

APPROVAL

This Project titled “**LandNant: An android app for landlord and tenant,**” submitted by **Md. Masud Rana, Md. Tushar Ahmed and Shushanto Kumar** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of science in Computer Science and Engineering (BSc) and approved as to its style and contents. The presentation has been held on 07th May, 2018.

BOARD OF EXAMINERS

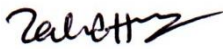
Dr. Syed Akhter Hossain
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Chairman



Dr. Sheak Rashed Haider Noori
Associate Professor and Associate Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Zahid Hasan
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor
Department of CSE
Jahangirnagar University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Nazmun Nessa Moon, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Nazmun Nessa Moon
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Masud Rana
ID: 142-15-3526
Department of CSE
Daffodil International University

Md. Tushar Ahmed
ID: 142-15-3983
Department of CSE
Daffodil International University

Shushanto Kumar
ID: 142-15-3795
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Nazmun Nessa Moon, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “An android app for landlord and tenant:Landland” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Pro. Dr. Akhter Hossain, Professor, and Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Our project titled “LandNant: An Android Apps For Landlord And Tenant” is an android app for security and crime management of Bangladesh. Using this app the tenant can send their personal information which is required by the government to the police station through the landlord. Now-a-days smart phone becomes a part of our life. We can’t think a single moment without our smart phone. We are more likely to use our phone for practical purpose in our daily life. Among these smart phone OS, android has majority percentage. Android OS is also an open source platform and it has large number of users. For this we thought that we should build something in android that has practical use to the user. And it will help them in difficult situations. Our app will manage all the information about the people living in an area. This will increase the security and decrease crime. It also reduce time. We no need to go to the police station to provide our information and it will reduce harassment. We use java and SDK for front-end design and firebase for back-end design. We also use firebase as our main database. This application is very user friendly. After finishing all the task and test process this application proved to be working effectively.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
Table of contents	v
List of figures	viii
List of tables	ix

CHAPTER	PAGE NO
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcomes	2
1.5 Report Layout	2
CHAPTER 2: BACKGROUND	3-5
2.1 Introduction	3
2.2 Related Work	3
2.3 Comparative Studies	4
2.4 Scope of Problems	4

2.5 Challenges	5
CHAPTER 3: REQUIRMENT SPECIFICATION	6-14
3.1 Business Process Modeling	6
3.2 Requirement Collection and Analysis	7
3.3 Use Case Modeling and Description	7
3.4 Logical Data Model	14
3.5 Design Requirements	14
CHAPTER 4: DESIGN SPECIFICATION	15-19
4.1 Front-end Design	15
4.2 Back-end Design	15
4.3 Interaction Design and UX	17
4.4 Implementation Requirements	18
4.4.1 Android Studio	18
4.4.2 The Emulator	18
4.4.3 Android SDK	18
4.4.4 Firebase	19
CHAPTER 5: IMPLEMENTATION AND TESTING	20-31
5.1 Implementation of Database	20
5.2 Implementation of Front-end Design	24
5.3 Implementation of Interaction	29
5.4 Testing Implementation	30
5.5 Test Results and Reports	31
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	32
6.1 Discussion and Conclusion	32

6.2 Scope for Further Development	32
REFERENCES	33
APPENDIX	34-35
Appendix A: Project Reflection	34
Appendix B: Related Diagrams	35
PLAGIARISM REPORT	36

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Business Process Modeling	6
Figure 3.2: Use Case Model	7
Figure 3.3: Logical Data Model	14
Figure 4.1: User List	16
Figure 4.2: Database Creation	16
Figure 4.3: Image of Tenant	17
Figure 5.1: Authentication Choice	20
Figure 5.2: Authentication Rule	21
Figure 5.3: Landlord Information	22
Figure 5.4: Tenant Information	23
Figure 5.5: Home page	24
Figure 5.6: User Registration	25
Figure 5.7 :Sign In	25
Figure 5.8: Account Creation	26
Figure 5.9: Information Form	27
Figure 5.10: Tenant Information	28
Figure 5.11: Tenant Verification	29
Figure A.1: Use case model	35
Figure A.2: Database	35
Figure: Plagiarism Report	36

LIST OF TABLE

TABLE	PAGE
Table 5.1: Test Case	30

CHAPTER 1

INTRODUCTION

1.1 Introduction

A majority of unpleasant issues faced by landlords and tenants seem to grow due to lack of communication. Fortunately, building a positive relationship doesn't take much effort and it all starts with one basic idea: communication. An android app can provide the solution by offering the ability to share issues between tenants and landlords which lead to rapid issue resolution and fewer misunderstandings. And to maintain security the government needs the information of the land owner and the tenant. This information will be maintained by the local police station to avoid any unwilling situation made by the local or other people. But we are not interested to give our information to the local police station by going there physically. So it's a demand of the age to develop an android app to manage and send information of the tenants to the nearest police station through the landlord.

1.2 Motivation

Now a days we see the argument between the owner of the house and the tenant because of disagree with each other which was agreed in the time of rent. They can't remember their terms and conditions. Sometimes they don't fill up the form because of laziness and thinking the harassment of police.

1.3 Objective

We want to make this app. This app will have contain these features.

1. To keep all essential information about the tenant.
2. To keep a legal document.
3. To provide security both of them.
4. To maintain security in an area.

1.4 Expected Outcome

By using this app a landlord can easily get all the information about the tenant such as national id number, picture and all previous record of the tenant .On the other hand tenant also easily fill up the information form which are easy and comfortable from the previous analog system and is saves their time. And one copy of the document will be sent by the landloard to the nearest police station that provides security.

1.5 Report Layout

Chapter 1: Introduction

In this chapter we will discuss about introduction, objectives, motivation and the expected outcome of our project.

Chapter 2: Background study

Chapter 2 contains the works related with our app, comparative studies and the challenges we have to face to implement this app.

Chapter 3: Requirement specification

In chapter 3 we will discuss about requirement specification such as business process modeling, requirement analysis and modeling, logical data model and design requirement.

Chapter 4: Design specification

Front-end design, back-end design interaction design and UX and implementation requirements are described in this section.

Chapter 5: Implementation and testing

In chapter 5 we discuss about the implementation of data base, implementation of front-end design, implementation of interaction, testing implementation and test results and reports.

Chapter 6: Conclusion and future scope

The subject to describe in chapter 6 are conclusion and the future scope of our project.

CHAPTER 2

BACKGROUND

2.1 Introduction

An android app can provide the solution by offering the ability to share issues and information between tenants and landlords which lead to rapid issue resolution and fewer misunderstanding. The goal of the app is to create a better relationship between tenants and landlord. The LandNant app is designed to support both tenants and landlords by enabling them to document and communicate repair issues, send automatic rent reminders, package notifications and emergency information. Effectively resolving the apartment issues is important to the tenant's long term future and this app will be an important tool for creating rental housing stability by helping tenants speak with greater credibility through initiating and documenting communications and building productive relationships with landlords.

2.2 Related Works

There are some android app almost similar with our app but these have many difference with our app. Some apps are:

Houzz App: This is an interior design application which provides the tools and information needed to complete home remodeling or decoration projects [1]. Rent Tracker: The app allows property manager to enter tenants and contractors [2]. The tenants can enter the rent payment and get a receipt by mail using this app. DIY Landlord: The app allows property manager to organize a property by unit, lease and tenant information [3]. The Inventory Manager: This app allows manager to keep track of items at their properties, such as furniture and sports supplies, by organizing them quickly and easily by location, category etc. [4].

2.3 Comparative Studies

Huzz app can only be used by the property manager. But in our app the user can be property manager or land lords and tenants. By using this app the land lord can change the entire information of the tenants but the tenants cannot update it without the approval of the landlord. In our app there are two log in page. One for admin and another for user. The user can be landlord or tenant. The tenant can pay his rent for the desired month. The payment information will be shared both tenant and landlord but they cannot update the payment information. Using our app the tenant get all information about the house that information provided by the landlord in the system. And the landlord can add new tenant in the system like DIY Landlord app [2]. In LandNant app the tenants can report, repair and maintains issues. They also have an option to take a picture of the issue and upload, which create a visual records accessible to the landlord instantly.

2.4 Scope of the problem

It's an android app, so if the landlord or the tenant does not have an android phone or if they do not know how to use an android phone it will be no effective for them. In this app the tenant needs to have same kind of bank account to pay the bill from his account and the payment get way method must be same at a given time period. We face problem to add the payment system in this app. If the landlord wants to send the tenant's information to the nearest police station, the nearest police station has to use this app and the landlord has to know the number of the nearest police station. If the tenant does not pay the bill in time he will be notify automatically but the app cannot handle it. The landlord has to give this notice by SMS. We want to add the tenant law in our app. But most of the people do not know how to implement this law in their day to day life.

2.5 Challenges

1. The main challenges is to language barrier. Because most of the people in our country do not know English.
2. We use English to develop this app.
3. The app user connected by using internet. So they must have net connection.
4. The next challenge is to implement this app in rural area.
5. The tenant has to keep track of all notification from the landlord as rent reminders.
6. Use a suitable database that store all the information and give security.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

BPM or Business Process Modeling is a process of constructing a structural view of a system or process. It includes some process, starts and symbol, condition as like a flow chart. In our system we construct a BPM model which shows how to send the tenant information to the landlord and if landlord verify the information then the information will send to the nearest police station. And it also shows that what will happen if the landlord dose not verify the information. If the information is correct then it will be saved in the tenant and landlord information database by the police station. Figure 3.1 shows business process modeling.

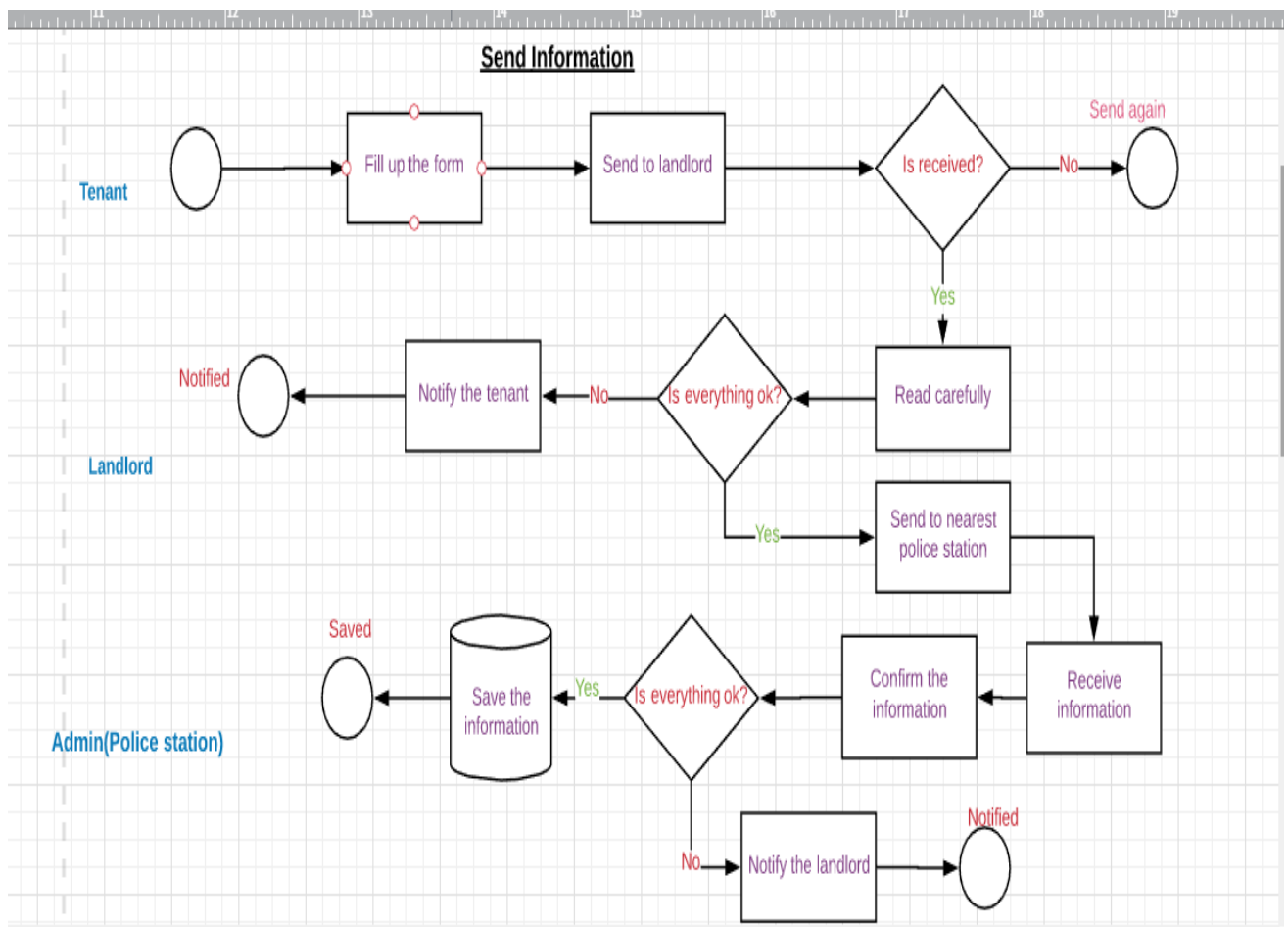


Figure 3.1: Business Process Model

3.2 Requirement Collection and Analysis

Requirement collection and analysis are very important term to develop a system or any kind of android application. It meets the goal of the user and the admin. As our application is concerned with the requirement of the landlord, tenant and the admin so, we went to them and tried to know what their requirements for this app are. Here the police station is the admin user. They all suggested to develop a simple app which they can use conveniently. To maintain the information of the landlord and the tenant this app will provide them a great service. Because they are deprive of facing the police. And they think that it will be a great process if they send their information to the police despite of going there.

3.3 Use Case Modeling

A use case model is a graphic description of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify and organizing system requirments. Figure 3.2 shows the use case modeling of the app.

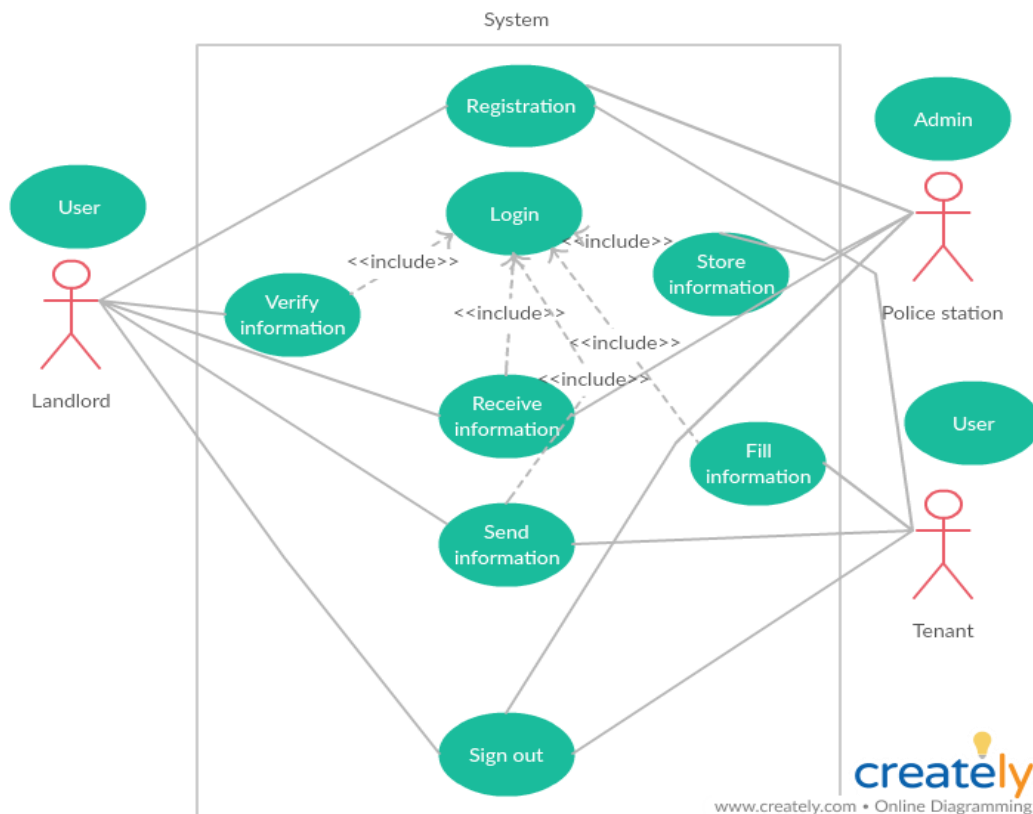


Figure 3.2: Use Case Model

Use Case : Admin Registration

Actor: Admin

Precondition: None

Primary Path:

1. Enter the name
2. Enter email address
3. Set a password

Alternate Path:

1. Enter valid email or password

Use Case: Admin Sign In

Actor: Admin

Precondition: Registration

Primary Path:

1. Enter Email
2. Enter Password
3. Click on “Sign In” button

Alternate Path:

1. Email or Password is incorrect

Use Case: Receive information

Actor: Admin

Precondition: Provide key

Primary Path:

1. Verify information

Alternate Path:

1. Information is not Received

Use Case: Store information

Actor: Admin

Precondition: Information received

Primary Path:

1. Check the accuracy
2. Manage database

Alternate path:

1. Inform the landlord

Use case: Admin sign out

Actor: Admin

Precondition: Sign In

Primary Path:

1. Click on “Sign Out” button

Alternate Path:

1. None

Use Case : Landlord Registration

Actor: Landlord

Precondition: None

Primary Path:

1. Enter the name
2. Enter email address
3. Set a password

Alternate Path:

1. Enter valid email or password

Use Case: Landlord Sign In

Actor: Landlord

Precondition: Registration

Primary Path:

1. Enter Email
2. Enter Password
3. Click on “Sign In” button

Alternate Path:

1. Email or Password is incorrect

Use Case: Create Account

Actor : landlord

Precndition: Sign In

Primary Path:

1. Enter Name
2. Enter a key
3. Enter password

Alternate Path:

1. Use valid key or password

Use Case: Verify And Send Information

Actor: Landlord

Precondition: Sign In as Landlord

Primary Path:

1. Enter landlord key
2. Verify information
3. Send information

Alternate Path:

1. Enter a valid landlord key

Use case: Landlord sign out

Actor: Landlord

Precondition: Sign In

Primary Path:

1. Click on “Sign Out” button

Alternate Path:

1. None

Use Case : Tenant Registration

Actor: Tenant

Precondition: None

Primary Path:

1. Enter the name
2. Enter email address
3. Set a password

Alternate Path:

1. Enter valid email or password

Use Case: Tenant Sign In

Actor: Tenant

Precondition: Registration

Primary Path:

1. Enter Email
2. Enter Password
3. Click on “Sign In” button

Alternate Path:

1. Email or Password is incorrect

Use Case: Fill Information

Actor: Tenant

Precondition: Sign In

Primary Path:

1. Set image
2. Enter full name
3. Enter father’s name
4. Enter date of birth
5. Select marital status

6. Enter parmanent address
7. Select profession
8. Select region
9. Enter email address
10. Enter national id number
11. Enter emergency contact information
12. Enter house Address
13. Save the information

Alternate Path:

1. Enter valid information

Use Case: Send Information

Actor: Tenant

Precondition: Fill up the form

Primary path:

1. Enter Landlord key
2. Click on “Send” button

Alternate Path:

1. The landlord key is incorrect

Use case: Tenant Sign Out

Actor: Tenant

Precondition: Sign In

Primary Path:

1. Click on “Sign Out” button

Alternate Path:

1. Null

3.4 Logical Data Model

Our project's logical data model has relational table named Admin, Landlord, Tenant and House information. Here, we can describe how these entities are connected with each other. The total relational model has shown on the following figure 3.3.

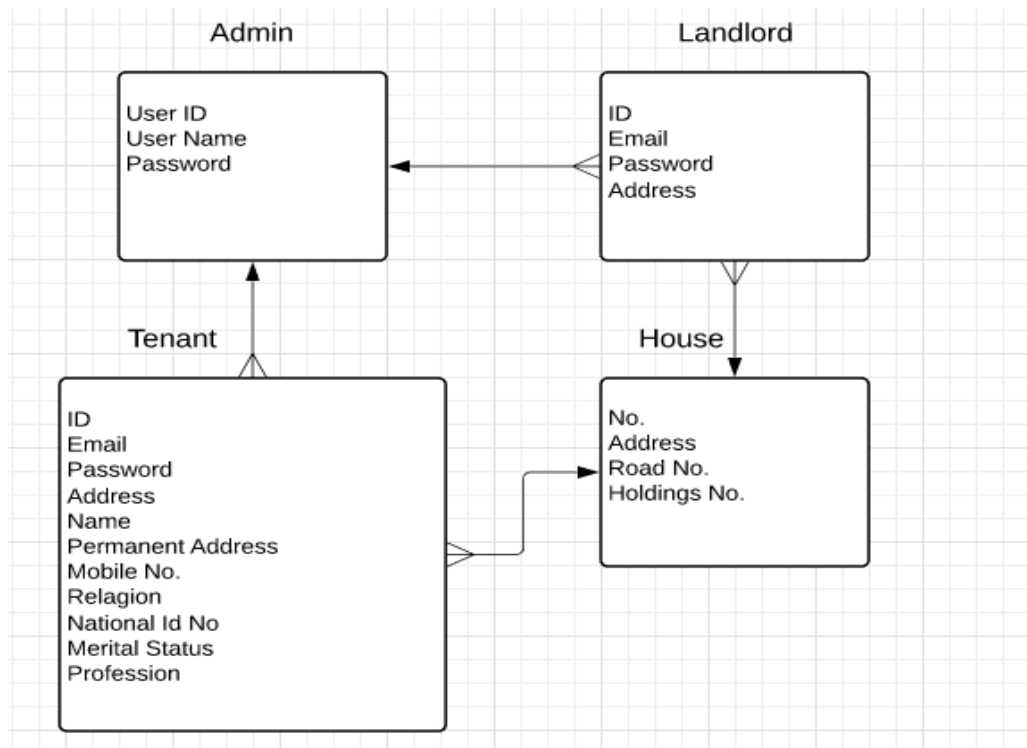


Figure 3.3: Logical Data Model

3.5 Design Requirement

A well design android app is pleasure to use. Our app provides service to the landlord, tenant and the government to maintain security in an area. Here, police station do their job as the admin of the database and the landlord and tenant are the field user. Though the users of our app are the landlord and the tenant so, we try to make the interface of the app convenient to the user. Well design of an android app is very essential but we should concentrate to the output design of the app first. User typically use this app to maintain their personal information which is required by the government. So simplifying this app by making user interface simple is our primary responsibility.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

Front page of an android application must draw the attention of the user. To make our application attractive to the user we use some Graphical User Interface (GUI). We design our project with easily understandable GUI so that, the user can easily access the app. In the front page we simply use a registration option and a sign in option. To sign in this app the user must complete the registration first.

4.2 Back-end Design

Back end design is only seen by the programmer. How will the application interact with the user is implemented in the back end. User cannot interact with the back end design. As our project maintain the information that send by the landlord to the police station via landlord so, we need to create a database to store these information. We also provide the unique id of the user. We use fire base to create the database. The following figures describes how we designed the back end of our project. Figure 4.1 shows the back end design for the user. This authentication function shows all the users who registered in the database by using our app. When a user sign in this app, a database is automatically generated for the user.

The screenshot shows the 'Authentication' section of the LandNant application. At the top, there are navigation links for 'USERS', 'SIGN-IN METHOD', 'TEMPLATES', and 'USAGE'. Below this is a search bar with the text 'Search by email address, phone number, or user UID' and an 'ADD USER' button. The main content is a table listing users with the following columns: Identifier, Providers, Created, Signed In, and User UID.

Identifier	Providers	Created	Signed In	User UID ↑
tusar123@gmail.com	✉	Mar 20, 2018	Mar 20, 2018	019ZjG7DwTOWsNrDZvj9f9jNYR2
rana@gmail.com	✉	Mar 21, 2018	Mar 22, 2018	1Wjj7UjDRRclzjhJd85ubvoleYF2
tusar@gmail.com	✉	Mar 21, 2018	Mar 21, 2018	2oqJnlJ477Q5nx06vOomUeUxleA3
liton@gmail.com	✉	Mar 22, 2018	Mar 22, 2018	3RlrpyVeSHUEy1KzP5SD3uT7ZWl3
masud3526@gmail.com	✉	Mar 23, 2018	Mar 23, 2018	BYTpbDHESON7e0mtKSU06lsdgx...
mshonyw60@gmail.com	✉	Mar 23, 2018	Mar 23, 2018	HMuQmwp8nxNuK6uaJUWl17Fcx...
susanto3795@gmail.com	✉	Mar 20, 2018	Mar 20, 2018	HczM9KbjQeSYbQKxjPt7q6VTJA3

Figure 4.1: User List

Figure 4.2 shows all the sub database we create in our project. All the sub database contains different types of information of the user. It provides the personal unique key for the each user by which the information of individuals are managed.

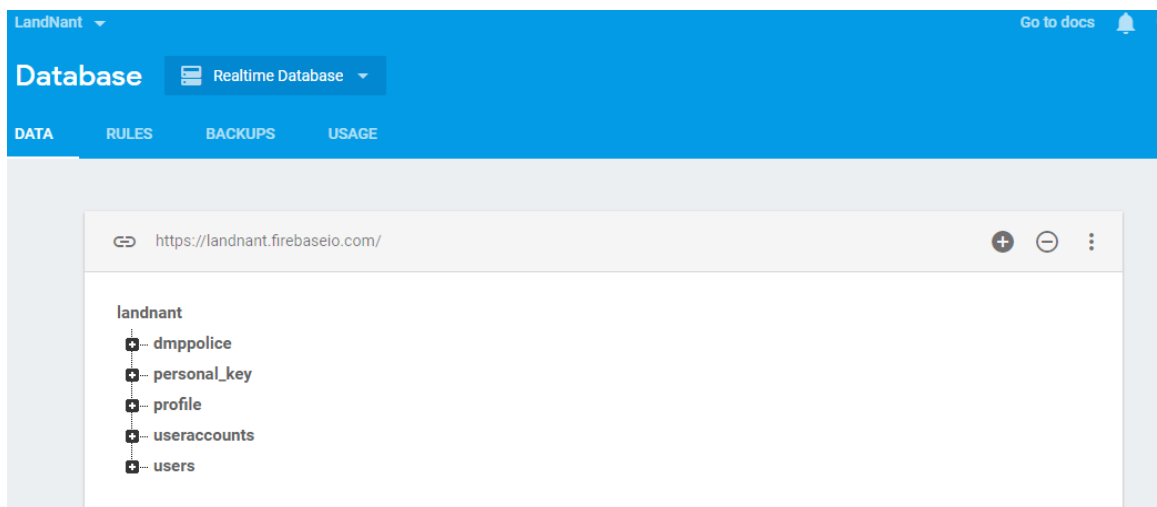


Figure 4.2: Database Creation

Figure 4.3 shows the table of image that are provided by the tenant. It contains every details of the tenant with their image.

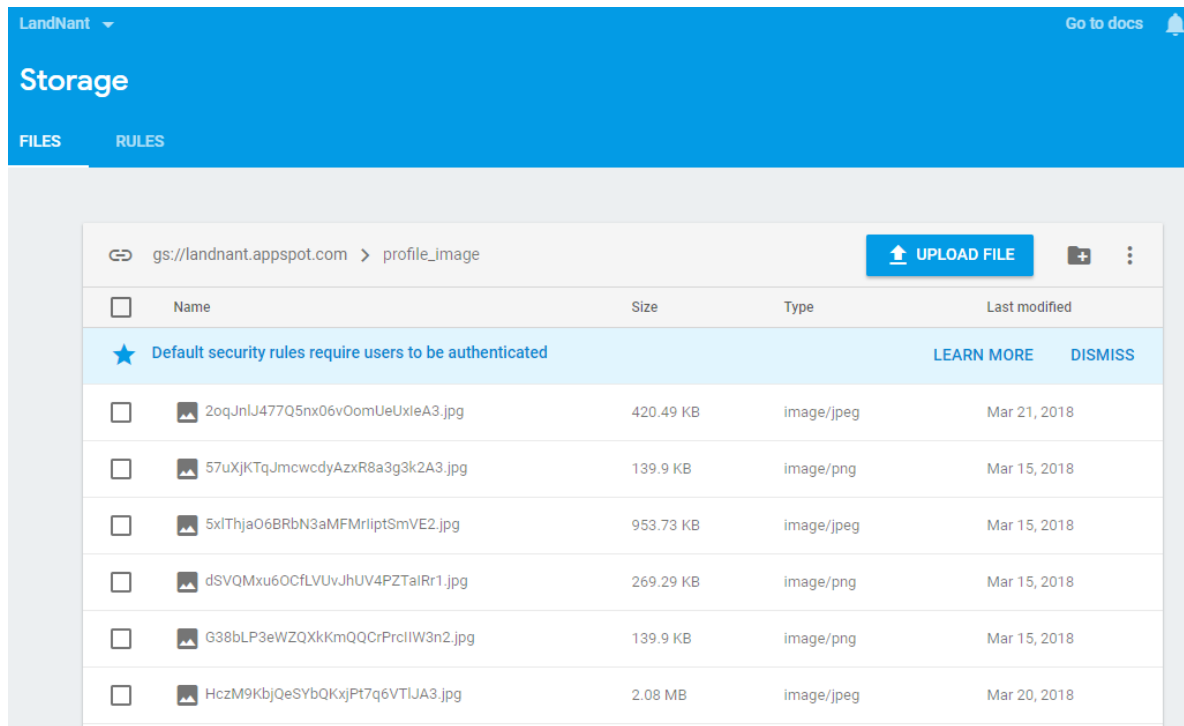


Figure 4.3: Image of Tenant

4.3 Interaction Design and UX

Interaction design describes the relationship between the application and the user. How much an android application will popular among the user that depends on the interaction of the app with the user. If the app is convenient to use and it provides a great outcomes to the user then it will gain the popularity. Experience is very essential for interaction design. Because mainly the user experiences focuses on the experience between the user and the application. From this experience we tried to make most of the features of our app that can interact with the user. Though ours is a security based android app so we tried to provide great security to them by using the registration option which required valid email id and password of the user.

UX design or user experience design is the most challenging part for a programmer. Always the need to focus on the experience of user of using android app and how they satisfy with a product. We pay attention on this sector and tried to implement an application that has satisfactory interface which should be accepted by the user.

4.4 Implementation Requirement

To implement a project we need different types of tools, components, and software. We also need some tools and component to implement our project. Ours is an android based application project so we used android based design tools and components to implement our application. Here is a short description of the tools and components that required to implement our application.

4.4.1 Android Studio

Android studio is a popular platform to implement any kind of android application. It provides the great features to the programmer to implement an android app.it provides almost all the components and tools and frame works for developing android app easily and quickly. We use android studio as our application IDE. It's an open source platform so, programmer prefers android studio to develop various types android application.

4.4.2 Emulator

Emulator enables to run an application on a physical device in which the application is not implemented. It runs the same code in the device based on actual device. Emulator helps us to use this in different version of android phone.

4.4.3 Android SDK

SDK is very essential tool to develop an android application. It contains all the design and component libraries for development and the simulator to test the application. We use java development kit in the java platform with the SDK.

4.4.4 Firebase

Fire base is the most popular real time database. We need a database to store the information of the user. We use firebase as the main database of our application to store and maintain the information of the user. We also had to maintain the security of the personal information of the tenant. It synchronizes the data in real time to every connected user in the application. If we build an apps that has a cross platform with iOS, Android, JavaScript SDKs, all of the user use the real time database instance and the user will automatically receive with the newest data by firebase.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

In this section we describe how we implemented our database. As we describe in the previous chapter that we use fire base as our main database. Fire base is a real time database so there is no need to use SQL query to perform the action. Fire base also has some other features like as hosting, crash reporting, cloud storage authentication, hosting and so on. How we use fire base to maintain our data is described below with proper diagram.

To store information in the database the user has to register and sign in. To sign in in our database fire base gives several choices to the user. In our app user can sign in by their email and in some case they can also use their phone number. Figure 5.1 shows the authentication option.

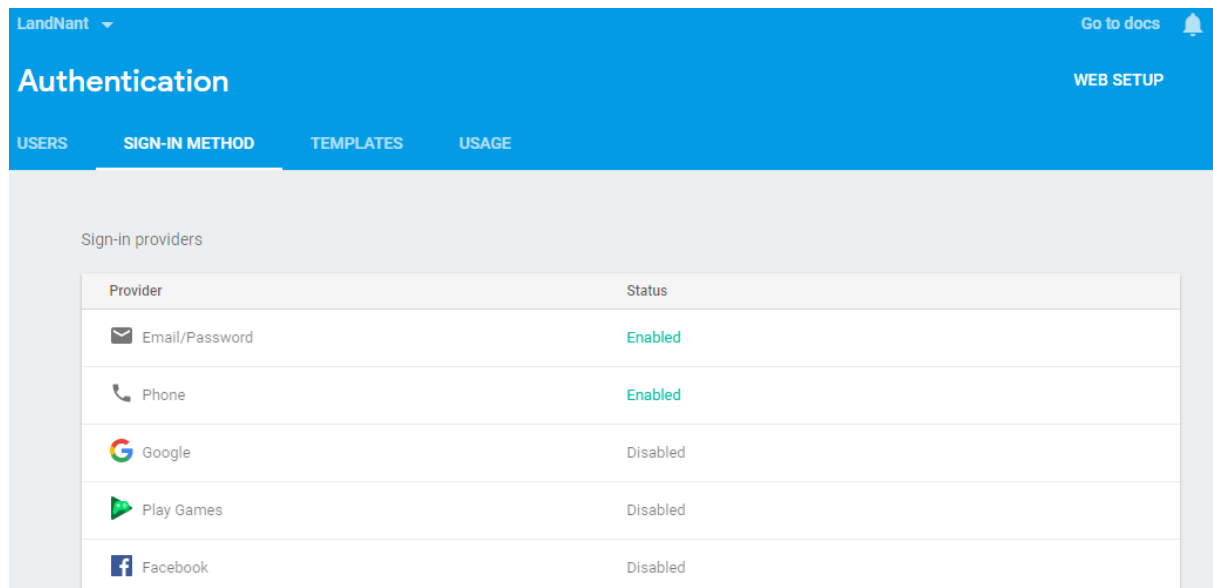


Figure 5.1: Authentication Choice

To sign in at the farther time if the user does not provide the valid user id or password then he cannot sign in. Firebase use an authentication rule if the email is valid or invalid. Figure 5.2 shows the authentication rule.

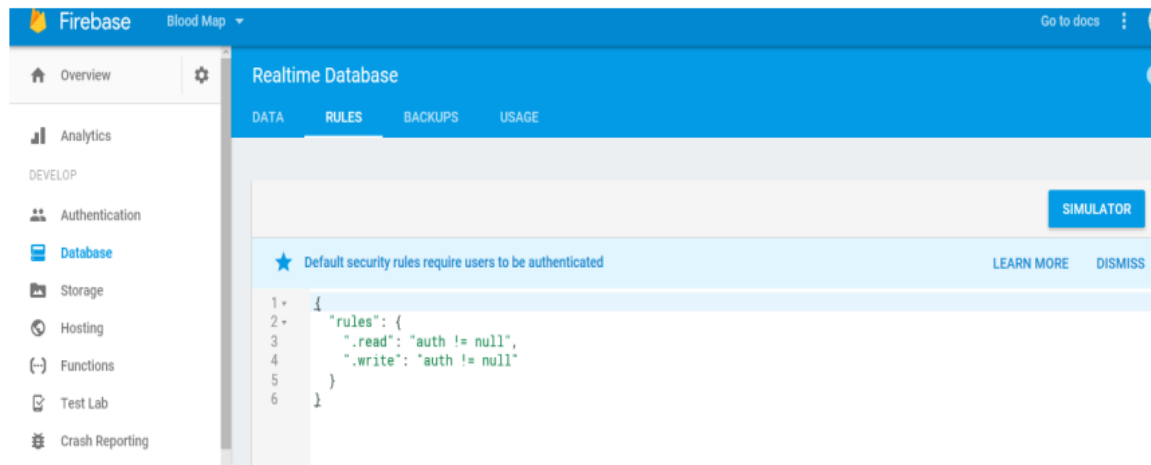


Figure 5.2: Authentication Rule

The user of our app are admin, landlord and the tenant. All the registered user have their own user table with unique user id. The information are stored in each unique table through the unique id. When a user sign in the database fire base gives a unique id. Landlord database and the tenant database are different. Each landlord has their own personal key. Figure 5.3 shows landlord information.

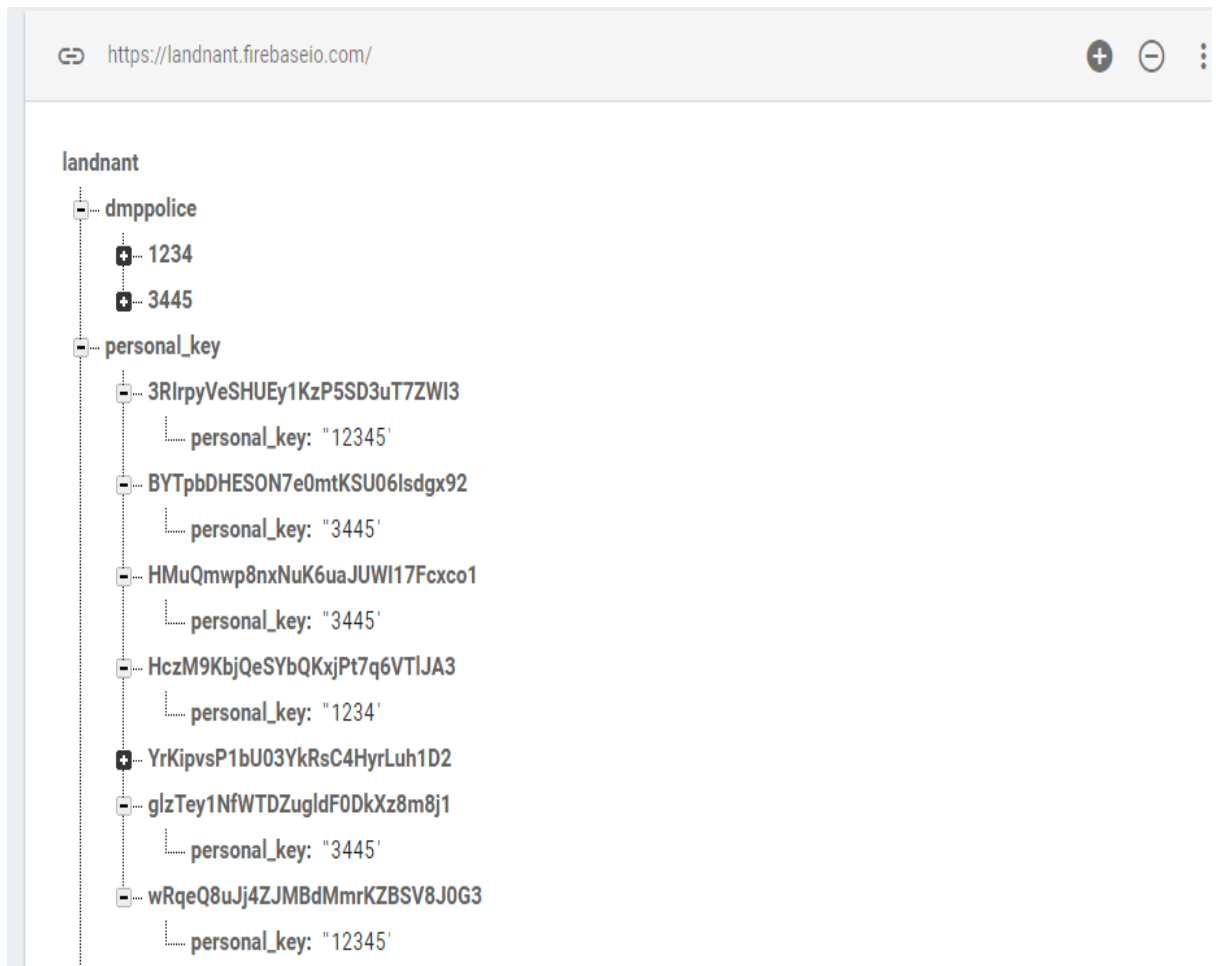


Figure 5.3: Landlord Information

A tenant sign in this database using the personal key of his landlord and firebase creates a sub key for the tenant which is also unique. Figure 5.4 shows the tenant information.

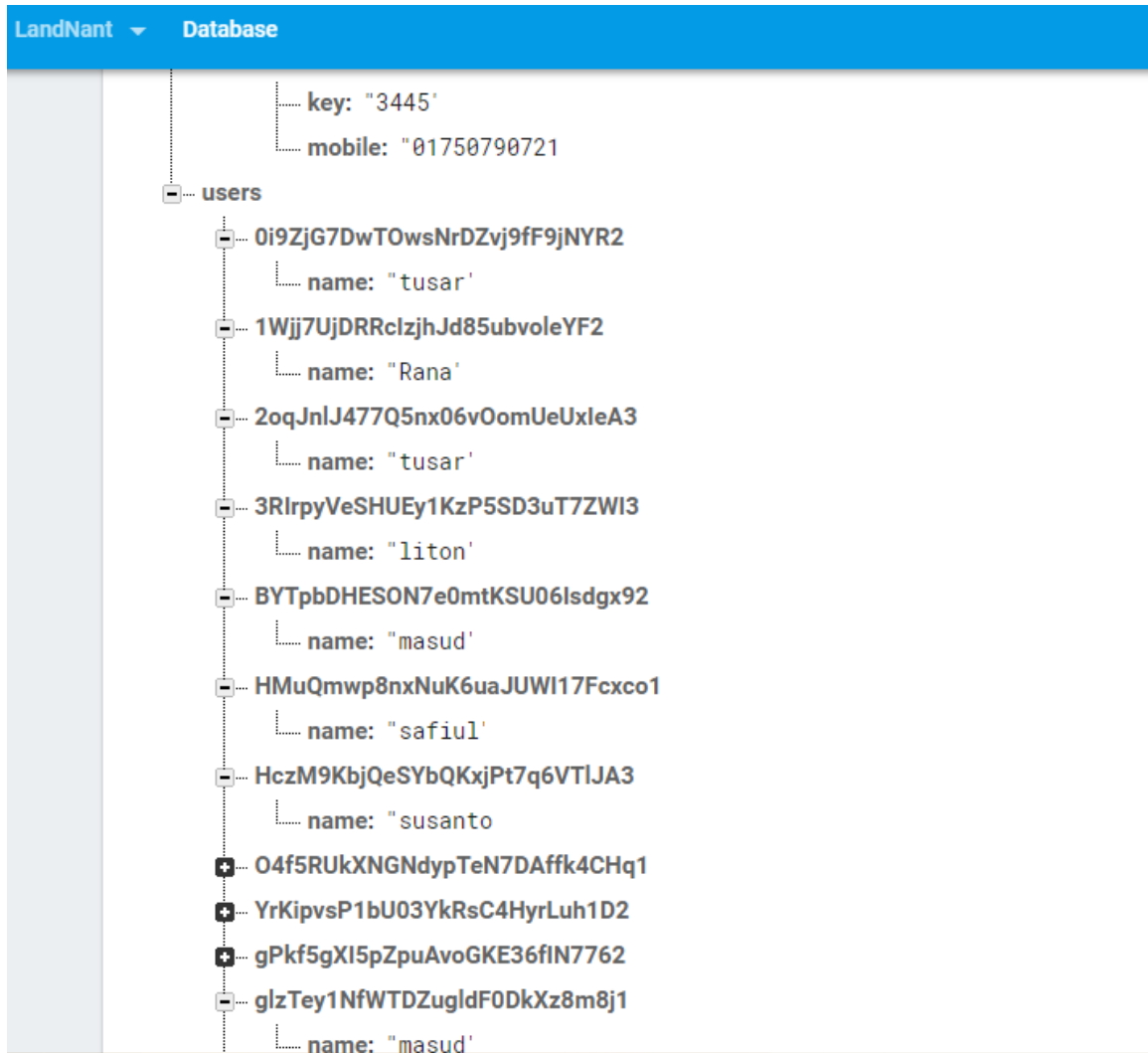


Figure 5.4: Tenant Information

5.2 Implementation of Front-end Design

Only the front end design is visible to the user. So it should affectionate to look at. It also should user friendly and convenient to use. We already have said that the user of our app are landlord, tenant and the admin. At the first page of our app we arrange registration and sign in option. Figure 5.5 shows how we implement the front-end of our app.



Figure 5.5: Home page

If any user wants to use our app at first he or she needs to register using register option. After clicking on the register option the following page will appear to the user. In figure we can see how a user complete the registration to use our app.

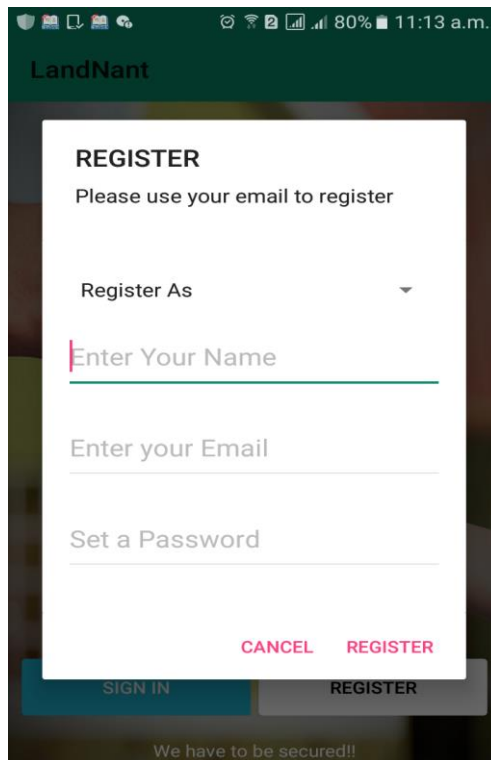


Figure 5.6: User Registration

User can registrar into the app by using name, email and password. After registration the user can sign in into the app as a landlord, tenant or admin and the have provide their email address and password here. Figure 5.7 shows user sign in option.

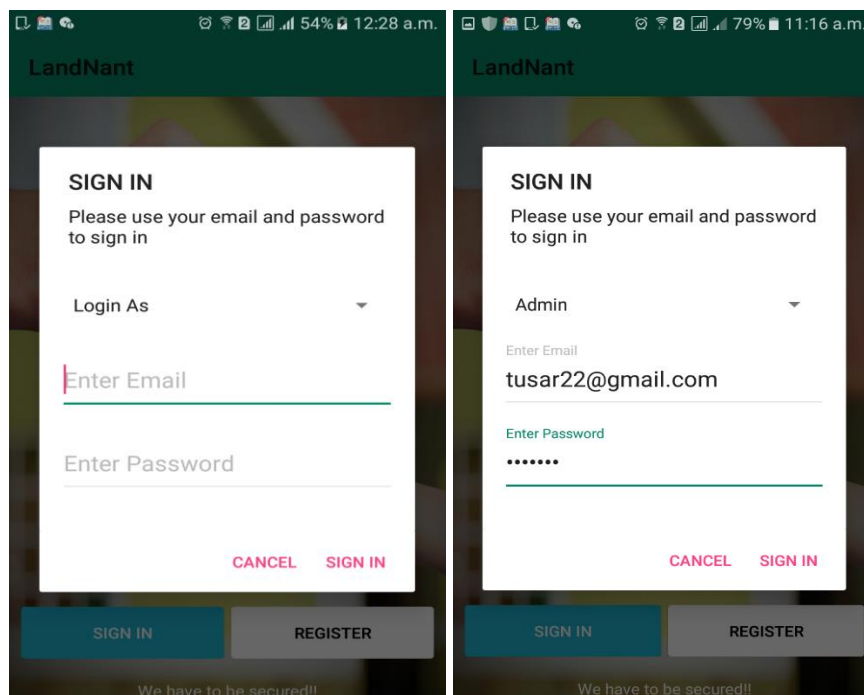


Figure 5.7 :Sign In

If the user sign as a tenant then he needs to choice **Tenant** option. In the same process the landlord and admin can sign in in the system. In figure 5.8 tenant will able create an account under a landlord key or sign in. to create a new tenant id the tenant has to set the mobile number, key and a verification key that is provided by the landlord. If the tenant verify himself as a tenant using the landlord key then he can simply sign up and fill his information and send to the landlord to verify.

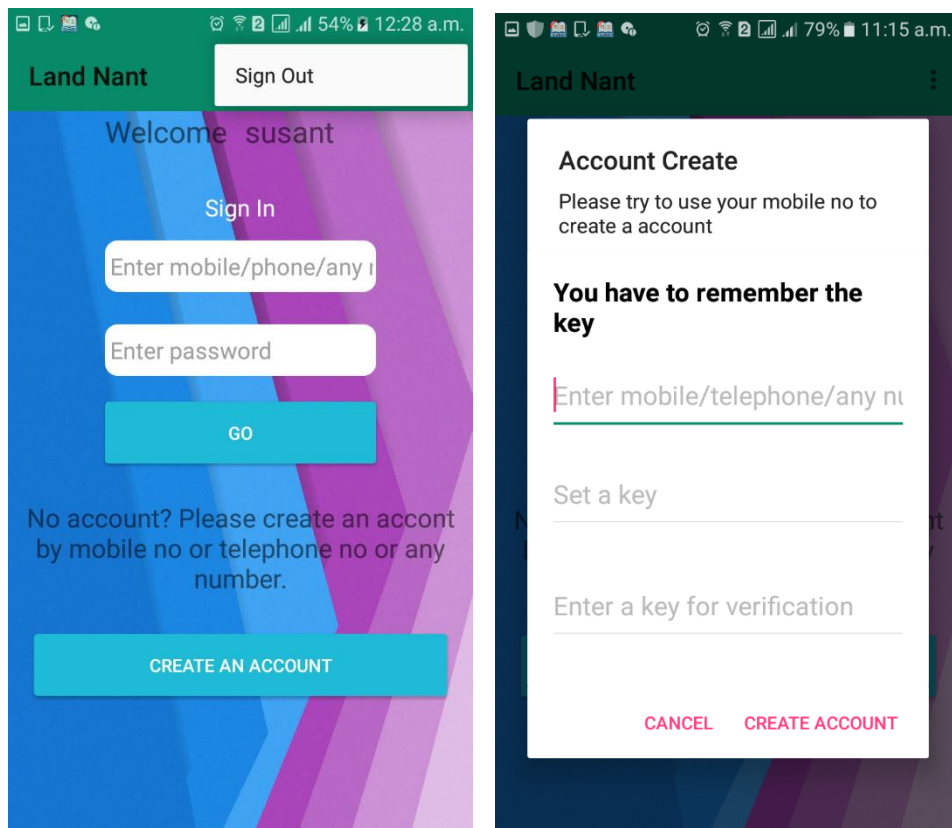
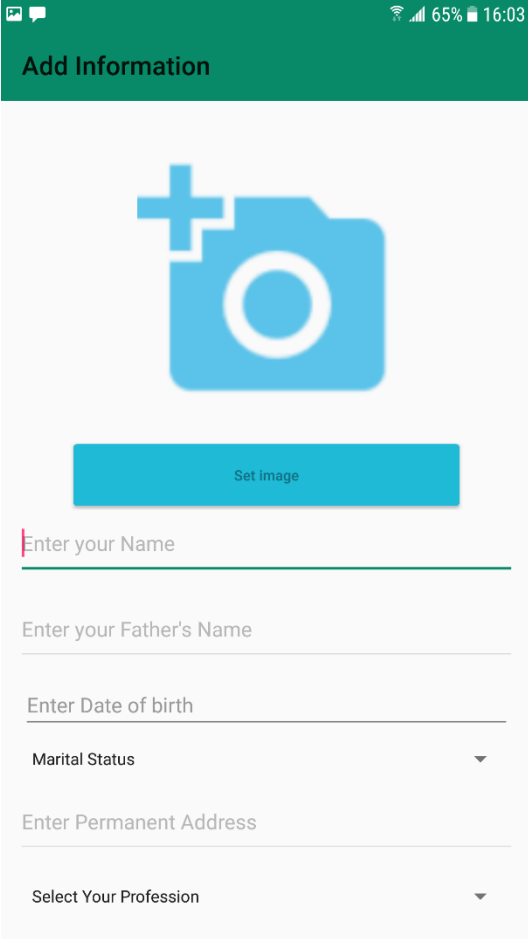


Figure 5.8: Account Creation

After login under a landlord the tenant can send his information to his landlord to verify. Figure 5.9 shows the form that the tenant and the landlord have to fill up.



The screenshot shows a mobile application interface with a green header bar containing the text "Add Information". Below the header is a large blue icon of a camera with a plus sign, indicating a photo upload function. Underneath the icon is a blue button labeled "Set Image". The form consists of several input fields: "Enter your Name", "Enter your Father's Name", "Enter Date of birth", "Marital Status" (a dropdown menu), "Enter Permanent Address", and "Select Your Profession" (a dropdown menu). The status bar at the top shows a signal strength icon, a Wi-Fi icon, 65% battery, and the time 16:03.

Figure 5.9: Information Form

Landlord can also sign in and create an account as landlord. In this page landlord generates a key which will provide to the tenant to send their information for verification. Landlord can see the number of tenant he has and who are verified by him who are waiting for verification. Figure 5.10 shows the number of tenant the landlord has.

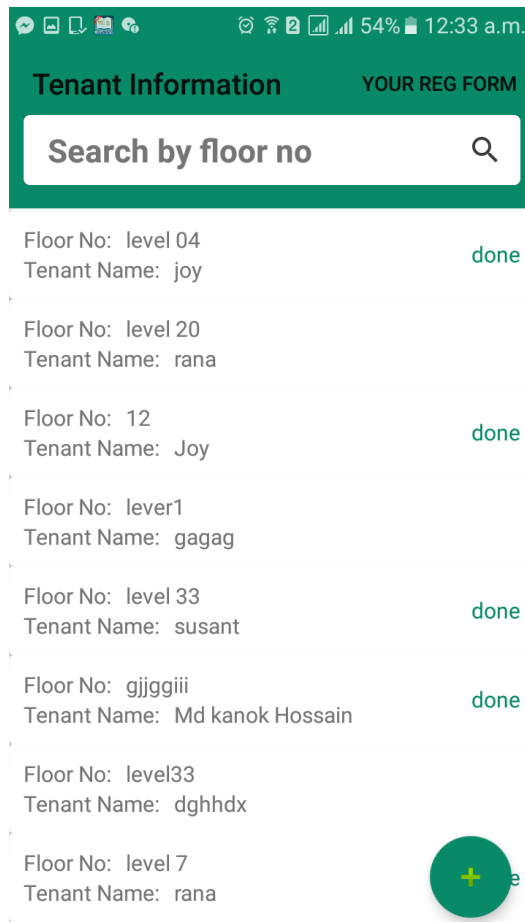


Figure 5.10: Tenant Information

By selecting a tenant the landlord can verify him and save the tenant's personal information in the database. To verify the tenant the landlord need to inter his key and if the key match then the verification and send option will appear. Otherwise he cannot save the information of his tenant in the database. Figure 5.11 shows the verification and send method.

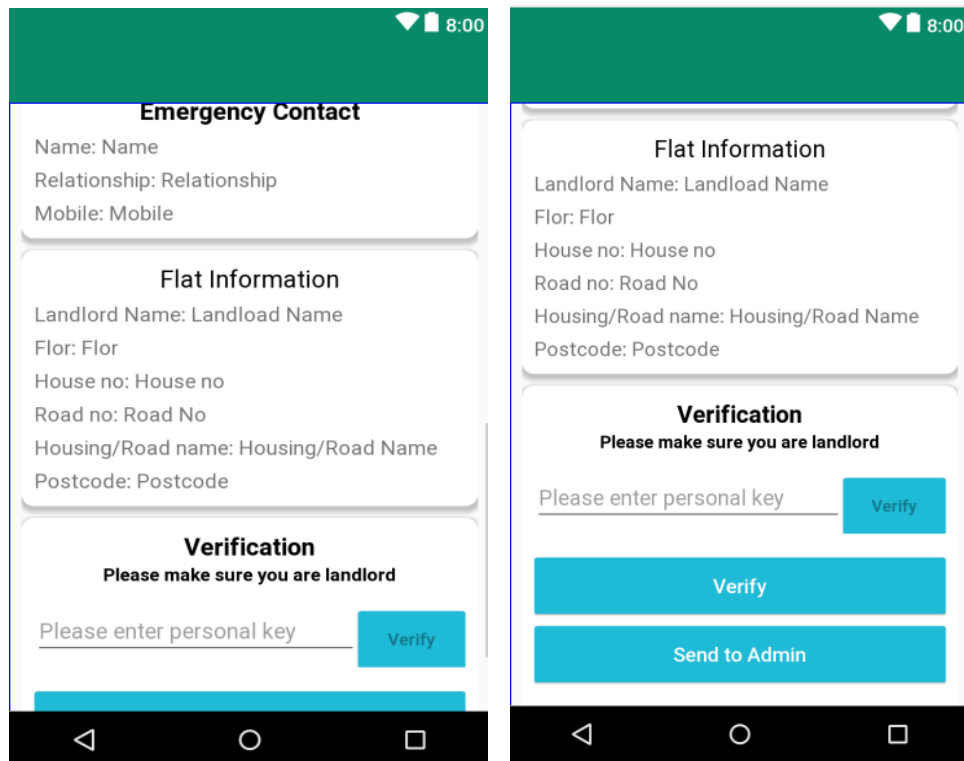


Figure 5.11: Tenant Verification

5.3 Implementation of Interaction

We implement our app for the user. Interaction makes a system attractive and popular to the user. So it's very important to interact with the user so that it can meet up their need. To interact with the user we try to make our app simple and easily usable. We implement our app with responsive UI for better user experience. To implement an easy interface we use easily understandable icon, text and button option.

Our app is successfully implemented with necessary equipment that can easily interact with the user. It has an impressive interaction with the user.

5.4 Testing implementation

Our goal is to design a series of test cases that has a high likelihood of finding errors. To uncover the errors software techniques are used. This techniques provide systematic guidance for designing test that

1. Exercise the internal logic of software components and
2. Exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

Table 5.1: Test case shows the following test case has been done for several time to detect errors.

Sno	Test Case Id	Test Case Name	Test Case Description	Step	Expected Result	Actual Result	Test Case Status Pass/Fail
1	Login Admin	Validate login	To verify that login name on login page	Enter the login name and password and click Sign in button	Login successful or an error message "Failed Sign in" must be shown	Login success-ful	Pass
2	Login User	Validate login	To verify that login name on login page	Enter the login name and password and click Sign in button	Login successful or an error message "Failed Sign in" must be shown	Login success-ful	Pass

3	Password	Validate Password	To verify that password on login page	Enter password and login name and click Sign in Button	An error message "password invalid" must be displayed	Passwo-rd is invalid	Fail
4	Sign Out	Sign out from the system	To log out from the account	Click on the Sign out button	Sign out from the system	Sign out Successful	Pass

5.5 Test Result and Reports

Test Reports represent the result of the test in a formal way. Report contains the data which we evaluated in a professional and organized manner. Report describe the operating condition and shows the result with test objective. By analyzing the testing report we can say whether the app is ready to use or not. In table 5.1 we describe the test case no, test case id, test case description, step of the test case, expected result and what is the actual result is and the test case status which can be pass or fail. Using these criteria we took several number of tests and got the accurate result. The accuracy is 100%. So, we can say that our app is error free and user acceptable.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

Now-a-days maintain security in a city is a big problem. To maintain security the government should have the information about the people of an area. Our app is implemented to maintain information about the people. So it will render a great effort to the government and the people of an area. In the age of computer science analog system for maintaining information is an old process. People can serve their personal information by using their android phone. It will also save their time. Using our app government can easily maintain all the information at a time in a single database.

If the user fill free to use our app and accept our work in a cordial heart then the implementation of the application will successful.

6.2 Future Scope

We cannot implement some important feature in our app. But our exertion won't be stop. We want to add some more features that will bring benefit to the landlord and the tenant. The main field user of our app are the tenant and the landlord. This app also gives service to them for communication. We want to add a payment system in our application. Using this payment method tenant can pay the rent in every month. The app will also keep a record of the transition. We are concern to implement a feature where landlord can give add for rent and the tenant can easily find his suitable place for living.

REFERENCES

- [1] <http://www.bickerstaffparham.com/posts/8-useful-mobile-apps-for-property-managers>
[last accessed: Mar 15, 2018]
- [2] <http://www.google.com/search?q=starpoint+tenant+screening+reviews&oq=starpoint+tenant&aqs=chrome.1.69i57j0l2.8707j0j7&sourceid=chrome&ie=UTF-8> [last accessed: Mar 15, 2018]
- [3] <http://www.balancepointpayroll.com/7-must-have-mobile-apps-for-property-managers/>
[last accessed: Mar 20, 2018]
- [4] <https://www.softwareadvice.com/property/management-apps-comparison/> [last accessed: Mar 20, 2018]

APPENDIX

Appendix A: Project Reflection

We started our journey from Fall 2017 to implement this android application. We tried to make a user friendly interface of our project. The main feature of our project is to sending the tenants information to the police station through the landlord. Our app will give the better service to maintain security and manage crime in a city area. It's very essential to keep the information of the landlord and the tenant to the government. Naw-a-days people are more likely to use the smartphone in their practical life. This app provides them a service in practical life. To implement this application first we build a model of our application then we implement our app step by step. After a long journey and hard work finally we able to reach our goal.

Our app provides service both of the user and the government. It also reduce time and the use of paper. Information can also be stored in a single database which helps to reduce data redundancy. Using this app people can easily send their personal information to save in the database. So, they will aware of providing information to the government and will not deprive of giving information. We believe that our application will be a positive and effective for users.

Appendix B: Related Diagrams

To implement our proposed application at first we build a model of our app. We use some diagrams and use case model to analyze how we can implement our app. The first diagram we create the use case diagram. From the use case diagram we can know about user of our app and what is their activity in the app. In our LandNant app there are three users. They are landlord, tenant and the police station. Police station acts as the admin. We also use business process model which acts a flow chart. In this section we describe the activity of the user and how they interact with the system. We implement our application using android studio and firebase. In this section we add some diagrams from the user interface and back-end. Figure A.1 shows the use case diagram.

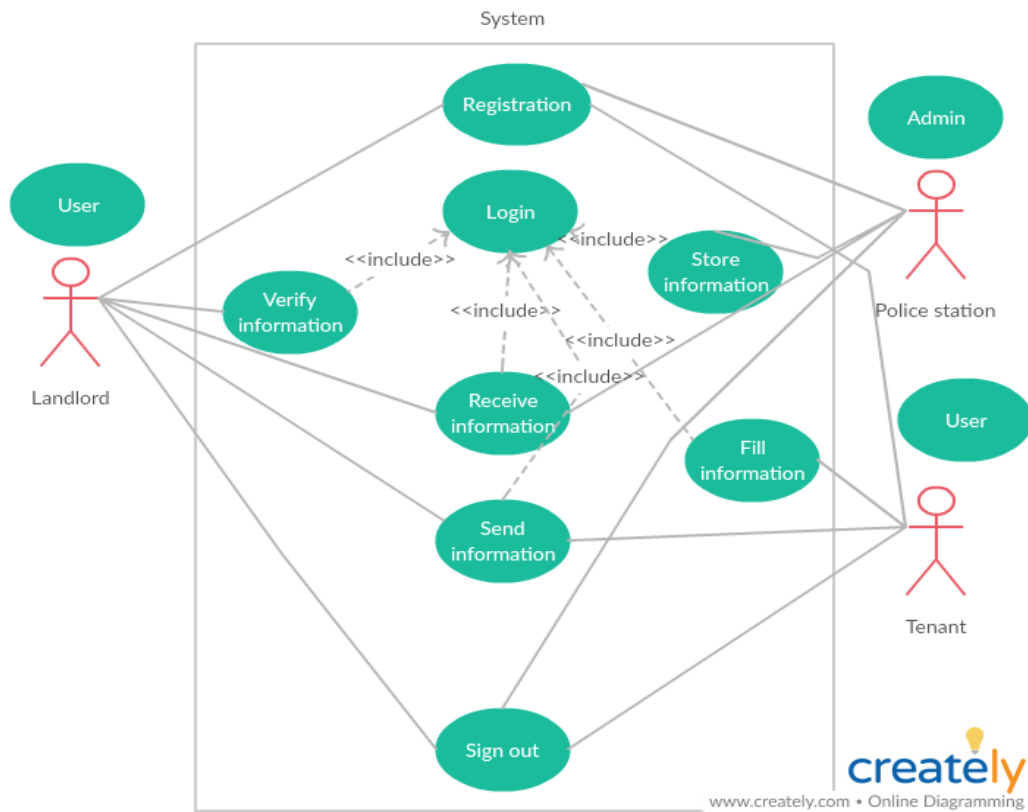


Figure A.1: Use Case Model

Figure A.2 shows the database of our application.

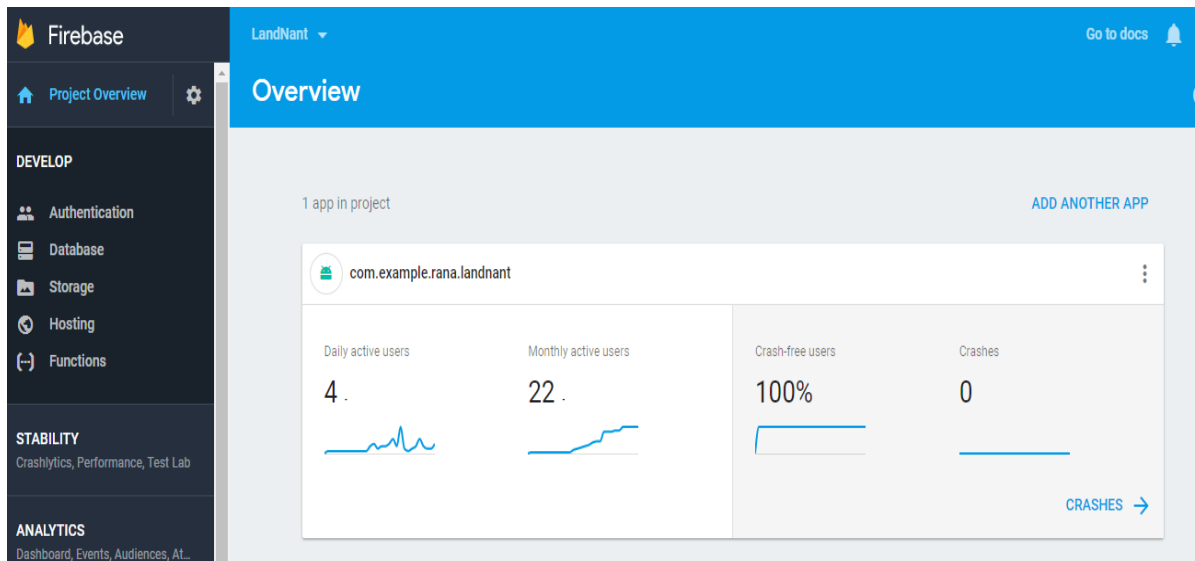


Figure A.2: Database

PLAGIARISM REPORT

Checked by: <https://my.plagamme.com/>

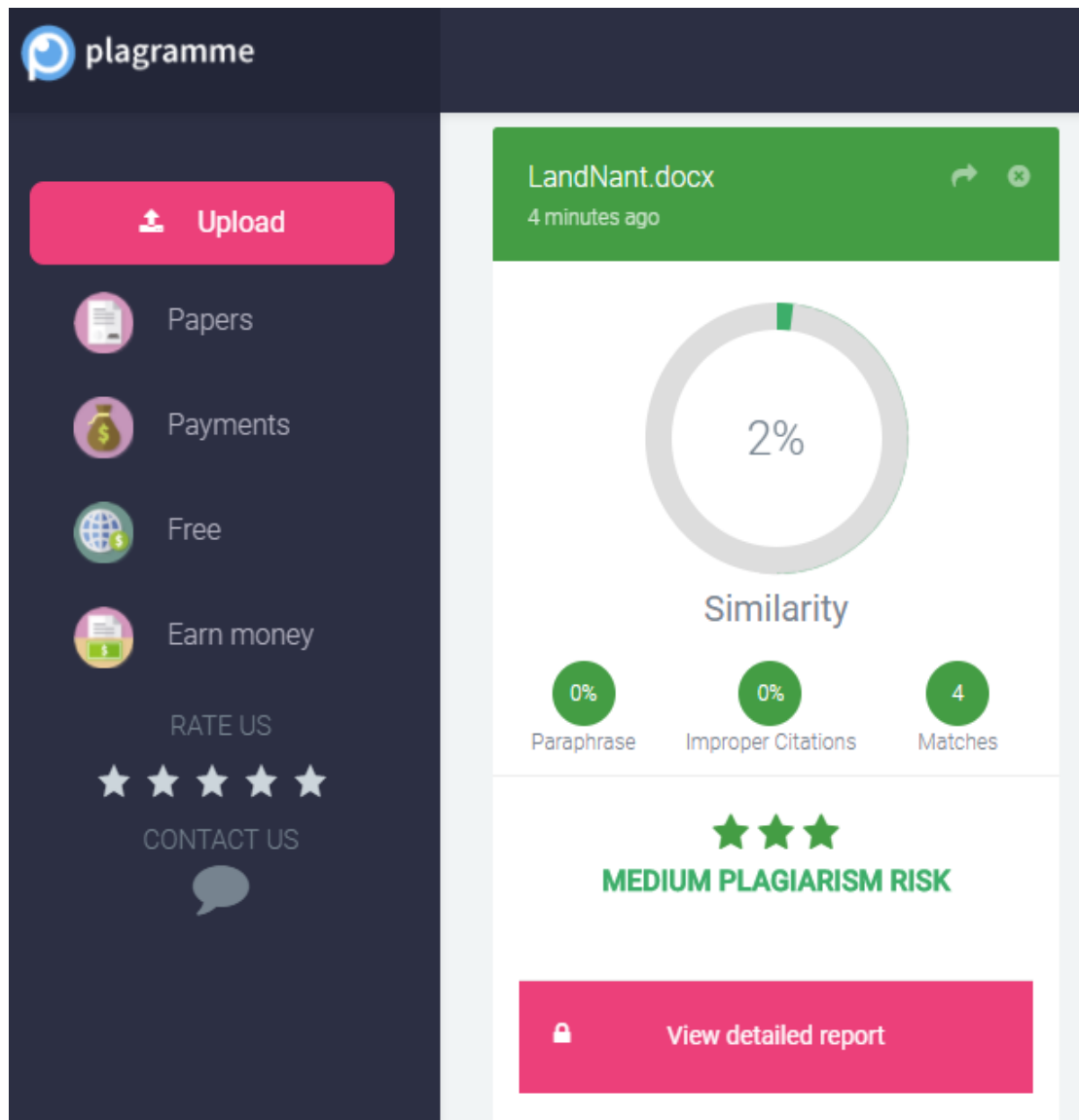


Figure: Plagiarism Report