A Major Project Report on

Rent Assist

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Engineering in Software Engineering at Pokhara University

By

AMIT PARAJULI

RABIN K.C.

SUNIL THAPA

Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(September, 2022)

A Major Project Report on

Rent Assist

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Engineering in Software Engineering at Pokhara University

By

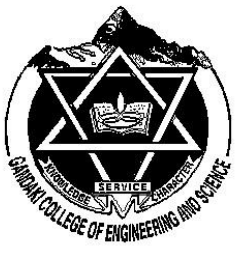
AMIT PARAJULI

RABIN K.C.

SUNIL THAPA

Supervisor

ER. KRISHNA KHADKA



Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(September, 2022)

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE
(Affiliated to Pokhara University)
P.O.Box 9, Lamachaur, Pokhara, Nepal

Ref. No.:

BONAFIDE CERTIFICATE

This is to certify that this project titled **RENT ASSIST** in partial fulfillment of the requirements for the degree of BACHELOR OF ENGINEERING IN SOFTWARE ENGINEERING is a bona fide work of **Amit Parajuli, Rabin KC** and **Sunil Thapa** under the supervision of **Er. Krishna Khadka**. It is further certified that this work doesn't form part of any other project work on the basis of which a degree or award was conferred on any earlier occasion on this by any other candidate.

Date of Evaluation: September 23rd, 2022

Er. Bal Krishna Nyaupane
(External Examiner)
Assistant Professor
Institute of Engineering,
Pashchimanchal Campus

Er. Shiva Ram Dam
(External Examiner)
Assistant Professor
School of Engineering,
Pokhara University

Er. Krishna Khadka
(Supervisor)

Er. Rajendra Bahadur Thapa
Project Head
Research Management Cell

Phone: +977-61-440866, 440414 | Fax: +977-61-440958 | URL: www.gces.edu.np | E-mail: gces@gces.edu.np

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and appreciation to all who gave us the opportunity to do this project. Completion of this project could not have been accomplished without the support of many individuals and the college administration. At first, we would like to give special gratitude to our project supervisor Er. Krishna Khadka and acting coordinator of Research Management Committee, Er. Rajendra Bahadur Thapa for providing necessary information regarding the project and also for the support while completing the project. Thanks to our parents for their encouragement and kind cooperation. We express our deepest gratitude to the principal Mr. Birendra Khadka and vice-principal Mr. Ashok Raj Parajuli for inspiring and supporting us during the entire project. Last but not least we are thankful to all our teachers, friends and seniors who have been always helping and encouraging us throughout this project.

Amit Parajuli

Rabin K.C.

Sunil Thapa

Gandaki College of Engineering and Science

Lamachaur, Pokhara

ABSTRACT

In this digital era rent collection is the piece of cake in developed countries, but in case of the developing country like Nepal there are no apps that target specific group having small to moderate group of tenants. The idea of this project is to set up a rent management system where the owner keeps digital track of his tenants and rent. Tenants is able to report problems, pay rent digitally and contact the owner through the app. Owners and tenants is able to view and manage transactions. Tenants is notified when the due date approaches. This project helps to calculate the electricity cost digitally. It includes OCR (Optical Character Recognition) for scanning the electricity sub-meter.

सारांश

यस परियोजनाको विचार एउटा भाडा व्यवस्थापन प्रणाली स्थापना गर्नु हो जहाँ मालिकले आफ्ना भाडामा लिने र भाडाको डिजिटल ट्र्याक राख्छ। भाँडादारहरूले उसको आफ्नै समस्याहरू निवेदन गर्न, बिधुतिया रूपमा भाडा तिर्न र एप मार्फत मालिकलाई सम्पर्क गर्न सक्षम छन्। मालिक र भाँडादारहरू बीचमा भाडामा लिने लेनदेन हेर्न र व्यवस्थापन गर्न सक्षम छन्। भाडावालहरूलाई सूचित गरिन्छ जब निर्धारित मिति नजिक आउँछ। यो परियोजनाले बिजुलीको लागत डिजिटल रूपमा गणना गर्न मद्दत गर्दछ। यसमा बिजुलीको सब-मिटर स्क्यान गर्नको लागि OCR (अप्टिकल क्यारेक्टर रिकग्निसन) समावेश छ।

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

1.1 BACKGROUND

This software will assist you by making the rent payment activities easy, mainly in the context of Nepal. The app sends an alert, information via push notification, to the tenants when the due is approaching. The Rent Assist app includes tenant payments, chat feature, notifications, report problems and scan electric meter. The main principle of this application is to make easy communication between owner and tenant and also make different types of payments (for example house/room rent, water and electricity) within a single app. The user-friendly interface and dynamic approach of the application make the experience easier and smooth for everyone.

1.2 PROBLEM STATEMENT

- Lack of proper payment details and information to tenants.
- The owner finds it hard to communicate with the tenant.
- Difficulties in providing different notices and announcements to tenants via traditional methods.
- Difficult calculation of electricity bill.
- No proof of payment or invoice.

1.3 OBJECTIVES

Rent assist is a mobile application that provides a platform for rent payment and easy communication for owner and tenants. This project will fulfil the following goals: -

- To make payment of rent digitally.
- To scan electricity sub-meter using Optical Character Recognition.
- To notify tenants of the due date, report problems and chat through app.

1.4 IMPLICATION

Tenants finds a problem with rent payments to the owner, including electricity and water bill. Our project deals with this problem. “Rent Assist” helps in easy payment, communication, delivering notices, reporting problems and many more. Our application helps tenants by providing a digital platform for payment and notices regarding the rent.

Chapter 2

LITERATURE REVIEW

Few similar products have been found that have already been developed which serve similar purposes as our system but not the same. Unlike all those products, "Rent Assist" is specifically designed for the context of Nepal for easy, hassle-free and digital calculation and payment of rent, electricity, water bill, etc.

Some applications are listed below:

1. **Buildium** (Buildium, 2004)

Built by property managers for property managers, Buildium's comprehensive service allows property owners to control every aspect of their business remotely.

Pros

- Build by property managers for property managers
- Automatic rent collection through a tenant portal
- Online ticket support during business hours
- A lot of training materials offered

Cons

- Expensive pricing plans
- Some customer support complaints
- Not a fit for single property managers

2. **TurboTenant** (turbotenant, 2015)

TurboTenant is a free option for the landlords whose pricing model puts all of its costs on the tenants.

Pros

- Free regardless of the number of units managed
- Great for DIY landlords
- 24/7 customer support through phone and online

Cons

- Tenants have to pay for it
- Limited advanced features.

3. **AppFolio** (appfolio, 2006)

AppFolio, popular among landlords with large rental portfolios, offers the best-advanced features to control your units remotely, and additional support and resources to grow your business and maximize its efficiency.

Pros

- Plans for residential landlords, community associations, and commercial real estate landlords
- Supports all types of units
- Has a mobile app

Cons

- Has a minimum monthly fee
- Not fit for few properties

4. **Propertyware** (Propertyware, 2001)

Propertyware is fairly simple to use, without complicated features commercial management software can have, making it the best choice for single-family home management. You'll benefit from Propertyware's management features, especially if you're a single-family home landlord.

Pros

- Manage large portfolios
- Has a mobile app

- Multiple pricing tiers

Cons

- Some customer support complains
- Fit only for large portfolios

5. **SimplifyEM** (SimplifyEm, 2006)

Designed by real estate professionals, SimplifyEm offers the ideal balance of price and features, making it the perfect choice for landlords that need help managing a few properties and don't want to pay the high cost of other software plans.

Pros

- Supports one to 2,000 units
- Designed by real estates professionals
- Has advanced features

Cons

- If you have more than 2,000 units, the platform can't grow with your portfolio

6. **Rent Assist**

Designed for the context of Nepal to make a rent payment procedure easy which offers features like Optical Character Recognition for Electric meter reading, in-built chat system, problem reporting and online payment.

Pros

- Suitable for the small and medium sized customer group focusing context of Nepal.
- Optical Character Recognition for electric meter reading and automatic price calculation.
- In built complaint report feature.

- Online payment feature.
- Free for users.

Cons

- Not feasible for large customer group.

Chapter 3

TOOLS AND METHODOLOGY

3.1 REQUIRED TOOLS

The following tools were used for the development of the application.

- VS Code – Code Editor
- Django – Backend
- Flutter – Mobile App
- Git – Version Control System
- PostgreSQL – Database
- Heroku – API deployment

3.2 METHODOLOGY

3.2.1 USE CASE DIAGRAM

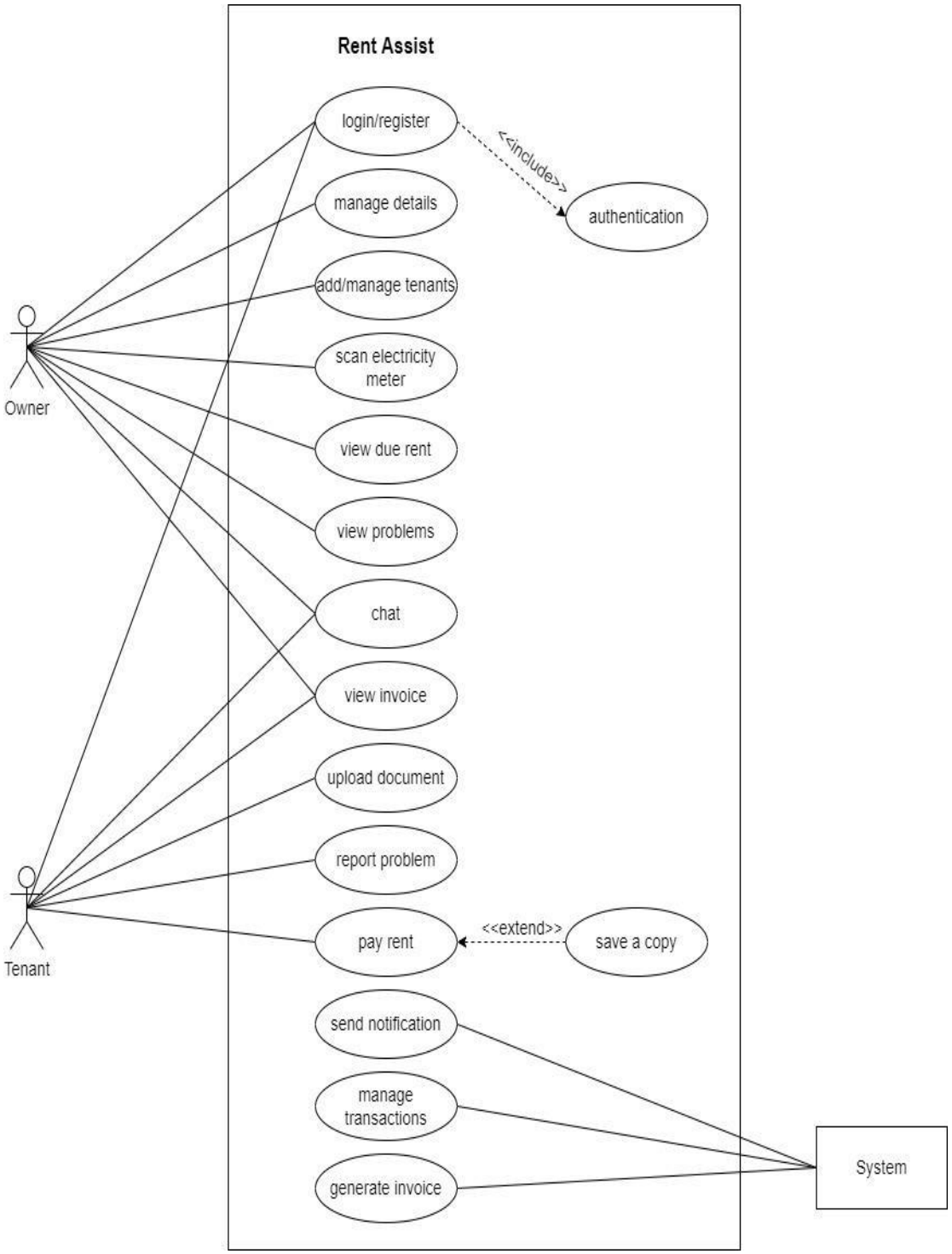


Figure 3. 1 Use Case Diagram

USE CASE UC1: REGISTER

Primary Actor: User (Tenants and owner)

Stakeholders:

- User: Wants to create a new account.
- System: Stores user information and updates database.

Preconditions: The user must have valid documents.

Postconditions: The user registration process is completed.

Basic Flow:

- The system asks the user to provide his/her credentials in the application.
- The system checks if all details have been entered and all the documents are provided.
- The system stores the info in the database.

USE CASE UC2: LOGIN

Primary Actor: User

Stakeholders:

- User: Wants to log in to the system.
- System: Checks the entered credentials for verification.

Preconditions: The user wants to use the application.

Postconditions: The user is logged in and greeted with the home screen.

Basic Flow:

- The user inserts his/her account credentials in the application.

- The system logs in the user if the credentials match the ones stored in the database.

USE CASE UC3: LOGOUT

Primary Actor: User

Stakeholders:

- User: Wants to log out of the system.
- System: Logs users out.

Preconditions: The user must have been logged in.

Postconditions: The logout process is completed.

Basic Flow:

- The user wants to log out.
- The system logs the user out.

USE CASE UC4: VIEW INFORMATION

Primary Actor: User

Stakeholders:

- User: Wants to view the information.
- System: Displays the information.

Preconditions: The user must be logged in.

Postconditions: Users can view the information.

Basic Flow:

- The user wants to view the information.
- The system displays the selected information.

USE CASE UC5: PAY RENT

Primary Actor: Tenant

Stakeholders:

- Tenant: Wants to pay the rent.
- System: Verifies transactions and generates invoices.

Preconditions: Tenant is logged in. The tenant has a due balance.

Postconditions: Tenant gets the invoice.

Due amount is cleared.

The owner gets notified.

Basic Flow:

- The tenant checks the due amount.
- The tenant chooses a payment method.
- The tenant makes payment.
- The system verifies the transaction.
- The system generates the invoice.
- The system notifies the owner.

Alternative Flow:

- Insufficient Balance, Network Error, Transaction gets cancelled, and an error message is displayed.

USE CASE UC6: SEND MESSAGE

Primary Actor: User

Stakeholders:

- User: Wants to send messages between tenant and owner.
- System: Checks if the provided details like username and password are correct.

Preconditions: The user wants to send a message.

Postconditions: Message is sent.

Basic Flow:

- Users log in to the system using their username and password.
- The user chooses to send a message and sends the message.
- Other users receive the message and can reply.

USE CASE UC7: MANAGE CHAT

Primary Actor: System

Stakeholders:

- User: Wants to send messages to each other.
- System: Manages chat sessions.

Preconditions: The user must be logged in.

Postconditions: Communication between tenant and owner.

Basic Flow:

- The system stores the messages in the database and then handles the chat sessions for multiple users.

USE CASE UC8: MANAGE TENANT

Primary Actor: Owner

Stakeholders:

- Owner: Wants to add, remove or edit tenants.
- System: Validates the information and updates the database.

Preconditions: The owner is logged in to the system.

Postconditions: Tenant is updated.

Basic Flow:

- The owner views the tenant.
- The owner modifies the tenant's information.
- The system validates and updates the database.

Alternative Flow:

Validation Error, Network Error, Error is displayed.

3.2.2 SYSTEM SEQUENCE DIAGRAM

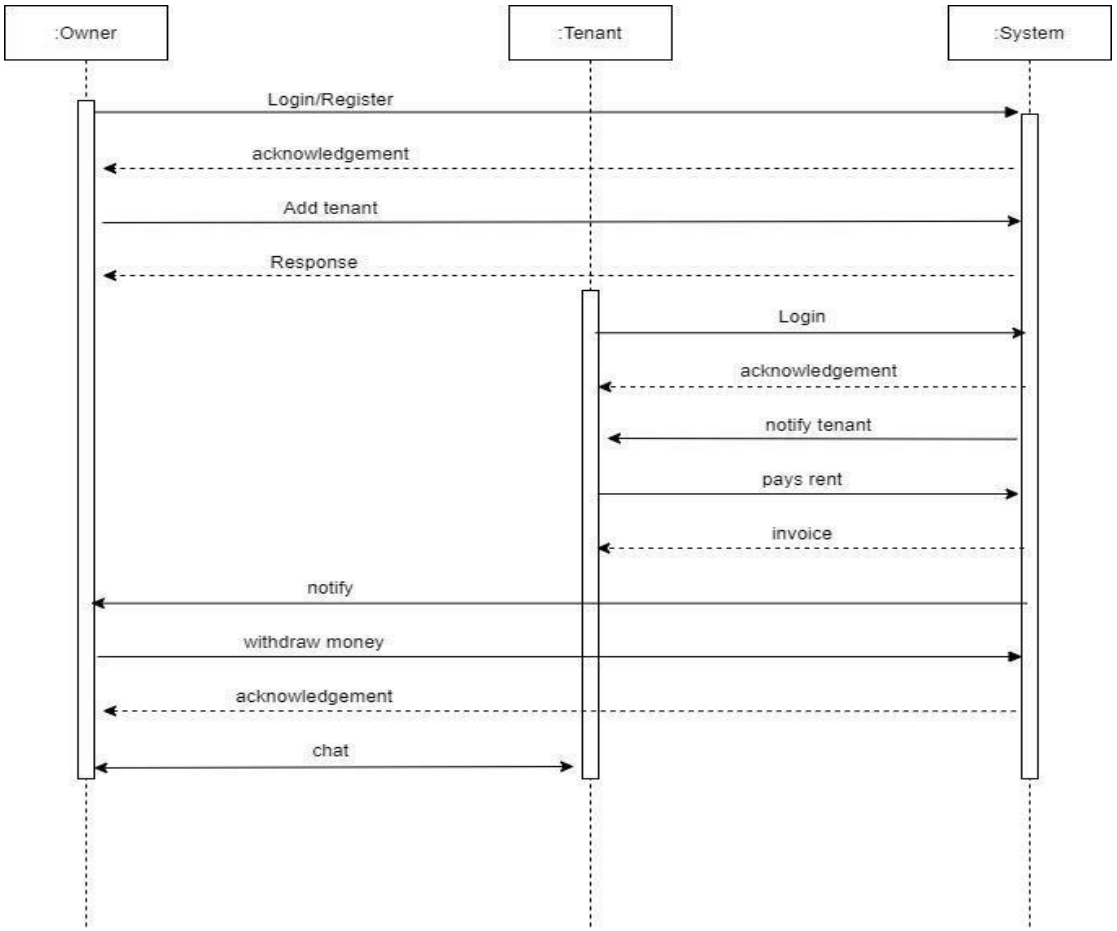


Figure 3. 2 Sequence Diagram

3.2.3 ENTITY RELATIONSHIP DIAGRAM

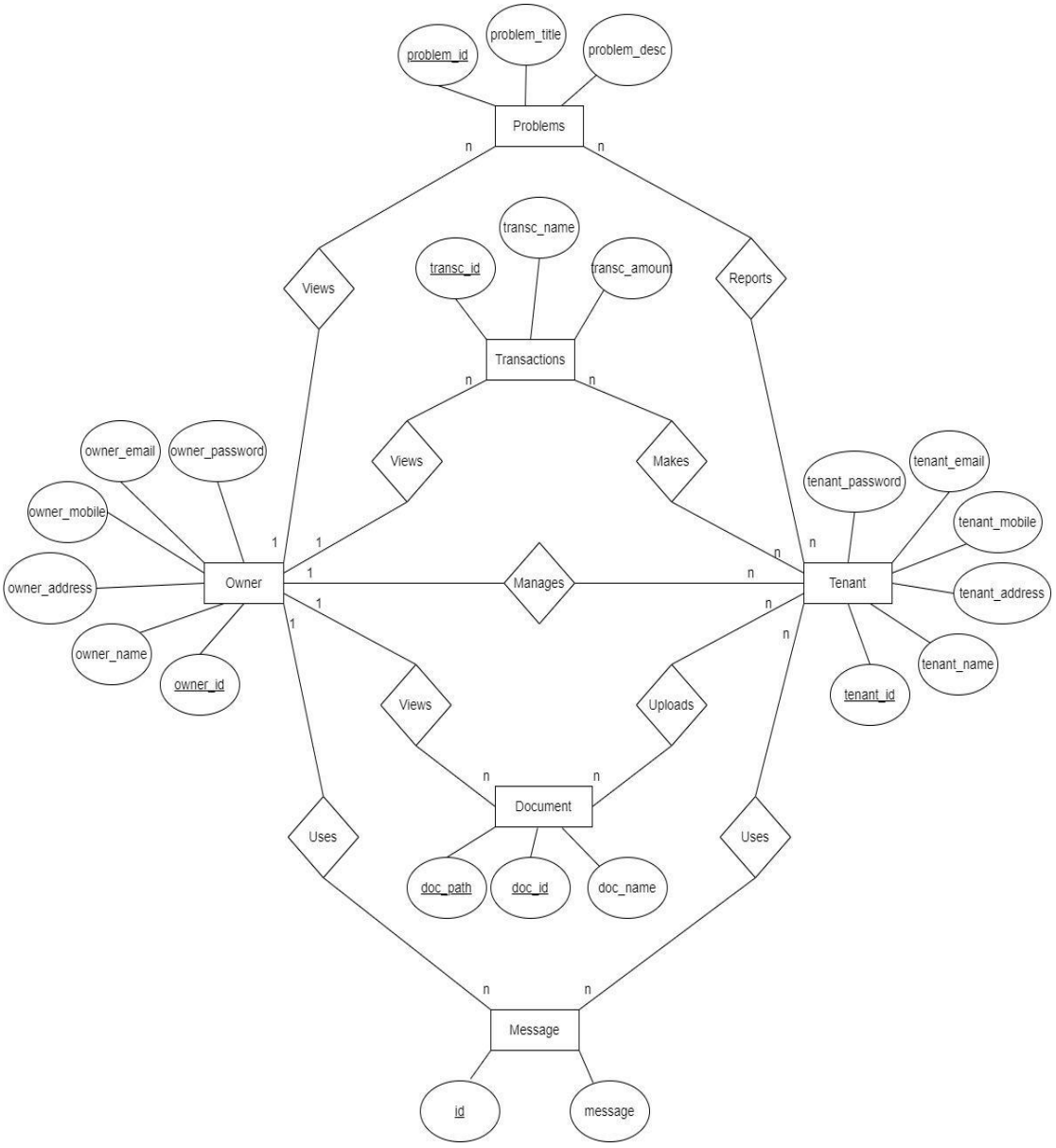


Figure 3. 3 Entity Relationship Diagram

Rent Assist will run in a sequence as follows. Owner login/register to the system. The system sends an acknowledgement. The owner then adds the tenant. The system sends a response. Tenants can then log in to the system. The system notifies tenants about some information related to rent. The tenant pays rent. The system sends an invoice. The system then notifies the owner. The owner withdraws money. The system sends an acknowledgement. The owner and tenant can chat with each other.

3.2.4 DESIGN CLASS DIAGRAM

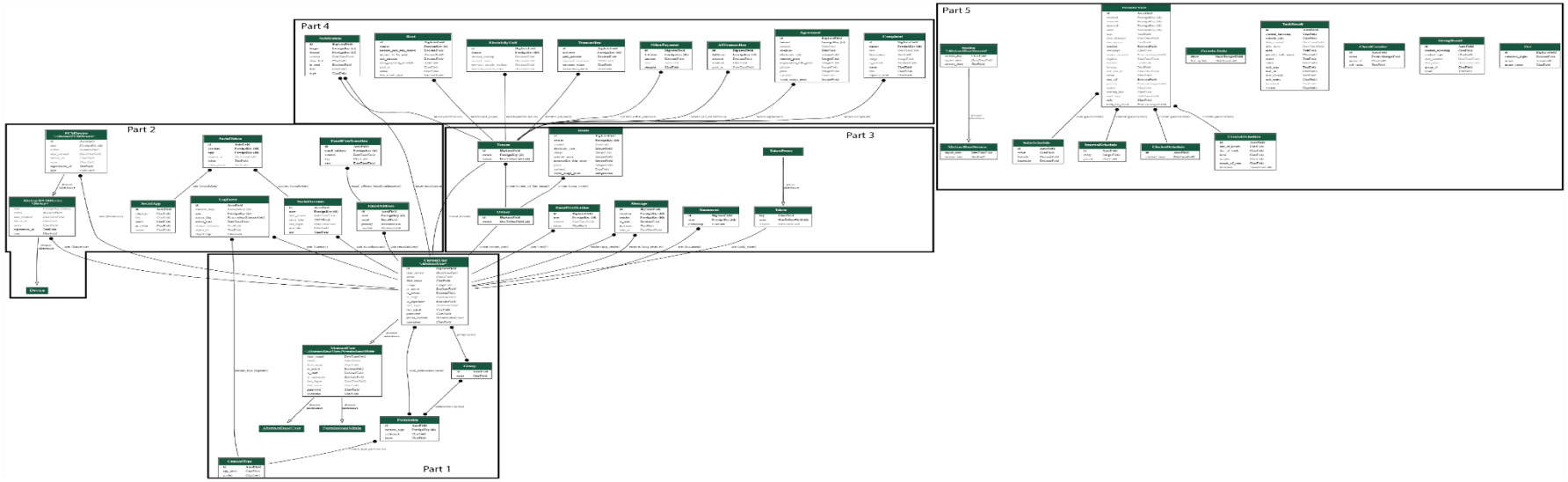


Figure 3. 4 Design Class Diagram (DCD)

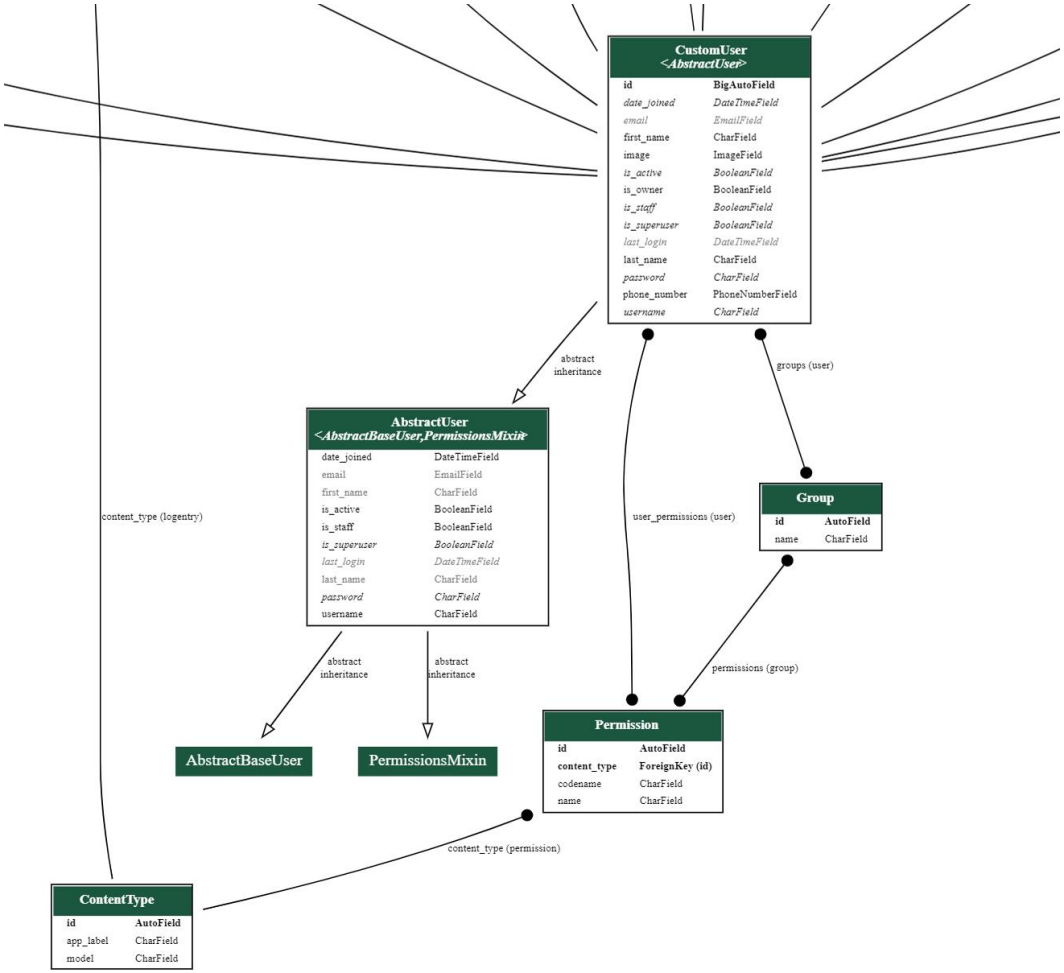


Figure 3. 5 DCD Part 1

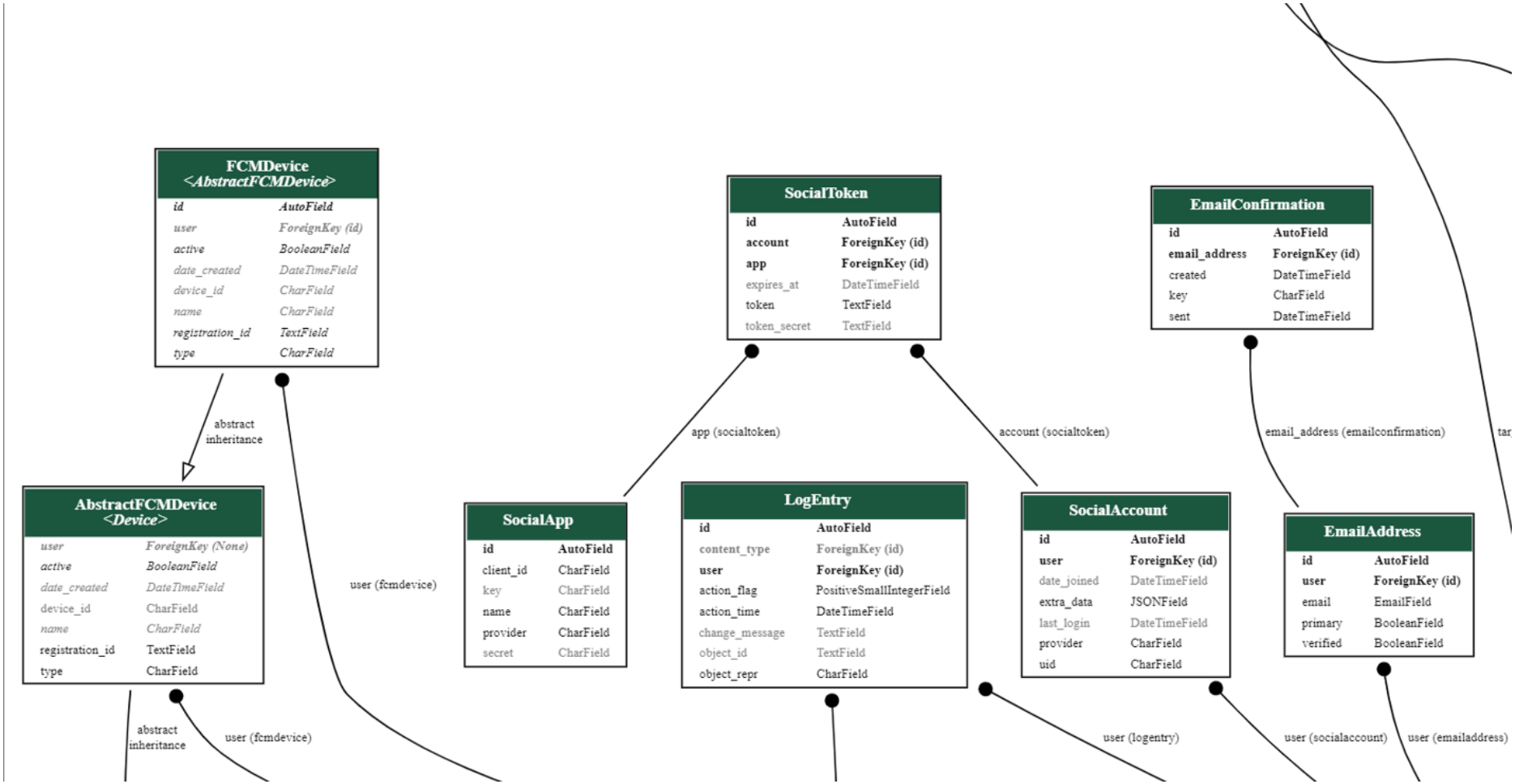


Figure 3. 6 DCD Part 2

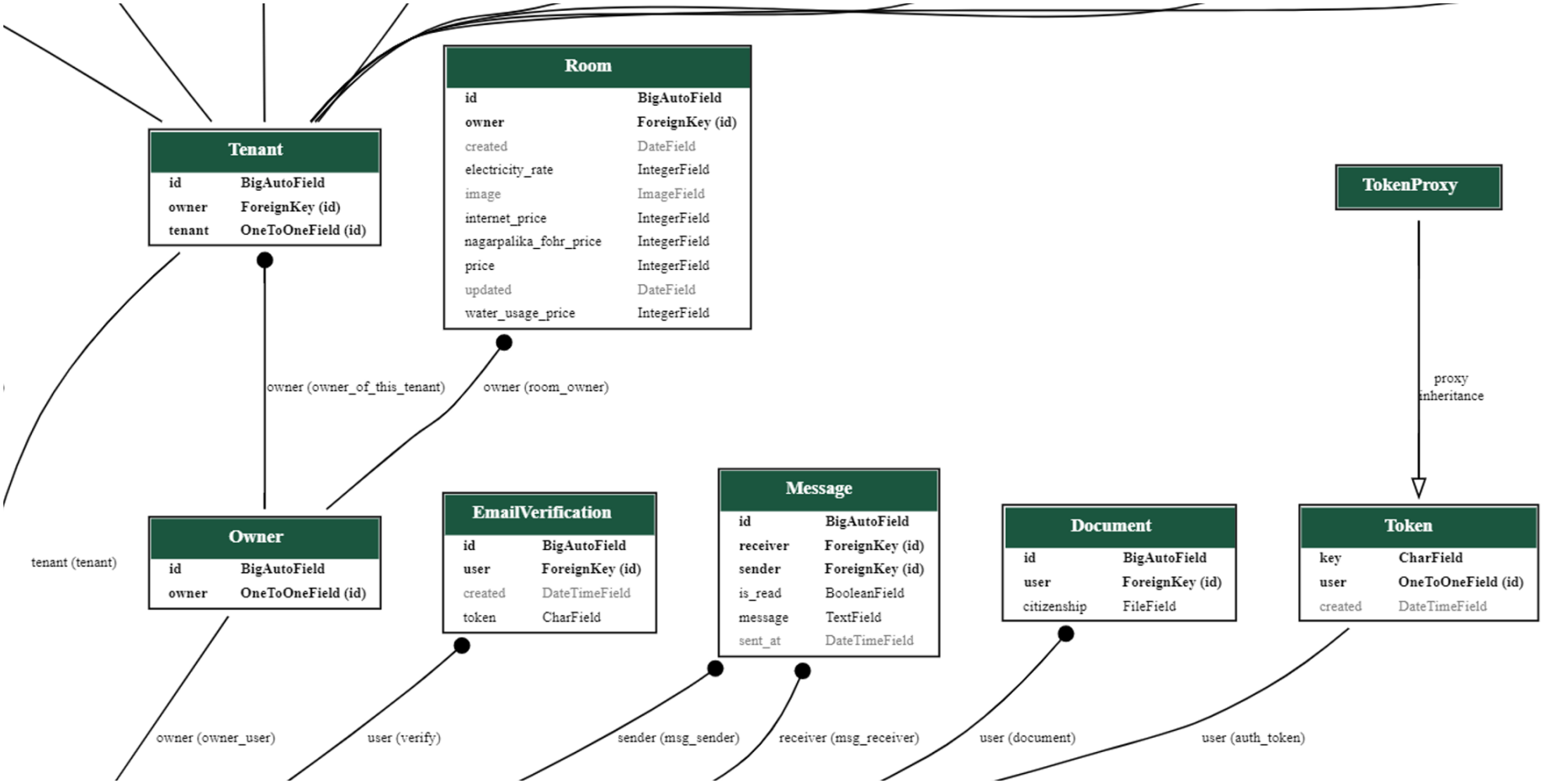


Figure 3. 7 DCD Part 3

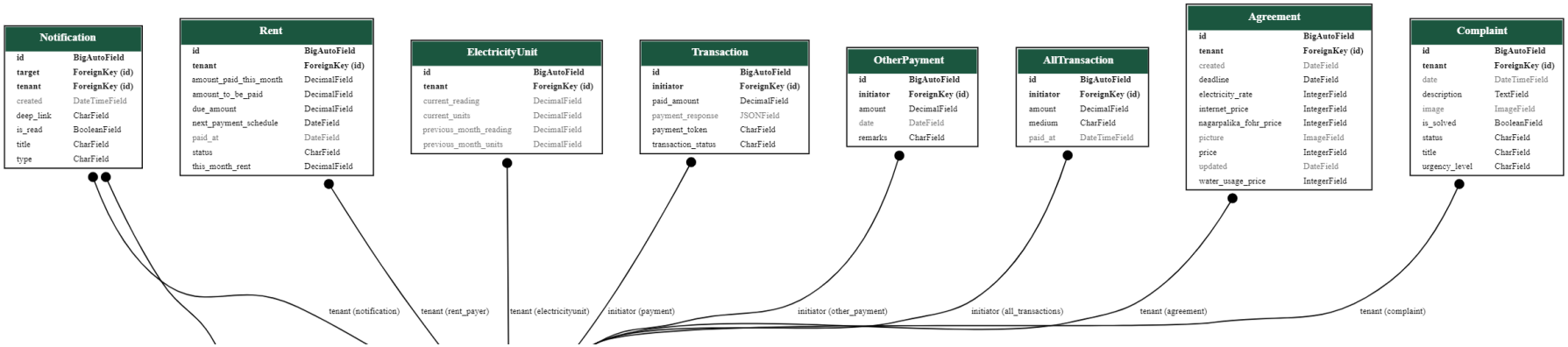


Figure 3. 8 DCD Part 4

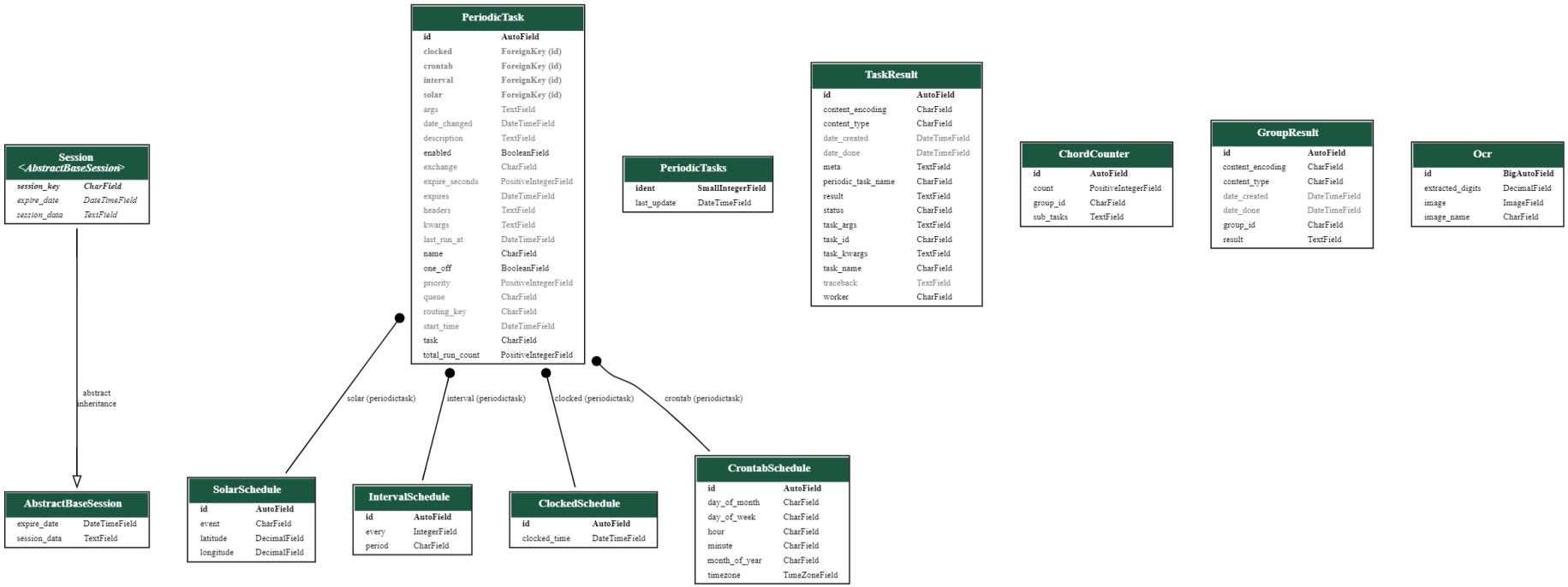


Figure 3. 9 DCD Part 5

3.3 APPROACH USED

Every software development methodology approach acts as a basis for applying specific frameworks for developing and maintaining the software. Some software development approaches like Waterfall Model, Agile Methodology, Rapid Application Development, Spiral Model, Incremental Model, etc. have been used since the origin of information technology. Since our project has defined needs, and we expect few changes in the final version, the iterative model development approach is used in this project. It is compatible with the size of our project and suited for our available time frame. It allows for further adding features necessary according to requirements.

Chapter 4

TEST CASES

4.1 SOFTWARE TESTING

Software testing is a process to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not. It also identifies the defects, bugs to ensure that the product is defect-free in order to produce a quality product.

4.2 TEST OBJECTIVES

The main objectives of testing Rent Assist were:

- To check whether the android application is built as per the set objectives or not.
- To ensure errors get fixed before deployment.
- To gain confidence in the level of quality of the system.
- To identify the testing standards and procedures thar were used on the project.
- Prepare and document the test scenarios and test cases.
- Manage defect tracking process.
- Provide test metrics/testing summary reports.

4.3 TEST RESULTS

Several tests were done while developing the system and the results were reviewed to identify and remove errors. The following table consists of the test results of testing which were performed to validate the system with respect to the requirements.

TEST CASE id	TEST CASES	EXPECTED	OBSERVED	RESULT

1.	Authentication for owner and tenant.	User with correct email and password and those who are verified can login.	User registers and email is sent for account verification.	PASS
2.	Agreement proposal	Owner proposes an agreement for a room. Tenant accepts the proposal if he agrees. Then an agreement is formed, else owner improves agreement and tenant accepts.	Email is sent to both tenant and owner that agreement is formed and Quick Response code is generated on the given data.	PASS
3.	Add tenant	The tenant scans the Quick Response code on the Owner App screen. After scanning QR he receives an agreement proposed by the owner. If the proposed agreement is acceptable to Tenant, he accepts the agreement and he is registered as tenant to Owner.	An email is sent to both Tenant and Owner that agreement is formed.	PASS

4.	Configure meter reading by Owner.	After successful registration of a Tenant to Owner, the Owner runs Optical Character Recognition on the electricity meter of the tenant apartment and configures the reading data.	A record is formed on the current reading and next scan date is scheduled.	PASS
5.	Chat	A tenant is able to have conversation with his owner and vice versa.	A message is sent and sender is able to see if the message is opened.	PASS
6.	Complaints	A tenant is able to complain on different problems that he/she is facing regarding the apartment.	The owner gets the notification about the complaint through email, push-notification and in App notification.	PASS

7.	Run Optical Character Rrecognition	When the scheduled deadline for monthly payment a notification is sent to both Owner and tenant. Owner scans the electricity meter of the tenant. He then confirms the electricity unit returned by the Optical Character Recognition.	The rent is calculated on the basis of agreement and last scanned electricity unit.	PASS
8.	Add documents	It is optional to add personal details unless owner requests for one.	Tenant is able to add documents.	PASS
9.	Payment	Two types of payment are available. That is online and cash payment. Online transaction is done through Khalti, Online banking, Connect IPS, etc	Owner is notified. Remaining due is calculated and saved.	PASS

10.	Integration/Continuous Deployment	Each changes/improvement made to our application is implemented with Continuous Integration/Continuous Deployment.	With each changes a new version of application is deployed.	PASS

Figure 4. 1 Test Cases

Chapter 5

RESULTS AND DISCUSSIONS

Our project Rent Assist helped us to learn in detail about the development of mobile application. It provided us the opportunity to get expertise in various mobile app development languages. Our project has fulfilled all its objectives. This project was completed within the estimated time and with the coordination of team members and our respected supervisor.

5.1 LIMITATIONS

This mobile app is developed to assist and make the rent payment process easy and hassle free, but there are some limitations which are listed below:

- Technical feasibility
  Internet access is required to use the app.
- Algorithm KNN classifier is not well trained.
- This app targets only specific customers, not suitable for large real estate customers.

5.2 FUTURE IMPROVEMENTS

Some of the improvements that can be implemented in the future are as follows:

- Model our application into subscription-based service.
- Train Optical Character Recognition more

Chapter 6

CONCLUSION

Rent Assist is a mobile application for easy payment of the rent for the tenant and rent collection for the owner. This app provides many features such as chat for easy communication between owner and tenant. Tenant can easily report the problems that he/she has been facing regarding the apartment. Owner can scan the electricity meter through the app for recent price of electricity without worrying about the past reading. The total rent is calculated automatically those which were included in the agreement. Both Owner and Tenant are notified on various action triggered by them through the email, push-notification and in-app notifications. The owner is notified through email, push-notification and in app notification when the rent is paid by tenant. Various other details such as due rent, contract information, owner/tenant information can be viewed through the app.

BIBLIOGRAPHY

Buildium(n.d.), Retrieved from Buildium: *https://www.buildium.com* accessed 25th May 2022.

TurboTenant(n.d.), Retrieved from TurboTenant: *https://www.turbotenant.com* accessed 25th May 2022.

Appfolio(n.d.), Retrieved from Appfolio: *https://www.appfolio.com* accessed 25th May 2022.

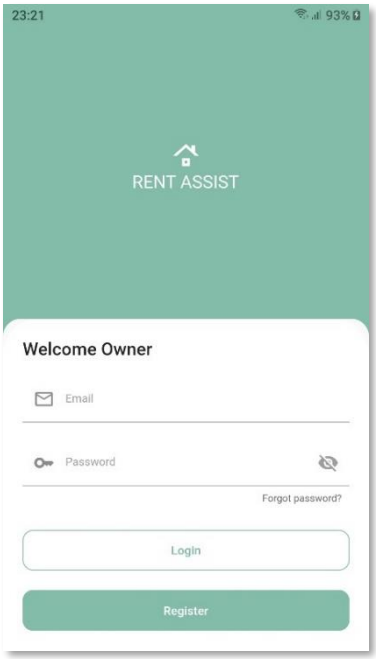
SimplifyEM(n.d.), Retrieved from SimplifyEM: *https://www.simplifyem.com* accessed 25th May 2022.

Propertyware(n.d.), Retrieved from Propertyware: *https://www.propertyware.com* accessed 25th May 2022.

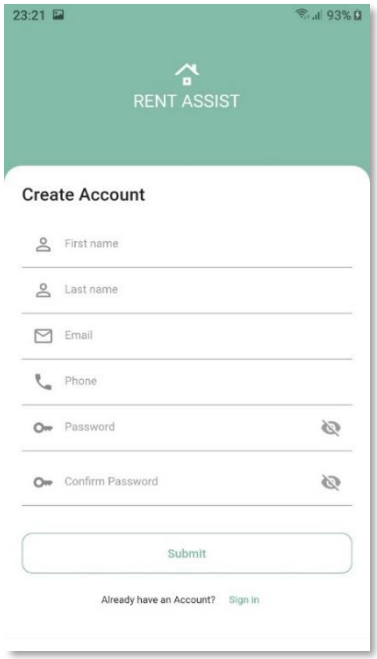
Investopedia(n.d.), Retrieved from Investopedia: *https://www.investopedia.com* accessed 25th May 2022.

Entity Relationship Diagram, Use Case Diagram and System Sequence Diagram Retrieved From *https://www.draw.io* accessed 30th May 2022.

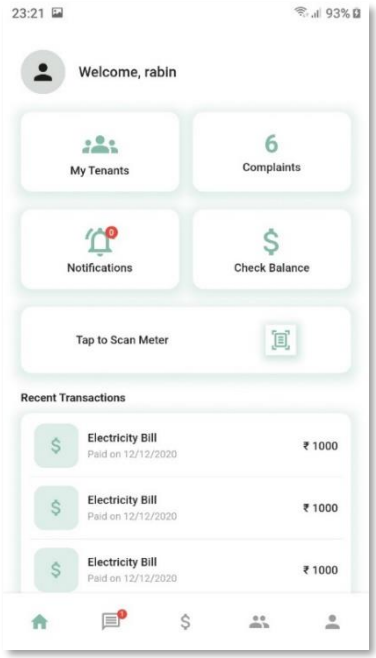
APPENDIX



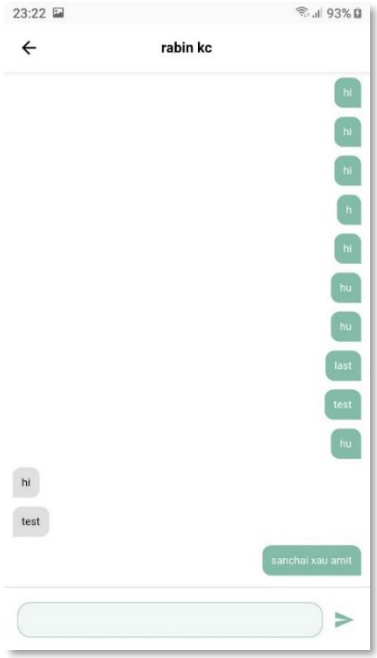
Appendix of Owner login screen



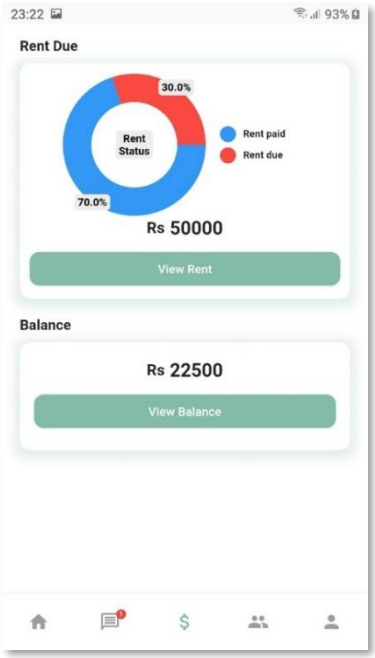
Appendix of Register screen

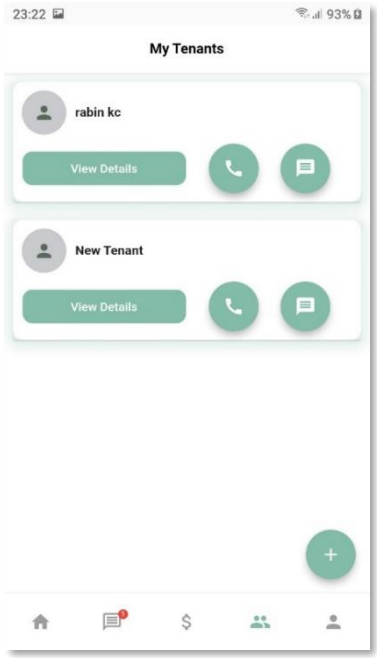


Appendix of Owner Homepage

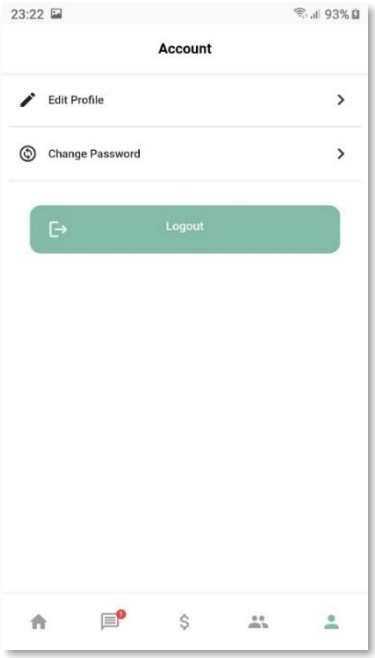


Appendix of Owner Chat Screen

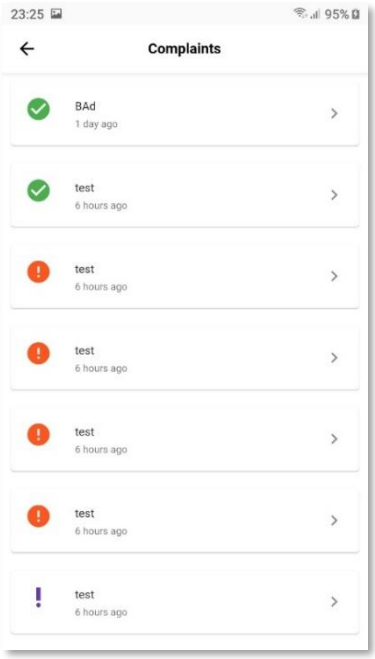




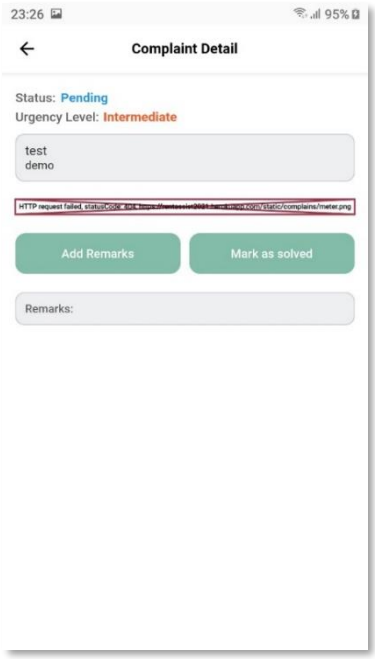
Appendix of Balance View Page **Appendix of Add Tenant Screen**



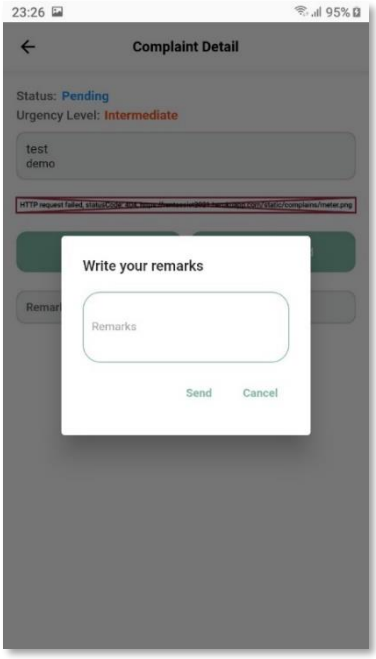
Appendix of Account Screen **Appendix of OCR Screen**



Appendix of Complaints Screen

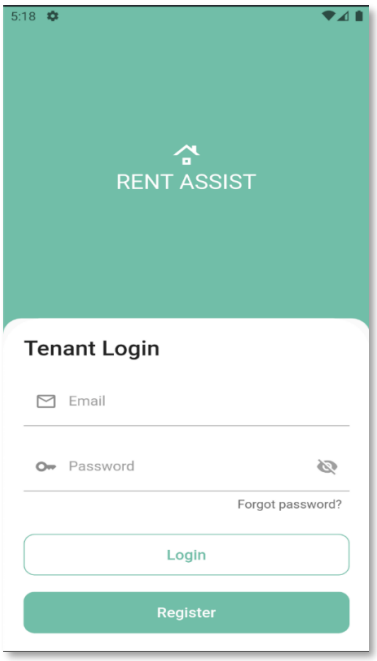


Appendix of Complaints Detail Screen

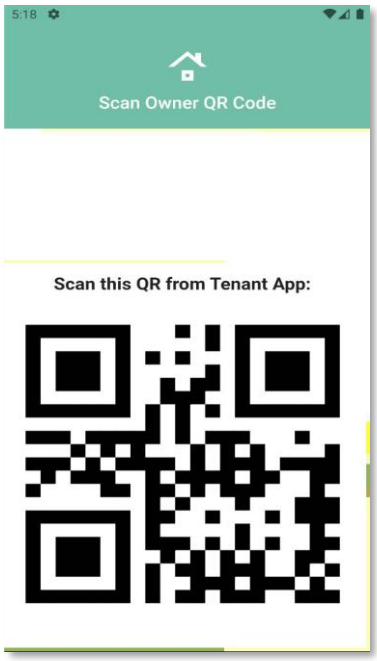


Appendix of Add Remarks Screen

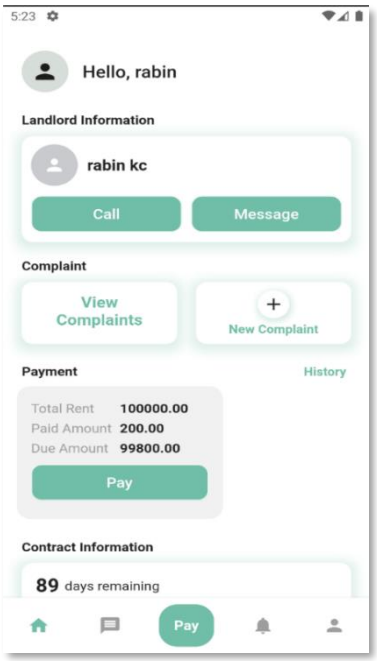
Appendix of Agreement QR Screen



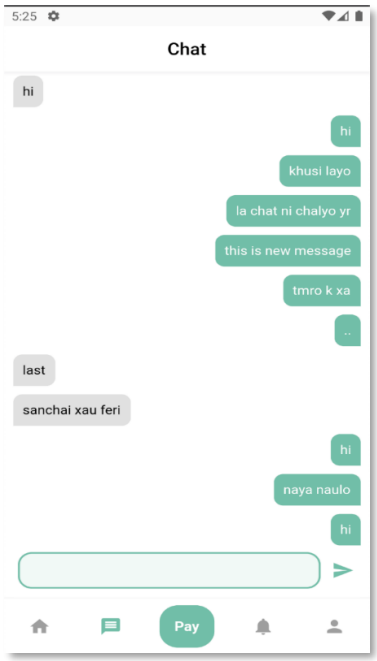
Appendix of Tenant Login Screen



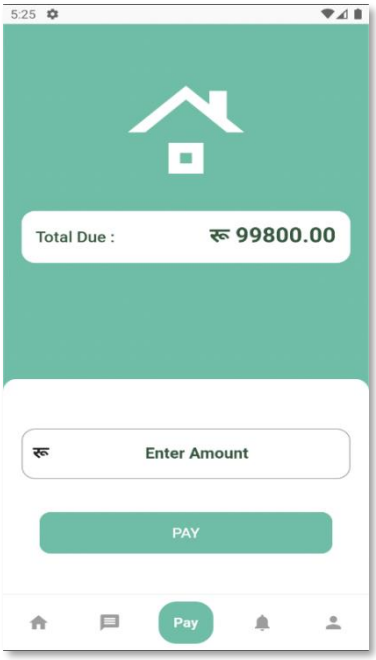
Appendix of Scan QR Screen



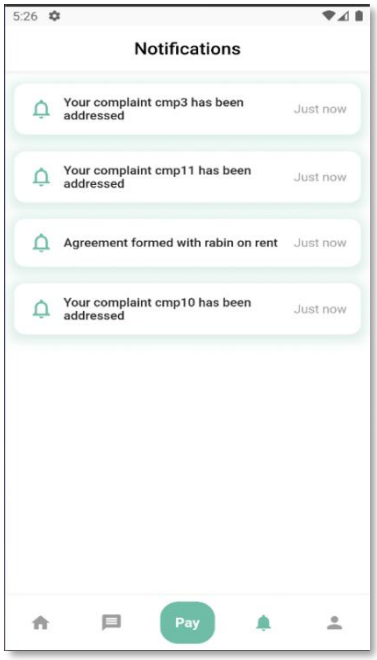
Appendix of Tenant Login Screen



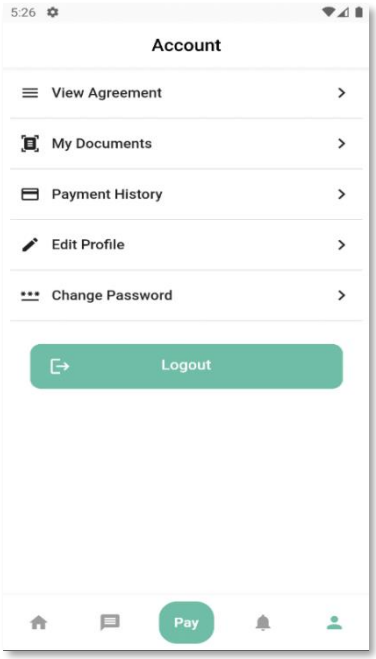
Appendix of Tenant Chat Screen



Appendix of Payment Screen

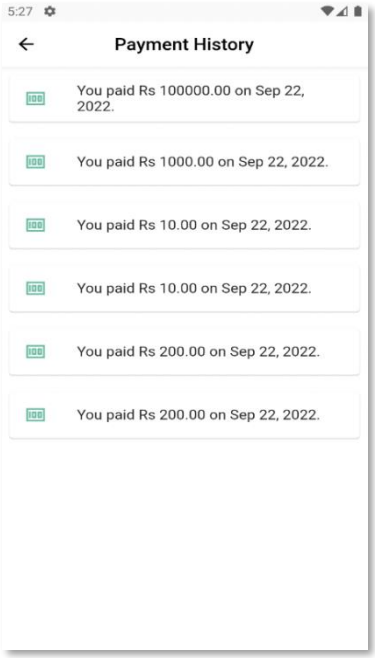


Appendix of Tenant Notification Screen

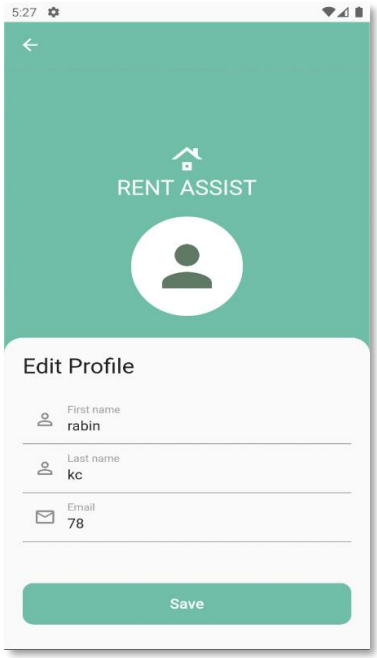


Appendix of Tenant Account Screen

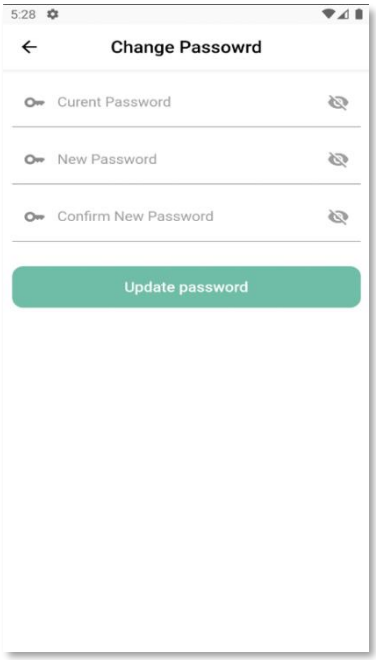
Appendix of View Agreement Screen



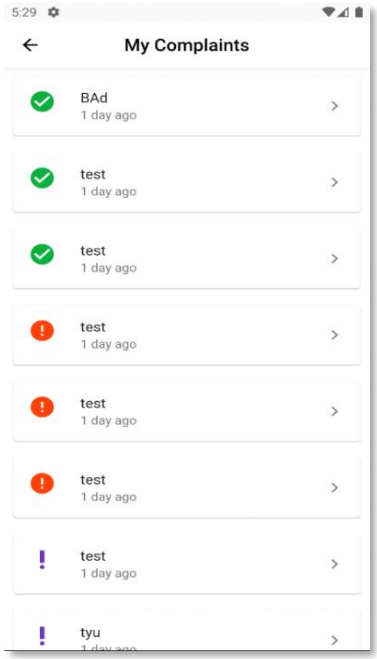
Appendix of Payment History Screen



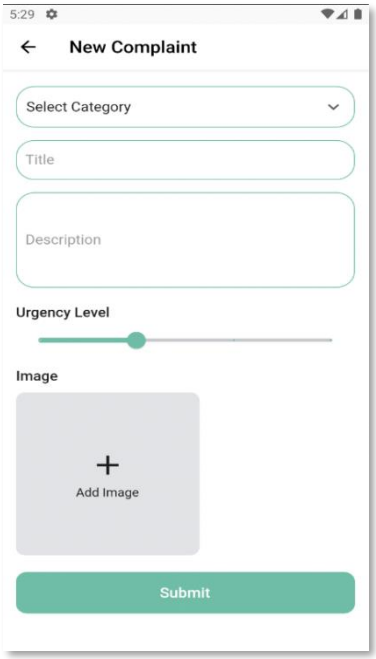
Appendix of Tenant Edit Profile Screen



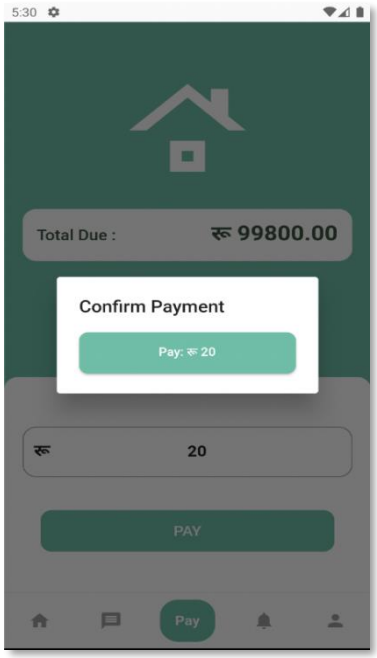
Appendix of Change Password Screen



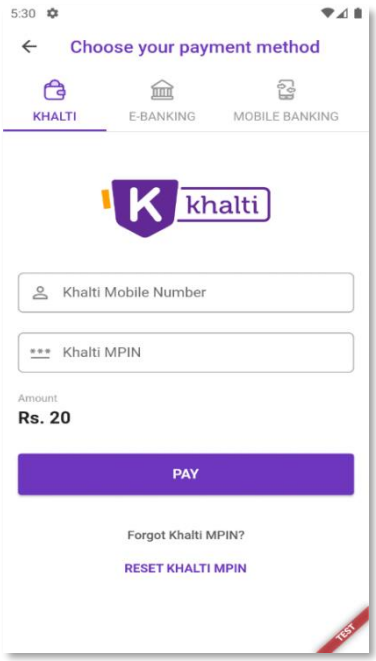
Appendix of My Complaints Screen



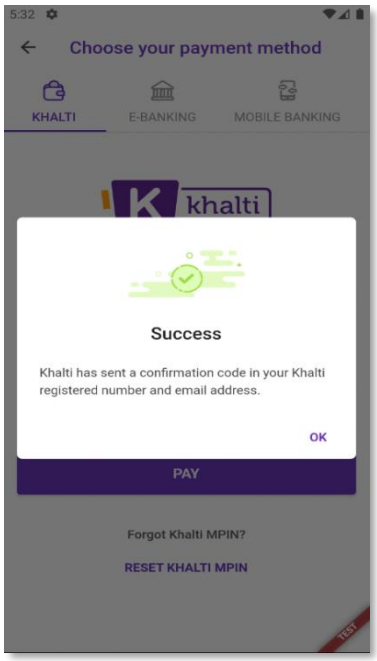
Appendix of Add Complaints Screen



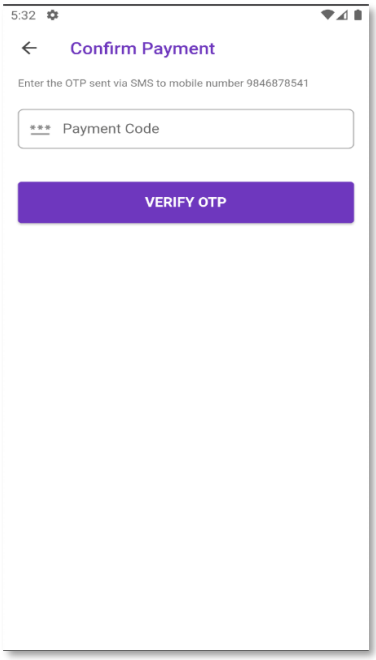
Appendix of Confirm Payment Screen



Appendix of Khalti Screen for Phone number



Appendix of Payment Success Screen



Appendix of OTP Confirmation Screen