

**VIVIR - A HOUSE RENTAL AND TENANCY
APPLICATION**

A THESIS

Submitted by

YOGENDRA - M150480CA

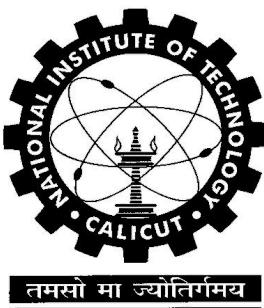
In partial fulfillment for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA)

Under the guidance of

Dr SUDEEP K S



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

National Institute of Technology Calicut
NIT Campus PO, Calicut
Kerala, India - 673601

MAY 2018

ACKNOWLEDGEMENT

Inspiration and guidance are invaluable in every aspect of life, especially in the fields of academics, which I have received from my guide **Dr. Sudeep K S** . I would like to thank him for his endless contributions of time, effort, valuable guidance and encouragement he has given to me.

I express my sincere thanks and gratitude to my additional mentor **Ms. Athira P K** for guiding and helping me during the whole project. Her guidance helped me a lot in comprehending the things in a better way.

Lastly It's family and friends specially Arsh, Avnish, Fahad and Sarvesh who can not be forgotten for their continuous support and motivations. Their help and supports are invaluable.

Date: 7th May 2018

YOGENDRA

Reg. No - M150480CA

DECLARATION

I hereby declare that the report entitled “House Rental and Tenancy Application (Vivir)” submitted by me, for the partial fulfilment of the degree of Master of Computer Applications is a record of the work carried out by me under the supervision of **Dr. Sudeep K S .**

I further declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature :

Place : Calicut

Date : 07/05/2018

Name : Yogendra

Reg. No : M150480CA

CERTIFICATE

This is to certify that the report entitled "House Rental and Tenancy Application - Vivir " submitted by Mr. Yogendra(M150480CA) to the National Institute of Technology Calicut towards partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications(MCA) is a bona fide record of the work carried out by him under my supervision and guidance.

Dr. SUDEEP K S
Assistant Professor
Department of Computer Science
and Engineering.
NIT Calicut

Place : Calicut

Date : 07/05/2018

Signature of Head of the Department

TABLE OF CONTENTS

	Page
Chapter 1 Introduction	
1.1 Purpose	9
1.2 Problem Definition	9
1.3 Background	10
1.4 Motivation	11
Chapter 2 Literature Survey	12
Chapter 3 Design and Implementation	
3.1 System Features	13
3.1.1 Common Features	13
3.1.2 Tenant Specific Features	14
3.1.3 Owner Specific Features	16
3.2 Non Functional Requirements	18
3.2.1 Performance Requirements	18
3.2.2 Security Requirements	18
3.2.3 Software Quality Attributes	18
3.3 Design	19
3.3.1 Activity Diagram	19
3.3.2 Use case Diagram	21
3.3.3 ER Diagram and Database Description	23
3.4 Operating and Development Environment	29
3.5 Implementation and Product Demo	30
Chapter 4 Conclusion	35
Chapter 5 Glossary	36
Chapter 6 References	37

ABSTRACT

House Rental and Tenancy is booking of apartment/flat and dealing all major activities during the whole stay at that apartment/flat. Every time when we shift (for long time) to a new city we face this rental and tenancy problem.

Searching and booking a flat is mostly one of the hectic task. Tenant has to search an appropriate flat and after that he can be able to book that. It can be time consuming and difficult to find the flat. Even after booking the flat he/she has to maintain many records and have to deal with other activities. As per Owner(flat) perspective reaching to the customer is also difficult and it can involve more expenditure.

So in the Era of Technology and smartphone dealing with all these problems can be more easier and efficient. “Vivir”- House rental and tenancy application will help in performing all these activities. It just need an android phone and internet connection to deal with all these problems.

This application involves two kind of users Tenant and Owner. Owner can add their apartment/flat details. After that Tenant can search for the apartment/flats and can request for booking the flat. After approval of the request from Owner and completing other formalities he can stay at that flat. During the whole stay he/she can manage related activities with this application itself.

Working with this application makes the House rental more easier and efficient. Tenant can search and book the flat more easily and with more number of choices of selection. For owner reaching to the customer(Tenant) is more easier and cost effective. It will also help in maintaining the good relationships between Tenant and Owner and would be environment friendly too.

LIST OF FIGURES

Figure No.	Figure	Page Number
Figure 1.1	Tenant	9
Figure 1.2	Owner	10
Figure 3.1	Activity Diagram for owner	19
Figure 3.2	Activity Diagram for Tenant	20
Figure 3.3	Use case Diagram for Owner	21
Figure 3.4	Use Case Diagram for Tenant	22
Figure 3.5	ER Diagram	23

LIST OF TABLES

#	Table Name	Page Number
I	user	25
II	tenant	25
III	owner	25
IV	request	26
V	complain	26
VI	notifications	27
VII	rent	27
VIII	booking	28
IX	apartment	28

1. INTRODUCTION

1.1 Purpose

House Rental and Tenancy application-” Vivir” is an android application . It aims to solve the daily life house rental problem. Its objective is to provide easy ,efficient , hassle free and cost effective services.

1.2 Problem Definition

Searching and Booking flat manually is a tedious task. House Rental and Tenancy application will deal this problem with ease and efficiency. A brief overview of it can be given as follows.

Major User – Tenant(Renter) and Owner

It is an android application for booking of apartment/flat and making easy stay for an user. Its working for each user can be described as follows -

- **Tenant -**

“Tenant” can search and book the flat. After booking he can do all transaction during the whole stay at that flat through this application. He can complain to owner by this application. He will get notification for all major activities. He has to request to Owner before leaving the flat.

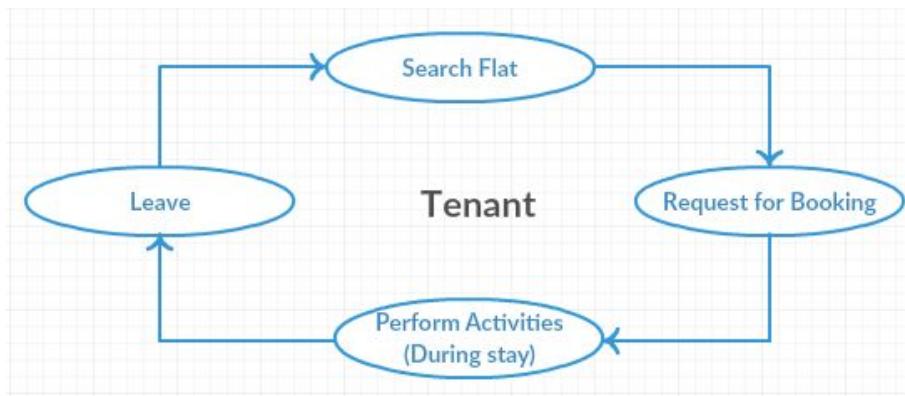


Figure : 1.1

- **Owner -**

Owner can upload and update flat/room details that he want to lend. He will approve all request of booking, rent and leaving. He will respond to complaints through this application. He can view notification of activities and request.

Each user have their own dashboard section. From there they can set/edit their profile and change password.

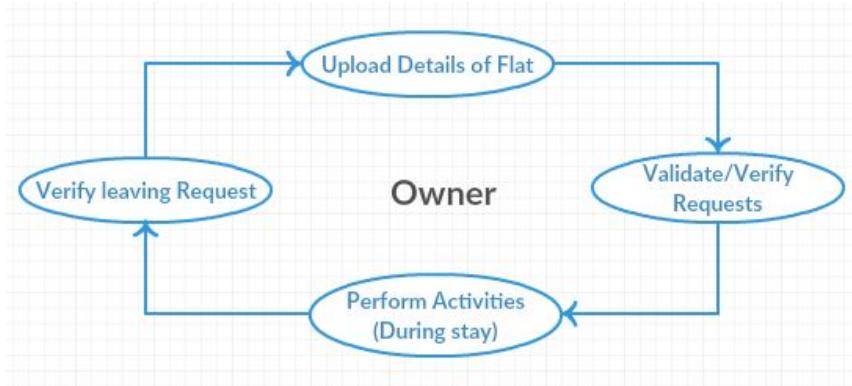


Figure : 1.2

→ **Input to the System :**

1. Flat details uploaded by the owner.
2. Users registration details during Signup.
3. Profile details like name, city etc.
4. Authentication details for sign in.
5. Text input for searching.
6. User complains and rent details.

→ **Output from the System :**

1. Limited and complete details of flat will be shown to user based on their registration.
2. User profile details.
3. Notifications of each major activities.
4. Complaints and their replies.
5. Owner will get rent details.
6. Message for each successful and unsuccessful activity.
7. Request Details can be seen by Owner.

1.3 Background

Searching and booking flat is one of the most general problem we face when we shift to a new city for living. We search for the flat manually by visiting and asking door to door or we contact to some agencies for that. Both of the way for booking the flat are time consuming and costly. Even after expending this much of time and money it may possible that we could not get the appropriate flat.

And as per flat owner perspective reaching to the tenant for lending their flats is also difficult and expensive.

For overcoming these problems we can make use of technology, i.e. an application that can do our job. "Vivir- a house rental and tenancy application" is one of such application who will help in dealing with these problems.

1.3 Motivations

Vivir is a standalone application. It aims to replace the existing manual searching/booking of flats/ rooms and activities associated with that. It just needs an android operating system and Internet connection to perform these activities. Thus, Vivir is a self contained application working with two types of peers - Tenant and Owner.

How it is useful ?

To Tenant :

- Easy and efficient searching and booking of flats/rooms.
- Get a large number of option for booking flats according to their need and choice.
- Removes the conflicts of rent and memorization of dues and dates.
- Makes the communication with owner easier.

To Owner :

- Application will provide an easy way to let their property.
- Approaching to the Customer/Tenant would be easier and efficient.
- Easy to perform all activities needed in room rental.
- Complaint handling with Tenant would be easier.
- No need to memorize/note anything, application will keep all necessary details.

To Social aspects :

- Cost and time efficient, needs only an internet connection and Android phone.
- Will help in maintaining the good relationship between Tenant and Owner.
- Transparency of rent and price.

2. LITERATURE SURVEY

This section will give a brief description of the literature review/survey needed for the early phase of this project. It will discuss the information from the website, and research that has been done.

1. Manual booking and Tenancy :

In most of the cities and places manual Flat/room searching and booking are working. User/Customer physically go to the place and search for their requirements. After booking the room all associated activities are done manually during the whole stay.

2. Booking Agencies and commission agents :

Nowadays several booking agencies are active for providing the details of available flats/rooms in the city. They charge with both parties (i.e Tenant and Owner) for these information.

3. Online Websites :

Several website are also active and working for room rental and tenancy facilities. Few of them are following.

<https://www.sulekha.com/> , <https://www.nestaway.com/> , <https://www.nestpick.com> ,
<https://www.eurooms.com/>

Working with Android application makes the Booking and Tenancy more easier. It just needs an android phone and internet connection and In today's world most of the users have that.

3. DESIGN AND IMPLEMENTATION

3.1 System Features

System has been devised to accomplish the following features and functionalities. These features can be divided in three categories based on the type of user.

- Common Features (Both for Tenant and Owner)
- Tenant Specific Features
- Owner specific Features

3.1.1 Common Features

1. Register

Description - System will store the authentication details of a user.

Response Sequence - System will provide a form for filling the details. After filling and submitting those details user will get a message of successful/unsuccessful registration.

Functional Requirements -

REQ-1 System shall provide a form for registration.

REQ-2 System shall validate the field type and mandatory details.

REQ-3 System should generate a message of successful or unsuccessful registration.

2. Login

Description - Verification of the authentic user (Registered user) and non authentic user.

Response Sequence - System will provide a form for filling the details. After filling and submitting those details system will verify for the authenticity of the user and authentic user will be redirected to their specific dashboard.

Functional Requirements -

REQ-1 System shall provide a form for registration.

REQ-2 System shall validate the field type and mandatory details.

REQ-3 System shall redirect to specific dashboard to the authentic user and shall generate a message for unauthorised users.

3. Change Password

Description - Changing of old password to new password.

Response Sequence - System will provide a form for filling old password, new password and password confirmation. After checking appropriate format and

authenticity system will change the password and will generate an appropriate message.

Functional Requirements -

REQ-1 System shall provide a form for filling the details.

REQ-2 System shall validate the new password and match it with the confirmation password.

REQ-3 If the entered old password matches with password stored in database then database password shall be updated to new password.

4. Set Profile -

Description - User profile details will be changed according to the details filled.

Response Sequence - User will get a form for filling the details and after submitting the details, user profile get updated.

Functional Requirements -

REQ-1 User shall get a form for filling the details.

REQ -2 Details should be stored in the database and changes should reflect on his profile.

5. View Notification -

Description - For all major activities user will get a notification.

Response Sequence - After clicking on view notification user will see the recent notification and after clicking on given floating button all notification will be displayed.

Functional Requirements -

REQ-1 System shall display the recent notification after clicking the notification button.

REQ-2 System shall display the all notification after clicking on given floating button.

3.1.2 Tenant Specific Features

1. Search Flat

Description - System will give the list of flats from the total flats according to the details entered in search box.

Response Sequence - Tenant will get a list of flats and after clicking on search icon a form for entering the search text will be displayed. After writing the text in the textbox filtered list of flats will be displayed.

Functional Requirements -

REQ -1 After clicking on search a list of flats shall be displayed.

REQ - 2 After clicking on search icon a text input box shall be displayed.

REQ-3 List of filtered flats shall be displayed based on the entered text.

2. Book Flat -

Description - After getting the flat details Tenant can request for booking.

Response Sequence - Tenant will get the flat details along with the button of booking. After clicking that button his/her booking request will be sent.

Functional Requirements -

REQ -1 Tenant shall get an interface with flat details and booking option.

REQ-2 After clicking on booking button system shall send a booking request to owner and tenant should get a booking confirmation message.

REQ-3 Flats that are already booked shall not appear and if any user is requesting for a flat again (that is not booked) a message of already sent request shall be displayed.

REQ - 4 After the approval of the request Tenant shall get a notification.

3. Upload Rent Details -

Description - Tenant will upload the details of rent paid.

Response Sequence - Tenant will get form for filling the rent details and after filling and submitting the details an approval request will be sent to owner.

Functional Requirements -

REQ - 1 Tenant shall get a form for filling the rent details.

REQ - 2 After filling and submitting the details approval request shall be sent to owner.

REQ - 3 After the approval of the request tenant shall get a notification.

4. Complain-

Description : Tenant can write complaints for any issue.

Response Sequence : Tenant will get a option for writing complaints and after successfully submitting those details, tenant will get a message of successful submission. He can also see the replies of the complaints.

Functional Requirements -

REQ-1 Tenant shall get a option for writing the complaints.

REQ- 2 Tenant shall be able to see the received replies.

5. Leave Flat -

Description : Tenant can request for leaving the flat.

Response Sequence - After clicking on leave flat button tenant will get form for filling the flat id and email. After verifying authentic email and flat id his/her request will be sent for approval. After approval of the request tenant will get a notification.

Functional Requirements -

REQ - 1 Tenant shall get a form for filling the flat id and Email.

REQ-2 After submitting those details a request of flat leaving shall be send to owner.

REQ -3 After approval of the request tenant shall get a notification.

3.1.3 Owner Specific Features**1. Upload flat/apartment Details**

Description : Owner will upload the details of flat.

Response Sequence - After clicking on floating button Owner will get a form for filling the apartment details. After submitting those details flat will be added.

Functional Requirements -

REQ-1 Owner shall get a form for filling the flat details.

REQ-2 After filling and submitting the form details added flat should appear in My Flats section.

2. Approve Booking Request

Description : When tenant will request for booking, owner will get a request.

Response Sequence - Owner will get a list of booking request, on clicking on that request details will be displayed with a option of reject and accept. After clicking on reject request will be deleted and after clicking on accept booking will be approved.

Functional Requirements

REQ-1 Owner shall get a list of booking request and after clicking on that request request details should be displayed.

REQ - 2 Request details page shall contain an accept and reject button.

REQ - 3 After clicking on reject request shall be deleted and after clicking on accept request shall be approved.

3. Resolve Complaints

Description : Owner can reply to the tenants complaints.

Response Sequence - Owner will see the complaints of tenants, and after that he can reply to their complaints by writing a text.

Functional Requirements

REQ - 1 Owner shall be able to see the complaints of tenants.

REQ - 2 Owner shall get the option for replying the complaints.

4. Approve rent

Description : When tenant will upload the rent details then Owner will get a request of approval.

Response Sequence - Owner will get a list of rent approval request, on clicking on that request details of request will be displayed with the options of reject and accept. After clicking on reject, request will be deleted and after clicking on accept, rent will be approved.

Functional Requirements

REQ-1 Owner shall get a list of rent approval request and after clicking on that request, request details should be displayed.

REQ - 2 Request details page shall contain an accept and reject button.

REQ - 3 After clicking on reject, request shall be deleted and after clicking on accept, request shall be approved.

5. Approve flat leaving request

Description : Owner can approve the request of leaving the flat.

Response Sequence - Owner will get a leaving flat request, and he/she can reject or accept that request. After approval of the request tenant will get a notification.

Functional Requirements

REQ-1 Owner shall get a list of leaving of flat leaving request and after clicking on that request request details should be displayed.

REQ - 2 Request details page shall contain an accept and reject button.

REQ - 3 After clicking on reject request shall be deleted and after clicking on accept request shall be approved.

6. Edit/Delete flat Details

Description : Owner can delete or update the flat.

Response Sequence - From flat details page Owner will get delete and edit flat option. After clicking on delete a confirmation dialog will popup, and actions will be taken according to the choice. After clicking on edit a form with previously filled details will be displayed, from there owner can edit and submit the details.

Functional Requirements

REQ-1 Owner shall get the option of edit and delete.

REQ - 2 After clicking on delete button owner shall get a option for confirmation.

REQ - 3 Details of flat shall be deleted after clicking on ok button and owner shall be redirected to MyFlat page

REQ - 3 Owner shall get a form with already filled flat details.

REQ-4 After submitting the form with new details flat data shall be updated.

3.2 Non Functional Requirements

3.2.1 Performance Requirements -

1. Application shall be easy to use and user friendly.
2. Result of each action shall be displayed within appropriate time.
3. Details shall be displayed in appropriate manner

3.2.2 Security Requirements

1. User details shall be stored with proper safety and shall be fetched with appropriate method.
2. Unauthorised user shall be able to access the limited features.

3.2.3 Software Quality Attributes

1. Usability - System shall arrange the features in such way that all features shall be usable and accessible in easy way.
2. Reliability - System should not crash and there should not be ambiguity in the informations.

3.3 DESIGN

3.3.1 Activity Diagram -

1. Activity Diagram For Owner

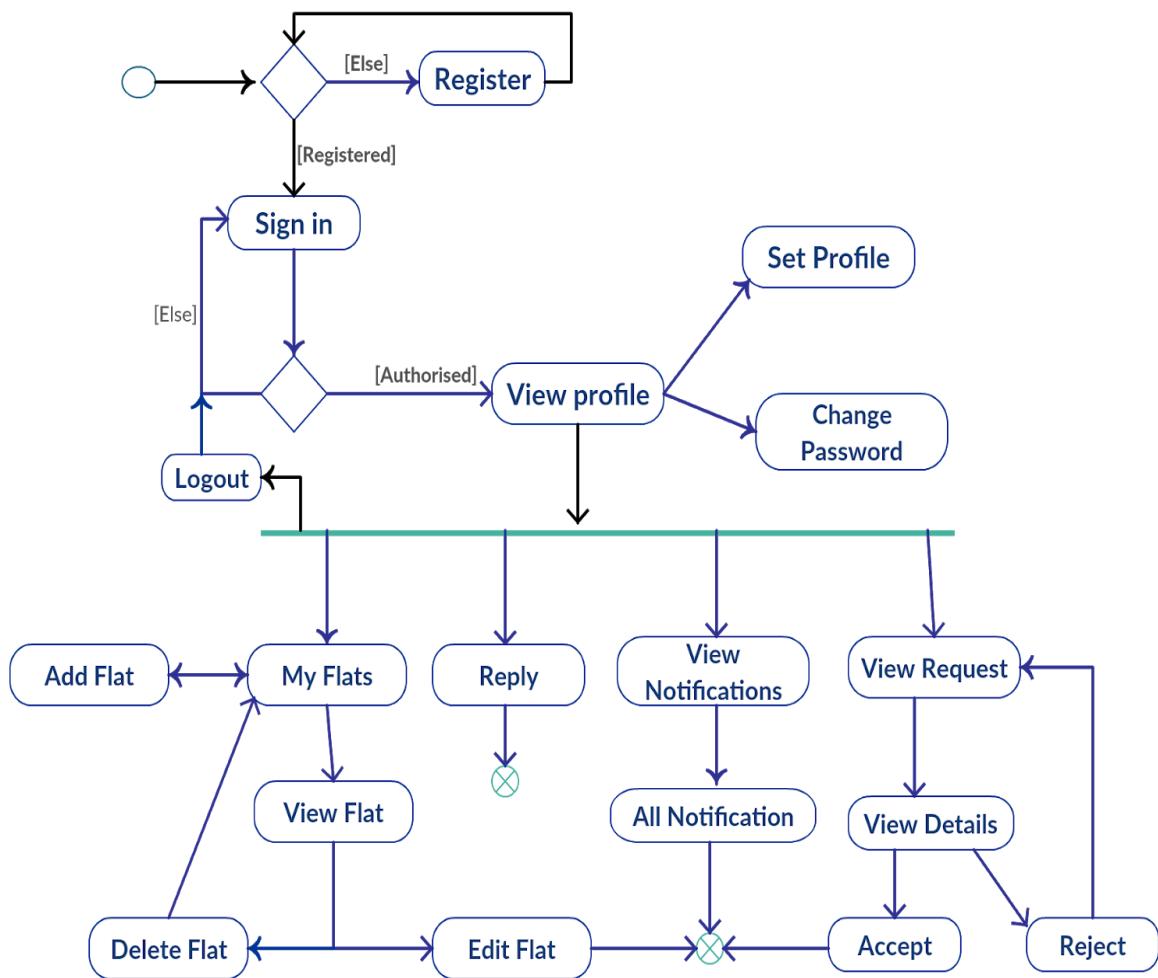


Figure 3.1

2. Activity Diagram For Tenant

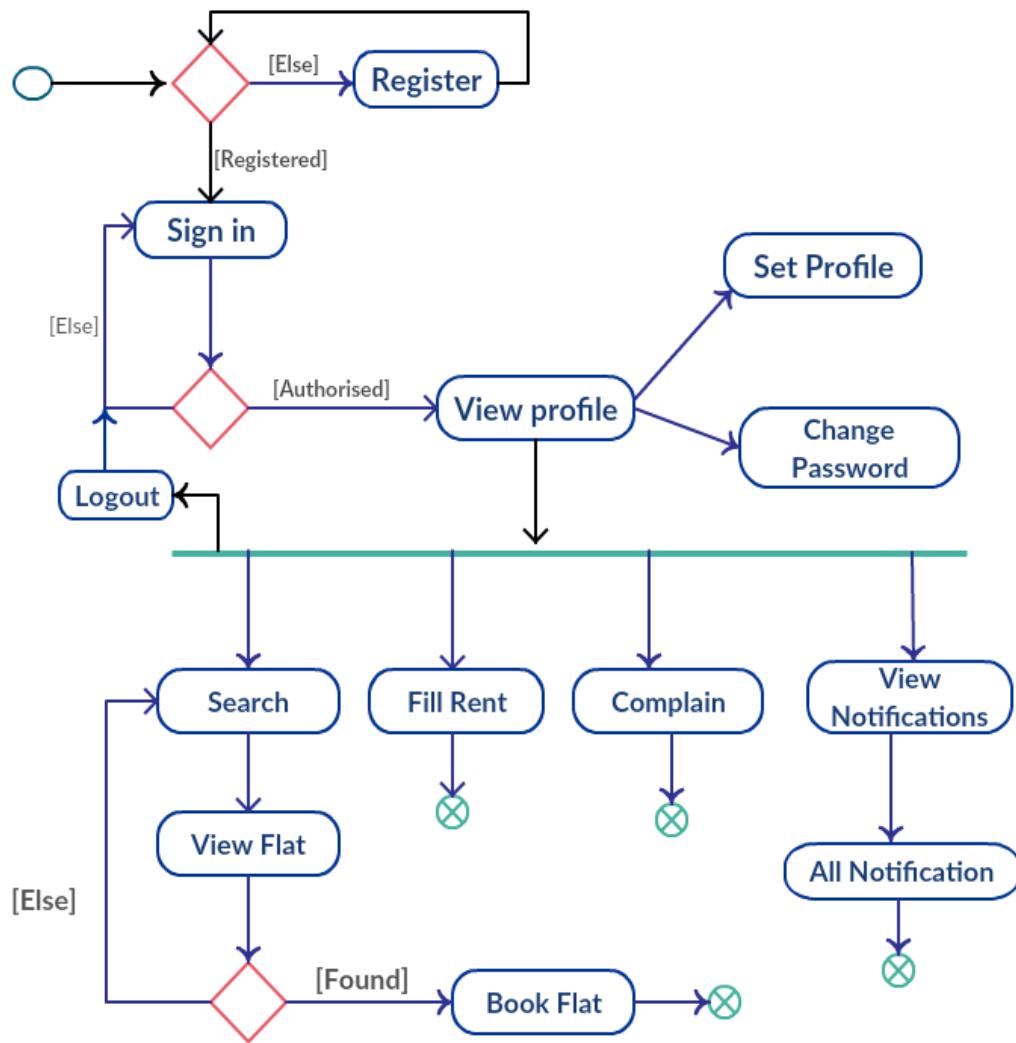


Figure 3.2

3.3.2 Use case Diagram

1 . Use Case Diagram for Owner

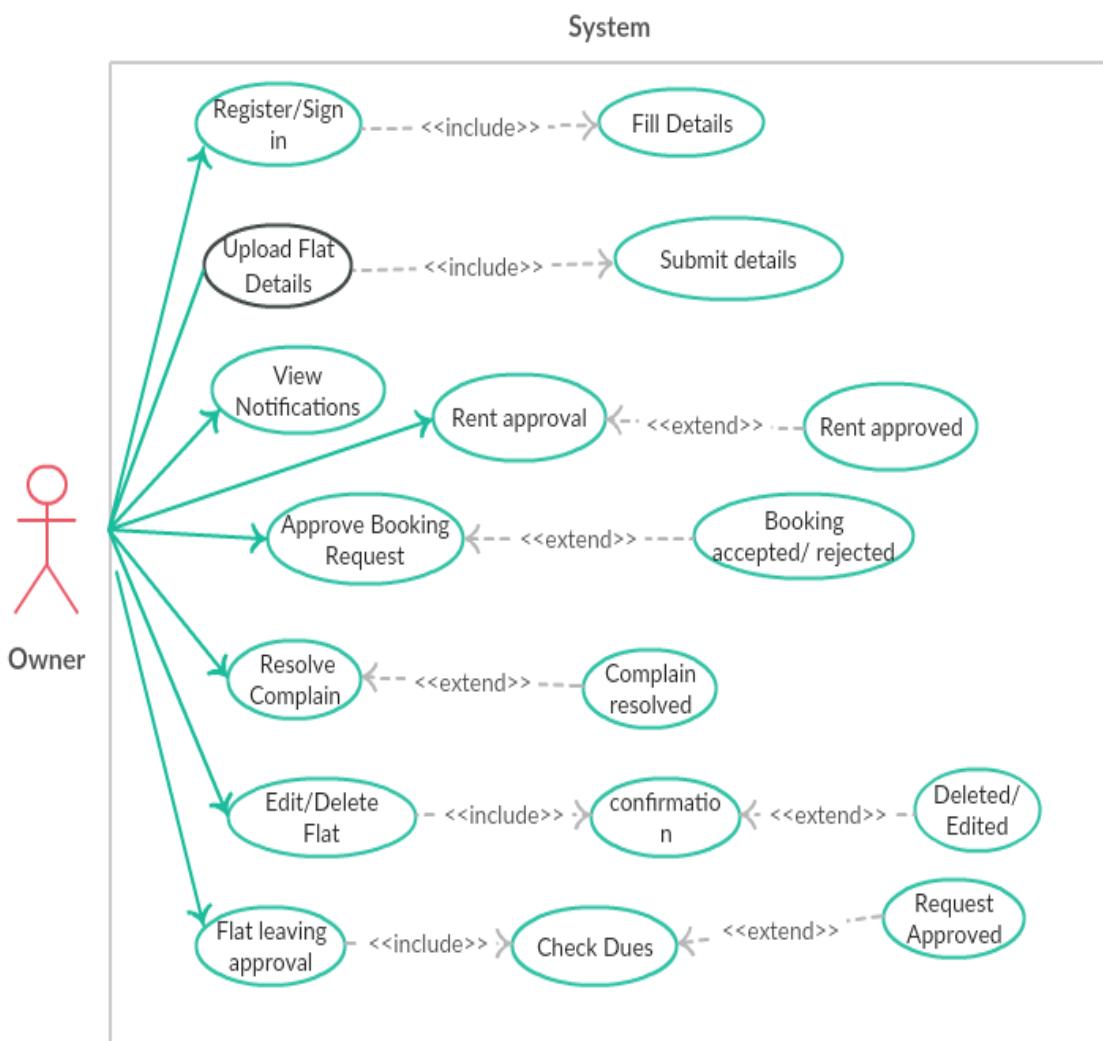


Figure 3.3

2 . Use Case Diagram for Tenant

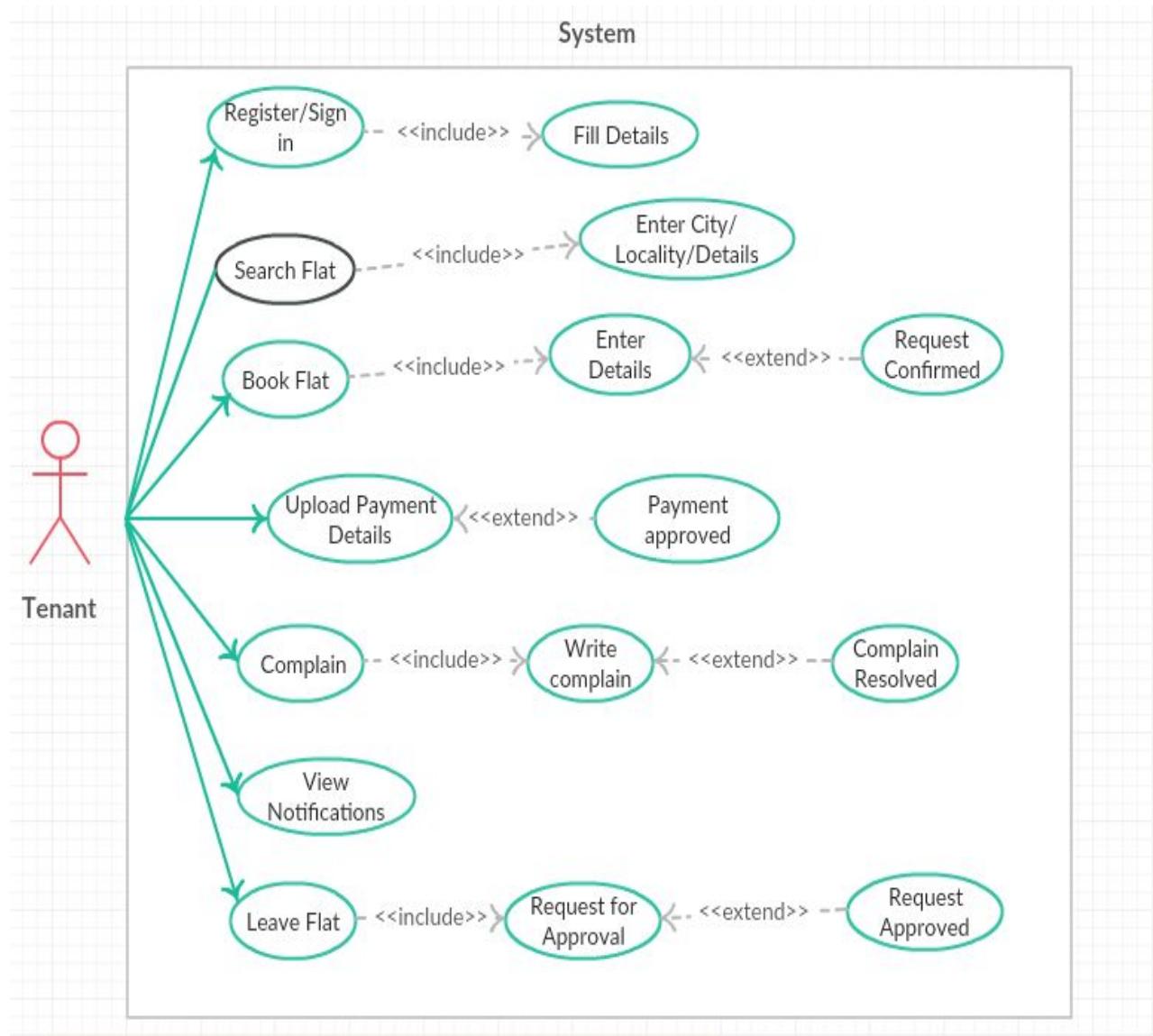


Figure 3.4

3.3.3 ER Diagram and Database Description

1 - Entity Relationship Diagram

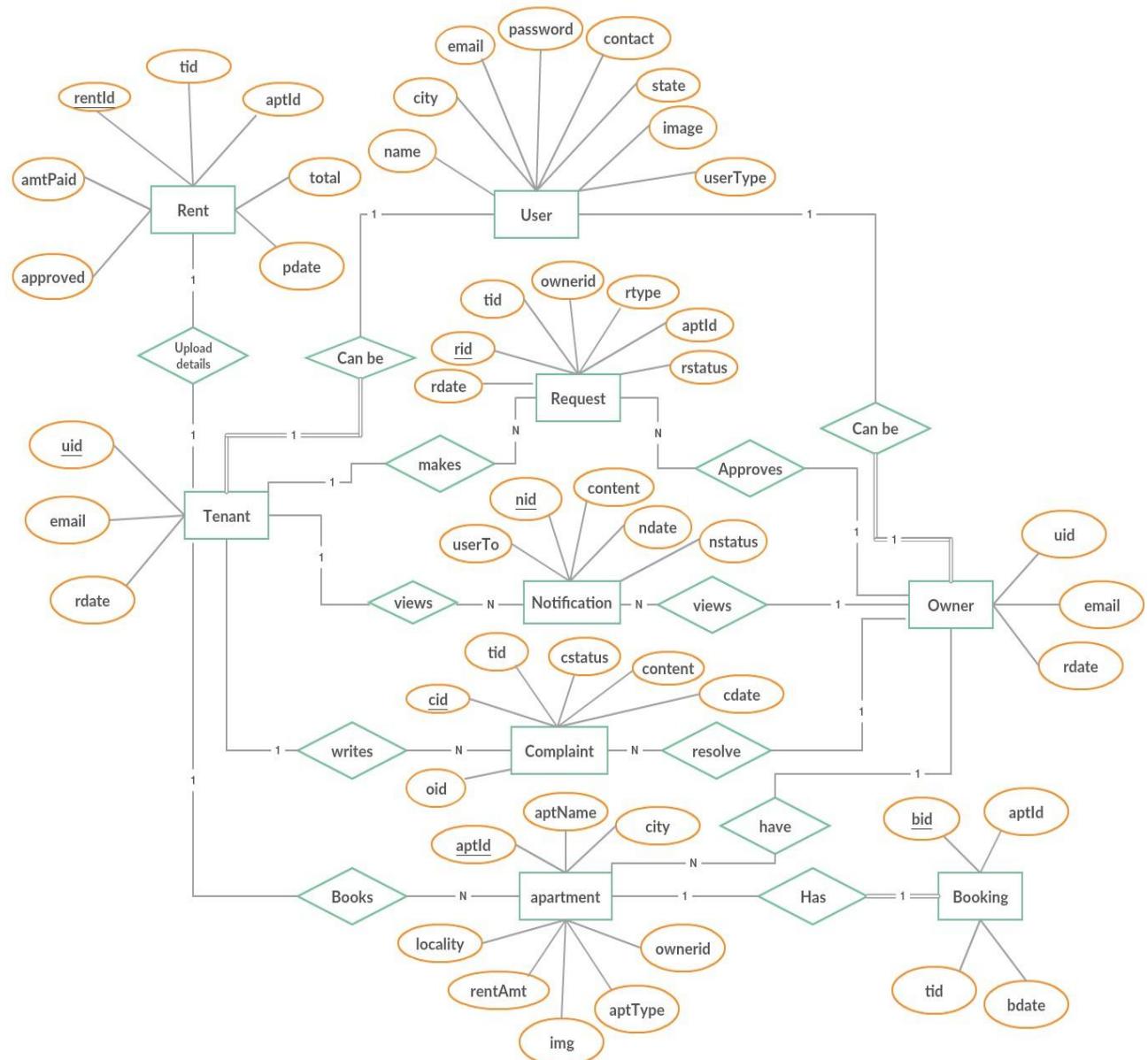


Figure 3.5

2 - Database Schema

The whole project includes following entities. Each entity have their attributes.

- user : email(Primary Key - varchar), password(varchar), name(text), image(varchar), city(varchar) , state(varchar), userType(varchar), contact(varchar)
- tenant : uid(Primary Key :varchar), email(varchar), rdate(date),
- owner : uid(Primary Key :varchar), email(varchar), rdate(date)
- Request : rid(Primary Key : varchar), tid(varchar), ownerid(varchar), rtype(varchar), aptId(varchar), rdate(date), rstatus(varchar)
- notifications : nid(Primary Key: varchar), userTo(varchar), content(varchar), ndate(date), nstatus(int)
- complain : cid(Primary Key : varchar), tid(varchar), oid(varchar), cdate(date), content (varchar), cstatus(varchar)
- rent : rentId(Primary Key : varchar), tid(varchar), aptId(varchar), amtPaid(varchar), total(varchar), pdate(date), approved(50)
- booking : bid (Primary Key : varchar), aptId(varchar), tid(varchar), bdate(date)

3 - Database Tables :

I. user

#	Name	Type	Default
1	<u>email</u>	varchar(50)	None
2	password	varchar(50)	None
3	name	varchar(50)	None
4	userType	varchar(50)	None
5	image	varchar(200)	DefaultUserImage.png
6	city	varchar(50)	N/A
7	state	varchar(50)	N/A
8	contact	varchar(15)	N/A

II. tenant

#	Name	Type	Default
1	<u>uid</u>	varchar(20)	None
2	email	varchar(50)	None
3	rdate	date	None

III. owner

#	Name	Type	Default
1	<u>uid</u>	varchar(50)	None
2	email	varchar(50)	None
3	rdate	date	None

IV. request

#	Name	Type	Default
1	<u>rid</u>	int(50)	None
2	tid	varchar(50)	None
3	ownerid	varchar(50)	None
4	rtype	varchar(50)	None
5	aptId	varchar(50)	None
6	rdate	date	0000-00-00
7	rstatus	varchar(50)	pending

V. complain

#	Name	Type	Default
1	<u>cid</u>	varchar(50)	None
2	tid	varchar(50)	None
3	oid	varchar(50)	None
4	cdate	date	None
5	content	varchar(1000)	None
6	cstatus	varchar(10)	None

VI. notifications

#	Name	Type	Default
1	<u>nid</u>	int(50)	None
2	userTo	varchar(50)	None
3	content	varchar(50)	None
4	ndate	date	None
5	nstatus	int(11)	None

VII. rent

#	Name	Type	Default
1	<u>rentId</u>	int(50)	None
2	tid	varchar(50)	None
3	aptId	varchar(50)	None
4	amtPaid	varchar(50)	None
5	total	varchar(50)	None
6	pdate	date	None
7	approved	varchar(50)	no

VIII. booking

#	Name	Type	Default
1	<u>bid</u>	int(50)	None
2	aptId	varchar(50)	None
3	tid	varchar(50)	None
4	bdate	date	None

IX. apartment

#	Name	Type	Default
1	<u>aptId</u>	varchar(50)	None
2	aptName	varchar(50)	None
3	locality	varchar(50)	None
4	city	varchar(50)	None
5	ownerid	varchar(50)	None
6	aptType	varchar(50)	None
7	rentAmt	varchar(50)	None
8	img	varchar(200)	default.jpg

3.4 Operating and Development Environment

- **Client Side Requirement**

1. Android OS (android 4.0 and above)
2. Connectivity: Good Internet connection

- **Server Side Technologies used**

1. Language : PHP 7.1
2. Database : MySQL

- **Development Environment used**

1. Android Studio and JDK
2. Languages : Java and XML
3. Windows/Linux System

3.5 Implementation and Product Demo

Source Code :

The implementation of the project has been done using above mentioned technologies. Source code of the whole project can be accessed from github from the below given link.

Github Link : <https://github.com/yogendra-nitc/Vivir>

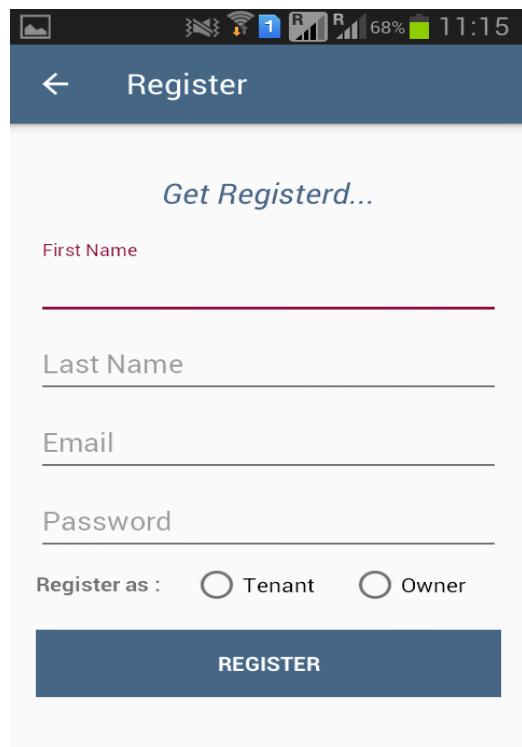
All commits and changes that have been made during whole development can be accessed by following the link : <https://github.com/yogendra-nitc/Vivir/commits/master>

Product Screenshots :

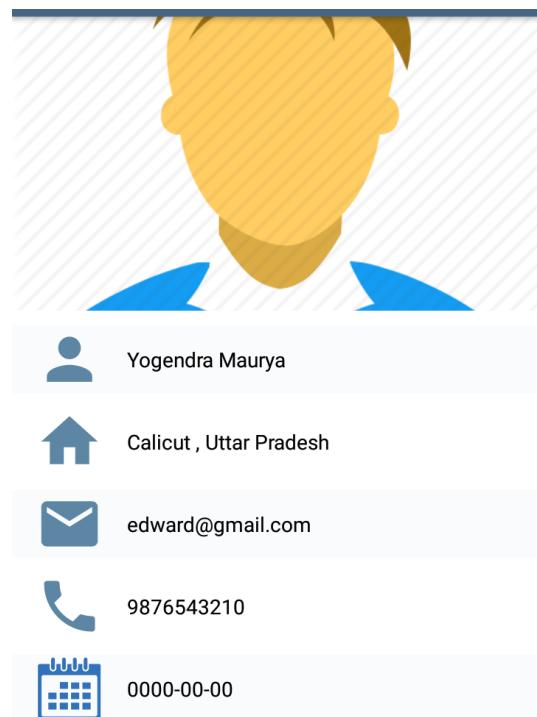
1 .

2 .

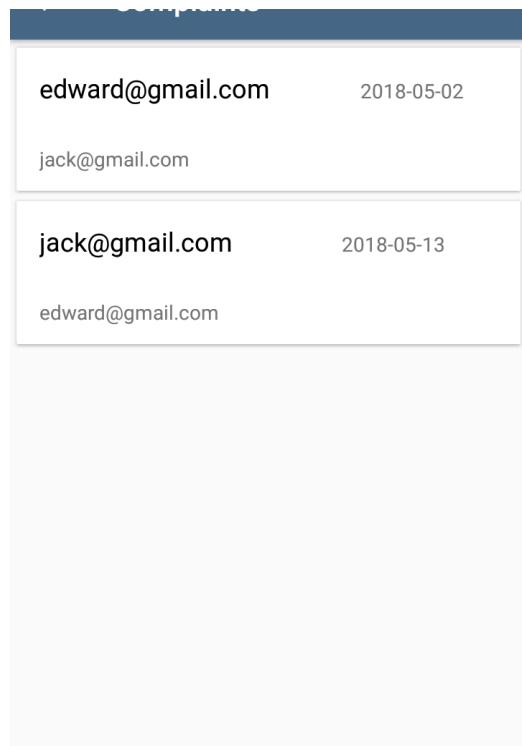
Property Image	Details
	dev, devnagar, Varanasi , 3BHK, Rs.7000/month
	Basant kunj, basant nagar, Varanasi , 4BHK, Rs.10000/month
	Doruk Villa, Kannoor, Calicut , 4BHK, Rs.12000/month
	Ashish Villa, Nizamabad, Azamgarh , 4BHK, Rs.8500/month
	Sarvesh Villa, Chandeshar, Azamqarh , 4BHK, Rs.15000/



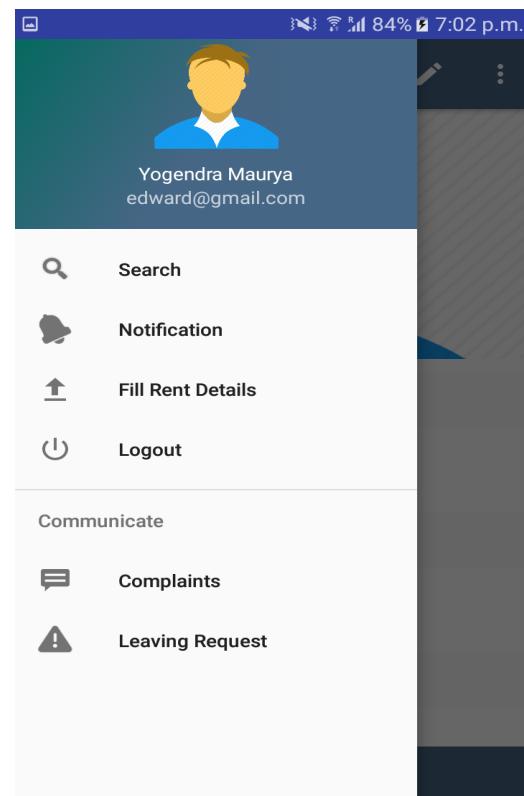
3.



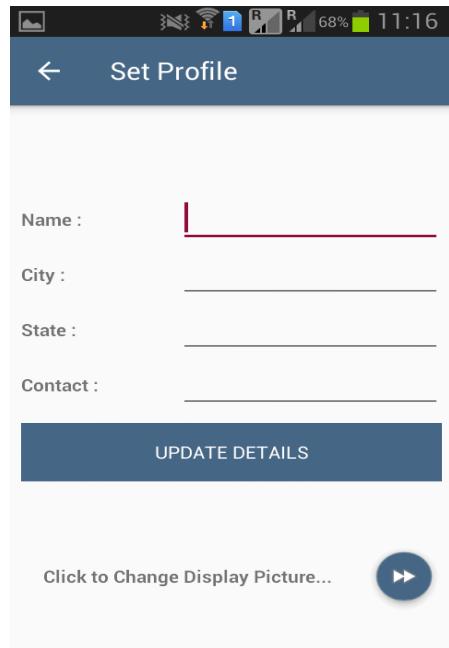
4.



5.



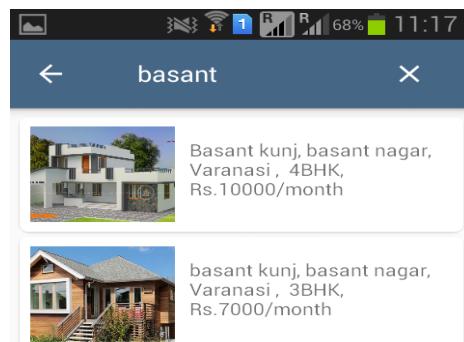
6.



7.



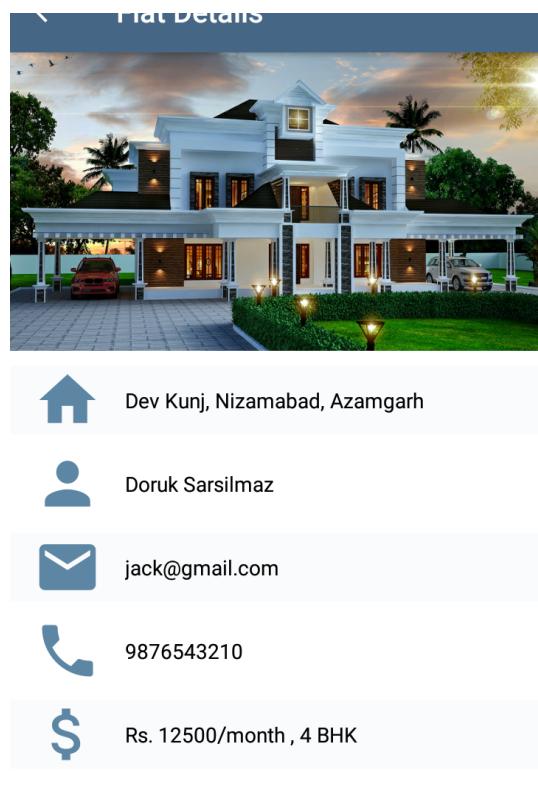
8.



9.



10.



11.

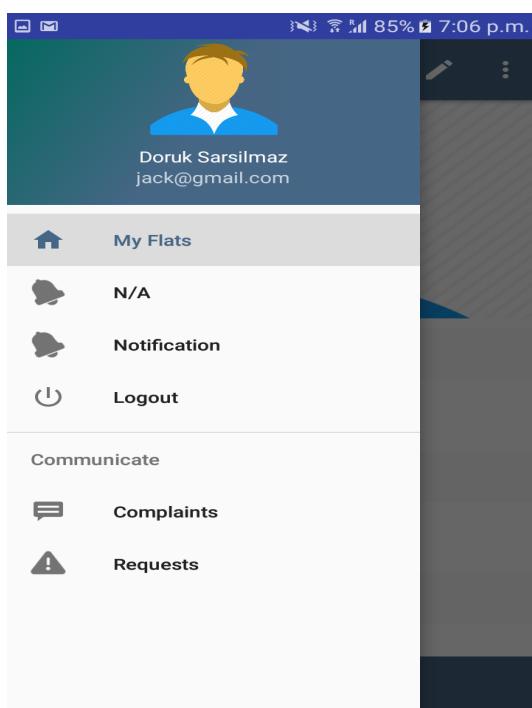
Rent Details

Use proper format for filling the data.

Apartment	APT##...V
Amount Paid	
Payment Date	0000-00-00

SUBMIT

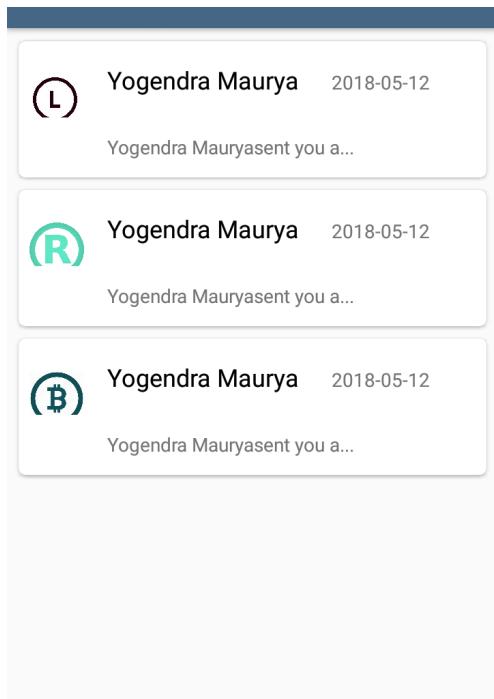
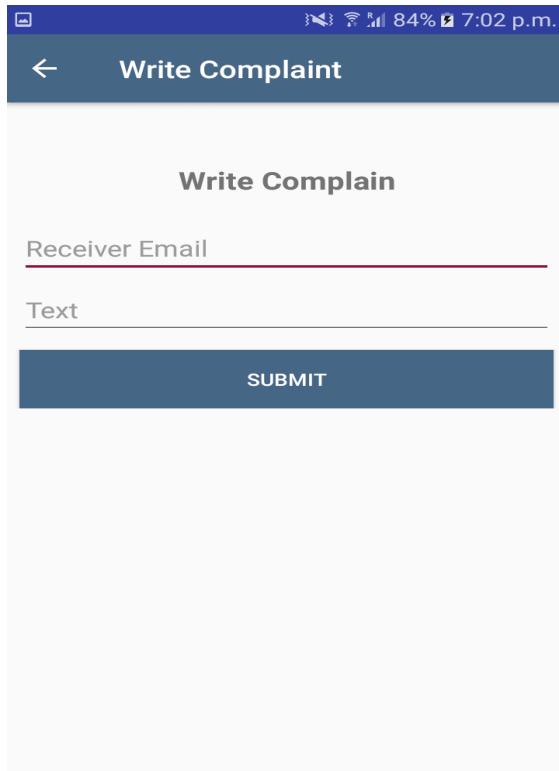
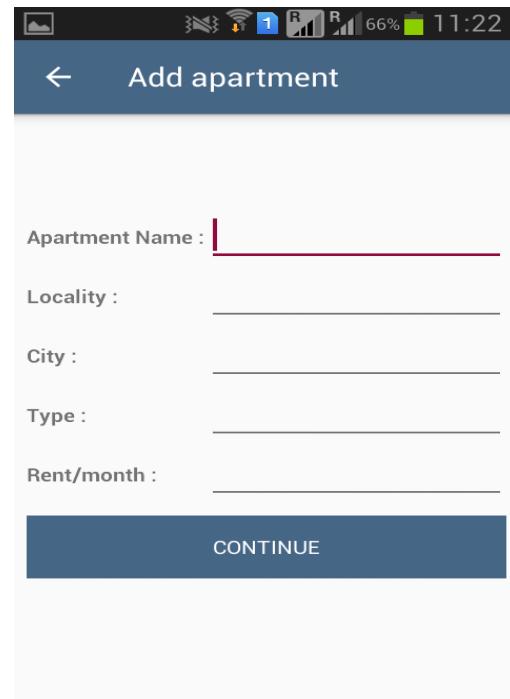
12.



- 13.
- Vivir
- Add Image
 - dev kunj, devnagar, Varanasi , 4BHK, Rs.10000/month
 - Dev Kunj, Nizamabad, Azamgarh , 4BHK, Rs.12500/month
 - dev kunj, devnagar, Varanasi , 4BHK, Rs.10000/month
 - Ashish Villa, Nizamabad, Azamgarh , 4BHK, Rs.8500/ month
 - Add Image
 - Abhi apt, Tanda, Ambedkar Nagar , 4BHK, Rs.4560/month
- 14.

13.

14.

**15.****17.****16.****18.**

4. CONCLUSION

House Rental and Tenancy Application - “ Vivir “ has been devised to make the flat/apartment searching and booking easier and efficient. By using this application tenant can easily search and book flats according to their need and choice. Communication and record maintaining will also become easier.

As per Owner perspective reaching out to the Tenant will also become very easy and effective. It will save a lot of time and money to both peers i.e. owner and Tenant.

This application will also help in maintaining the transparency of rent and good relationship between owner and tenant and in the era of technology using this application will cost nothing compared to the other ways of dealing House rental and Tenancy.

5. GLOSSARY

Tenant - Person who is going to book the flat

Owner - Person who is lending the flat

REQ - Requirement

OS - Operating System

JDK - Java Development Kit

XML - Extensible markup language

PHP - Hypertext Preprocessor

System Features - Expected functionalities from the application

6. REFERENCES

- [1] **Android Studio:** <http://developer.android.com/studio/index.html> (05/05/2018)
- [2] **Udacity :** <https://in.udacity.com/>(20/04/2018)
- [3] **Java - The Complete Reference** by Herbert Schildt , 9th Edition
- [4] **XML** - www.tutorialspoint.com
- [5] https://en.wikipedia.org/wiki/Android_Studio
- [6] https://www.w3schools.com/php/php_mysql_intro.asp
- [7] Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems sixth edition - 2011.
- [8] Java Code Conventions - <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>
- [9] <https://creately.com/app/#>](09/05/2018)
- [10] <https://www.sulekha.com/>(Feb,2018)
- [11] <https://www.nestaway.com/>(Feb,2018)