North South University
Department of Electrical & Computer Engineering
Project Report
Date Prepared: 19.09.2021
Course No: CSE311 **Sec: 04**

Faculty: Intisar Tahmid Naheen
Lab Instructor: Nazmul Alam Dipto

Project Name: House Rental Management System

Member 1
Name: Samil Abdullah
ID: 1611408642
Email: samil.abdullah@northsouth.edu

Member 2
Name: Mohammad Sazzad Alam
ID: 1611200642
Email: sazzad.alam@northsouth.edu

Introduction

House Rental Management System is a platform where a person can easily find a suitable house within his requirements (Price, Number of bedrooms, Number of Toilets, Common Space etc.). When a tenant fills up his requirements through website, the website will show some suggest house which can be suitable for him. He can also message the owner for confirmation. After that the tenant can pay all the bills including house rent from this platform including Bkash / Rocket. The tenant can also complain about anything about his apartment through this website. The tenant has to upload all his information and document through this website before 1 month of rent a house. An owner of a house can upload his requirements to rent his house. He can upload his appointment picture and upload it through website. He can show all the information of the tenant and he can show all the transaction history of payment.

Software Use:

We will use:

1. VS Code
2. Adobe Photoshop
3. Flask Framework for backend
4. SQLite for database
5. CSS
6. HTML

VS Code is for writing the code of HTML, CSS, FLASK and all. We are used Photoshop for some background image. SQLite is used for local hosting server.

Project Specification

Rental Management

We will be creating an application on the web platform that will enable the landlords and the tenants to keep a track of their monthly bills and accommodate them by interacting in times of need. Our application mainly focuses on management. In order to do so, it will open fields for the landlords to alert the tenants while the tenants can also invoke any sort of actions that has to be taken by the landlord regarding the house. The application will provide a user-friendly environment both in terms of the interface and user usability.

Our goal is to provide a user-friendly web application where the landlord and the tenants will have equally important and interesting features to work on. We are looking forward to create something that is easy to use and that does not crash.

Our application is intended to be used by the Landlords and tenants in order to resolve their reluctance while communicating with one another.

As students, our goal for now is to set up a working application that will be a demo of what is about come in the future. We feel that there is a broader aspect of this application. The need for a rental management application is definitely growing in a developing country like ours.

Scope

Our application will help the landlords to clearly identify his/her tenants with access to all the information associated with the tenant.

The landlords can keep track of the bills that are cleared and check on history.

Landlords can Invoke alerts to the tenants if the bills aren't cleared for a given month. He/she will have the access to send personal messages through this application.

Tenants can also keep track of his/her payment history.

Tenants will have the opportunity to make complains regarding household utility or others.

Business Strategy

In our region, there is large increasing of house renting business. For helping this business, our application makes it easier and portable for clients of this business. We choose three kinds of strategy for getting monetize-

1. Affiliate Programs: Our apps generate revenues via transection commission. We try to create affiliate programs that gives the company on every transection the clients would do that is done by usage of the app.
2. Advertising: We add most renowned monetizing system like pay per click model (ppcm) in affiliation with GOOGEL AdSense.

The APPLICATION:

Home Page

This page would contain the links to Create Account, Login for the tenants and the landlords, about page and the settings.

Create Account

There will two kinds of user. The landlord and the tenant. Users would have to provide with the following information:

Landlord:

User name, First Name, Last Name, Address (special fields will be created for specific floor details), Cell Phone Number, Email Address, Password, No. of apartments, No. of floors, Monthly fare Utility bills

Tenant:

Username, First Name, Last Name, Father's Name, Address (special fields will be created for specific floor details), Cell Phone Number, Email Address, Password, Rent bill.

The details may vary over the course. Attachment option may get included for the tenant.

Login

Login section would require two fields to for both the types of users. And in order to login they would require to provide:

Username

Password

landlord login:

He / She will come up with a page where the user can get access to any of the tenants registered in his building. Information regarding the tenant will pop up when any of those links are pressed upon. The landlord will be able to delete any existing tenant account. And an option to alert the tenant to clear any missing due will be provided.

Tenant login:

In case of the tenants, they will find a complain box field along with his/her own information after logging in. Tenants will be provided option to clear payment through bkash or cash in hand.

The Management

In the management section there will be the account id, date of payment, and the amount paid. For the paid tenants a check will indicate whether any due is not cleared.

About

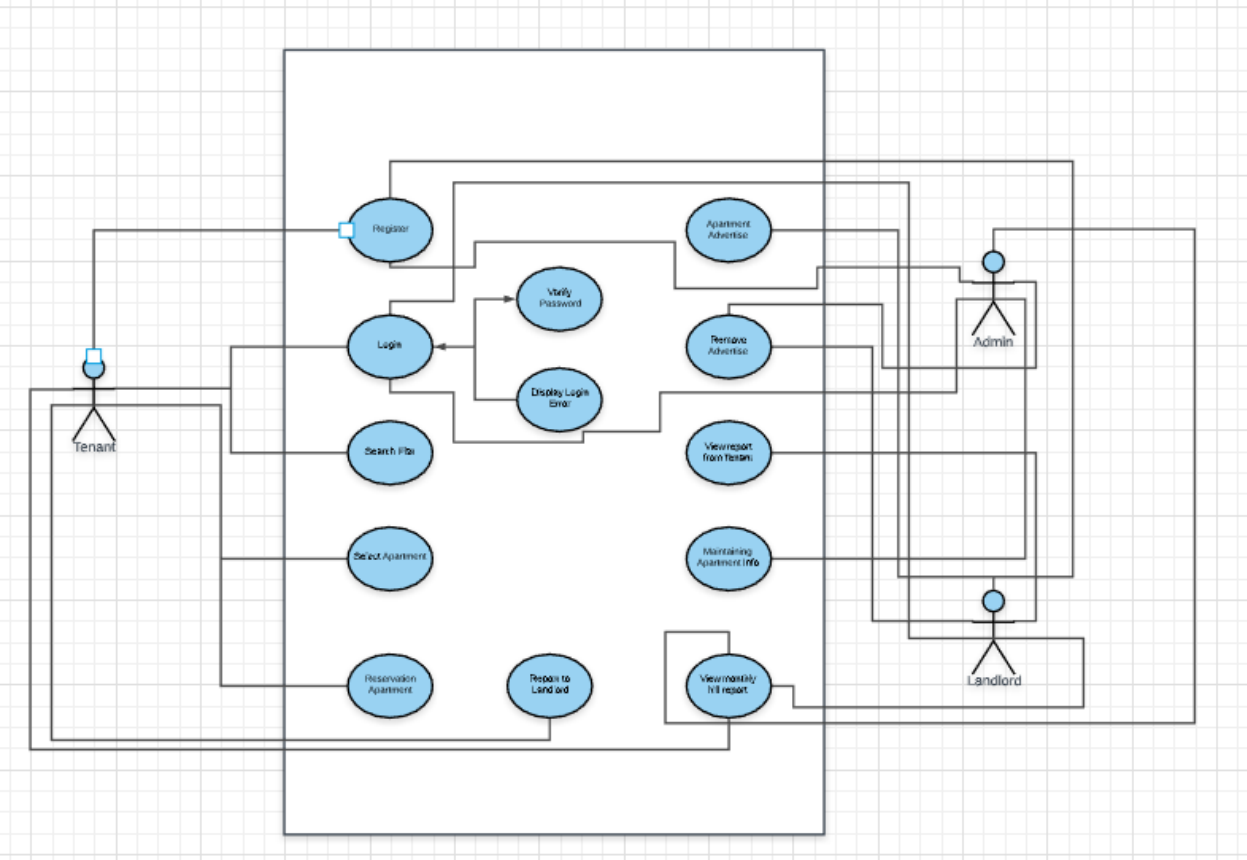
This page section would contain a small description about the company, and the products. There would be some statistics about the company and paragraphs about how they started and what their objective is. This might include what changes the company is bringing in the near future and so on.

Contact Us

This section would contain some useful contact links, such as email addresses of the managing bodies, or location to the stores and their contact information.

UML Use-Case Diagram:

Our project use case diagram looks like this.



Here, 3 types of Actors. Which is Tenant, Landlord and Admin.

UML expended use case for login:

1. Use Case Name: Login.

- 2. Type: Primary, Secondary.

- 3. Actor(s): Admin, Landlord, Tenant.

- 4. Brief Overview: When a user login, he come to his dashboard and if he is a new user then the system pushes his to register dashboard and after the register, he can login.

Typical Course of Events:

Actor Actions	System Responses
1. Actor Goes to www.hrms.com 2. Actor enters user name and password. 3. User is New. 4. Password Incorrect.	1. System responds with login page 2. System check it and push him to his dashboard if password correct. 3. System push him to registration dashboard for registration 4. System push his to reset password page

UML Expand Use Case for Registration:

- 6. Use Case Name: Registration.

- 7. Type: Primary.

- 8. Actor(s): Admin, Landlord, Tenant.

9. Brief Overview: Every user must register to use this system. He has to first register as Landlord or Tenant. The registration page is different for tenant and landlord.

10. Typical Course of Events:

Actor Actions	System Responses
1. Actor clicks to registration. 2. Fill up different requirements 3. Enter password 4. Actor enter Save	1. System responds to registration page. 2. System tell his who to fill up 3. System cannot show his password in screen 4. System save all his information into database and push him to login page

UML Expand Use Case for Search Apartment:

- 11. Use Case Name: Apartment.

- 12. Type: Primary.

- 13. Actor(s): Tenant.

- 14. Brief Overview: When a user search for an apartment the system asks his to fill up his requirements like rent amount, number of rooms, number of washrooms, place and after that the system show suitable apartments.

15. Typical Course of Events:

Actor Actions	System Responses
1. Actor click Search Apartment	1. System ask him to fill up his requirements for search.
2. Actor fill up requirements.	2. System show him some apartments for his requirements.
3. Actor click reservation	3. System push him to select apartment

UML Expand Use Case for Select Apartment:

- 16. Use Case Name: Select Apartment.

- 17. Type: Primary.

- 18. Actor(s): Tenant.

- 19. Brief Overview: When a tenant selects an apartment he likes, he shows all the information of the apartment like rent, rooms pictures, etc. If he chooses the apartment then he can confirm his reservation though reservation button.

20. Typical Course of Events:

Actor Actions	System Responses
1. Actor so to select apartment. 2. Actor click back option. 3. Actor click Reservation.	1. System shows him all the details of the apartment. 2. System push this to search apartment page 3. System push him to reservation page to re

UML Expand Use Case for Reservation Apartment:

- 21. Use Case Name: Reservation Apartment.

- 22. Type: Primary.

- 23. Actor(s): Tenant.

- 24. Brief Overview: When a tenant wants to reserve an apartment, he first has to fill up all his information with NID number and after fill up the information he has to press confirm button, then a message goes to landlord and he contact with the tenant.

25. Typical Course of Events:

Actor Actions	System Responses
1. Actor goes to Reservation apartment. 2. Actor fill up the information and confirm.	3. System responds with reservation page. 4. System save the information to database and auto send a sms to the landlord with the information of the tenant for comfirm

UML Expand Use Case for Report to Landlord:

- 26. Use Case Name: Report to Landlord.

- 27. Type: Primary.

- 28. Actor(s): Tenant.

- 29. Brief Overview: Tenant can complain about anything of the apartment through report option.

30. Typical Course of Events:

Actor Actions	System Responses
1. Actor click report to landlord 2. Actor fill up his complain and click send	1. System responds with report option. 2. System store his complain information to database and inform landlord about it

UML Expand Use Case for Apartment Advertise:

1. Use Case Name: Apartment Advertise.

2. Type: Primary.

3. Actor(s): Landlord.

4. Brief Overview: Only landlord can upload or edit apartment advertise using this option.

5. Typical Course of Events:

Actor Actions	System Responses
1. Actor clicks to apartment advertise. 3. Actor enters information of apartment advertise.	2. System responds it. 4. System check it and return to dashboard.

UML Expand Use Case for Remove Advertise:

6. Use Case Name: Remove Advertise.

7. Type: Primary.

8. Actor(s): Admin, Landlord.

9. Brief Overview: Landlord and Admin can remove an advertise from this option.

10. Typical Course of Events:

Actor Actions	System Responses
1. Actors clicks to remove advertise. 3. Actors remove any information and press ok.	2. System responds to remove advertise page. 4. System check it and return to dashboard.

UML Expand Use Case for View Report from Tenant:

11. Use Case Name: View report from tenant.

12. Type: Primary.

13. Actor(s): Landlord.

14. Brief Overview: If tenant can any report to landlord, then landlord can solve this.

15. Typical Course of Events:

Actor Actions	System Responses
1. Actors click to view report option. 3. Actor see the information.	2. System responds to view report option. 4. System check it.

UML Expand Use Case for Maintaining Apartment Info:

16. Use Case Name: Maintaining apartment information.

17. Type: Primary.

18. Actor(s): Admin.

19. Brief Overview: Only admin can maintain this. If admin upload any wrong info, then admin can change it easily.

20. Typical Course of Events

Actor Actions	System Responses
1. Actors click to maintain apartment info.	2. System responds to apartment info.
3. Actor can edit and press ok.	4. System check it and return to dashboard.

UML Expand Use Case for View Monthly Bill Report:

21. Use Case Name: View monthly bill report.

22. Type: Primary.

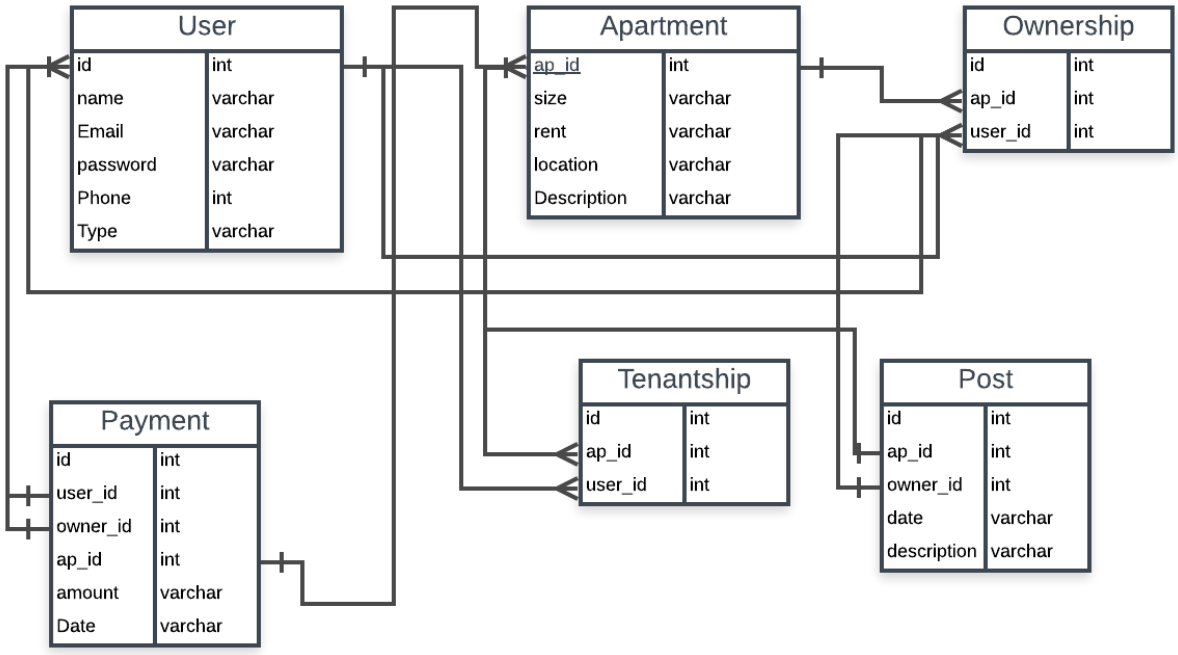
23. Actor(s): Admin, Tenant, Landlord.

24. Brief Overview: Landlord can show all the payment. After see that landlord can contact to tenant.

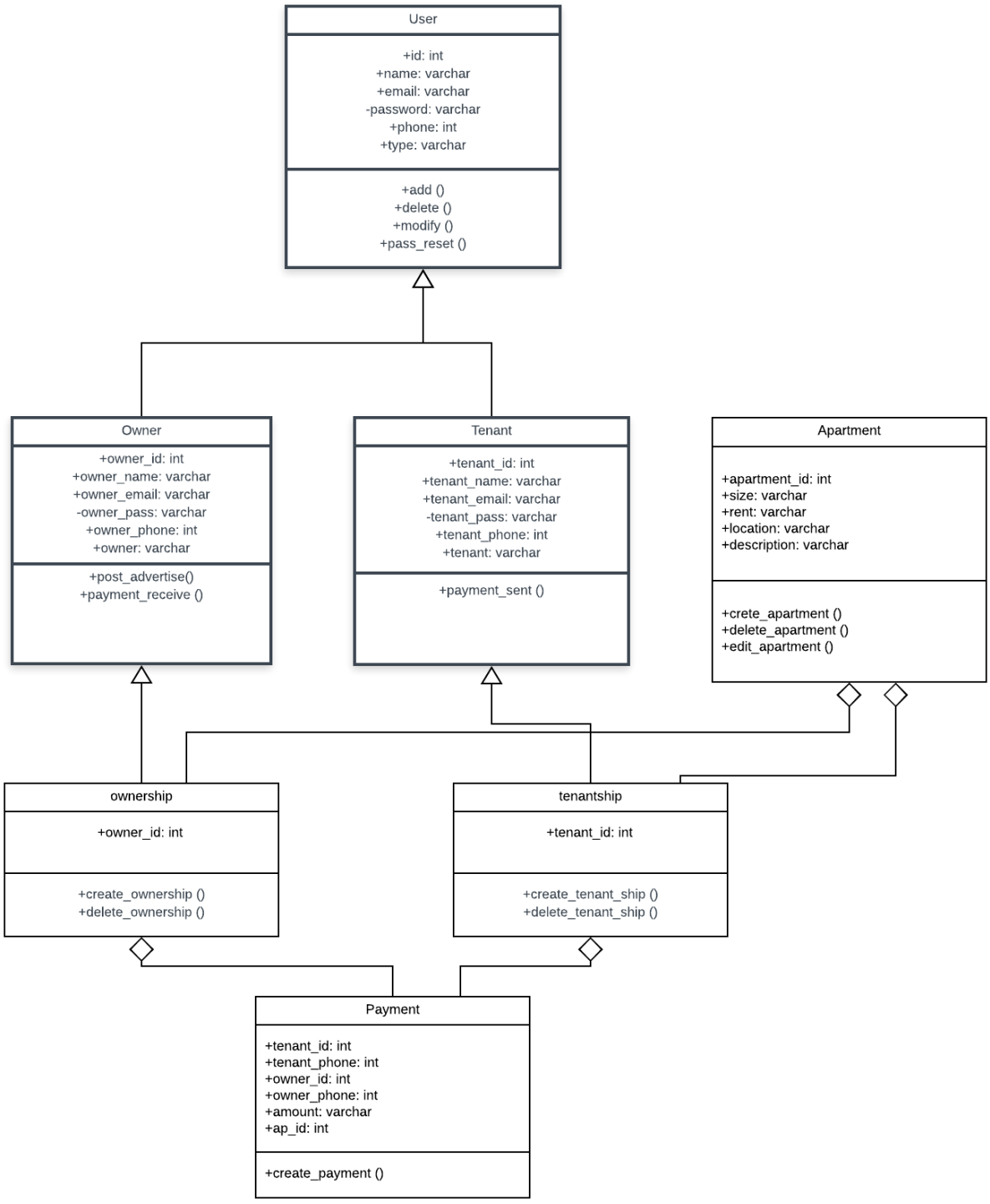
25. Typical Course of Events:

Actor Actions	System Responses
1. Actors click to monthly bill report.	2. System responds to monthly bill report option.
3. Actor can edit and press ok.	4. System check it and return to dashboard.

ER Diagram:

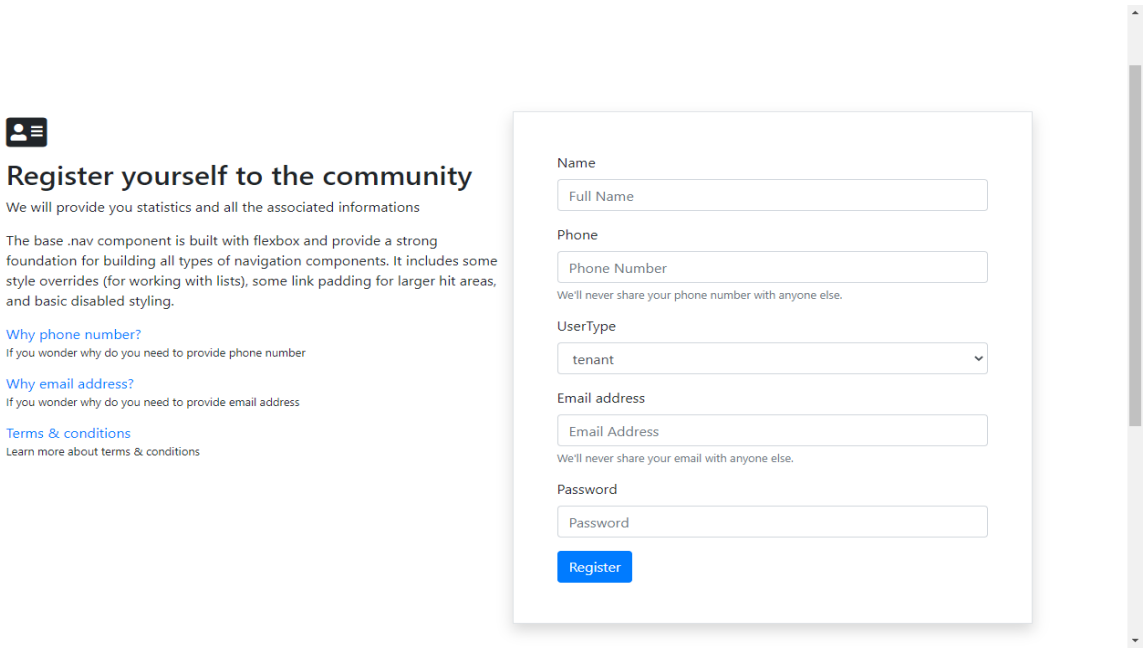

User
id int
name varchar
Email varchar
password varchar
Phone int
Type varchar
Apartment
ap_id int
size varchar
rent varchar
location varchar
Description varchar
Ownership
id int
ap_id int
user_id int
Tenantship
id int
ap_id int
user_id int
Post
id int
ap_id int
owner_id int
date varchar
description varchar
Payment
id int
user_id int
owner_id int
ap_id int
amount varchar
Date varchar

EER Diagram:

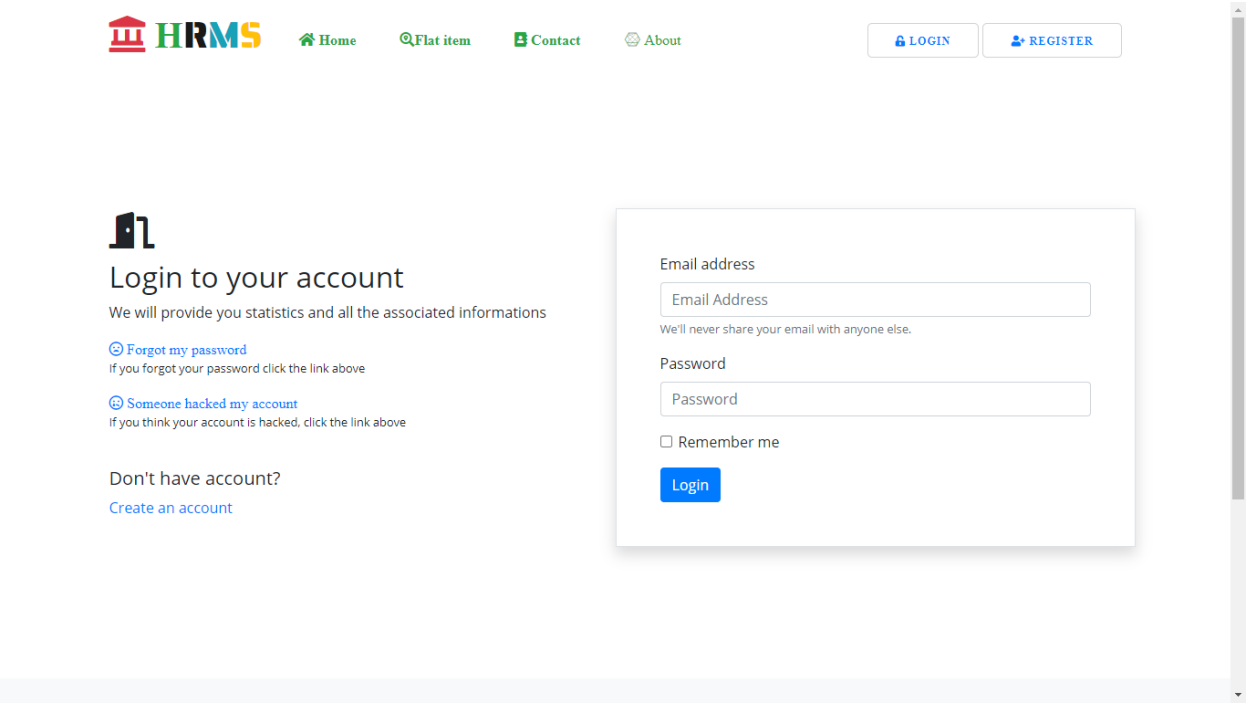

User
+id: int
+name: varchar
+email: varchar
-password: varchar
+phone: int
+type: varchar
+add ()
+delete ()
+modify ()
+pass_reset ()
Owner
+owner_id: int
+owner_name: varchar
+owner_email: varchar
-owner_pass: varchar
+owner_phone: int
+owner: varchar
+post_advertise()
+payment_receive ()
Tenant
+tenant_id: int
+tenant_name: varchar
+tenant_email: varchar
-tenant_pass: varchar
+tenant_phone: int
+tenant: varchar
+payment_sent ()
Apartment
+apartment_id: int
+size: varchar
+rent: varchar
+location: varchar
+description: varchar
+crete_apartment ()
+delete_apartment ()
+edit_apartment ()
ownership
+owner_id: int
+create_ownership ()
+delete_ownership ()
tenantship
+tenant_id: int
+create_tenant_ship ()
+delete_tenant_ship ()
Payment
+tenant_id: int
+tenant_phone: int
+owner_id: int
+owner_phone: int
+amount: varchar
+ap_id: int
+create_payment ()

Some pictures of our web site:

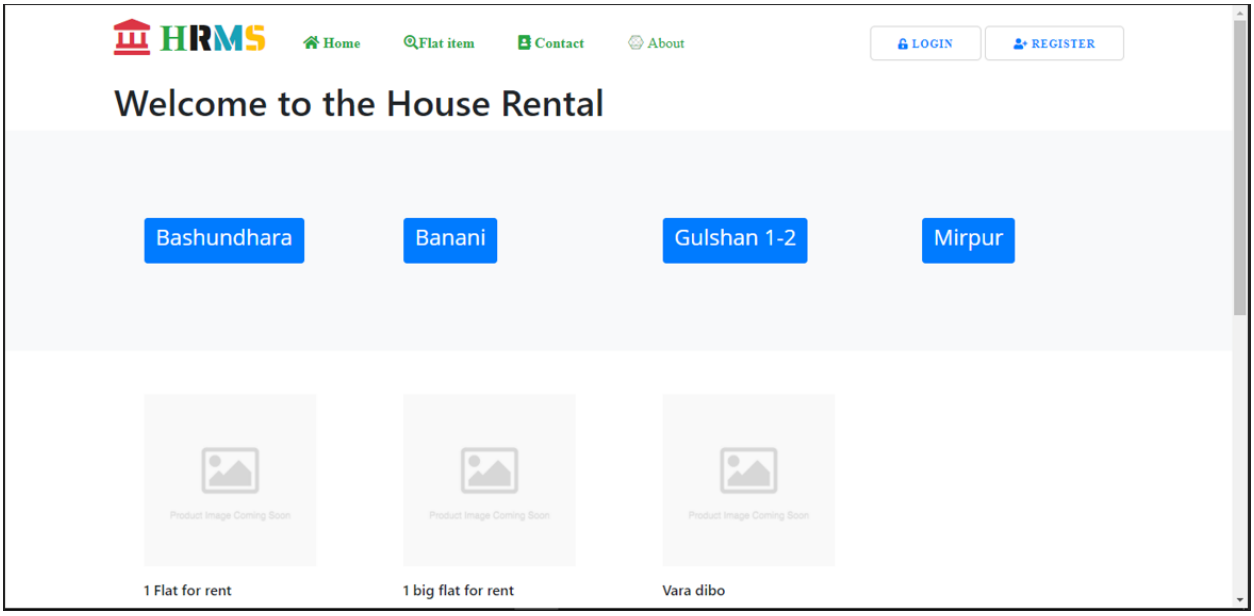
Home Page



Register Page



Login Page



Flat Item search page

Future Work:

We expect to endure better House Rental site for the Bangladeshi peoples. In future people can study about our project and they can add more plans to our project so that we can update more.