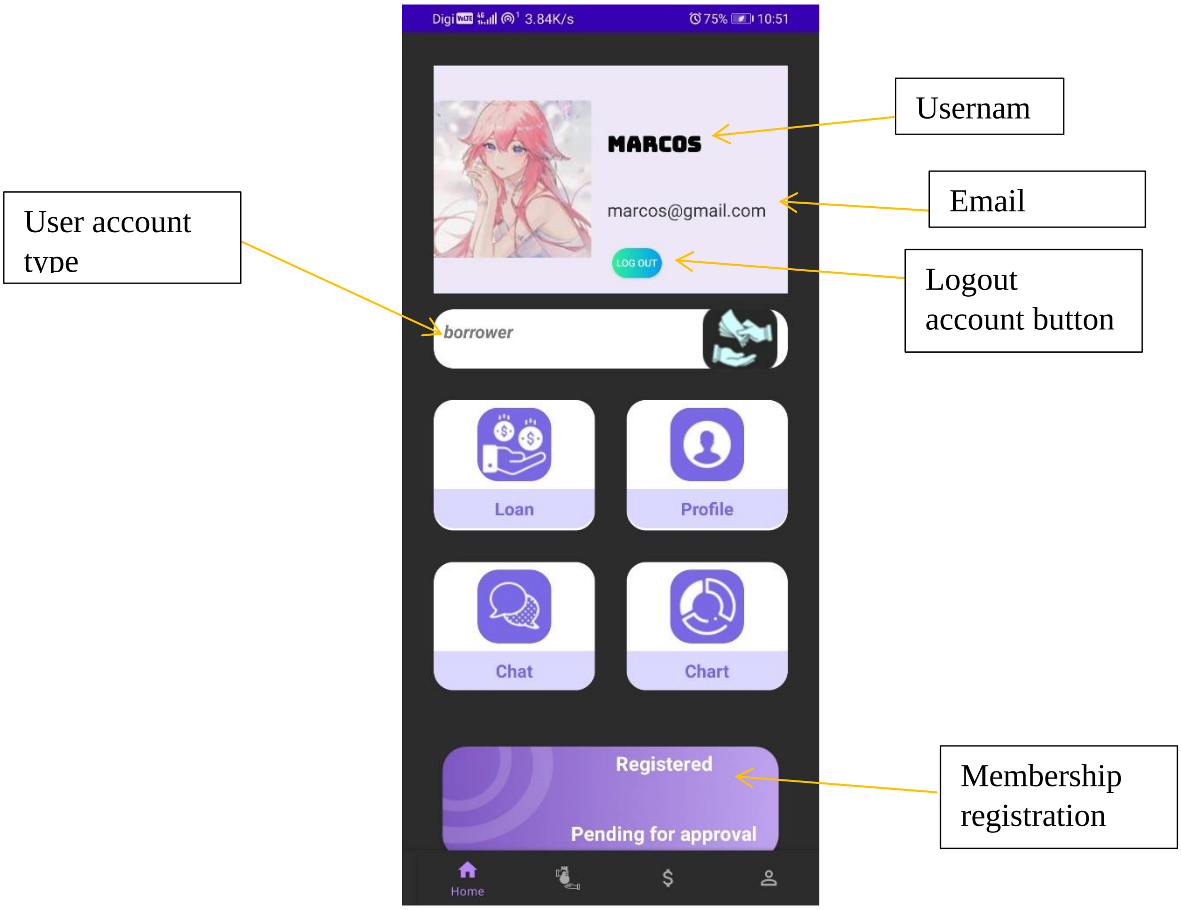
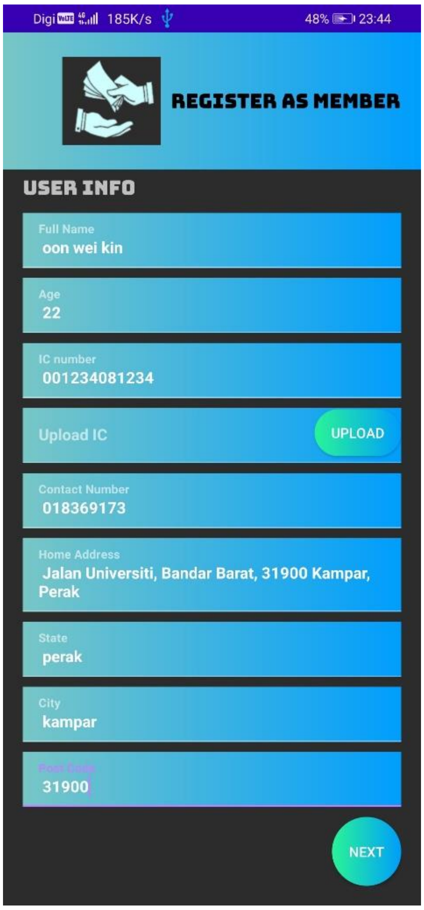


Figure 5.4. 6 Main Menu (borrower)

On top of the main page, it will display username, email address and a log out button for user to log out their account. At the bottom of this page, it has the bottom navigation bar to navigate to main page, loan page, transaction page and profile page. User can have quick access of all page by clicking the four buttons at the middle of this page. The four buttons will redirect user to loan page, profile page, chat page and chart page. To get start with this loan account, user have to register as a member of our lending mobile application.



REGISTER AS MEMBER

USER INFO

Full Name: oon wei kin

Age: 22

IC number: 001234081234

Upload IC — UPLOAD

Contact Number: 018369173

Home Address: Jalan Universiti, Bandar Barat, 31900 Kampar, Perak

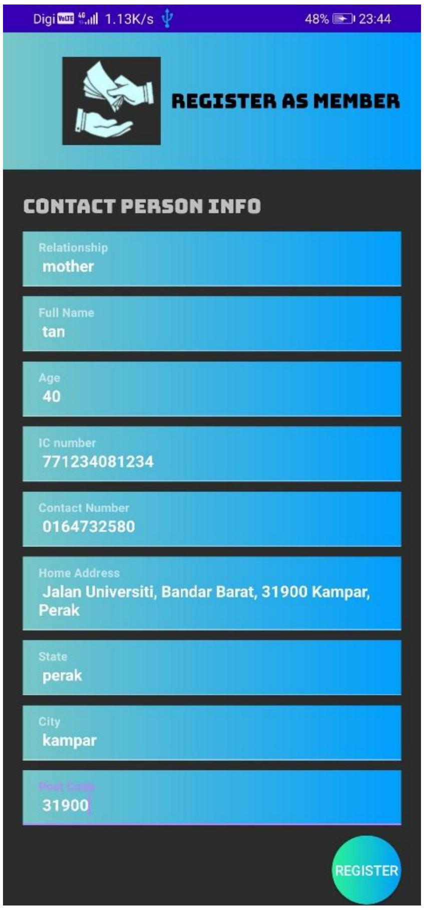
State: perak

City: kampar

Post Code: 31900

NEXT

Figure 5.4. 7 Membership Registration Page (user info)



REGISTER AS MEMBER

CONTACT PERSON INFO

Relationship: mother

Full Name: tan

Age: 40

IC number: 771234081234

Contact Number: 0164732580

Home Address: Jalan Universiti, Bandar Barat, 31900 Kampar, Perak

State: perak

City: kampar

Post Code: 31900

REGISTER

Figure 5.4. 8 Membership Registration (contact person info)

User have to register as member of M lending (fintech P2P lending mobile application) in order to get full user permission. Without full user permission, user can only use the page with Main Menu and Profile page. All the information was required, user have to provide all information and upload their identity card picture. M lending collect the contact person information of user for verification purpose. All the information will store in Firebase and user have to wait for 2-3 working days of the membership registration request approval or reject.

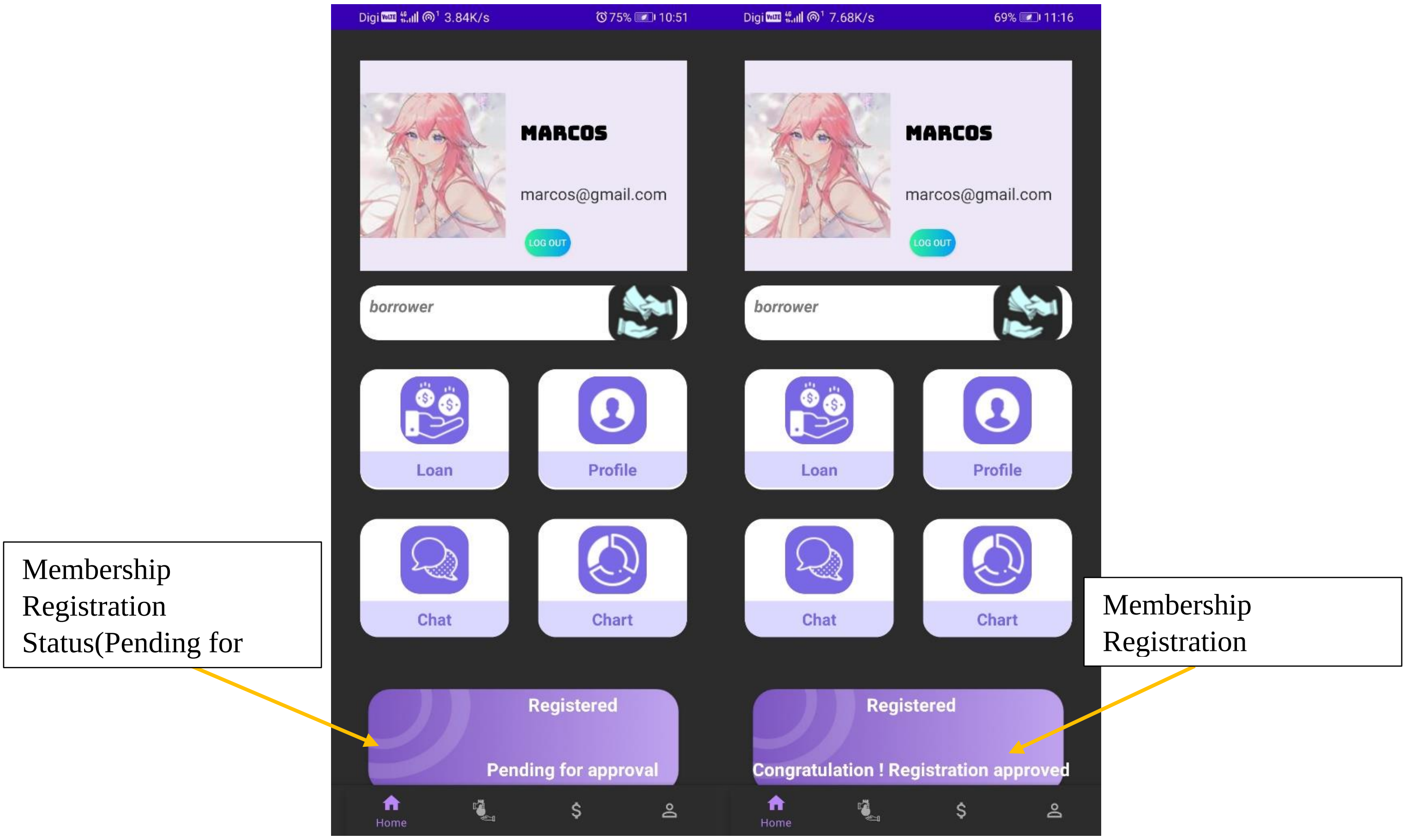
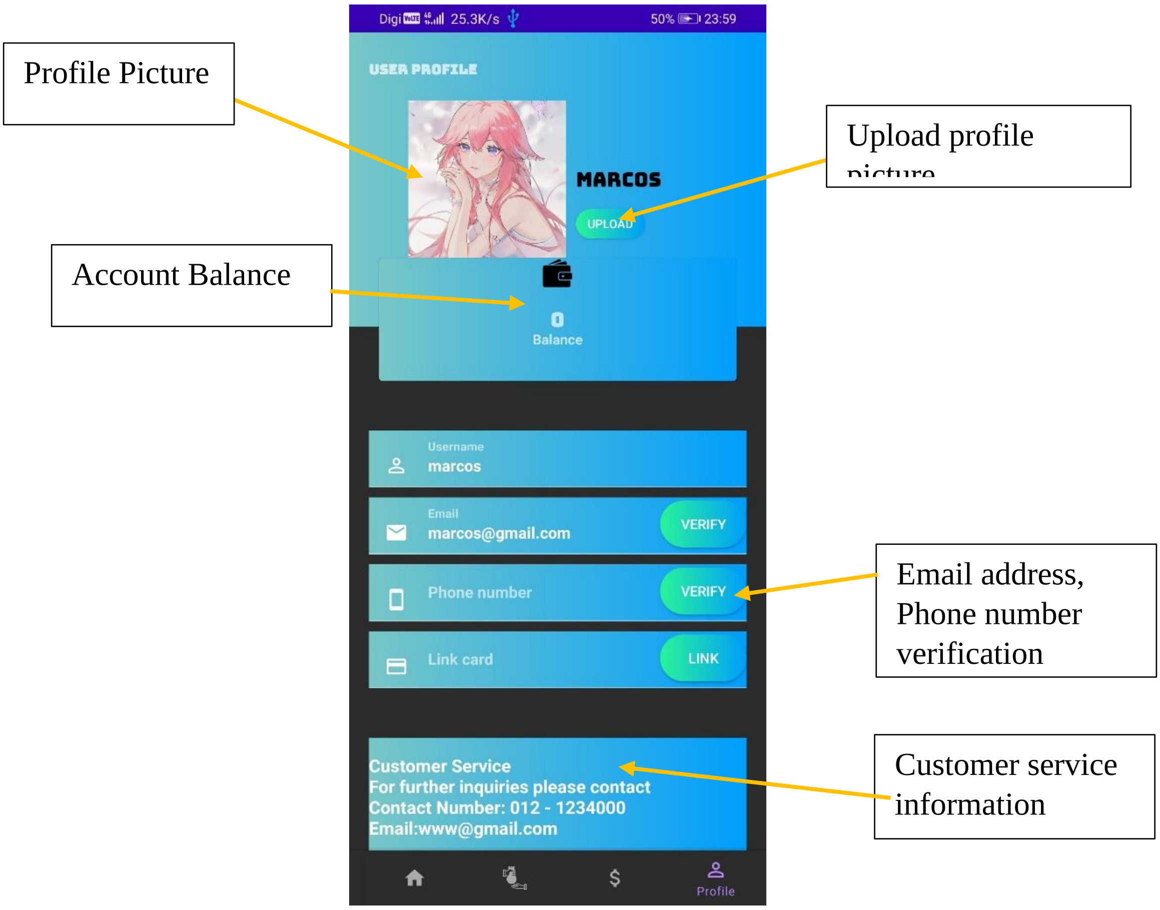


Figure 5.4. 9 Main Menu (Membership Registration)

The purpose of membership registration is to handle the process of user identity verification. After register successful, the text view at the bottom of this page will display “pending for review” .If user was not registered or pending for approval, user can only access to main page and profile page. After the administrator reviewed and approved for the member registration, the text view will display “Registration approved” .Then, user can access to all the page in this application (loan, invest, transaction and chart).



Username
marcos

Email
marcos@gmail.com

Phone number

Link card

Customer Service
For further inquiries please contact
Contact Number: 012 - 1234000
Email:www@gmail.com

Figure 5.4. 10 User Profile Page

Figure 5.4.10 shows the user profile page that contained profile picture, upload button, account balance, verify button, username, email address, phone number, link card and customer service information. User is able to upload and change his/her profile picture. The profile picture will save in Firebase Storage. The account balance display user's account balance and user can top up or withdraw the balance by clicking the account balance view. Users may verify their email address and phone number as an added layer of security to this account. The customer services information was shown at the bottom in user profile page.

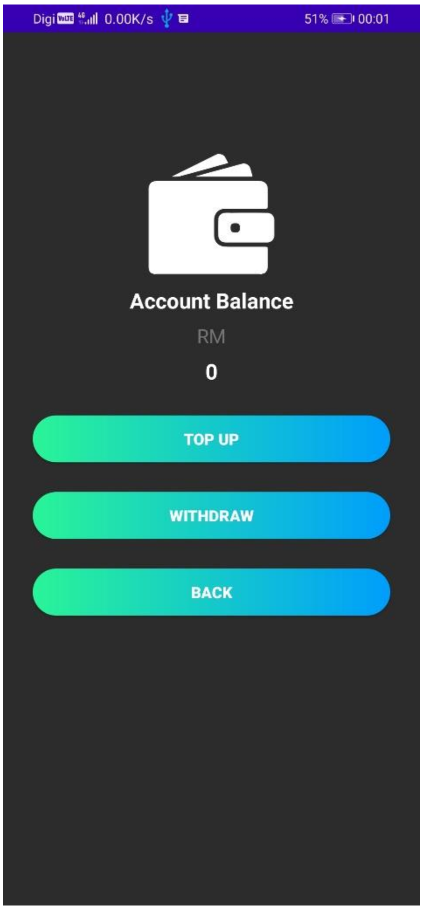


Figure 5.4. 11 Account Balance

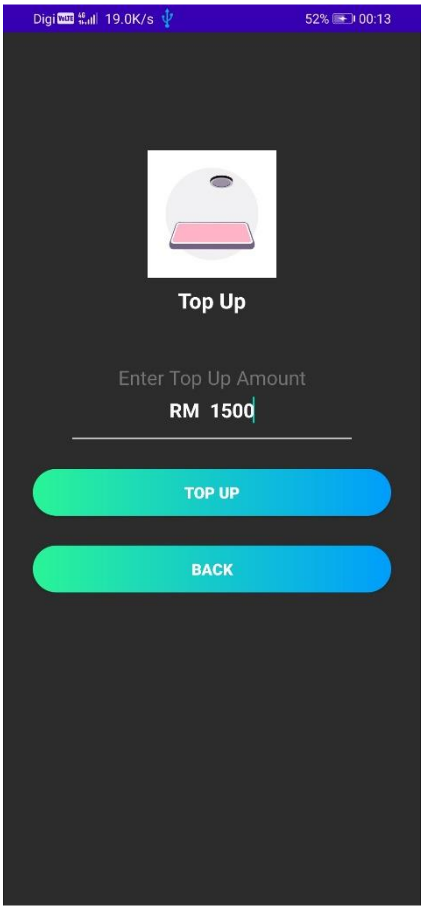
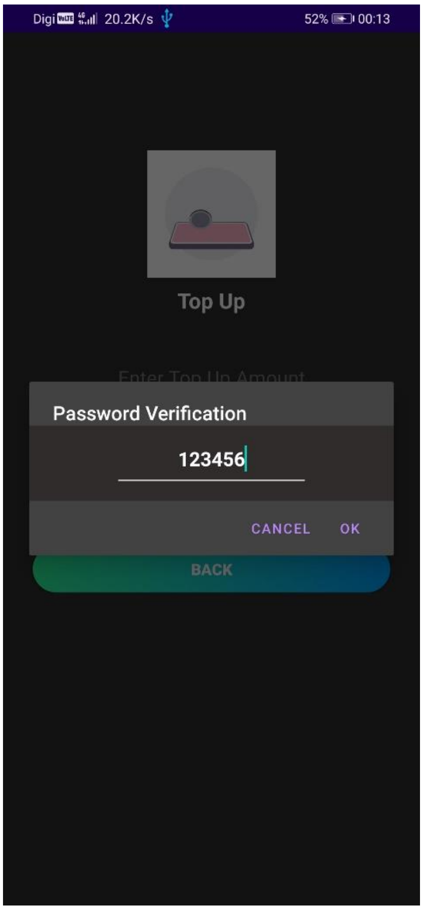


Figure 5.4. 12 Top Up

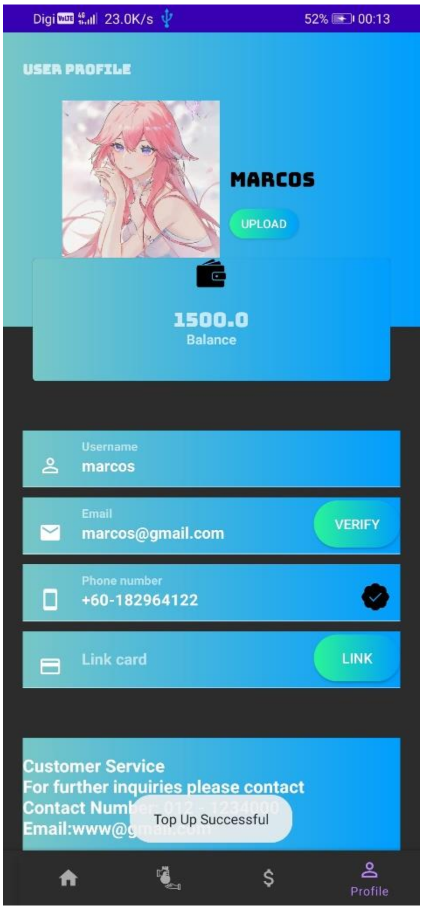
After Clicking the account balance view (refer to Figure 5.4.10), it will redirect user to Account Balance page (refer to Figure 5.4.11) . In this page, user can perform top up and withdraw action, the current account balance of current user is RM 0. In order to perform top up and withdraw action, user have to link their bank card first. If linked, these two functions will be available for user. Else, top up and withdraw button will set to not clickable (refer to Figure 5.4.10 – link card). Figure 5.4.12 shows Top Up page with an inserted amount of RM 1500.



Password Verification

123456

Figure 5.4. 13 Top Up 1



MARCOS

1500.0
Balance

Username
marcos

Email
marcos@gmail.com

Phone number
+60-182964122

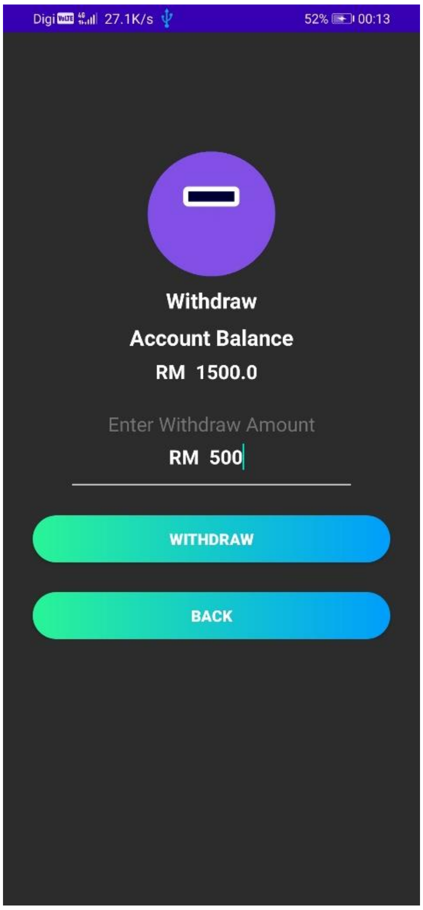
Link card

Customer Service
For further inquiries please contact
Contact Number: 012 - 1234000
Email:www@gmail.com

Top Up Successful

Figure 5.4. 14 User Profile 1

After user inserted the top up amount and click the top up button (refer to Figure5.4.12), a password verification will pop out and request user to key in his/her password to confirm the top up request. Figure 5.4.14 shows “Top Up successful” after the password is verified, the user’s account balance has been updated and save in Firebase.



Digi 27.1K/s 52% 00:13

Withdraw

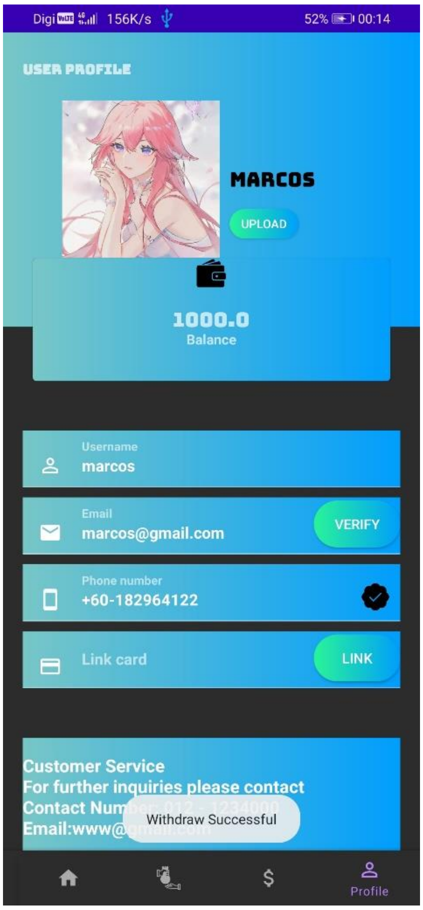
Account Balance

RM 1500.0

Enter Withdraw Amount

RM 500

Figure 5.4. 15 Withdraw



Digi 156K/s 52% 00:14

USER PROFILE

MARCOS

1000.0
Balance

Username
marcos

Email
marcos@gmail.com

Phone number
+60-182964122

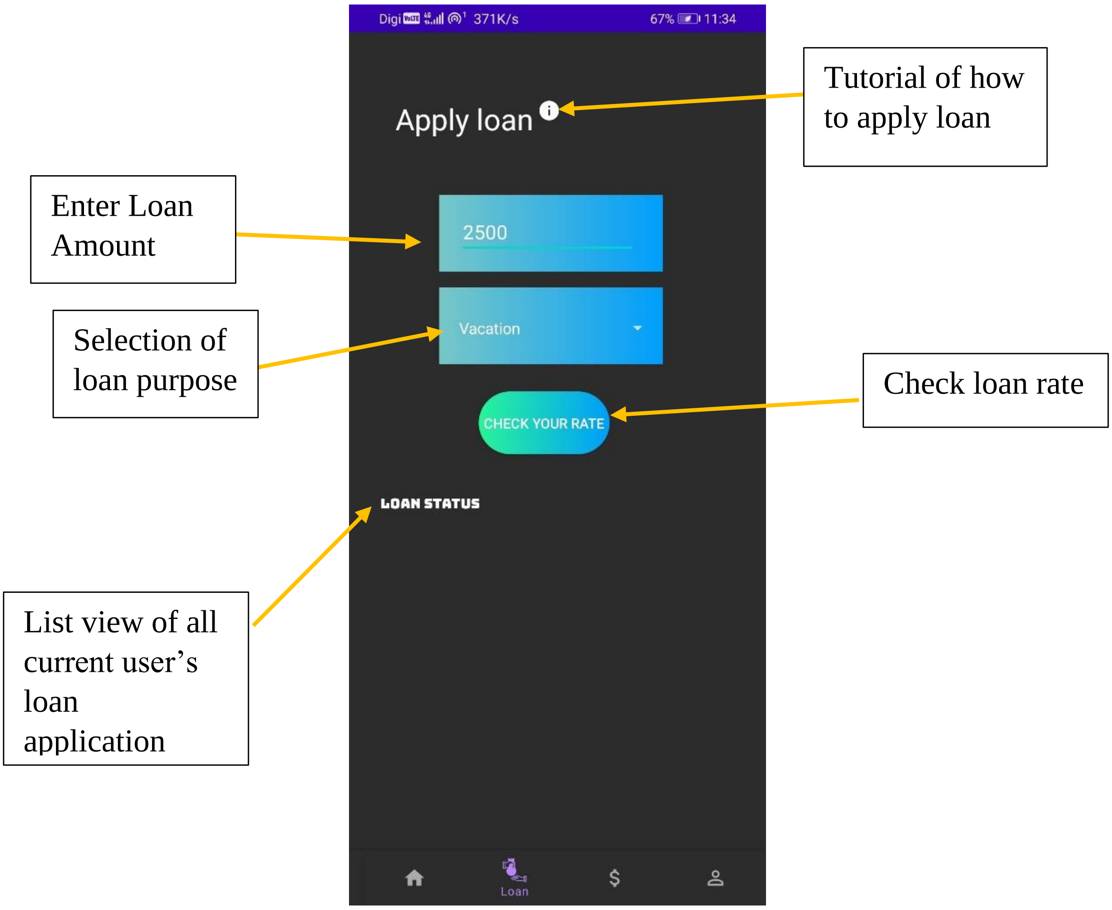
Link card

Customer Service
For further inquiries please contact
Contact Number: 012 - 1234000
Email:www@gmail.com

Withdraw Successful

Figure 5.4. 16 User Profile 2

Figure 5.4.15 shows the current account balance, which is RM 1500. The entered withdraw amount is RM 500. When user click the withdraw button, the system will retrieve current account balance and check whether the current balance is greater than or equal to the entered amount. If the amount is sufficient to perform withdraw action, password verification (refer to Figure 5.4.13) will prompt. After completed verification, the latest account balance will be updated and save in Firebase.



Tutorial of how to apply loan

Enter Loan Amount

Selection of loan purpose

Check loan rate

List view of all current user’s loan application

Figure 5.4. 17 Apply Loan page

This is the loan page of M Lending that online available for borrower account type. This page contained a “Apply loan” title with a tutorial icon, two columns to insert loan application details and a list view to display all current user’s loan application status.

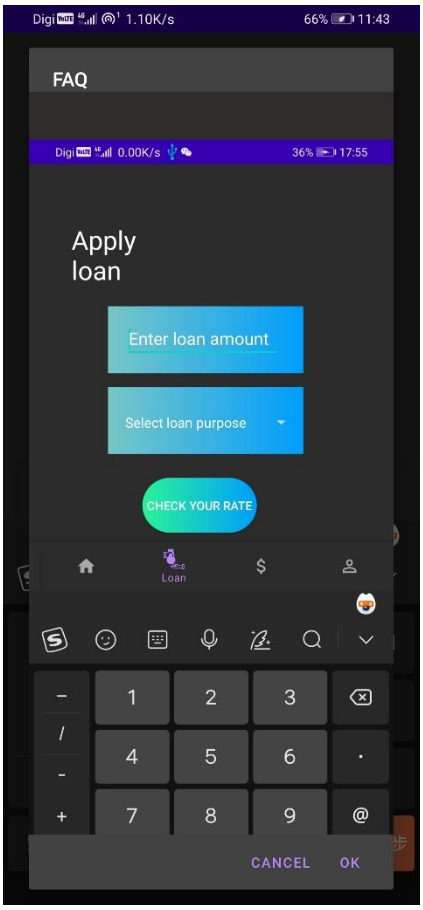
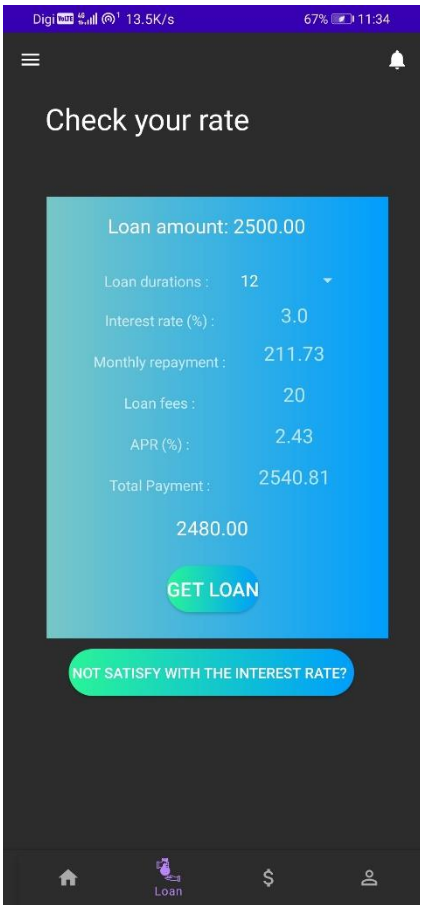


Figure 5.4. 18 Loan Tutorial

Users who are unsure about how to utilize the loan function will find a tutorial button on the loan page (refer to Figure 5.4.18). After pressing the button, the loan page will provide a pop-up dialogue with a brief lesson of the loan application process to give the user a general idea of how to carry out the work.



Check your rate

Loan amount: 2500.00

Loan durations : 12

Interest rate (%) : 3.0

Monthly repayment : 211.73

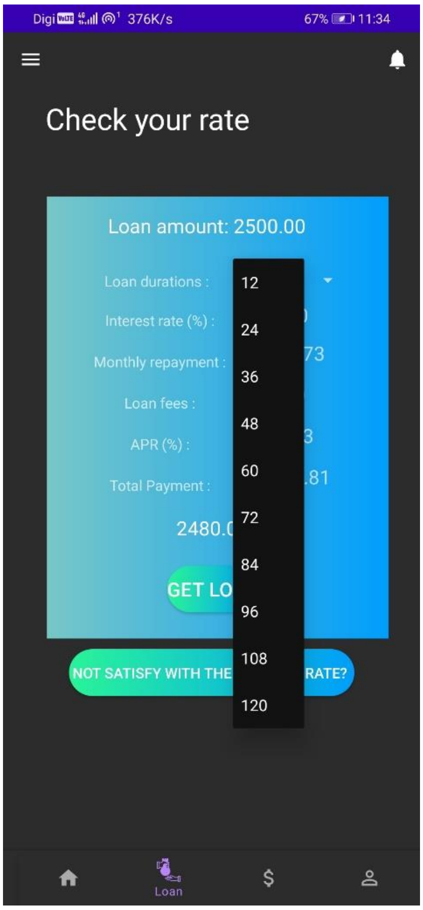
Loan fees : 20

APR (%) : 2.43

Total Payment : 2540.81

2480.00

Figure 5.4. 19 Check Loan Rate



Check your rate

Loan amount: 2500.00

Loan durations :

Interest rate (%) :)

Monthly repayment : 73

Loan fees :

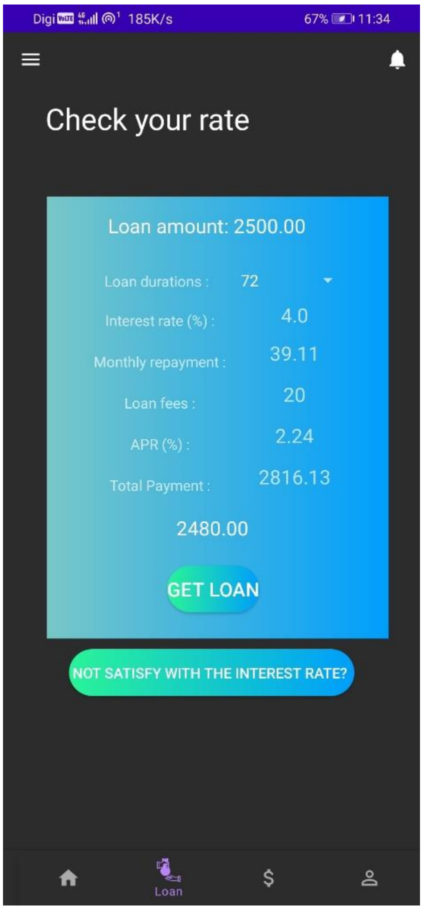
APR (%) : 3

Total Payment : .81

2480.(

Figure 5.4. 20 Check Loan Rate 1

M Lending 3.0% basic interest rate for all type of loan with 12 months loan duration. The loan rate will increase 0.2% for every 12 months of loan duration. For example, 3.2% interest rate for 24 months loan duration, 3.4% interest rate for 36 months duration. Figure 5.4.19 shows a loan rate calculator, the calculator will calculate all the loan details such as interest rate, month repayment, loan fees, annual percentage rate, total payment after user inserted loan amount and selected loan purpose in loan page (refer to Figure 5.4.18). Figure 5.4.19 shows a month selection spinner for user to select loan duration in between 1 to 10 years. The default value of loan duration is set to 12 months, after user select the new loan duration, the loan calculator will recalculate the loan details based on the loan duration (refer to Figure 5.4.21).



Check your rate

Loan amount: 2500.00

Loan durations : 72

Interest rate (%) : 4.0

Monthly repayment : 39.11

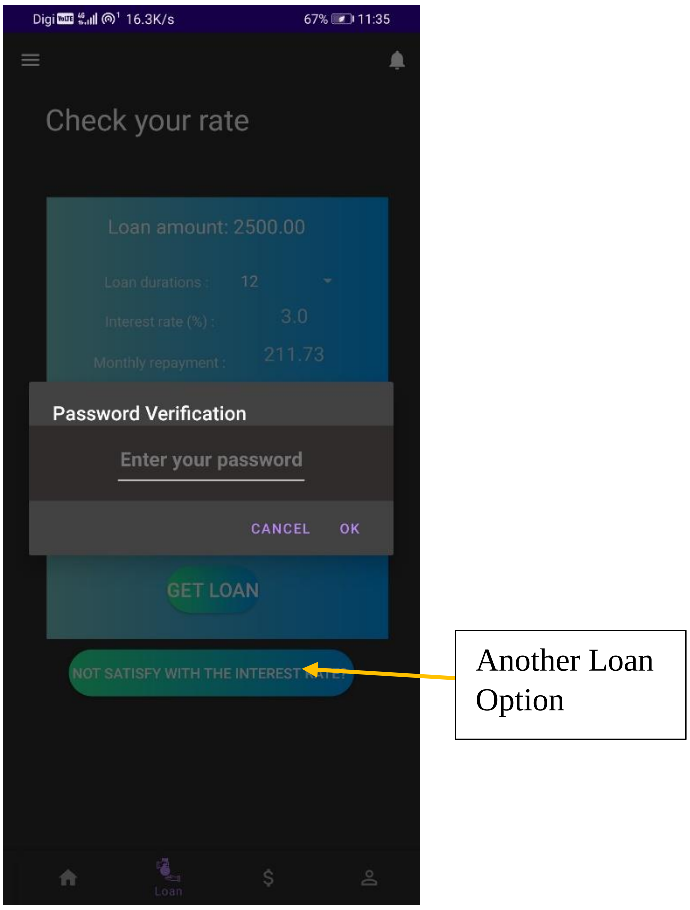
Loan fees : 20

APR (%) : 2.24

Total Payment : 2816.13

2480.00

Figure 5.4. 21 Check Loan Rate 2



Check your rate

Loan amount: 2500.00

Loan durations : 12

Interest rate (%) : 3.0

Monthly repayment : 211.73

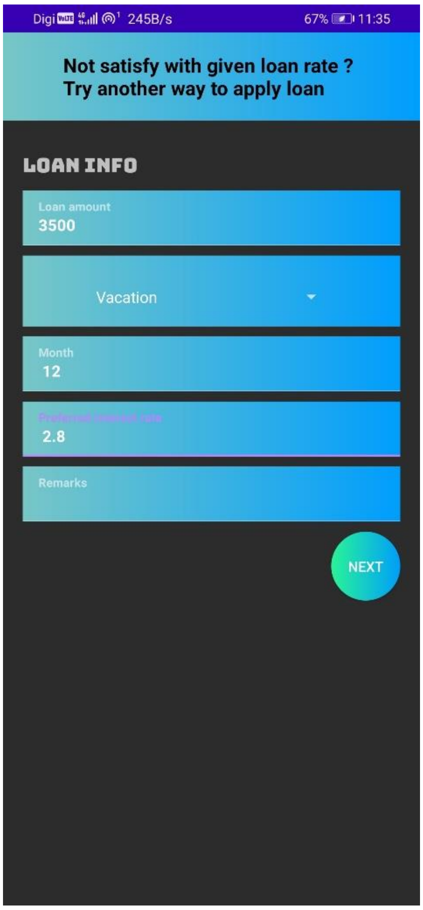
Password Verification

Enter your password

Another Loan Option

Figure 5.4. 22 Check Loan Rate 3

Figure 5.4.21 shows the calculated loan details after user select a loan duration of 72 months. If user is satisfied with the calculated loan rate, user can click “GET LOAN” button and the password verification will pop out to let user enter his/her password for further account verification. If user was not satisfied with the rate given, there are a second option for user to apply for a loan. User can click the button below of “GET LOAN” to apply with second loan method (refer to Figure 5.4.22). The loan application of check loan rate will be sending the platform management for further review and approval.



Not satisfy with given loan rate ?
Try another way to apply loan

LOAN INFO

Loan amount
3500

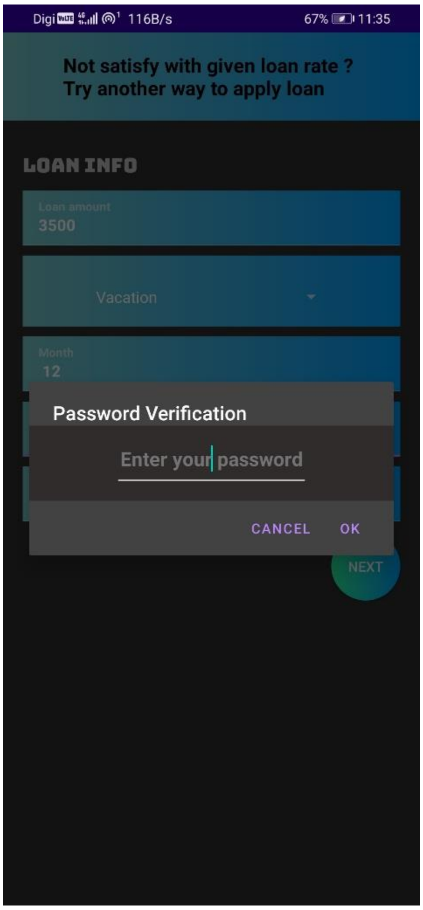
Vacation

Month
12

Preferred interest rate
2.8

Remarks

Figure 5.4. 23 Another Loan 1



Not satisfy with given loan rate ?
Try another way to apply loan

LOAN INFO

Loan amount
3500

Vacation

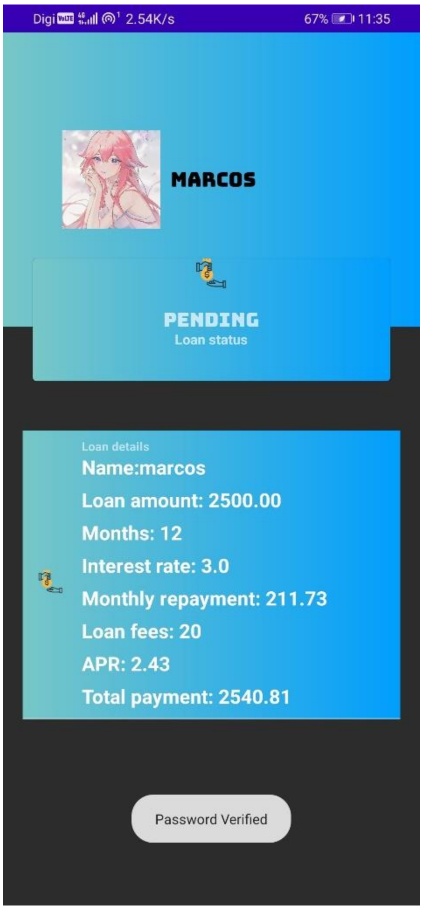
Month
12

Password Verification

Enter your password

Figure 5.4. 24 Another Loan 2

After user select for another loan application option, user can enter the loan duration, interest rate and remarks (refer to Figure 5.4.23). This loan application will send to platform management to review and post in Loan Market for investor to make investment. After user confirm to apply the loan, a dialog of password verification will prompt to let user enter password (refer to Figure 5.4.24).



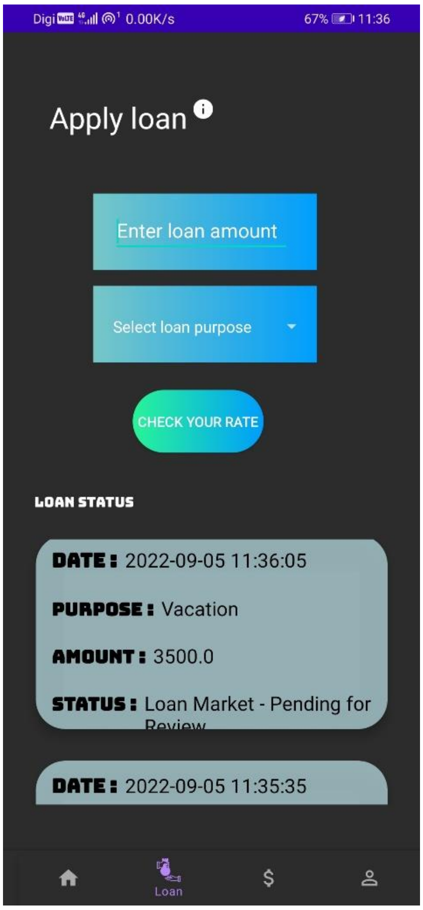
MARCOS

PENDING
Loan status

Loan details
Name:marcos
Loan amount: 2500.00
Months: 12
Interest rate: 3.0
Monthly repayment: 211.73
Loan fees: 20
APR: 2.43
Total payment: 2540.81

Password Verified

Figure 5.4. 25 Loan View



Apply loan

Enter loan amount

Select loan purpose

CHECK YOUR RATE

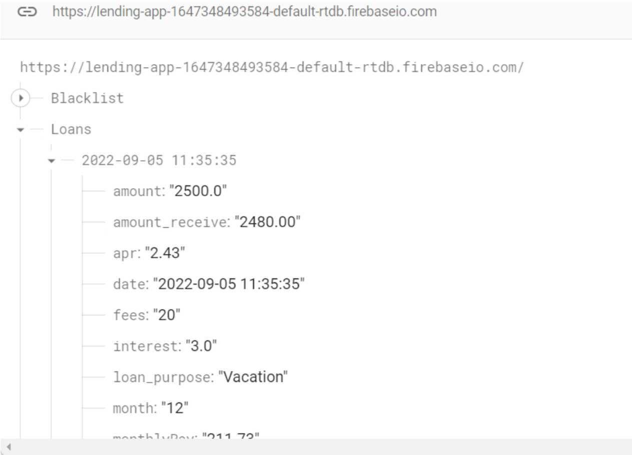
LOAN STATUS

DATE : 2022-09-05 11:36:05
PURPOSE : Vacation
AMOUNT : 3500.0
STATUS : Loan Market - Pending for Review

DATE : 2022-09-05 11:35:35

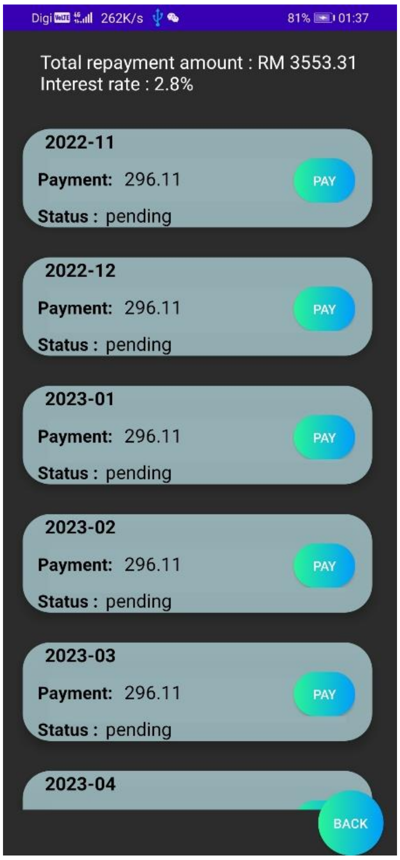
Figure 5.4. 26 Apply Loan Page 1

After password verify successful, a "Password Verified" message will prompt and the system will redirect user to Loan View page that display all the loan application details and loan status (refer to Figure 5.4.25). Back to the apply loan page (Figure 5.4.26), the loan status list view has been updated after user apply loan successful (refer to Figure5 .4.22 and Figure 5.4.24). The loan application's status will be updated after the platform manager reviewed the loan application. All the loan details will be saved in Firebase.



```
https://lending-app-1647348493584-default-rtdb.firebaseio.com

https://lending-app-1647348493584-default-rtdb.firebaseio.com/
  Blacklist
  Loans
    2022-09-05 11:35:35
      amount: "2500.0"
      amount_receive: "2480.00"
      apr: "2.43"
      date: "2022-09-05 11:35:35"
      fees: "20"
      interest: "3.0"
      loan_purpose: "Vacation"
      month: "12"
      monthlyPay: "211.73"
```

Figure 5.4. 27 Realtime Database (store loan application details)



Total repayment amount : RM 3553.31
Interest rate : 2.8%

2022-11
Payment: 296.11
Status : pending

2022-12
Payment: 296.11
Status : pending

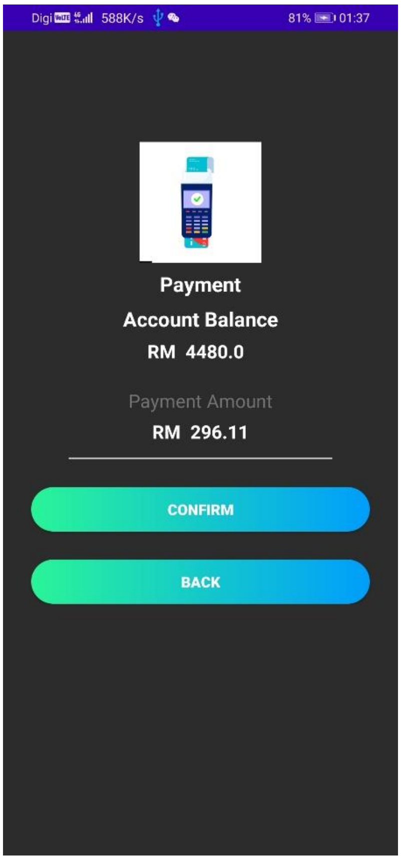
2023-01
Payment: 296.11
Status : pending

2023-02
Payment: 296.11
Status : pending

2023-03
Payment: 296.11
Status : pending

2023-04

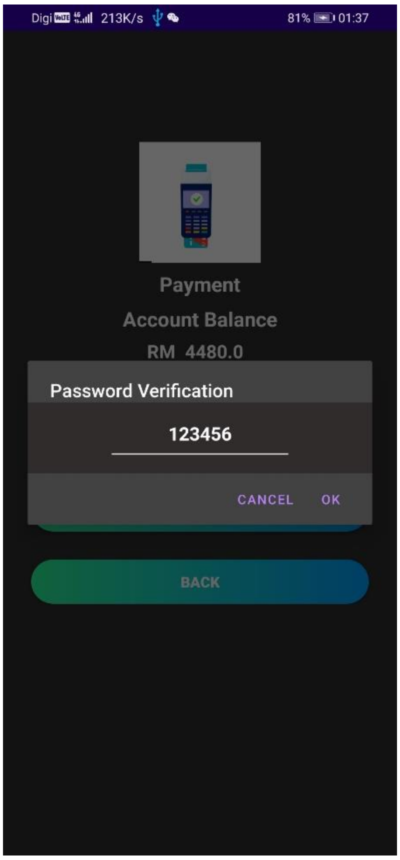
Figure 5.4. 28 Repayment List



Payment
Account Balance
RM 4480.0

Payment Amount
RM 296.11

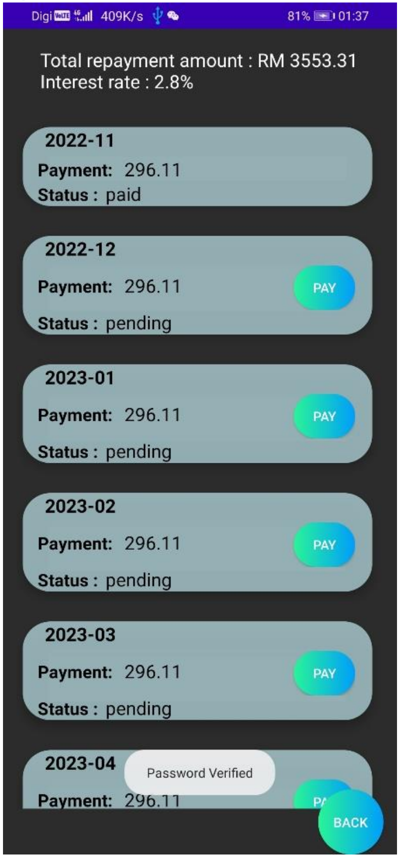
Figure 5.4. 29 Payment



Payment
Account Balance
RM 4480.0

Password Verification
123456

Figure 5.4. 30 Payment 1



Total repayment amount : RM 3553.31
Interest rate : 2.8%

2022-11
Payment: 296.11
Status : paid

2022-12
Payment: 296.11
Status : pending

2023-01
Payment: 296.11
Status : pending

2023-02
Payment: 296.11
Status : pending

2023-03
Payment: 296.11
Status : pending

2023-04
Payment: 296.11

Password Verified

Figure 5.4. 31 Repayment List 1

Figure 5.4.28 shows the repayment list of loan with months. User can click the pay button to pay the payment of related month. The system will redirect user to payment page to process the payment action (refer to Figure 5.4.29). This page will display user's current account balance and the amount of monthly repayment. A dialog of password verification if user click the confirm button to pay the payment (refer to Figure 5.4.30). If the password inserted is valid, a "password verified" message will toast and the system will redirect user to the repayment list page. The pay button will disappear if the payment was made for the month listed (refer to Figure 5.4.31).

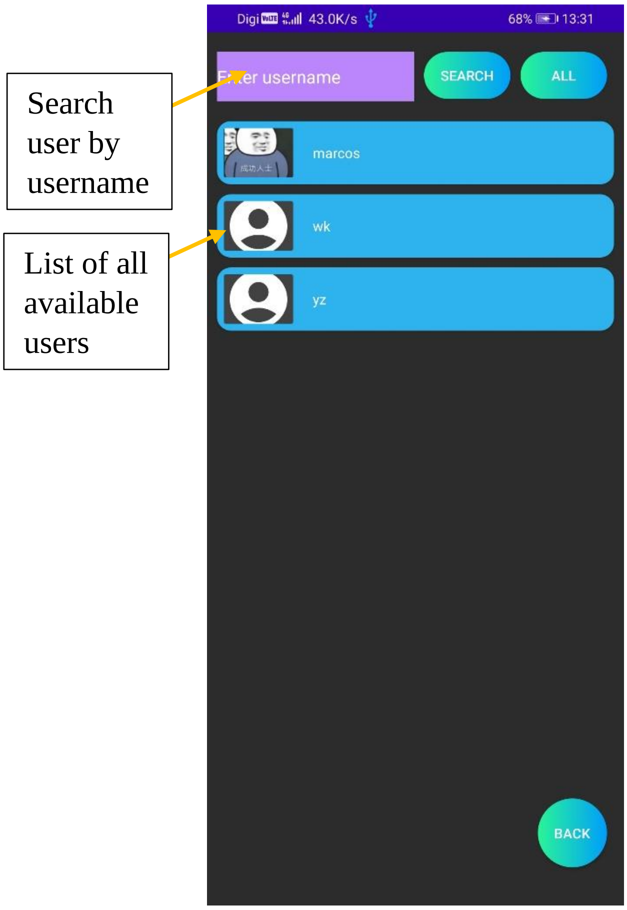
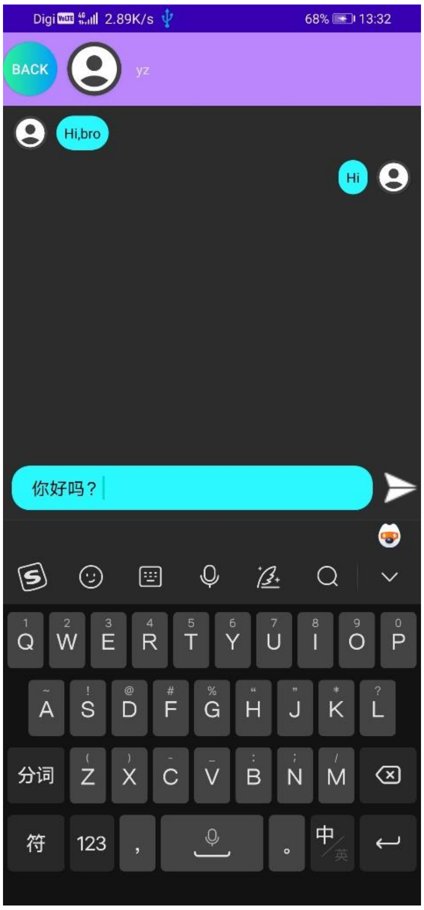


Figure 5.4. 32 Chat Page (user list)



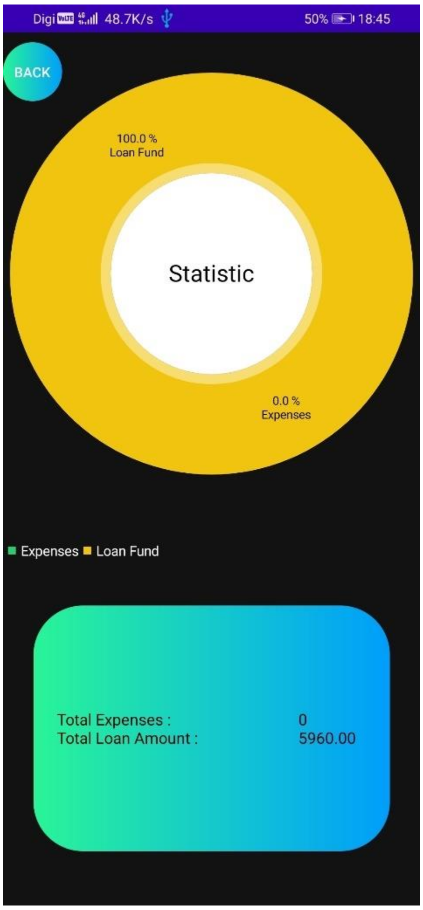
Hi,bro

Hi

你好吗？

Figure 5.4. 33 Message Page

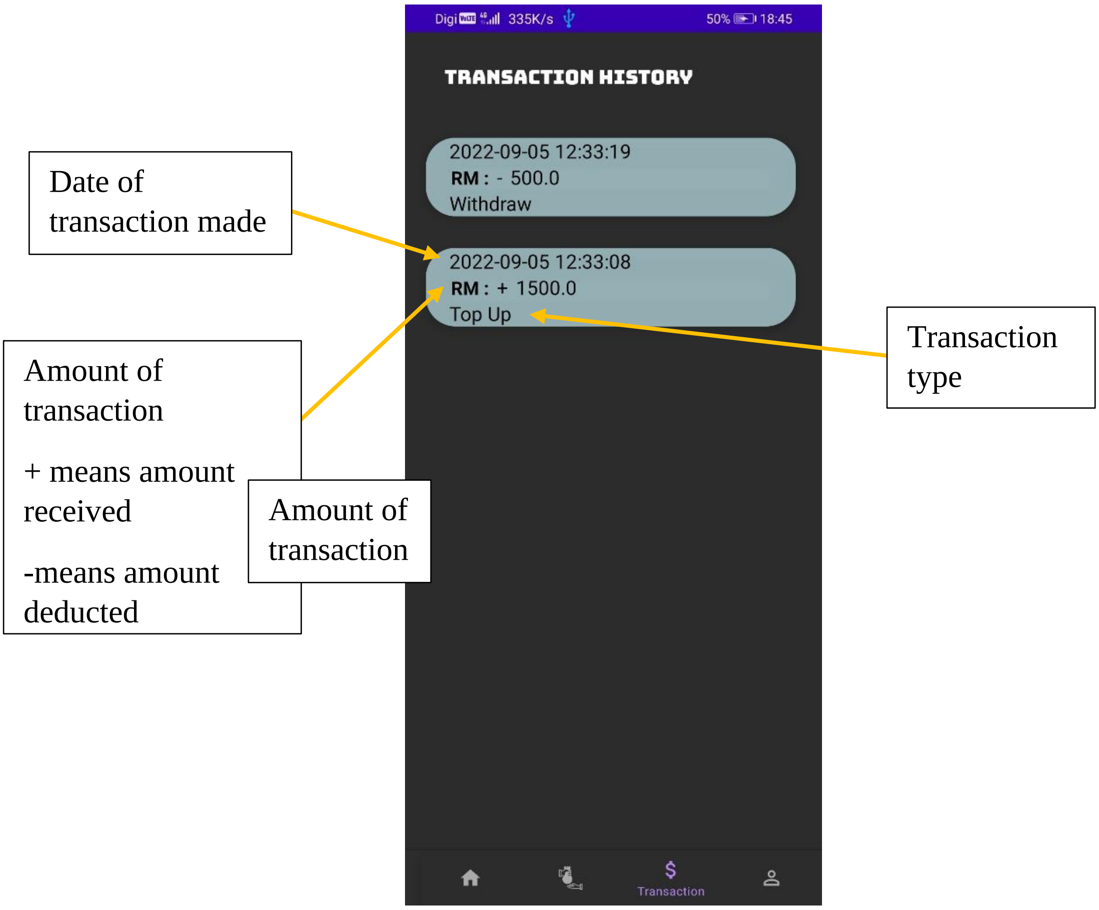
Figure 5.4.32 shows Chat Page that listed all users. Current user can use the search function to search users by using their username. When a user is searched for, all other users are removed from the list and just the person under searched is shown. User can chat with others by just clicking the column in Figure 5.4.32, the system will redirect user to message page (refer to Figure 5.4.33). Users may interact with others to discuss loans or share information on peer-to-peer lending. The conversation history will be saved since all sent messages will be recorded in Firebase. Users may always look back at their previous messages since the system will retrieve them and show on the message page.



Total Expenses : 0
Total Loan Amount : 5960.00

Figure 5.4. 34 Statistic Page

The statistic page shown in figure 5.4.34 will display the total expenses and total loan amount of current user in a pie chart. The expenses included the monthly repayment of loan while the loan amount refer to total loan amount that user applied. Charts are efficient visual aids. due to their rapid and simple information presentation. With this chart, user can know how much they spent in loan repayment and how much loan they applied quickly.



TRANSACTION HISTORY

2022-09-05 12:33:19
RM : - 500.0
Withdraw

2022-09-05 12:33:08
RM : + 1500.0
Top Up

Date of transaction made

Amount of transaction

+ means amount received

-means amount deducted

Amount of transaction

Transaction type

Figure 5.4. 35 Transaction History

Figure 5.4.35 shows transaction history page with two transaction records that recorded in figure 5.4.12 and figure 5.4.15. The transaction records will be saved in Firebase when the transaction process is performed successful. Transaction history record transaction type such as top up, withdraw, loan repayment, invest, and loan funds. The + sign represent amount received while – sign represent amount deducted from account balance. All of the transaction history will be recorded by date and time that made the transaction and save in Firebase.

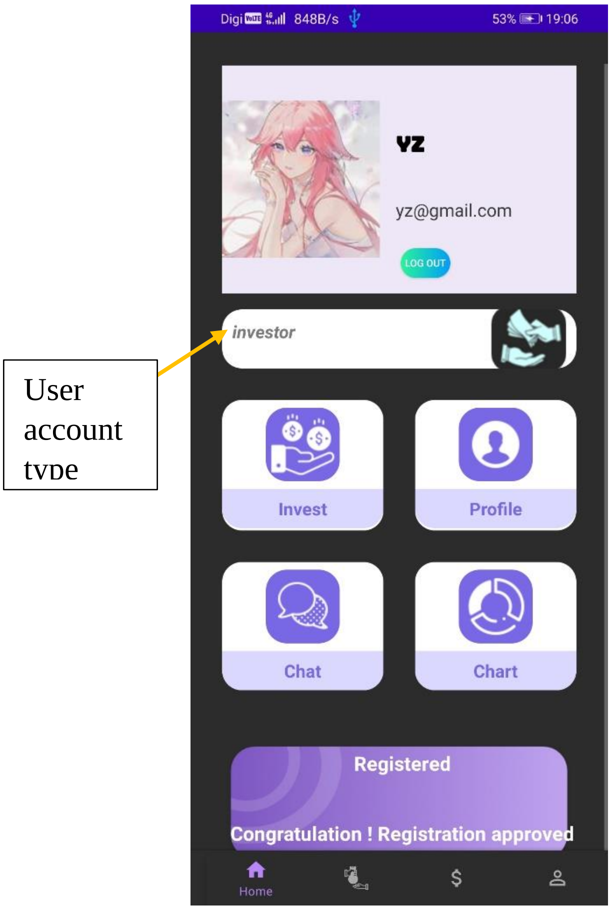
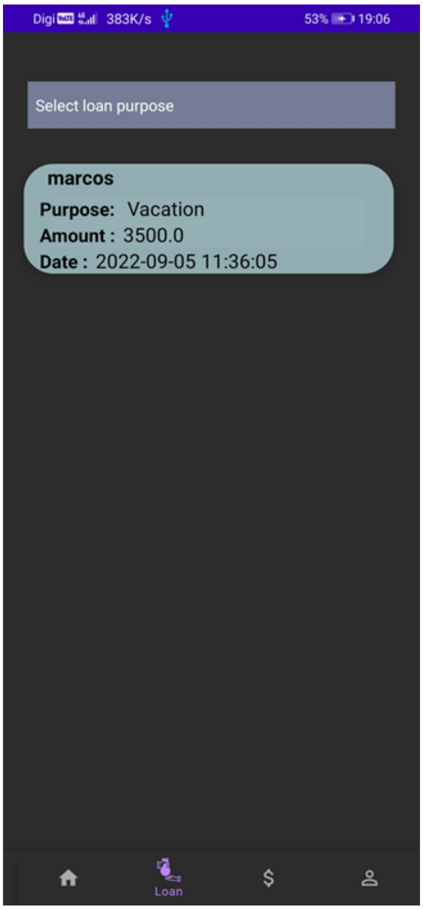


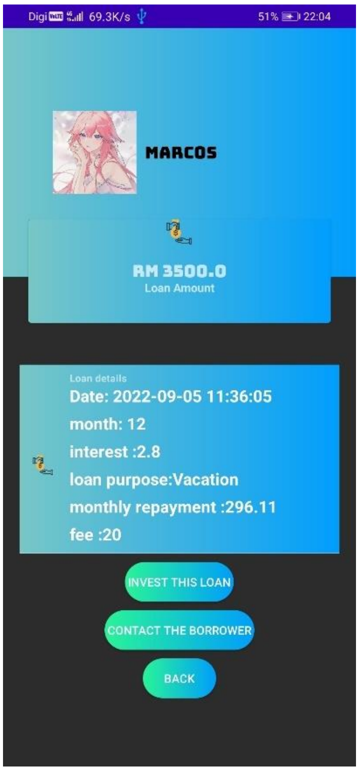
Figure 5.4. 36 Main Menu (investor account)



marcos
Purpose: Vacation
Amount : 3500.0
Date : 2022-09-05 11:36:05

Figure 5.4. 37 Investment page (loan market)

Figure 5.4.36 shows an investor user account, the first main button in the middle of the page have been changed to invest button (refer to Figure 5.4.6). All of the other functions of investor account are same as borrower except the invest function which is only available for investor only. After user click the invest button, system will redirect user to investment account (refer to Figure 5.4.37). The investment page is also known as loan market which listed all of the loan application that applied with second method (refer to Figure 5.4.23). The loans posted in loan market were reviewed by platform manager. User (investor) can filter the loans by their loan type and make investment. Investor can earn repayment of loan of this investment.

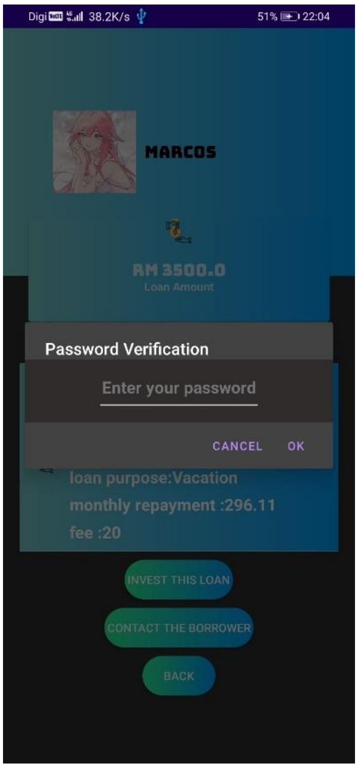


MARCOS

RM 3500.0
Loan Amount

Loan details
Date: 2022-09-05 11:36:05
month: 12
interest :2.8
loan purpose:Vacation
monthly repayment :296.11
fee :20

Figure 5.4. 38 Investment 1



MARCOS

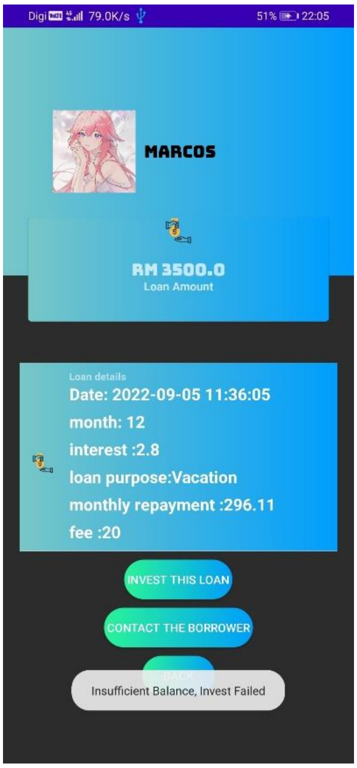
RM 3500.0
Loan Amount

Password Verification

Enter your password

loan purpose:Vacation
monthly repayment :296.11
fee :20

Figure 5.4. 39 Investment 2



MARCOS

RM 3500.0
Loan Amount

Loan details
Date: 2022-09-05 11:36:05
month: 12
interest :2.8
loan purpose:Vacation
monthly repayment :296.11
fee :20

Insufficient Balance, Invest Failed

Figure 5.4. 40 Investment 3

Figure 5.4.38 shows loan details and applicant's name after user (investor) selected their loan to invest (refer to Figure 5.4.37). There are three main buttons in this page which are "invest this loan" button, "contact the borrower" button, and back button. If user confirm to invest the selected and pressed the "invest this loan" button, a password verification dialog will pop out and request user to key in their password for further identity verification (refer to Figure5.4.39). The system will check user's account balance before invest to the loan. If user' balance is lesser than the loan amount, system will toast "Inssuficient balance" to notify user invest failed (refer to Figure 5.4.40). If user click the "contact the borrower" button, the system will direct the user to the chat page, which allows them to search the loan applicant. As a result, the investor may communicate with the borrower to know more about the loan (refer to Figure 5.4.32).

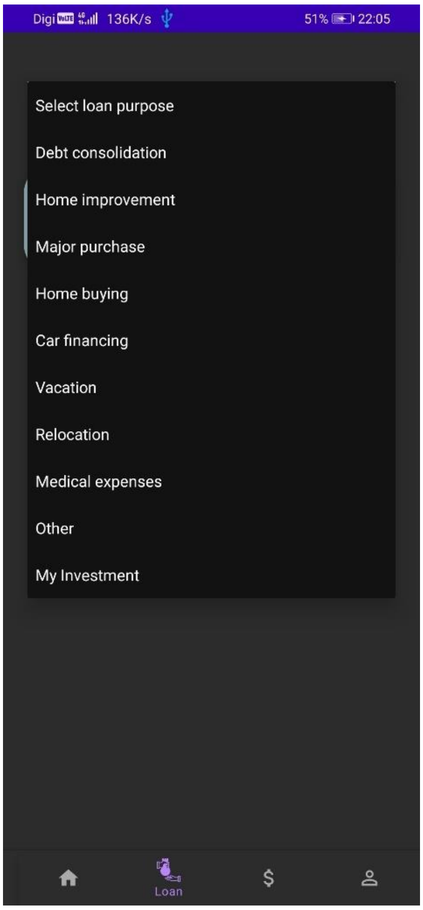
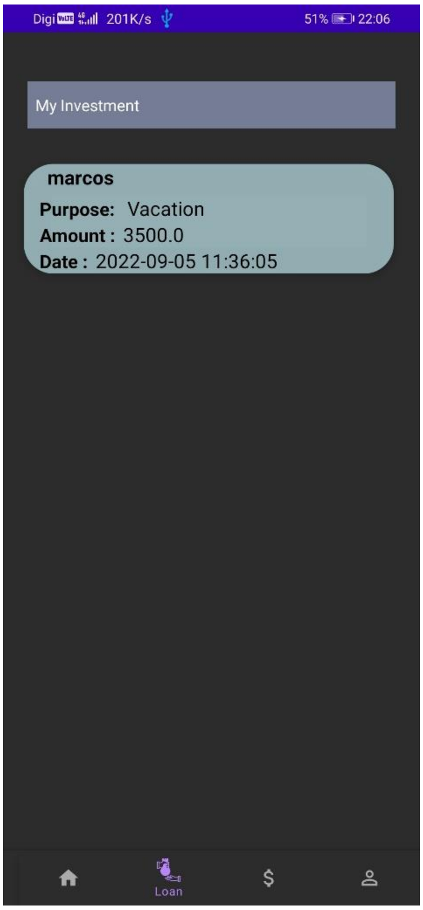


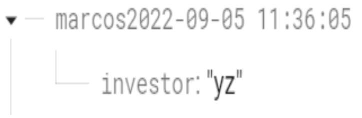
Figure 5.4. 41 Investment page 1



marcos
Purpose: Vacation
Amount : 3500.0
Date : 2022-09-05 11:36:05

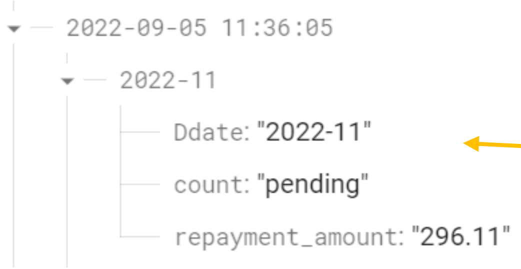
Figure 5.4. 42 Investment page 2

After the user invest a loan successful (refer to Figure 5.4.39), the system will redirect user to investment page. User can use the filter to sort the list view of loan to find out the invested loan. When user selected "My Investment" in the filter, the system will clear other listed loans except the invested loans (refer to Figure 5.4.42). After the investment, the borrower will get the loan money and one month after receiving the loan funds, the borrower must begin making monthly repayments. All of these data will be updated and stored in Firebase.



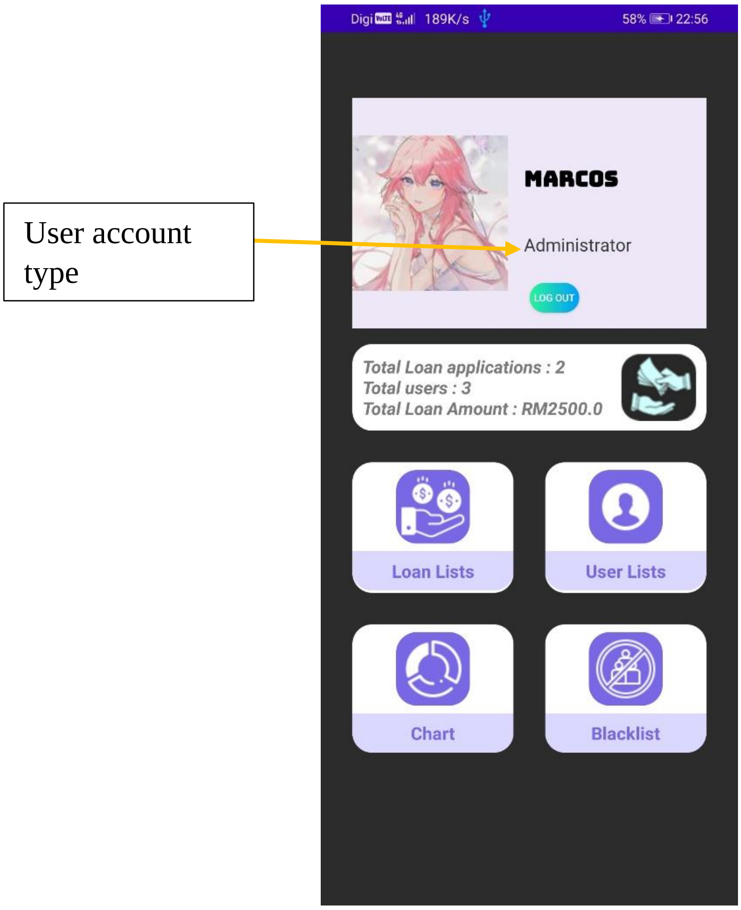
```
marcos2022-09-05 11:36:05
    investor: "yz"
```

Firebase stored investor's username of the loan



```
2022-09-05 11:36:05
    2022-11
        Ddate: "2022-11"
        count: "pending"
        repayment_amount: "296.11"
```

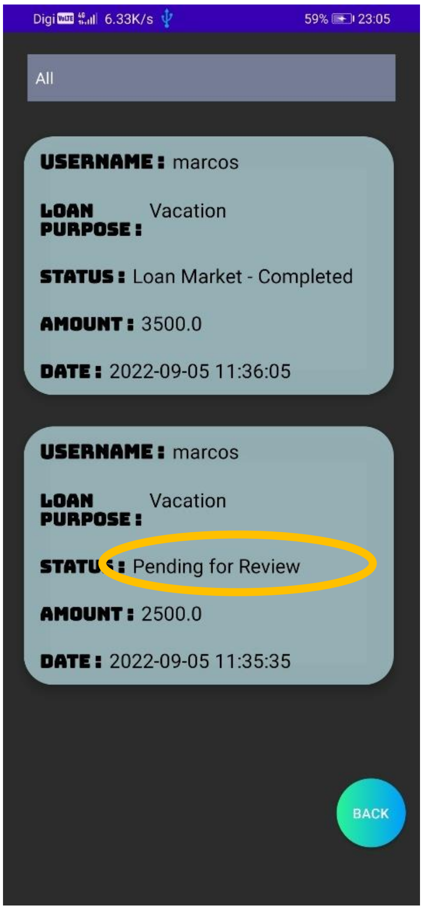
The system will calculate all the repayment month and save in Firebase.



Total Loan applications : 2
Total users : 3
Total Loan Amount : RM2500.0

Figure 5.4. 43 Admin Menu Page

This is the admin module of M Lending which only available for administrator, the platform manager of M Lending to manage the platform. From Figure 5.4.43, it shows a calculation of total loan, total users and total loan amount of the platform. The loans from loan market were excluded in total loan amount as the loans were not given by the platform. However, all the loans applied will be count in total loan application. There are four main buttons (main functions) for admin module such as loan lists (refer to Figure 5.4.44), user lists, chart and blacklist.



USERNAME : marcos

LOAN PURPOSE : Vacation

STATUS : Loan Market - Completed

AMOUNT : 3500.0

DATE : 2022-09-05 11:36:05

USERNAME : marcos

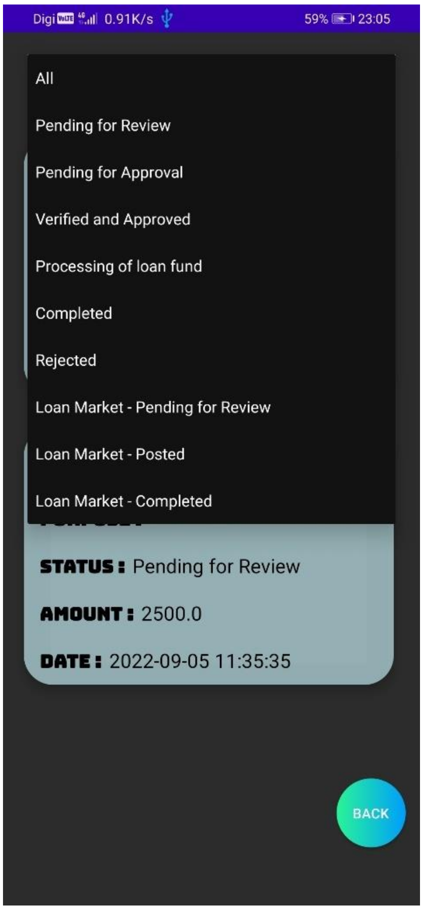
LOAN PURPOSE : Vacation

STATUS : Pending for Review

AMOUNT : 2500.0

DATE : 2022-09-05 11:35:35

Figure 5.4. 44 Loan lists



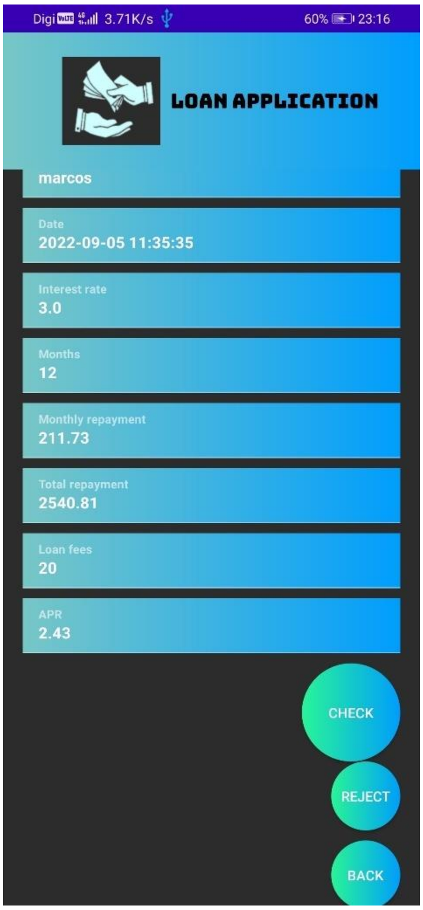
STATUS : Pending for Review

AMOUNT : 2500.0

DATE : 2022-09-05 11:35:35

Figure 5.4. 45 Loan lists 1

Figures 5.4.44 shows a list of loans for admin (platform manager) to manage all the loan application and update their status. The system retrieved all the loan applications details from Firebase and display it in a list. The platform can find loans by filter the list with loan status (refer to Figure 5.4.45). Platform manage can click the loan and the system will navigate user to loan view page to get more information about the loan (refer to Figure 5.4.46).



marcos

Date
2022-09-05 11:35:35

Interest rate
3.0

Months
12

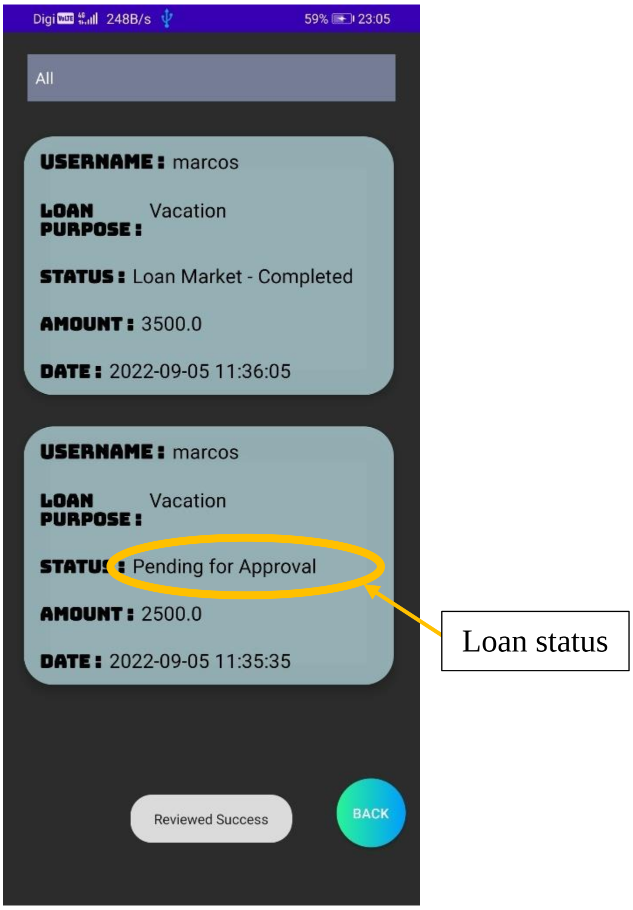
Monthly repayment
211.73

Total repayment
2540.81

Loan fees
20

APR
2.43

Figure 5.4. 46 Loan View Page

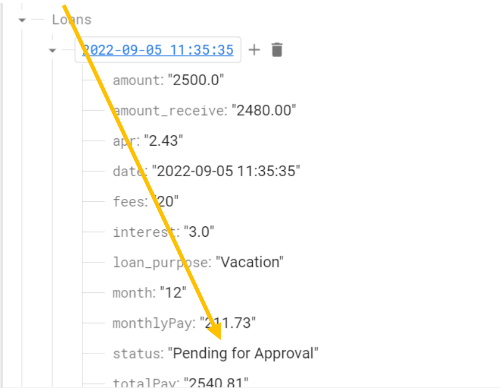


USERNAME : marcos
LOAN PURPOSE : Vacation
STATUS : Loan Market - Completed
AMOUNT : 3500.0
DATE : 2022-09-05 11:36:05

USERNAME : marcos
LOAN PURPOSE : Vacation
STATUS : Pending for Approval
AMOUNT : 2500.0
DATE : 2022-09-05 11:35:35

Figure 5.4. 47 Loan lists 1

After user click the loan from loan list, system will redirect user (platform manager) to loan view page. This page will retrieve the loan data that user clicked on from Firebase. The platform manager can update the loan status by clicking the "check" button in figure 5.4.46. When "check" button is clicked, the system will check user blacklist in the Firebase to check whether the applicant is blacklisted or not. If not, "Reviewed Success" message will toast. The loan status has been updated to "pending for Approval" compared to loan status "pending for review" in figure 5.4.44. The latest loan status will be stored in Firebase.



```
Loans
  2022-09-05 11:35:35
    amount: "2500.0"
    amount_receive: "2480.00"
    apr: "2.43"
    date: "2022-09-05 11:35:35"
    fees: "20"
    interest: "3.0"
    loan_purpose: "Vacation"
    month: "12"
    monthlyPay: "211.73"
    status: "Pending for Approval"
    totalPay: "2540.81"
```

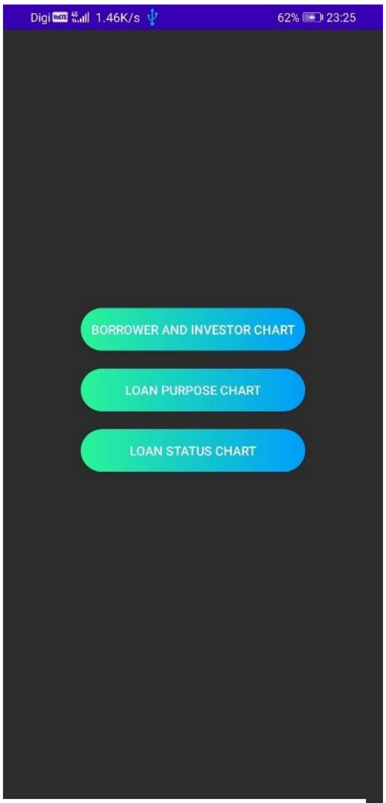
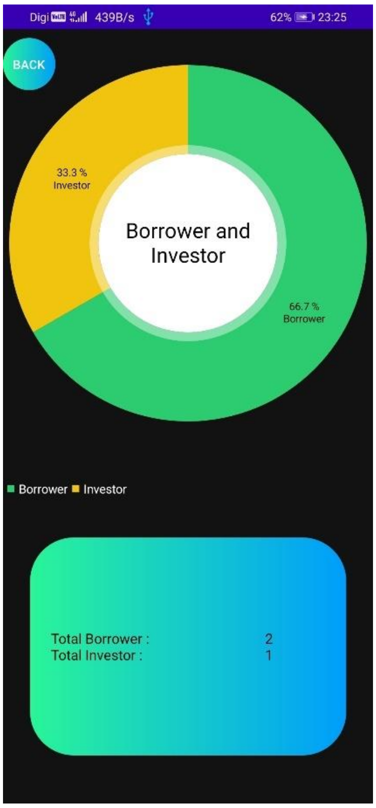
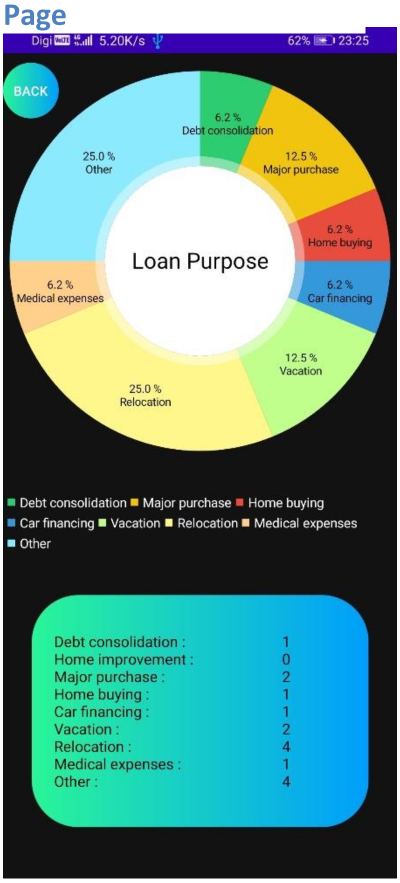


Figure 5.4. 48 Admin Chart Page



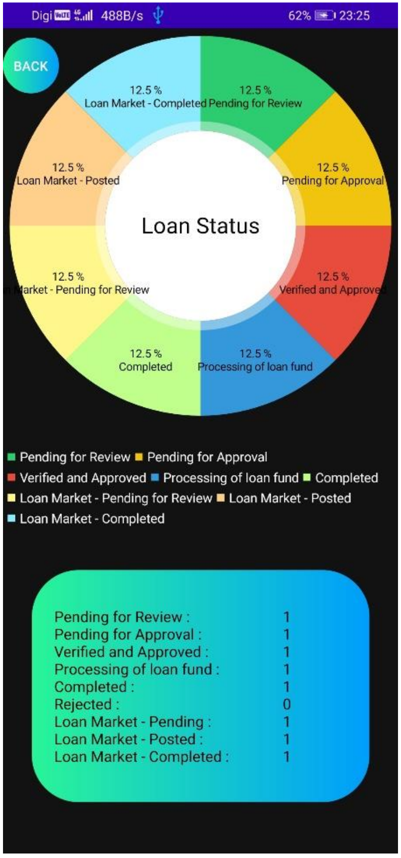
Total Borrower : 2
Total Investor : 1

Figure 5.4. 49 Account type chart



Debt consolidation : 1
Home improvement : 0
Major purchase : 2
Home buying : 1
Car financing : 1
Vacation : 2
Relocation : 4
Medical expenses : 1
Other : 4

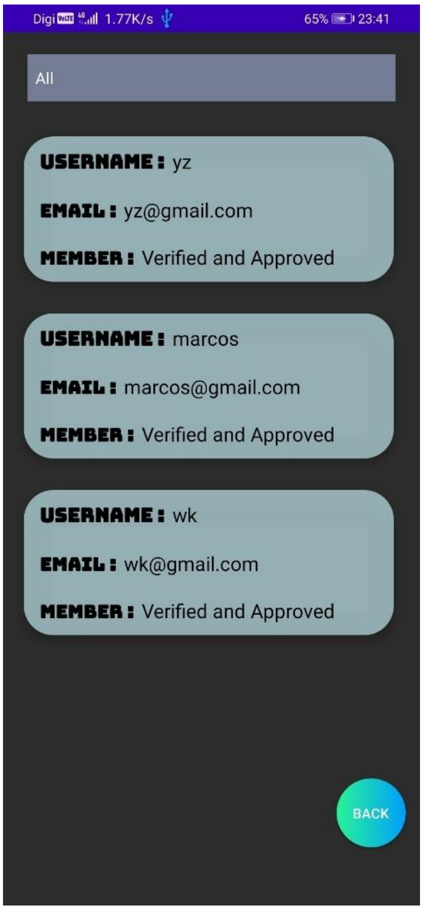
Figure 5.4. 50 Loan purpose chart


Digi 488B/s 62% 23:25
BACK
Loan Status
12.5 % Loan Market - Completed
12.5 % Pending for Review
12.5 % Pending for Approval
12.5 % Verified and Approved
12.5 % Processing of loan fund
12.5 % Completed
12.5 % Market - Pending for Review
12.5 % Loan Market - Posted
🟩 Pending for Review 🟨 Pending for Approval
🟥 Verified and Approved 🟦 Processing of loan fund 🟩 Completed
🟨 Loan Market - Pending for Review 🟧 Loan Market - Posted
🟦 Loan Market - Completed

Pending for Review : 1
Pending for Approval : 1
Verified and Approved : 1
Processing of loan fund : 1
Completed : 1
Rejected : 0
Loan Market - Pending : 1
Loan Market - Posted : 1
Loan Market - Completed : 1

Figure 5.4. 51 Loan status chart

Figure 5.4.48 shows an admin chart page that display all the available chart. The platform manager can select the related chart button and the system will redirect user to related chart page such as account type chart (refer to Figure 5.4.49), loan purpose chart (refer to Figure 5.4.50) and loan status chart (refer to Figure 5.4.51). The account type chart shown in figure 5.4.49 conclude and categorize user's account type into borrower and investor. The account type chart displays the total count of borrower and investor account. Figure 5.4.50 shows loan purpose chart that conclude total loan applications and categorized them with loan purpose. This chart displays the count of loan purpose for all loan applications. The last chart which is loan status chart categorize all the loans by their loan status. This chart displays the count of every loan with loan status.



USERNAME : yz

EMAIL : yz@gmail.com

MEMBER : Verified and Approved

USERNAME : marcos

EMAIL : marcos@gmail.com

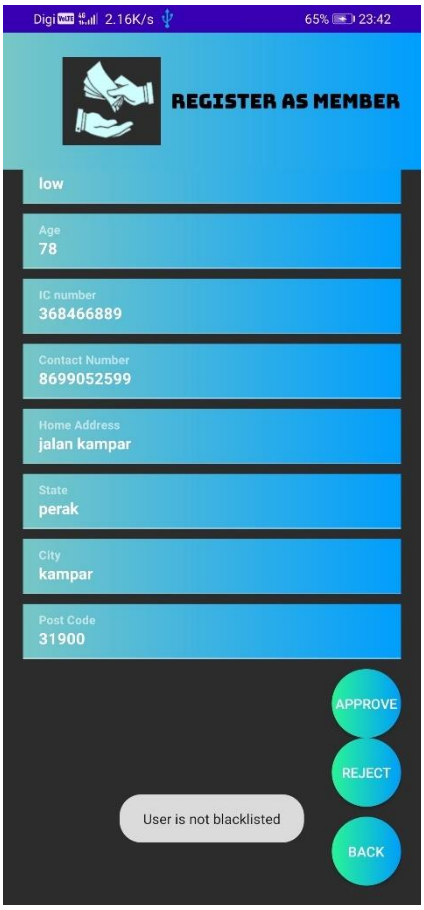
MEMBER : Verified and Approved

USERNAME : wk

EMAIL : wk@gmail.com

MEMBER : Verified and Approved

Figure 5.4. 52 Users list page



low

Age
78

IC number
368466889

Contact Number
8699052599

Home Address
jalan kampar

State
perak

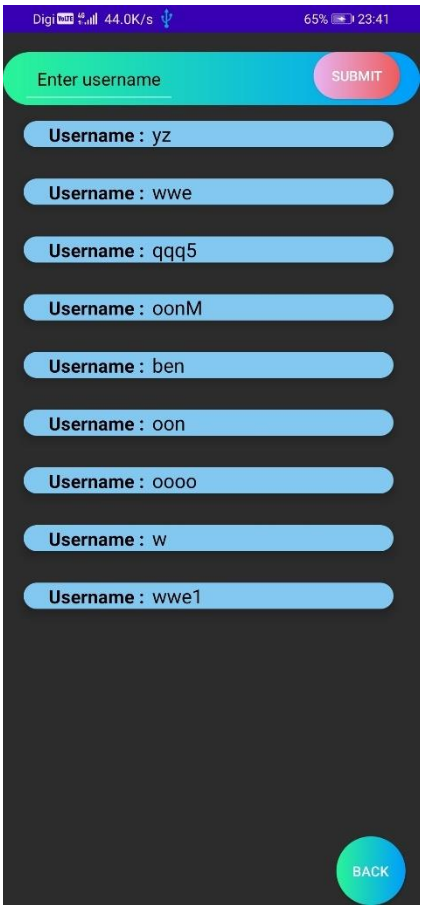
City
kampar

Post Code
31900

User is not blacklisted

Figure 5.4. 53 User View Page

Figure 5.4.52 shows all the users of M Lending in a list view, the platform manager is able to filter the users by their membership status. In order to change the membership status of user, platform manager can click on the user view and the system will redirect user to the User View Page (refer to Figure 5.4.53). Platform manager can click the buttons at the bottom right corner of the page to perform certain tasks like check, approve, reject and back. When user click the check button, the system will check blacklist to verify the username of user is blacklisted or not. If not, a "user is not blacklisted" message will toast and the check button will change to approve button. After user click the approve button, the membership status of the user will be changed and save in Firebase.



Username : yz

Username : wwe

Username : qqq5

Username : oonM

Username : ben

Username : oon

Username : oooo

Username : w

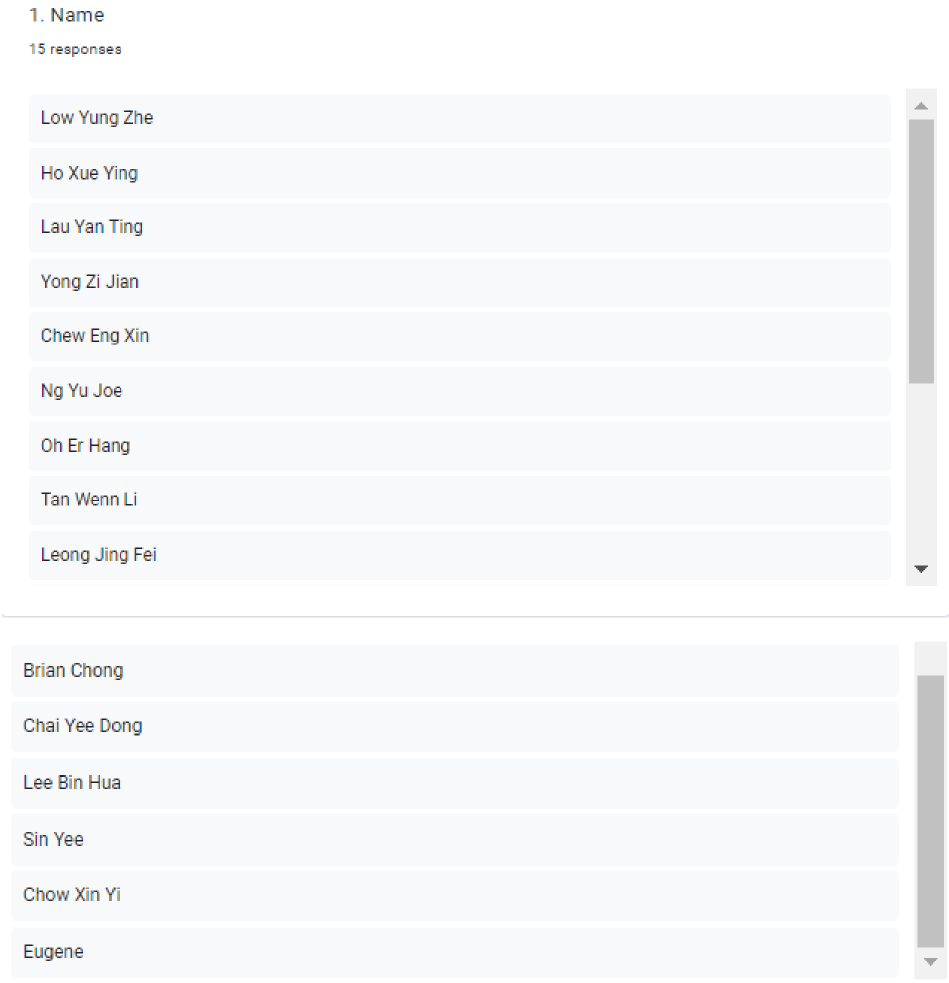
Username : wwe1

Figure 5.4. 54 Black List

Figure 5.4.54 shows a black list page that display all blacklisted username. All blacklisted username will be saved in Firebase. The platform manager can add new username to the black list and platform manager can check the username for the approval of user's membership registration request and loan application request.

6 Chapter 6 System Evaluation and Discussion

6.1 System Evaluation Survey Results



1. Name

15 responses

Low Yung Zhe

Ho Xue Ying

Lau Yan Ting

Yong Zi Jian

Chew Eng Xin

Ng Yu Joe

Oh Er Hang

Tan Wenn Li

Leong Jing Fei

Brian Chong

Chai Yee Dong

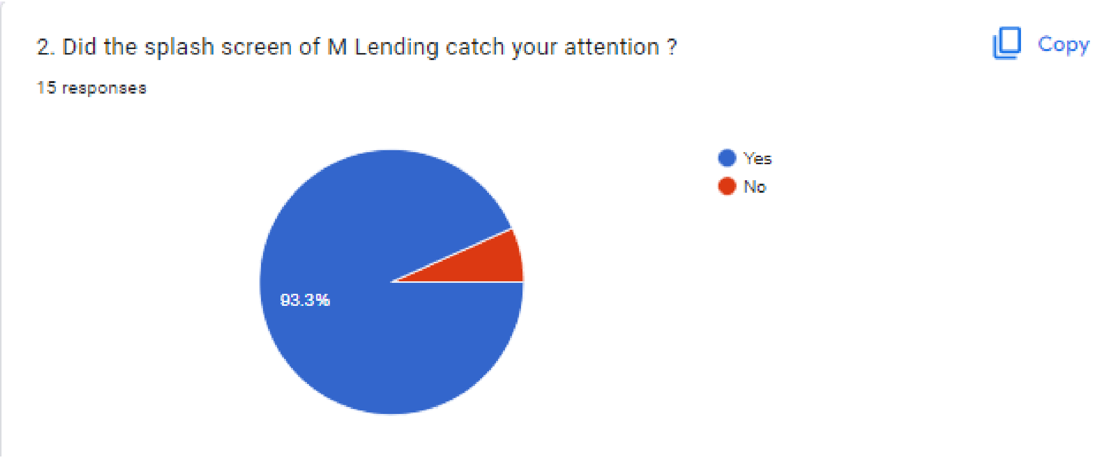
Lee Bin Hua

Sin Yee

Chow Xin Yi

Eugene

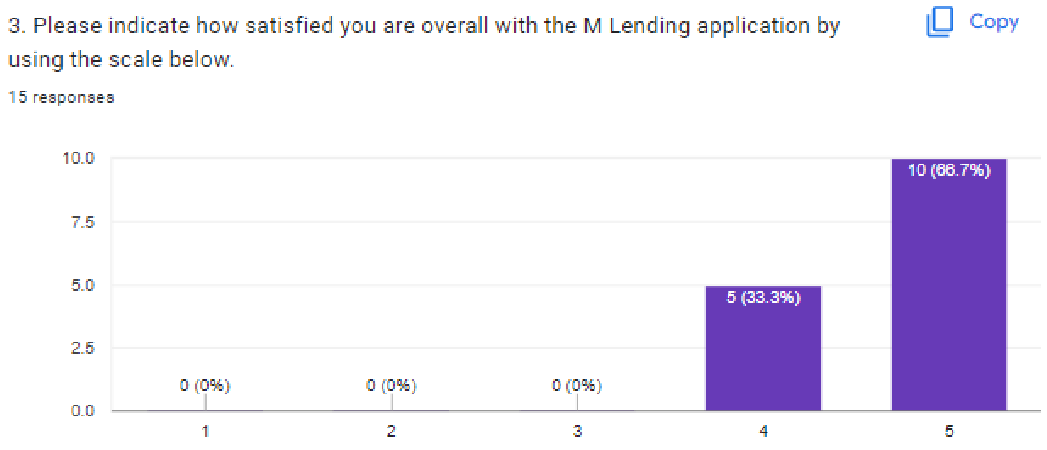
Figure 6.1. 1 Name of all respondents



2. Did the splash screen of M Lending catch your attention ?

15 responses

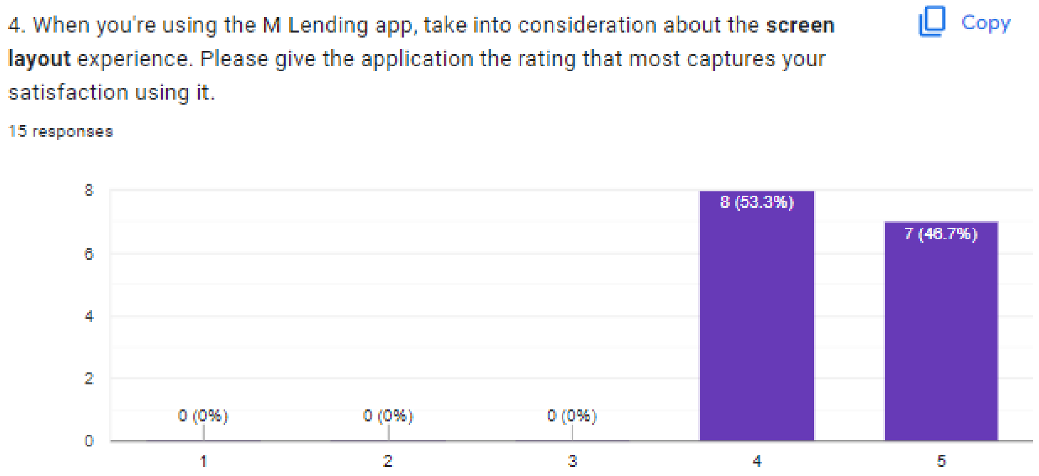
Figure 6.1. 2 Splash screen evaluation



3. Please indicate how satisfied you are overall with the M Lending application by using the scale below.

15 responses

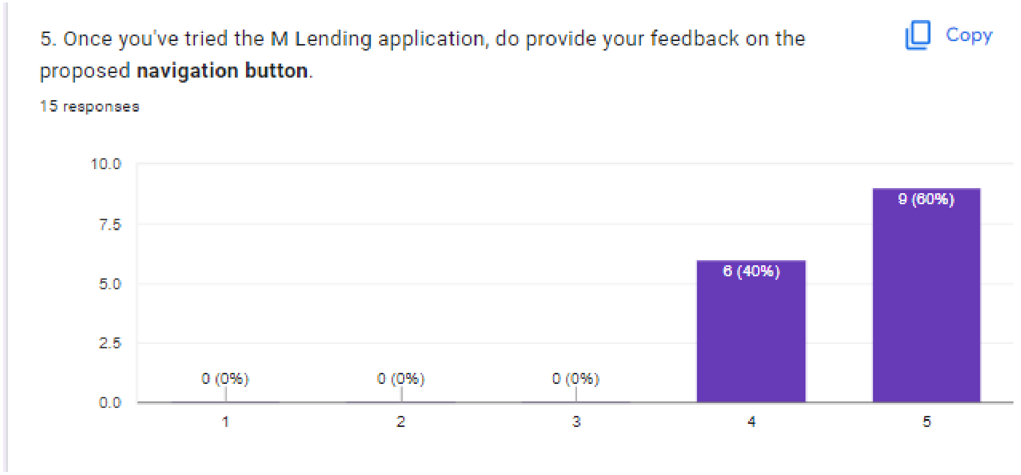
Figure 6.1. 3 Overall Satisfaction Evaluation



4. When you're using the M Lending app, take into consideration about the **screen layout** experience. Please give the application the rating that most captures your satisfaction using it.

15 responses

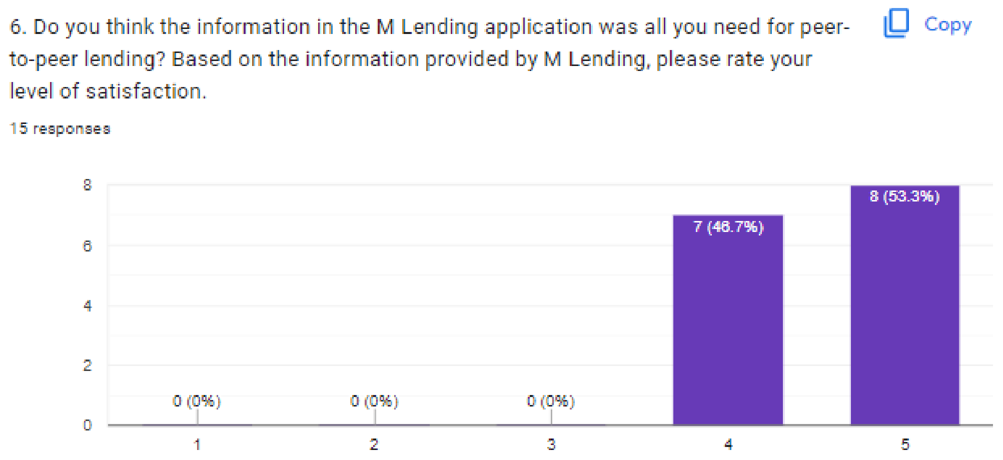
Figure 6.1. 4 Screen Layout Evaluation



5. Once you've tried the M Lending application, do provide your feedback on the proposed **navigation button**.

15 responses

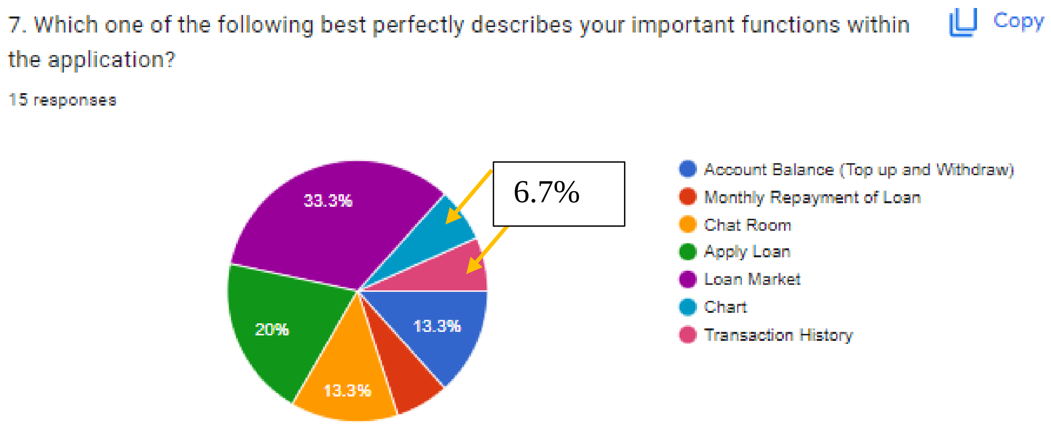
Figure 6.1. 5 Navigation Button Evaluation



6. Do you think the information in the M Lending application was all you need for peer-to-peer lending? Based on the information provided by M Lending, please rate your level of satisfaction.

15 responses

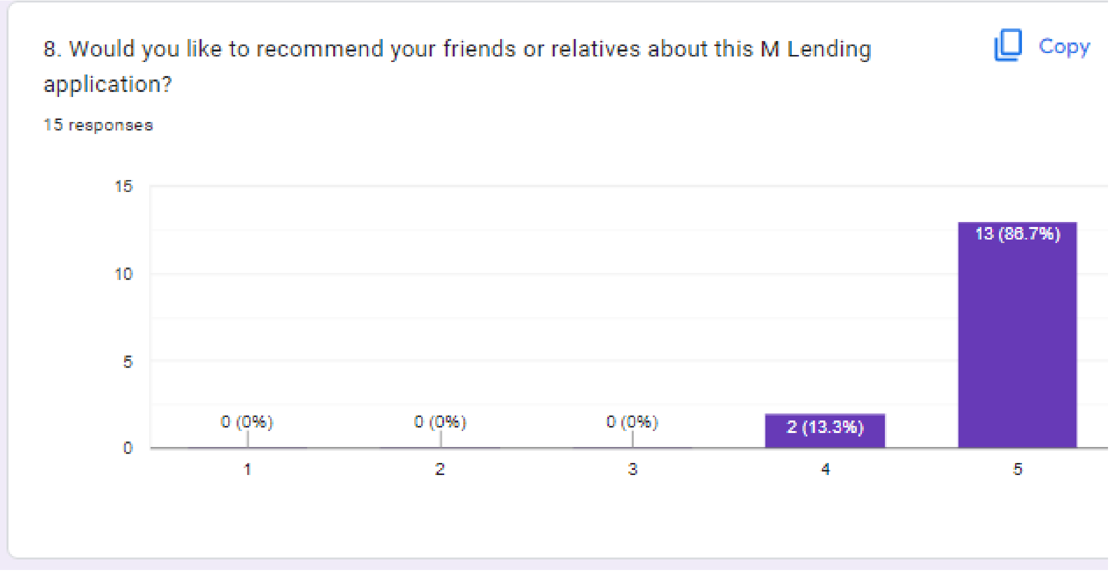
Figure 6.1. 6 Evaluation on information provided



7. Which one of the following best perfectly describes your important functions within the application?

15 responses

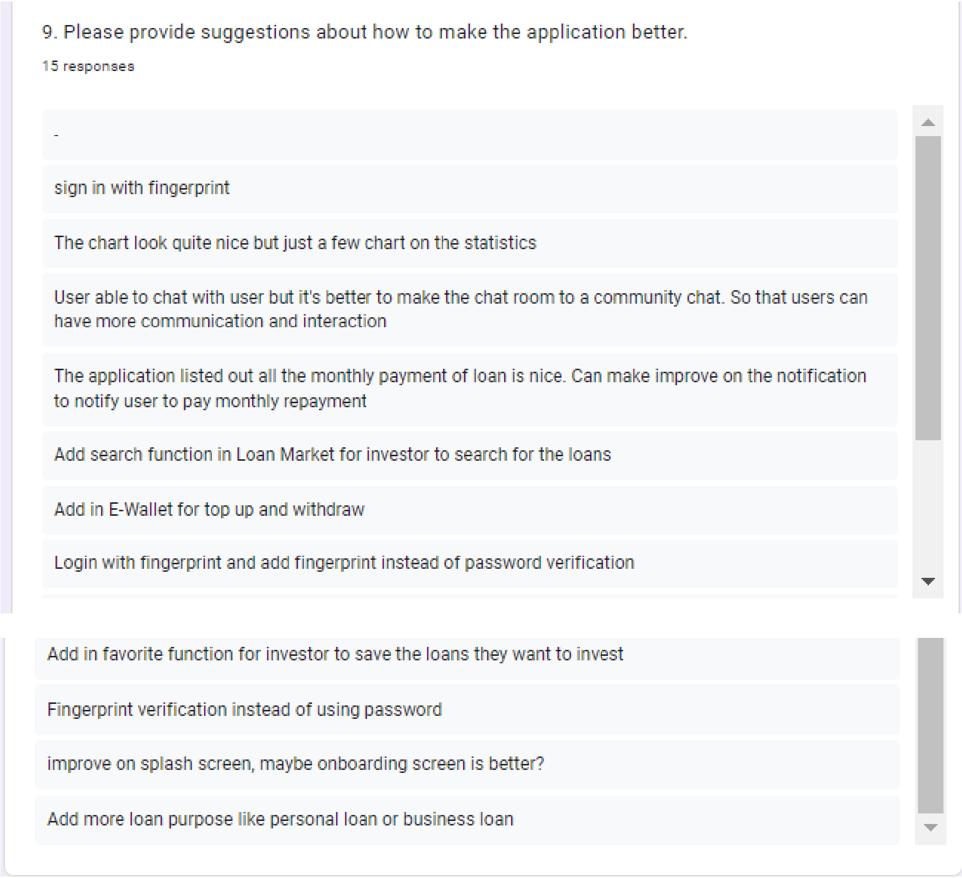
Figure 6.1. 7 Evaluation on Top feature



8. Would you like to recommend your friends or relatives about this M Lending application?

15 responses

Figure 6.1. 8 Recommendation opinion of respondents



9. Please provide suggestions about how to make the application better.

15 responses

-

sign in with fingerprint

The chart look quite nice but just a few chart on the statistics

User able to chat with user but it's better to make the chat room to a community chat. So that users can have more communication and interaction

The application listed out all the monthly payment of loan is nice. Can make improve on the notification to notify user to pay monthly repayment

Add search function in Loan Market for investor to search for the loans

Add in E-Wallet for top up and withdraw

Login with fingerprint and add fingerprint instead of password verification

Add in favorite function for investor to save the loans they want to invest

Fingerprint verification instead of using password

improve on splash screen, maybe onboarding screen is better?

Add more loan purpose like personal loan or business loan

Figure 6.1. 9 Recommendation

There are 15 respondents involved in this system evaluation survey, whose name are shown in Figure 6.1.1. Based on Figure 6.1.2, 93.3% respondents feel that the splash screen is catching their attention and only 6.7% of respondents think that the splash screen need to make improvement. This evaluation of splash screen is to know that whether the splash screen provided enough information and can it attract user to use the application. Based on Figure 6.1.3, 66.7% of respondents give 5 marks on the satisfaction for this application and 33.3% of respondents give 4 marks. From the results shown in Figure 6.1.4, 46.7% of respondents were very clear and satisfied for the screen layout experience of this application and 53.3% of respondents give 4 marks on this. 9 respondents think that the proposed navigation buttons were very clear and 6 of the respondents thought that the navigation buttons provided is clear only (refer to Figure 6.1.5). All of the respondents were satisfied and extremely satisfied for the lending information provided in this application, 53.3% of respondents are extremely satisfied while 46.7% are satisfied (refer to Figure 6.1.6). Based on the result in Figure 6.1.7, the ranking for all features from highest to lowest are Loan Market, Apply Loan, Chat Room, Account Balance, Transaction history, Monthly Repayment Loan and chart, which followed by 33.3%, 20%, 13.3%, 13.3%, 6.7%, 6.7% and 6.7% correspondingly. Figure 6.1.8 shows that 13 respondents are very likely to recommend this application while 2 respondents are likely to recommend it. Based on the results shown in Figure 6.1.9, the respondents suggested to add in fingerprint function, E-wallet function, notification function, favourite function, improve on search, splash screen, loan purpose and chart

6.2 Testing Setup and Result

6.2.1 Unit Testing 1 – Login

Objective: To ensure user to login with valid username and password

Input	Expected Output	Actual Output
Login by entering the current username and password.	The system verified user and allow user to login.	The user login successfully.
Login by entering incorrect username or password.	The system rejects the login request.	The user cannot log into the system.
Login with blank input.	The system focuses on the input column and request user to insert value	The user cannot log into the system.
Click "Forgot Password".	Redirect user to forgot password page to reset password.	The user is able to reset password.
Click "Forgot Password".	The system redirects user to register page.	The user is able to register an account.

Table 6.2. 1 Unit Testing 1 - Login

6.2.2 Unit Testing 2 – Register

Objective: To ensure user is able to register an account with valid input.

Input	Expected Output	Actual Output
Filled in all the edit text fields with valid input.	The registration process was successfully.	The user registered account successfully.
Filled in all the edit text fields with invalid input.	The registration process was unsuccessfully.	The user failed to register an account.
Register with blank input.	The system focuses on the input column and request user to insert value	The user cannot register an account.

Table 6.2. 2 Unit Testing 2 - Register

6.2.3 Unit Testing 3 – Main Menu

Objective: To ensure the user can use the functionalities provided.

Input	Expected Output	Actual Output
Click the logout icon beside the profile picture	The system allows user to logout account.	The user can logout the account.
Click the loan button	The system redirects the user to Loan Page	The user is able to go to the Loan Page.
Click the Profile button	The system redirects the user to Profile Page	The user is able to go to the Profile Page.
Click the chat button	The system redirects the user to Chat Page	The user is able to go to the Chat Page.
Click the chart button	The system redirects the user to Chart Page	The user is able to go to the Chart Page.
Click the transaction bottom navigation button	The system redirects the user to Transaction History Page	The user is able to go to the Transaction History Page.

Table 6.2. 3 Unit Testing 3 - Main Menu

6.2.4 Unit Testing 4 - Profile

Objective: To ensure user can view account balance and account details

Input	Expected Output	Actual Output
Click the upload button	The system allow user to upload profile picture	The user can upload new profile picture.
Click the Account Balance View	The system redirects the user to Account Balance Page	The user is able to go to the Account Page.
Click the verify email address button	The system sends	The user is able to go to the Profile Page.
Click the chat button	The system redirects the user to Chat Page	The user is able to go to the Chat Page.
Click the chart button	The system redirects the user to Chart Page	The user is able to go to the Chart Page.
Click the transaction bottom navigation button	The system redirects the user to Transaction History Page	The user is able to go to the Transaction History Page.

Table 6.2. 4 Unit Testing 4 - Profile

6.2.5 Unit Testing 5 – Top Up

Objective: To ensure user can top up account balance

Input	Expected Output	Actual Output
Insert the top up amount and click top up button	The system pops out a password verification dialog	The user is able to insert password in the pop out dialog
Insert password in password verification dialog edit text column and click ok button	The system checks password validation and close the dialog	The user is able to top up account balance.
Click the back button	The system redirects user to the Account Balance page	The user is able to go Account Balance page.

Table 6.2. 5 Unit Testing 5 - Top Up

6.2.6 Unit Testing 6 – Withdraw

Objective: To ensure user able to withdraw account balance

Input	Expected Output	Actual Output
Insert the withdraw amount and click withdraw button	The system pops out a password verification dialog	The user is able to insert password in the pop out dialog
Insert password in password verification dialog edit text column and click ok button	The system checks password validation and account balance amount.	The user is able to withdraw account balance.
Click the back button	The system redirects user to the Account Balance page	The user is able to go Account Balance page.

Table 6.2. 6 Unit Testing 6 - Withdraw

6.2.7 Unit Testing 7 – Apply Loan

Objective: To ensure user can make a loan application

Input	Expected Output	Actual Output
Check loan rate by Inserting loan amount and select loan purpose	The system verified input validation and redirects user to check rate page	The user is able to go to check rate page
Check loan rate without inserting any input	The system focuses on the input column and request user to insert value	The user is unable to go to check rate page
Click on list view item	The system redirects user to Loan View Page that display related list view item details	The user able to go to Loan View Page.

Table 6.2. 7 Unit Testing 7 - Apply Loan

6.2.8 Unit Testing 8 – Check Rate

Objective: To ensure user can check actual loan rate

Input	Expected Output	Actual Output
Select loan duration from the month drop down list	The system gets the month duration and calculate actual loan rate	The user is able to view the actual loan rate
Click the get rate button	The system pops out a password verification dialog	The user is able to key in password and go to Loan View Page
Click another loan button	The system redirects user to another loan page	The user is able to go to another loan page

Table 6.2. 8 Unit Testing 8 - Check Rate

6.2.9 Unit Testing 9 – Apply Another Loan

Objective: To ensure user able to apply loan with loan market

Input	Expected Output	Actual Output
Get loan by inserting loan amount, loan purpose, duration, interest rate	The system verified input validation and pop out a password verification dialog	User able to get loan and key in password
Get loan with blank input	The system focuses on blank input columns	User unable to get loan
Insert correct password in password verification dialog	The system redirect user to Apply Loan Page	The user able to go to Apply Loan Page

Table 6.2. 9 Unit Testing 9 - Another Loan

6.2.10 Unit Testing 10 – Loan Repayment

Objective: To ensure user able to pay his/her repayment by monthly

Input	Expected Output	Actual Output
Pay loan repayment by clicking pay button	The system redirect user to payment page	User able to go to payment page
Click confirm button to perform the payment action	The system checks user's account balance and pop out a password verification dialog	User able to key in password
Insert correct password in the dialog	The system verifies the password and redirect user to loan repayment list page	User able to go to loan repayment list page
Insert incorrect password in the dialog	The system detects incorrect password and toast "incorrect password" message	User failed to pay the loan

Table 6.2. 10 Unit Testing 10 - Loan Repayment

6.2.11 Unit Testing 11 – Chat

Objective: To ensure that user can use chat function normally

Input	Expected Output	Actual Output
Insert username and click search button	The system displays related user's chat room	User able to search user by username
Click the all button	The system displays all user's chat room	User able to view all chat room
Click the back button	The system redirect user to main menu page	User able to go to main menu page
Click on user's chat room	The system redirect user to related user's message page	User able to go to message page and chat with other user

Table 6.2. 11 Unit Testing 11 - Chat

6.2.12 Unit testing 12 – Investment

Objective: To ensure user can view loan application in loan market

Input	Expected Output	Actual Output
Filter the loan market list by selecting loan purpose	The system allows results matching the filtered item to be displayed	User can view result showing loan purpose that matched the filtered item
Click any of the loan	The system redirect user to Market View Page of related loan	User can go to Market View Page that displayed the related loan details

Table 6.2. 12 Unit Testing 12 - Investment

6.2.13 Unit Testing 13 – Market View

Objective: To ensure user can invest on loan application

Input	Expected Output	Actual Output
Click invest this loan button	The system pops out a password verification dialog	User can key in password in the dialog
Key in correct password in password verification dialog	The system verifies password and display "Invest successful" message	User able to go to Investment page
Click contact the borrower button	The system redirect user to Chat page	User able to go to Chat page
Click back button	The system redirect user to Investment page	User able to go to Investment page

Table 6.2. 13 Unit Testing 13 - Market View

6.2.14 Unit Testing 14 – Transaction

Objective: To ensure user can view transaction history

Input	Expected Output	Actual Output
Click the transaction icon from the navigation bar	The system redirect user to Transaction page displayed all transaction history	User able to view all transaction history

Table 6.2. 14 Unit Testing 14 - Transaction

6.2.15 Unit Testing 15 – Admin Menu

Objective: To ensure user(admin – platform manager) to use the functionalities provided

Input	Expected Output	Actual Output
Click logout button	The system log user out of the account	The user able to log out his/her account
Click loan list button	The system redirect user to loan list page	The user able to go to loan list page
Click user list button	The system redirect user to user list page	The user able to go to user list page
Click chart button	The system redirect user to chart page	The user able to go to chart page
Click black list button	The system redirect user to black list page	The user able to go to black list page

Table 6.2. 15 Unit Testing 15 - Admin Menu

6.2.16 Unit Testing 16 – Loan List

Objective: To ensure user (admin) can view all the loan application

Input	Expected Output	Actual Output
Select loan purpose from the filter	The system displays loan application that matching the selected item from filter	User able to search loan application by loan purpose
Click the loan application view	The system redirect user to loan list view page and display related loan details	User able to go to loan list view page and view loan details

Table 6.2. 16 Unit Testing 16 - Loan List

6.2.17 Unit Testing 17- Loan List View

Objective: To ensure user (admin) can edit and update the user's membership status

Input	Expected Output	Actual Output
Click review button	The system changes loan status and redirect user to loan list page	User able to update loan status and go to loan list page
Click reject button	The system changes loan status and redirect user to loan list page	User able to reject loan and go to loan list page
Click back button	The system redirect user to loan list page	User able to go to loan list page

Table 6.2. 17 Unit Testing 17 - Loan List View

6.2.18 Unit Testing 18 – User List

Objective: To ensure user (admin) can view all user information

Input	Expected Output	Actual Output
Select membership status from the filter	The system displays users that matching the selected item from filter	User able to search users by loan purpose
Click the user view	The system redirect user to user list view page and display related user information	User able to go to user list view page and view user details

Table 6.2. 18 Unit Testing 18 - User List

6.2.19 Unit testing 19 – User List View

Objective: To ensure user (admin) can edit and update all user's membership status

Input	Expected Output	Actual Output
Click review button	The system changes membership registration status and redirect user to user list page	User able to update membership status and go to user list page
Click reject button	The system changes membership status and redirect user to user list page	User able to reject membership registration and go to user list page
Click back button	The system redirect user to user list page	User able to go to user list page

Table 6.2. 19 Unit Testing 19 - User List View

6.2.20 Unit Testing 20 – Black List

Objective: To ensure user (admin) able to view blacklist and add new username to blacklist

Input	Expected Output	Actual Output
Insert username and click submit button	The system adds new username in to black list	User able to add new black list username to the black list
Click the back button	The system redirect user to Admin Menu Page	User able to go to Admin Menu Page

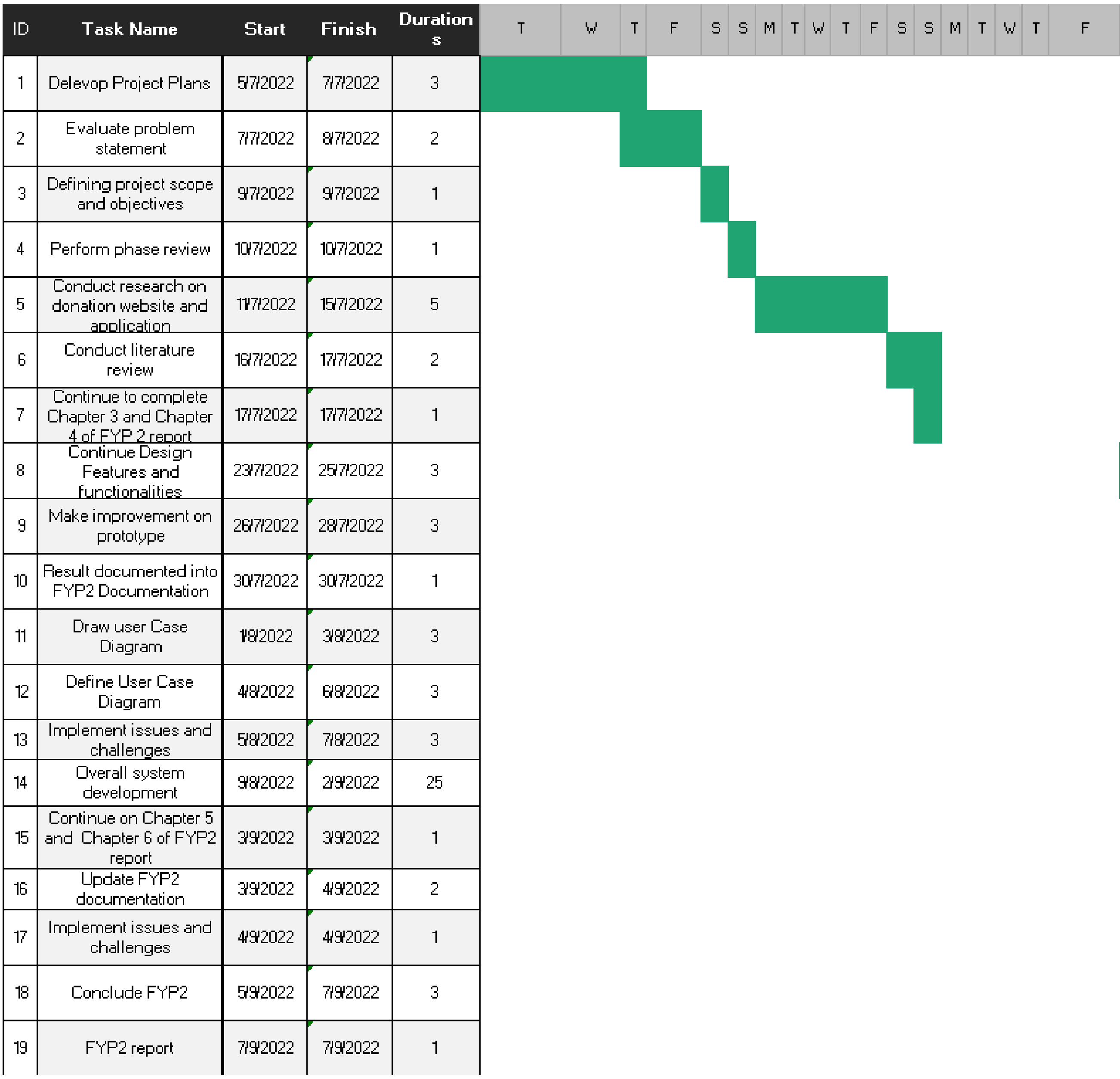
Table 6.2. 20 Unit Testing 20 - Black List

6.3 Implementation Issues and Challenges

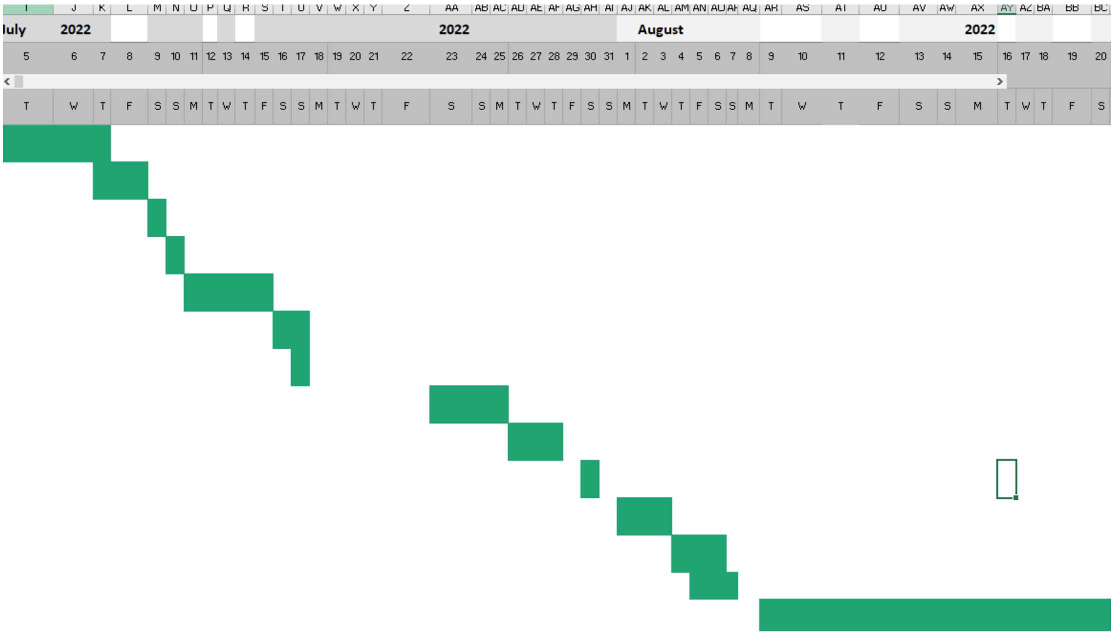
There are some issues and challenges faced during the development and implementation of this project, Fintech peer-to-peer lending mobile application. The development of this project involved two main technology and knowledge that I am highly unfamiliar with, which is financial technology and mobile application development. I spend more time and efforts to learn fintech and mobile application development.

The challenges that I faced currently was the trustworthy of people through peer-to-peer lending. In my country, peer-to-peer lending were still considered not to have reached such a mature level because the people afraid of using new thing. Therefore, in my country, there are not many peer-to-peer platform or company. However, peer-to-peer lending was accepted by majority of residents in US, Singapore, China, UK as an alternative way to get loan and as an investment platform to get extra income. So, I believed that my country will accept peer-to-peer lending start from this peer-to-peer mobile application.

6.4 Timeline



ID	Task Name	Start	Finish	Durations	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
1	Delevop Project Plans	5/7/2022	7/7/2022	3	🟩	🟩	🟩															
2	Evaluate problem statement	7/7/2022	8/7/2022	2			🟩	🟩														
3	Defining project scope and objectives	9/7/2022	9/7/2022	1					🟩													
4	Perform phase review	10/7/2022	10/7/2022	1						🟩												
5	Conduct research on donation website and application	11/7/2022	15/7/2022	5							🟩	🟩	🟩	🟩	🟩							
6	Conduct literature review	16/7/2022	17/7/2022	2												🟩	🟩					
7	Continue to complete Chapter 3 and Chapter 4 of FYP 2 report	17/7/2022	17/7/2022	1													🟩					
8	Continue Design Features and functionalities	23/7/2022	25/7/2022	3																		
9	Make improvement on prototype	26/7/2022	28/7/2022	3																		
10	Result documented into FYP2 Documentation	30/7/2022	30/7/2022	1																		
11	Draw user Case Diagram	1/8/2022	3/8/2022	3																		
12	Define User Case Diagram	4/8/2022	6/8/2022	3																		
13	Implement issues and challenges	5/8/2022	7/8/2022	3																		
14	Overall system development	9/8/2022	2/9/2022	25																		
15	Continue on Chapter 5 and Chapter 6 of FYP2 report	3/9/2022	3/9/2022	1																		
16	Update FYP2 documentation	3/9/2022	4/9/2022	2																		
17	Implement issues and challenges	4/9/2022	4/9/2022	1																		
18	Conclude FYP2	5/9/2022	7/9/2022	3																		
19	FYP2 report	7/9/2022	7/9/2022	1																		


July 2022
2022
August
2022
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S

6.5 Objective Evaluation

The suggested system effectively accomplishes its goals by expediting the loan application process and offering best peer-to-peer lending services to both borrowers and lenders. The suggested solution allows users who want to get a loan without completing the necessary paper work and waiting in line at a commercial bank to do it more quickly. Users may immediately take use of all peer-to-peer lending services offered by this proposed system, regardless of time or location, after having their membership and account registration validated. The suggested approach effectively enables all users to apply for loans and invest in loans. The processing of the loan review only takes two to three business days. The suggested approach enables customers to swiftly get loans for a variety of uses. The suggested approach offered cheap interest rates on each loan application and more opportunities for investors to invest on borrowers' loans. In addition, after users have entered the loan amount and loan purpose, the suggested system may automatically compute the loan rate, including interest rate and monthly payback amount. The suggested system had two crucial modules, namely the conversation module and the loan market module, in an effort to entice more investors to participate in loans. While the loan market module gives investors the freedom to choose loans, the chat module makes it simple for investors to communicate with borrowers.

7 Chapter 7 Conclusion and Recommendation

7.1 Conclusion

The proposed project is a financial technology lending mobile application. Today, the global peer-to-peer lending market has grown significantly in 2015-2020. This is because when compared to traditional funding options, P2P lending is fast, paperless, offers higher returns, has no impact on the credit rating and is more flexible. However, P2P lending was not as familiar to our country Malaysia. So, the objective I proposed this project is to make it possible for Malaysians to make themselves known in terms of peer-to-peer lending.

There are three main ***problems*** stated which are the complicated loan application process, strict eligibility requirements and lack of qualitative investment opportunity. The preparation of a commercial loan application has a long and lengthy process. Even though the Malaysia's commercial bank have provided loan application services via online but the applicant still needs to submit the loan documentation to the branches of the bank. This means that the applicant must always visit the bank branches while the loan application is being processed and the applicant is required to prepare sufficient documentation each time. If there are missing documents, the application process will be postponed until the applicant has prepared sufficient documents. Based on this problem, the proposed project can accelerate the application process by completing the loan application process fully online and mobile. All necessary documentation can simply be submitted through the proposed project by uploading the documentation and the management team will review the documentation for authenticity. The second problem that commercial bank has strict eligibility requirements for loan application. Credit rating is a major issue in obtaining a commercial bank loan. The proposed project was an alternative lending platform for applicants who are not successful in obtaining a traditional loan and allow a greater number of eligible individuals to obtain loans. The eligibility criteria were more flexible than those of commercial banks, because there are many lenders on that platform, but commercial banks are the only lender. When there are multiple lenders, the eligibility criteria will be different for individuals, so the applicant has the option of obtaining a low interest loan with a minimum of eligibility criteria. Additionally, this proposed project can also address the lack of quality investment opportunities. For commercial banks, any loan they make can be considered a bank investment. The bank can get financial return from borrower. Therefore, providing a loan was an investment by the lender or investor to the borrower. That was some sort of investment opportunity. Fintech peer-to-peer lending brings diversity and a broad range of investors and borrowers on a unique platform to match the offer and demand. The lender and invest can have more opportunity to seek for an investment on loan in the loan market of this platform while the borrower has more chances to get a loan. It was a win-win situation for both the lender and the borrower, who both have what they need.

Besides, I have researched and analyzed on the other peer-to-peer lending platform such as Sofi, LendingCrowd, Funding Societies, Zopa and LendingClub. I did highlight the functionalities, features, strengths, and weaknesses of these peer-to-peer lending platform together with my proposed project. I conducted a comparison table after analyzing all these existing P2P lending platforms. I have a clear idea about the functionalities and features that should implement in my proposed project This can end up helping proposed project become more competitive in the marketplace. I planned to prepare a near-complete prototype to test and get feedback from them in the final year project 2 to find out the weaknesses of the proposed project and make improvements. The methodology used in proposed project is prototype model which consisted five steps such as requirement gathering, quick design, build a prototype, user evaluation of prototype and refine prototype, implementation and testing.

In summary, the proposed project will be able to resolve all of the issues identified in the Problem Statement. The proposed project will also include the features and functionality identified in the comparison table.

7.2 Recommendation

The system evaluation survey is used to identify room for improvement in the proposed system. The proposed system was an early completed prototype that provided basic functionalities. Different users will have different viewpoints and the viewpoints can help developer to understand the weaknesses of the proposed project. Thus, the developer can make improvement by adding new features in proposed project in the future. According to the evaluation survey results, fingerprint, notification and favorite are the features that the most respondents recommend to include in the system. For fingerprint feature, the respondents suggest to add in as a verification like password verification. Respondents also suggest to login in the application with fingerprint as it is convenience compared to key in username and password to login. For the notification feature, respondents suggest to add in to notify user the loan repayment. While the favorite feature is for the investor to bookmark the loan application in Loan Market. So, the investor can always find back the loan application they want. All of these suggestions will be considered in future update versions of the proposed system.

8 REFERENCES

[1] Bessette, C. (2022). *SoFi Banking (formerly SoFi Money) review - NerdWallet*. [online] Available at: <https://www.nerdwallet.com/article/banking/sofi-money-review> [Accessed 5 February 2022].

[2] GPayments. (2022). *Strong Customer Authentication under PSD2 guidelines.* [Online] Available at: https://www.gpayments.com/about/psd2-strong-customer-authentication-3d-secure-2/#:~:text=Strong%20customer%20authentication%20(SCA)%20is,(something%20the%20user%20is) [Accessed 09 February 2022].

[3] Malaymail. (2021). *Banks in Malaysia to remain cautious on loan approvals*. [Online] Available at: https://www.malaymail.com/news/money/2021/05/03/banks-in-malaysia-to-remain-cautious-on-loan-approvals/1971216 [Accessed 02 February 2022].

[4] Miller, D. & Houston, R. (2022). *SoFi Invest review: A worthwhile app for beginners without a ton of money to invest.* [Online] Available at: https://www.businessinsider.com/personal-finance/sofi-invest-review [Accessed 04 February 2022].

[5] Philps, R. (2021). *Bank Loan Advantages and Disadvantages for Small Businesses*. [Online] Available at: https://www.nerdwallet.com/uk/business-loans/pro-and-cons-of-bank-loans-for-small-businesses/#what-are-the-disadvantages-of-bank-loans [Accessed 03 February 2022].

[6] Renton, P. (2011). *An Introduction to the Lending Club Trading Platform.* [Online] Available at: https://www.lendacademy.com/an-introduction-to-the-lending-club-trading-platform/ [Accessed 10 February 2022].

[7] The Star. (2021). *Housing loan approval rate on the way up.* [Online] Available at: https://www.thestar.com.my/business/business-news/2021/07/07/housing-loan-approval--rate-on-the-way-up [Accessed 02 February 2022].

[8] Tretina, K. (2022). *Peerform allows eligible applicants with fair credit to borrow up to $25,000.* [Online] Available at: https://www.investopedia.com/peerform-personal-loans-review-4842542 [Accessed 05 February 2022].

[9] Tretina, K. (2022). *Best Peer-to-Peer Lending.* [Online] Available at: https://www.investopedia.com/articles/investing/092315/7-best-peertopeer-lending-websites.asp [Accessed 05 February 2022].

[10] Trotman, J. (2020). *Recommendations For P2P Mobile Payment App Development.* [Online] Available at: https://www.nimbleappgenie.com/blogs/recommendations-for-p2p-mobile-payment-app-development/ [Accessed 04 February 2022].

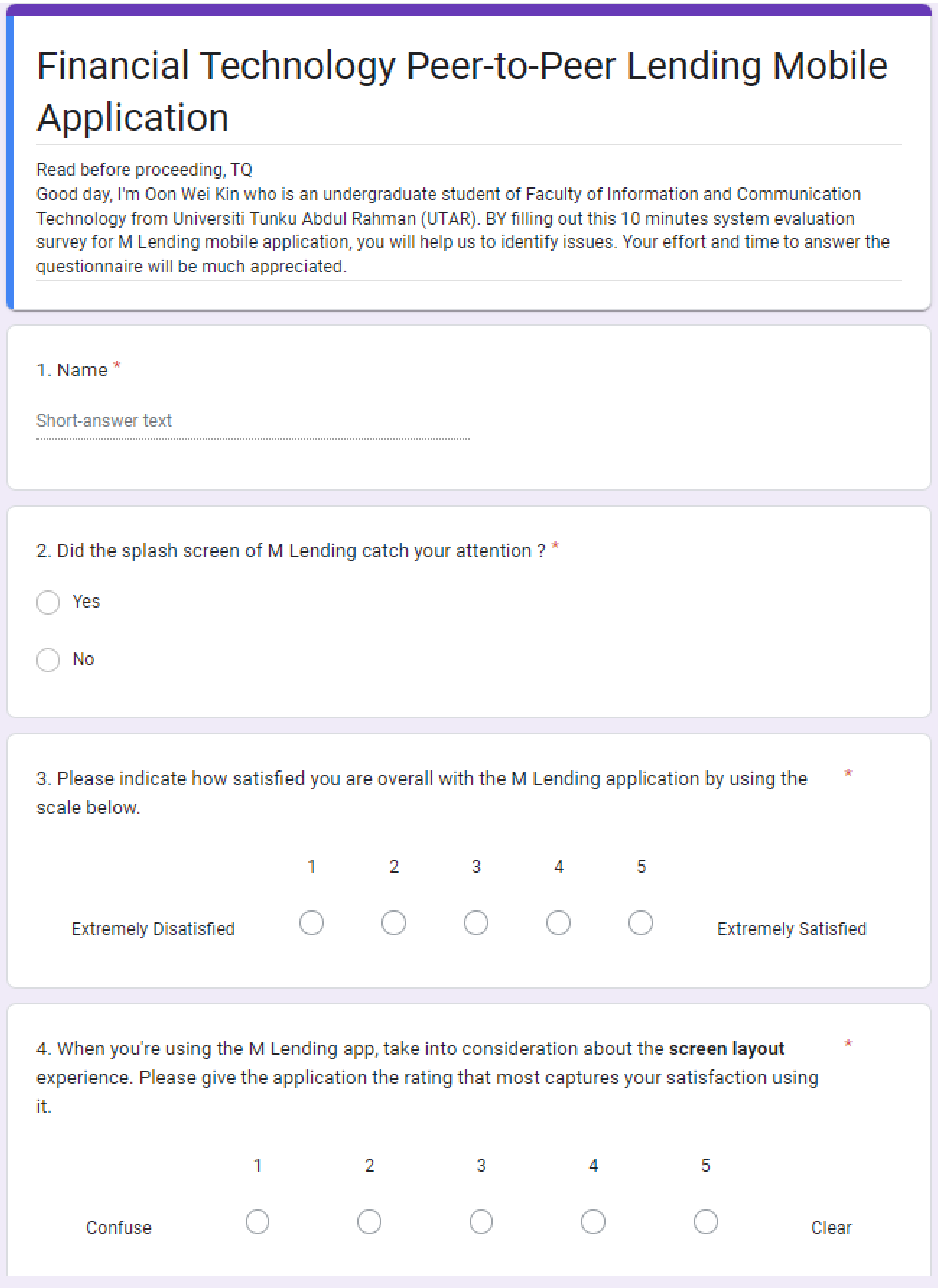
[11] Trontman, J. (2021). *The Only Features You Need In Your Peer to Peer Lending App.* [Online] Available at: https://www.nimbleappgenie.com/blogs/the-only-features-you-need-in-your-peer-to-peer-lending-app/ [Accessed 03 February 2022].

[12] Wilbert, S. (2022). *Sofi Review: Is it the Best for Advanced Banking Products?* [Online] Available at: https://www.dumblittleman.com/sofi-review/ [Accessed 06 February 2022].

[13] Zinn, Dori. (2021). *What is peer-to-peer lending.* Available at: https://www.bankrate.com/loans/personal-loans/peer-to-peer-lending-vs-bank-loan/ [Accessed 04 February 2022].

9 APPENDIX

9.1 System Evaluation Survey Form



Financial Technology Peer-to-Peer Lending Mobile Application

Read before proceeding, TQ
Good day, I'm Oon Wei Kin who is an undergraduate student of Faculty of Information and Communication Technology from Universiti Tunku Abdul Rahman (UTAR). BY filling out this 10 minutes system evaluation survey for M Lending mobile application, you will help us to identify issues. Your effort and time to answer the questionnaire will be much appreciated.

1. Name *

Short-answer text

2. Did the splash screen of M Lending catch your attention ? *

○ Yes

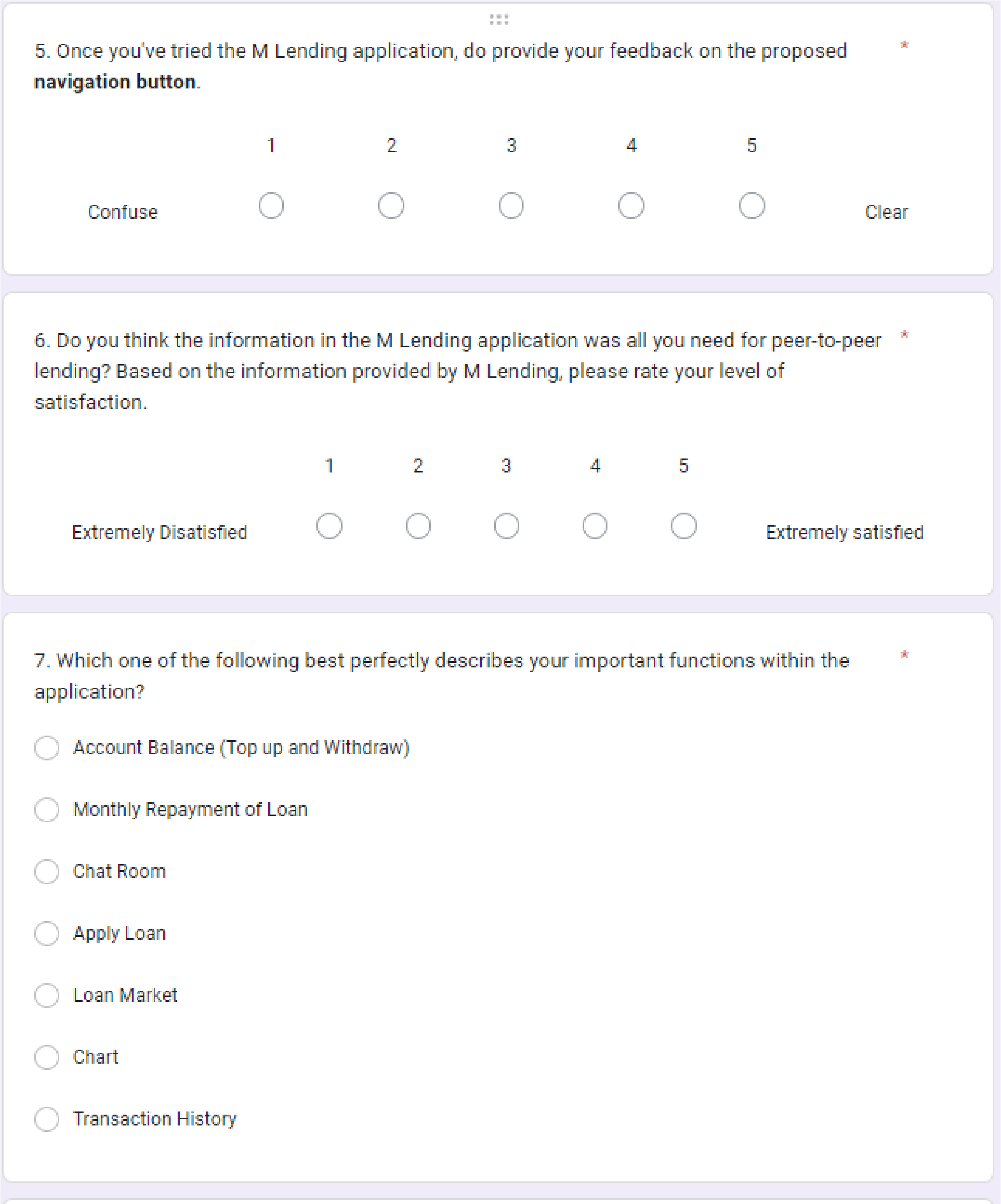
○ No

3. Please indicate how satisfied you are overall with the M Lending application by using the scale below. *

	1	2	3	4	5	
Extremely Disatisfied	○	○	○	○	○	Extremely Satisfied

4. When you're using the M Lending app, take into consideration about the **screen layout** experience. Please give the application the rating that most captures your satisfaction using it. *

	1	2	3	4	5	
Confuse	○	○	○	○	○	Clear



5. Once you've tried the M Lending application, do provide your feedback on the proposed **navigation button**. *

	1	2	3	4	5	
Confuse	○	○	○	○	○	Clear

6. Do you think the information in the M Lending application was all you need for peer-to-peer lending? Based on the information provided by M Lending, please rate your level of satisfaction. *

	1	2	3	4	5	
Extremely Disatisfied	○	○	○	○	○	Extremely satisfied

7. Which one of the following best perfectly describes your important functions within the application? *

○ Account Balance (Top up and Withdraw)

○ Monthly Repayment of Loan

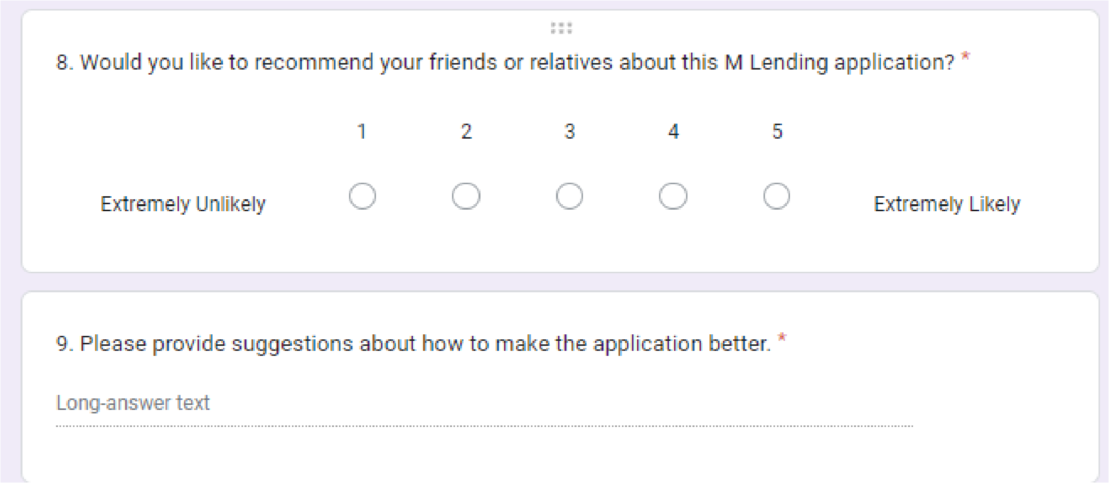
○ Chat Room

○ Apply Loan

○ Loan Market

○ Chart

○ Transaction History



8. Would you like to recommend your friends or relatives about this M Lending application? *

	1	2	3	4	5	
Extremely Unlikely	○	○	○	○	○	Extremely Likely

9. Please provide suggestions about how to make the application better. *

Long-answer text

9.2 FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 4**
Student Name & ID: Oon Wei Kin 18ACB05279	
Supervisor: Mr. Su Lee Seng	
Project Title: Fintech Peer-to-Peer Lending mobile application development	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Completed Chapter 1: Introduction and Chapter 2: Literature Review in FYP2 report.

2. WORK TO BE DONE

Chapter 3: Proposed Method/Approach and determine the design and functionalities of proposed project.

3. PROBLEMS ENCOUNTERED

No problem encounter in week 4.

4. SELF EVALUATION OF THE PROGRESS

The starting progress is smooth as I have completed Chapter 1 and Chapter 2 before week 4.

SuLee

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 6**
Student Name & ID: Oon Wei Kin 18ACB05279	
Supervisor: Mr. Su Lee Seng	
Project Title: Fintech Peer-to-Peer Lending mobile application development	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Complete Chapter 3 System Methodology/Approach for FYP2 report.

2. WORK TO BE DONE

- Continue to develop prototype, coding for the rest of the features like chat room, loan market, statistics, upload profile picture, top up, withdraw, and payment.

- Continue to work on Chapter 4 System Design for FYP2 report.

3. PROBLEMS ENCOUNTERED

No problem encounter in week 6.

4. SELF EVALUATION OF THE PROGRESS

Still need to solve the bug and improve in the prototype development.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 8**
Student Name & ID: Oon Wei Kin 18ACB05279	
Supervisor: Mr. Su Lee Seng	
Project Title: Fintech Peer-to-Peer Lending mobile application development	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Done the development of investor module, the loan market, payment and account balance features.

-Completed Chapter 4 System Design for FYP2 report.

2. WORK TO BE DONE

- Continue to complete the Chapter 5 System Implementation & Chapter 6 System Evaluation in the report.

- Continue to explore how to code each function so that it can be functioning on backend.

-Continue to code the remaining features.

3. PROBLEMS ENCOUNTERED

No problem encounter in week 8.

4. SELF EVALUATION OF THE PROGRESS

Still need to learn more to develop a good peer-to-peer lending mobile application

SuLee

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 10**
Student Name & ID: Oon Wei Kin 18ACB05279	
Supervisor: Mr. Su Lee Seng	
Project Title: Fintech Peer-to-Peer Lending mobile application development	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Completed Chapter 5 System Implementation in the report.

-Completed the Admin module and the function of admin module for the prototype.

2. WORK TO BE DONE

-Make improvement on the mobile application base on supervisor's suggestion.
- Continue to complete the whole report (Chapter 6 and 7).

3. PROBLEMS ENCOUNTERED

No problem encounter in week 10.

4. SELF EVALUATION OF THE PROGRESS

Still need to learn more financial technology to integrate proposed project.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 12**
Student Name & ID: Oon Wei Kin 18ACB05279	
Supervisor: Mr. Su Lee Seng	
Project Title: Fintech Peer-to-Peer Lending mobile application development	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

- Completed the draft of FYP2 report.

- Completed the improvement of the mobile application.

- Completed the development of prototype.

-Completed the testing of prototype.

2. WORK TO BE DONE

- Ready to submit FYP 2 report
-Continue to make the protype more well, make improvement on some function.

3. PROBLEMS ENCOUNTERED

No problem encounter in week 12.

4. SELF EVALUATION OF THE PROGRESS

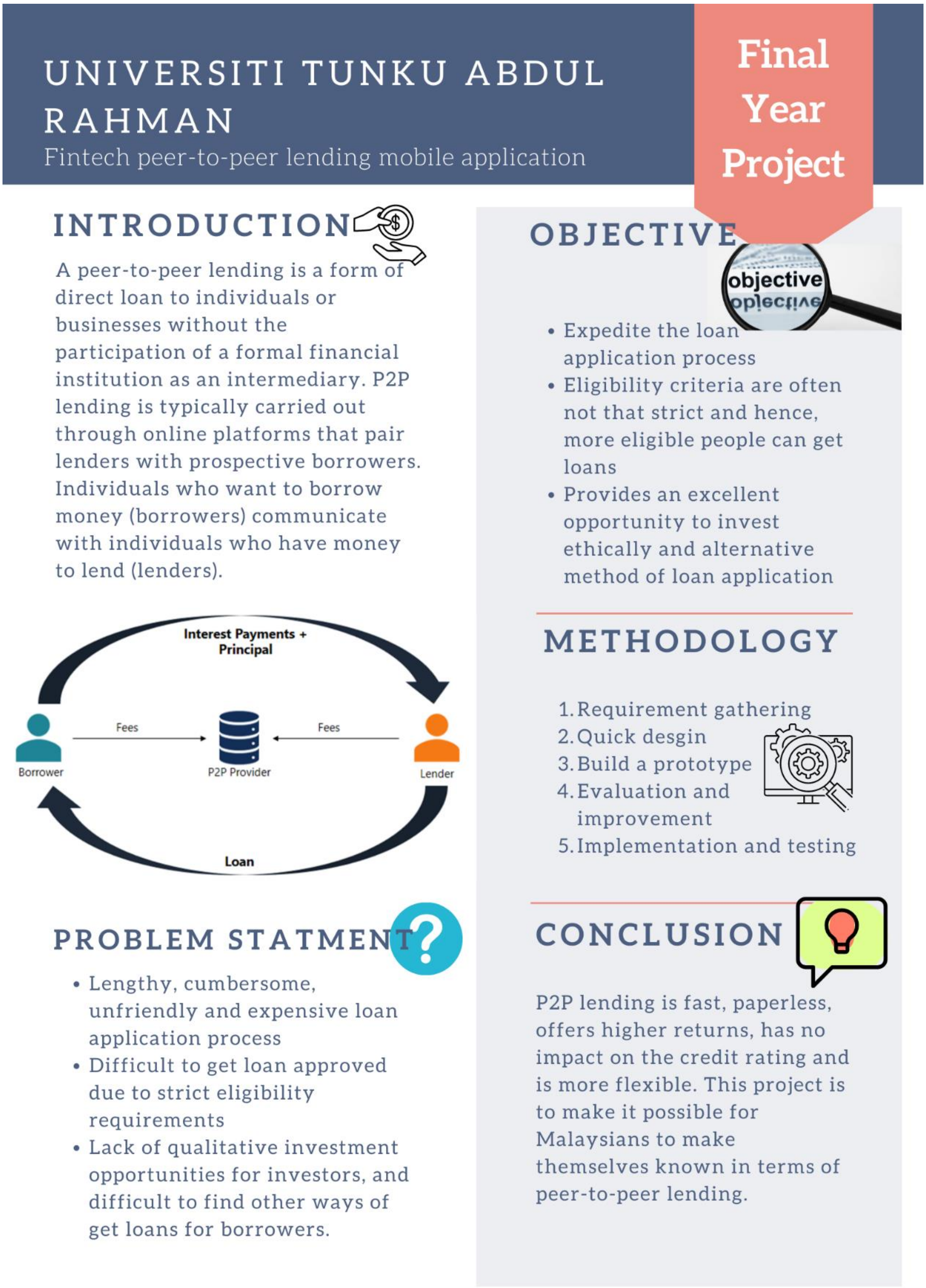
Still need to make some correction on FYP2 report writing and learn on how to develop a complete mobile application.

SuLee

Supervisor's signature

Student's signature

9.3 POSTER



UNIVERSITI TUNKU ABDUL RAHMAN

Fintech peer-to-peer lending mobile application

Final Year Project

INTRODUCTION

A peer-to-peer lending is a form of direct loan to individuals or businesses without the participation of a formal financial institution as an intermediary. P2P lending is typically carried out through online platforms that pair lenders with prospective borrowers. Individuals who want to borrow money (borrowers) communicate with individuals who have money to lend (lenders).

PROBLEM STATMENT

- Lengthy, cumbersome, unfriendly and expensive loan application process
- Difficult to get loan approved due to strict eligibility requirements
- Lack of qualitative investment opportunities for investors, and difficult to find other ways of get loans for borrowers.

OBJECTIVE

- Expedite the loan application process
- Eligibility criteria are often not that strict and hence, more eligible people can get loans
- Provides an excellent opportunity to invest ethically and alternative method of loan application

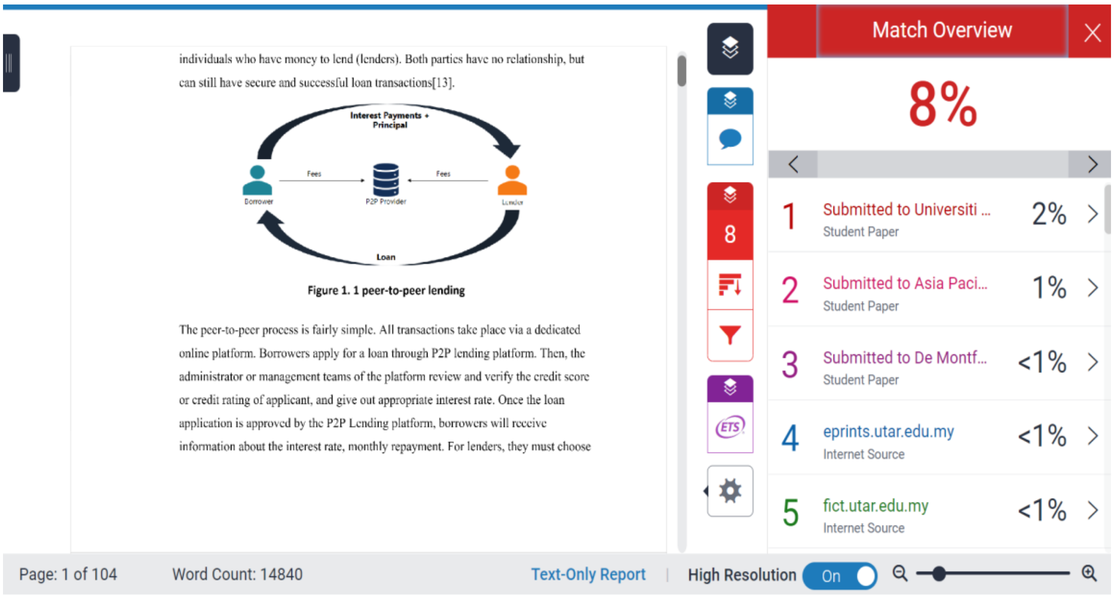
METHODOLOGY

1. Requirement gathering
2. Quick desgin
3. Build a prototype
4. Evaluation and improvement
5. Implementation and testing

CONCLUSION

P2P lending is fast, paperless, offers higher returns, has no impact on the credit rating and is more flexible. This project is to make it possible for Malaysians to make themselves known in terms of peer-to-peer lending.

10 PLAGIARISM CHECK RESULT



individuals who have money to lend (lenders). Both parties have no relationship, but can still have secure and successful loan transactions[13].

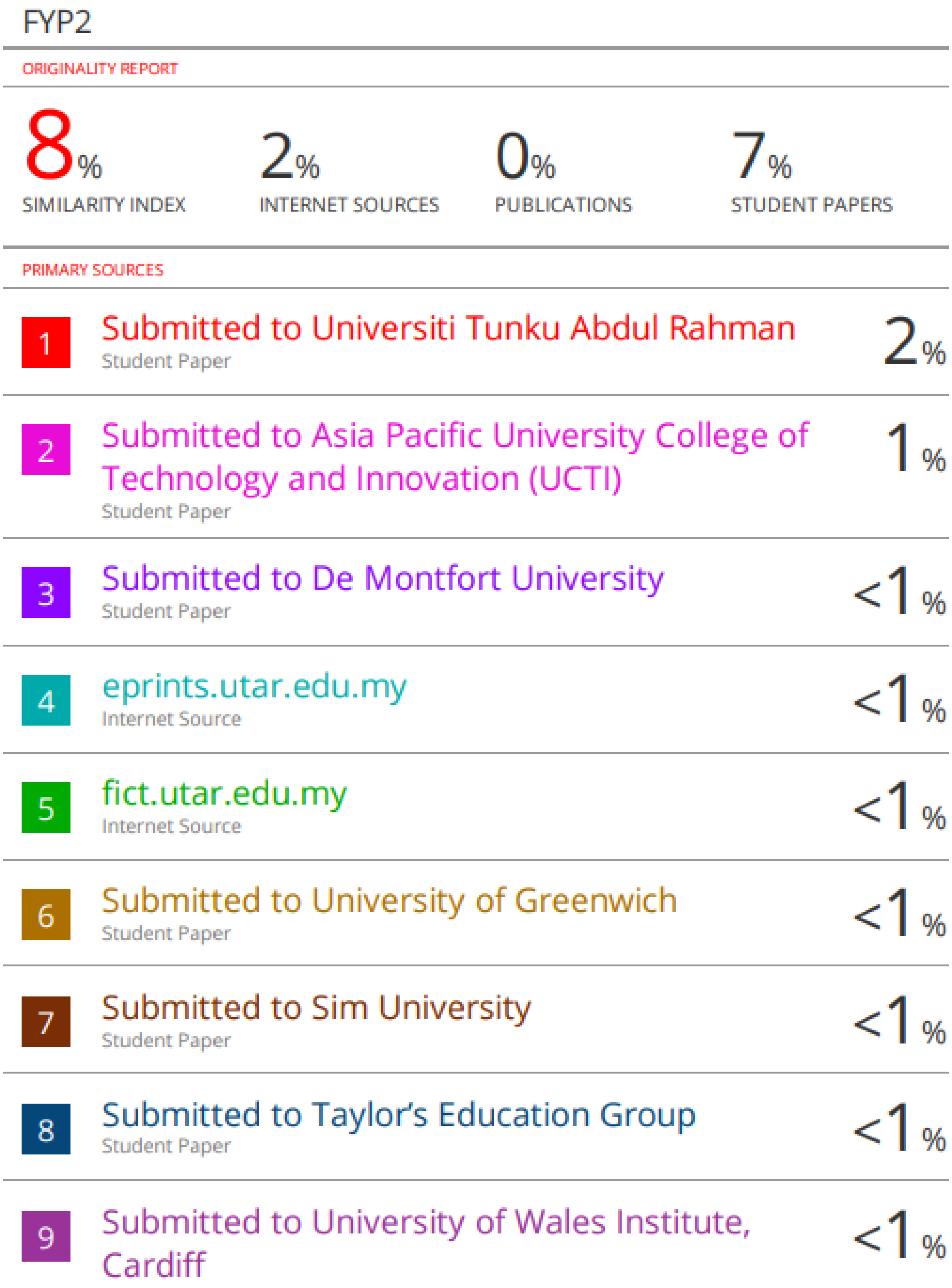
Figure 1. 1 peer-to-peer lending

The peer-to-peer process is fairly simple. All transactions take place via a dedicated online platform. Borrowers apply for a loan through P2P lending platform. Then, the administrator or management teams of the platform review and verify the credit score or credit rating of applicant, and give out appropriate interest rate. Once the loan application is approved by the P2P Lending platform, borrowers will receive information about the interest rate, monthly repayment. For lenders, they must choose

1	Submitted to Universiti ... Student Paper	2%
2	Submitted to Asia Paci... Student Paper	1%
3	Submitted to De Montf... Student Paper	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	fict.utar.edu.my Internet Source	<1%

Page: 1 of 104 Word Count: 14840

PLAGIARISM CHECK RESULT



FYP2

ORIGINALITY REPORT

8%	2%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
3	Submitted to De Montfort University Student Paper	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	fict.utar.edu.my Internet Source	<1%
6	Submitted to University of Greenwich Student Paper	<1%
7	Submitted to Sim University Student Paper	<1%
8	Submitted to Taylor's Education Group Student Paper	<1%
9	Submitted to University of Wales Institute, Cardiff	<1%

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Oon Wei Kin
ID Number(s)	18ACB05279
Programme / Course	IB
Title of Final Year Project	Fintech Peer-To-Peer Lending Mobile Application Development

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index:___8__ **% Similarity by source** Internet Sources: ___2 % Publications: __0___% Student Papers: ___7___ %	
Number of individual sources listed of more than 3% similarity: 0___	

Parameters of originality required, and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

SuLee

Signature of Supervisor	Signature of Co-Supervisor
Name: ___Su Lee Seng______	Name: _________________________
Date: _07/09/2022____________________	Date: ___________________________

11 FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student ID	18ACB05279
Student Name	Oon Wei Kin
Supervisor Name	Su Lee Seng

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgment
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)
Date: 07/09/2022