Independent University **Bangladesh (IUB)**

IUB Academic Repository

Internship Reports Summer 2023

2023-09-21

An Undergraduate Internship/Project on 'House Rent Management System' software

Syam, Imran

Independent University, Bangladesh

https://ar.iub.edu.bd/handle/11348/692

Downloaded from IUB Academic Repository



An Undergraduate Internship/Project on Topic

By

Imran Syam

Student ID: **1921731**

Summer, 2023

Supervisor:
Dr. Ashraful Islam

Assistant Professor

Department of Computer Science & Engineering

Independent University, Bangladesh

September 21, 2023

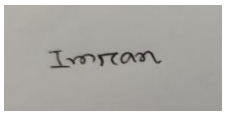
Dissertation submitted in partial fulfillment for the degree of
Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I, Imran Syam (ID: 1921731), hereby certify that this paper, titled "House Rent Management System" is part of my requirements for the Computer Science and Engineering Degree at Independent University, Bangladesh (IUB). I'd like to thank my supervisor, Dr. Ashraful Islam, for his constant advice and support throughout my internship. His remarkable thoughts and experience were critical in executing this project effectively. Moreover, I promise you that the information contained in this report is fully unique, representing my personal insight and expertise obtained throughout my time there. Any third-party resources used in this work have been properly recognized as well as attributed.



Signature Date

19-10-2023

Imran Syam

Name

Acknowledgement

I'd like to thank God for providing me with the opportunity to successfully complete my internship. I'd also like to express my appreciation to family members, classmates, and colleagues for their unwavering reassurance and assistance during my academic career and internship. Their affection and belief in me have served as a constant source of motivation for me.

I'm grateful to the computer science and software development professionals and experts who willingly offered their expertise and resources via academic publications and internet forums. Their input considerably improved my comprehension and skills.

I'd like to say thank you to Dr. Ashraful Islam who is an Assistant Professor of Computer Science and Engineering at Independent University in Bangladesh. He guided me and advised me during my internship and report writing. I'm also grateful to the Independent University in Bangladesh.

Khairat Hossin, a Senior Officer at Rupali Bank Limited, was really polite, friendly, and helpful during my internship. He pushed me to achieve more than I thought I could, and I'm glad I got to work with him.

I appreciate all of the assistance and guidance I received throughout my internship. This experience, I believe, will help me with my future aspirations in computer science. I'm grateful to everyone who has assisted me and been a part of my path.

Letter of Transmittal

01.10.23
Dr. Ashraful Islam
Assistant Professor
Department of Computer Science and Engineering,
School of Engineering and Computer Science.

Subject: Graduation Internship Report Submission

Dear Sir,

I'm eager to turn in my Internship Report, which is required for my Bachelor of Science in Computer Science and Engineering. I'd want to thank you for all of your help and guidance throughout my internship. This report chronicles my experiences during a three-month internship at "Rupali Bank Limited," where I had the privilege of working under the supervision of Mr. Khairat Hossin, the bank's Senior Officer.

This report was written to convey what I learnt and did throughout my internship. I got to experience how the real world operates and put my academic knowledge to use. I followed the guidelines and addressed all of the questions.

I believe that you will are pleased and find this report beneficial. Please let me know your thoughts and suggestions on how I may improve. I want to ensure that I performed a decent job and met the requirements of my internship program.

I'd like to thank you for your continuing assistance and direction throughout my internship. I've learned a lot from your guidance, and I'm looking forward to hearing your opinions on the report. Thank you for this invaluable opportunity.

Sincerely yours,
Imran Syam
ID: 1921731
Department of Computer Science and Engineering,
Independent University, Bangladesh.

Evaluation Committee

Supervision Panel

Academic Supervisor	Industry Supervisor

Panel Members

Panel Member 1	Panel Member 2

Office Use

Program Coordinator	Head of the Department

Abstract

Rupali Bank Limited, a big bank, offered me an internship chance. It has numerous branches, offices, and ATMs throughout the country. They must appropriately manage the locations they rent for their offices.They perform numerous responsibilities connected to renting out space, such as opening new branches. renewing leases and relocating offices are two examples. They also have to maintain their licenses up to date.

During doing these types of activities it is often found that the rent proposal is not sent from the concerned divisional office in due time. Even the proposals are received far later than the expiry date of the rent deed agreement. Staying in a building after the expiration of the agreement is illegal and risky which may be the cause of unexpected situations. Sometimes Building owners take the opportunity of expiration of agreement and demand an excessive rate of rent in comparison with the rent of other financial institutions of the same area. In this situation sometimes authority shifts the branch under compulsion. .

The bank intends to develop 'House Rent Management System' software to monitor their rental agreements in order to address these difficulties.This is the task they have allocated to me. To build this project, I'm using Laravel and React. This combination lets you construct modern, engaging, and robust online apps by combining the capabilities of both technologies.This software will assist them in keeping track of all rental information and will alleviate all of their troubles with this issue.

Contents

Attestation **2**
Acknowledgement **3**
Letter of Transmittal **4**
Evaluation Committee **5**
Abstract **6**
Introduction **1**
1.1 Overview/Background of the Work 1
1.2 Objectives 2
1.3 Scope 2
Literature Review **3**
2.1 Relationship with Undergraduate Studies 3
2.2 Related works 3
Project Management & Financing **4**
3.1 Work Breakdown Structure 4
3.2 Process/Activity wise Time Distribution 5
3.3 Gantt Chart 6
3.4 Process/Activity wise Resource Allocation 7
3.5 Estimated Costing 8
Methodology **9**
4.1 Why Agile for this project? 10
Body of the Project **11**
5.1 Work Description 11
5.2 Requirement Analysis 11
5.2.1 Functional Requirements 12
5.2.2 Non-Functional Requirements 16
5.3 System Analysis 17
5.3.1 Six Element Analysis 17
5.3.2 Feasibility Analysis 17
5.3.3 Problem Solution Analysis 18
5.3.4 Effect and Constraints Analysis 18
5.4 System Design 19
5.4.1 UML Diagrams 19
5.4.2 Architecture 23
5.5 Implementation 25
5.5.1 Frontend 25
5.5.2 BackEnd 25
5.5.3 Routing 26
5.5.4 Database 26
5.6 Testing 26
5.6.1 Input 27
5.6.2 Output 28
5.6.3 Designing Test Cases 29
5.6.4 Test Results 29

Results & Analysis 30
Chapter 7 31
Project as Engineering Problem Analysis 31
7.1 Sustainability of the Project/Work 31
7.2 Social and Environmental Effects and Analysis 31
7.3 Addressing Ethics and Ethical Issues 31
Lesson Learned 32
8.1 Problems Faced During this Period 32
8.2 Solution of those Problems 32
Future Work & Conclusion 33
9.1 Future Works 33
9.2 Conclusion 33
Bibliography 34

List of Figures

Figure 3.1 Work Breakdown Structure 4
Figure 3.3 Gantt Chart 6
Figure 4.1 Agile Development 9
Figure 5.1 Rich Picture 11
Figure 5.2 Entity Relationship Diagram 19
Figure 5.3 Use Case Diagram 20
Figure 5.4 Activity Diagram for add listings 20
Figure 5.5 Activity Diagram for updating listings 21
Figure 5.7 Activity Diagram for all listings 22
Figure 5.8 Activity Diagram for critical listings 22
Figure 5.9 Activity Diagram for expired listings 23
Figure 5.10 Model View Controller Architecture 24
Figure 5.11 Frontend Screenshots 25
Figure 5.12 Backend Screenshots 25
Figure 5.13 Routing Screenshots 26
Figure 5.14 Database Screenshots 26
Figure 5.15 Input test case TC001 Screenshoot 27
Figure 5.17 Output test case TC001 Screenshoot 28
Figure 5.18 Output test case TC001 Screenshoot 28
Figure 5.19 API endpoint 30

List of Tables

Table 3.1 Process/ Activity wise Resource Allocation..... 7
Table 3.2 Estimated Costing 8
Table 5.1 Functional Requirement 15
Table 5.2 Six element analysis..... 17
Table 5.2 Input table with Field 27
Table 5.3 Output table with Field..... 28
Table 5.4 Designing Test case Input/output Pass/Fail table 29

Chapter 1

Introduction

Internships are an important way for students to obtain practical experience, match their career goals, and realize their full potential. They help students improve their skills by bridging the gap between academic understanding and real-world application. Independent University in Bangladesh offers internships as part of its "Computer Science and Engineering" degree, which is required for graduation. As an undergrad, I recently finished a three-month internship at the prestigious "Rupali Bank Limited." In this report, I will discuss my internship at Rupali Bank, including details on the work I did, the valuable lessons I learned, and how this experience aided my professional development.

1.1 Overview/Background of the Work

I Joined Rupali Bank Limited as an Intern on 31 July 2023. As part of my internship, I got the opportunity to work on one of their projects, House Rent Management System. This project is built using Laravel, React, Shadcn/ui, MySQL and other javascript libraries and packages. I used React for the frontend part of the project, Laravel for the backend, MySQL for the database and Shadcn/ui for styling.

The component-based architecture of React enables for the construction of reusable UI components, making complex user interfaces easier to build and maintain. Laravel is a robust and well-documented PHP framework that includes capabilities such as routing, authentication, database connections with Eloquent, and request middleware. Laravel is great for creating RESTful APIs that the React frontend can utilize to communicate with the backend. MySQL is a well-known relational database management system that is well-suited for structured data storage and retrieval. Shadcn/ui provides pre-designed UI components and styles that can be changed to meet the visual design requirements of the project.

So there are almost 586 offices of this bank around the country. Most of those offices are rented. They often face problems. Sometimes, they get rent proposals very late from the divisional office, even after their rental agreements have expired. To solve these issues, the bank wants to create House Rent Management software to monitor their rental agreements.

1.2 Objectives

This system was designed with specific goals in mind, they are as follows:

- Create a central data archive to manage and supervise all rent-related actions.
- Knowing which office's rental agreement needs to be renewed beforehand.
- Digitalize the rental management process making it more easy and efficient.
- Developing a super friendly application for the user.

1.3 Scope

The scopes of this project are as follows:

Data Archive: Create a single data repository to manage and oversee all rent-related activities at Rupali Bank Limited. This archive will contain rental information on branches, sub-branches, collecting booths, and ATM booths.

Risk-Based Classification: Implementing a risk-based classification system (Red, Yellow, Green) for branches based on the expiration dates of rent agreements. This classification will assist in prioritizing activities and decision-making by identifying branches that are more likely to face illegal or operational challenges as a result of expiring agreements.

Rent Renewals on Time: Ensure that branch rent renewal agreements are completed on time and before the present agreements expire. This goal attempts to prevent illegal building occupations and minimize operating disruptions.

Authentication: The program provides a simple and user-friendly sign-in experience. After you sign in, you may simply view the services that are accessible for your unique user role.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

During my bachelor's degree studies at IUB, I received essential information and abilities that were crucial in the development of the "House Rent Management System." Working on this project would have been far more difficult without the solid foundation I got from these classes. Some of the most important courses that influenced my project work include:

Object-Oriented Java Programming (CSE213): This course gave me a good foundation in object-oriented programming (OOP) ideas.

Web Application Development (CSE309): In this course I learned Laravel, PHP, Javascript. This course helps me to get started with the language and framework that I'm using in this project.

System Analysis and Design (CSE307): The principles I learnt in this course, notably requirement analysis and diagram creation, aided my comprehension of customer requirements significantly. This understanding helped me design the project appropriately.

Database Management (CSE304): This course introduced me to fundamental concepts like entity-relationship diagrams, MySQL and database design. In my project, I used these concepts while creating the project's database utilizing MySQL.

Data Structures and Algorithms (CSE203 & 211): These courses helped me better understand core programming ideas, which came in handy while dealing with the architecture of a new programming language during the project.

2.2 Related works

This project is just for internal use within Rupali Bank and is not intended for public use. There are a lot of projects similar to this. Some of them are mentioned below:

1.
   - Source code: https://github.com/Shahed-Chy-Suzan/property-listing
   - Language: PHP, Blade, Shell
2.
   - Source code: https://github.com/Shahed-Chy-Suzan/HouseRental
   - Language: PHP, Blade, Digital command Language, others
3.
   - Source code: https://github.com/sheikhshakil/rentopedia-public
   - Language: EJS, javaScript, CSS

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

There are several jobs that must be completed as part of a project. The Work Breakdown Structure (WBS) is a diagram that depicts how various tasks are connected and arranged. It is beneficial to divide the project into smaller, more manageable chunks. The work breakdown structure (WBS) can help project managers plan and manage their projects more effectively. The work breakdown structure (WBS) may help with things like allocating resources, assigning tasks to team members, and monitoring project status. The WBS may also be used in conjunction with a Gantt chart, which is a diagram that depicts when each job should begin and end. This manner, the team will know what to do and when to do it. The work breakdown structure (WBS) is critical for project management since it aids in planning, doing, and regulating.

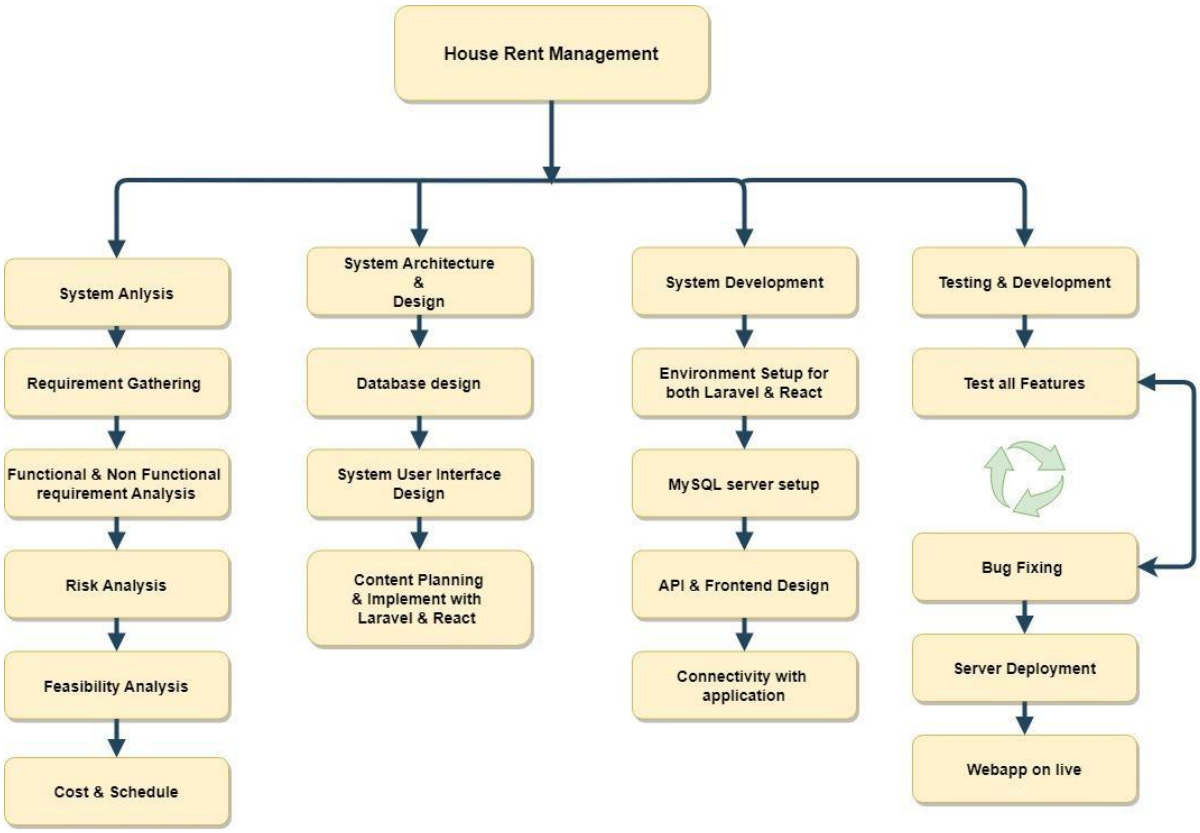


Figure 3.1 Work Breakdown Structure

3.2 Process/Activity wise Time Distribution

When determining the duration of a project, it's critical to break it down into all of the distinct activities that must be completed. One important aspect of this is determining the "critical path." This is the most crucial sequence of jobs that needs to be fion time to ensure that the entire project stays on track. The critical route is defined by the fact that each job in it is dependent on the one before it being completed. As a result, if one item in the critical route is delayed, it might disrupt the entire project timeframe. It's like a chain reaction where everything has to happen in the proper order and on time for the project to be completed.

The project consists of a detailed list of activities and processes that must be completed in a specified order to achieve the intended result. Project managers may efficiently plan and assign resources to ensure the project stays on track by looking at the critical path and understanding how activities rely on each other in terms of time and completing its tasks on schedule.

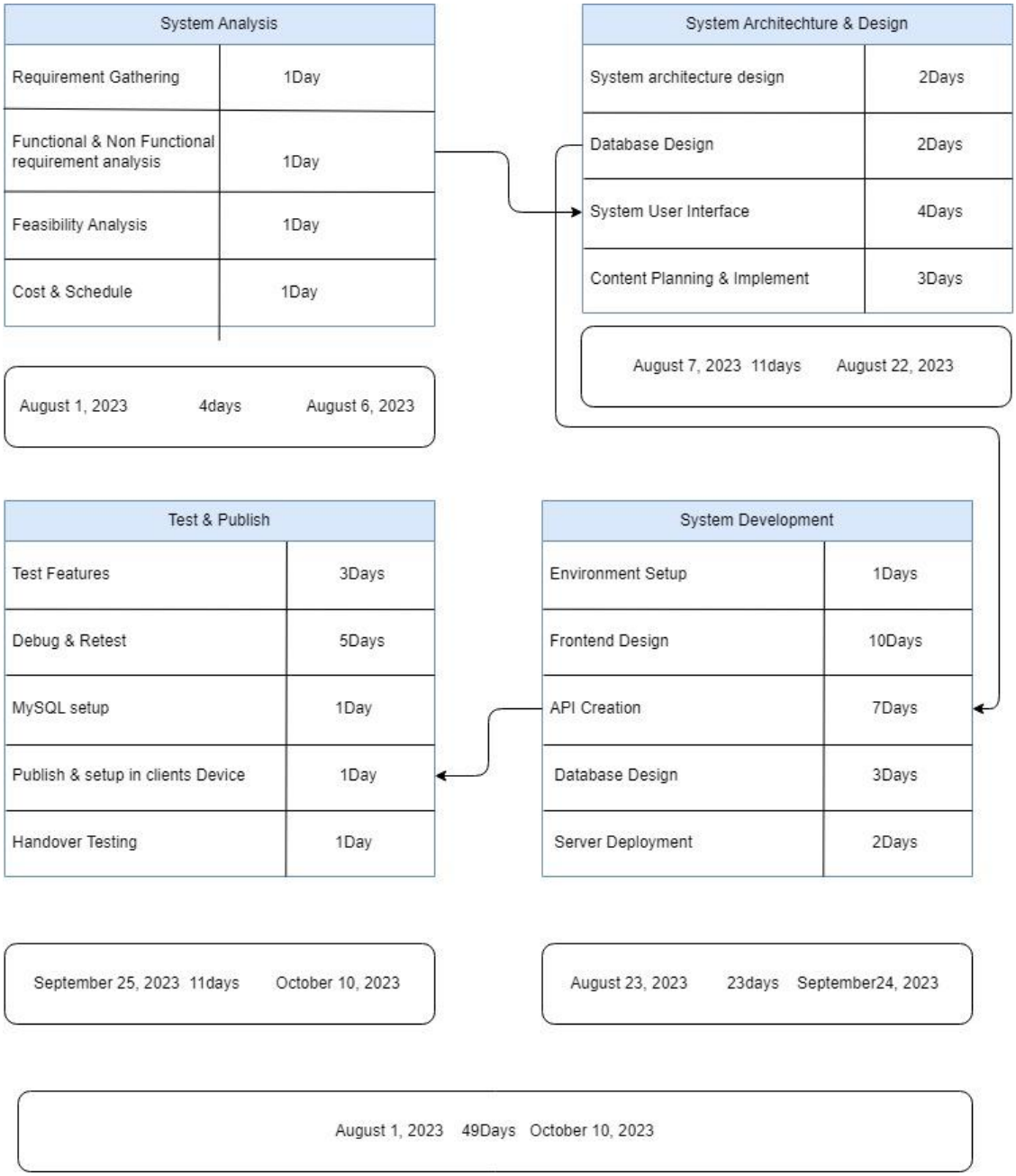


Figure 3.2 Time Distribution

3.3 Gantt Chart

During the application project's planning phase,Gantt chart is used for monitoring and visualizing project timelines. Gantt charts provide useful information about job dependencies, progress monitoring, prioritizing, and resource allocation. These bar chart-style charts give a clear and simple overview of jobs as well as their corresponding deadlines. The primary resources selected for our project were the developers, and the critical assets comprised office PCs and servers necessary for execution. Each team member was allocated distinct duties with related dates, and their coordinated efforts resulted in the project's successful completion.

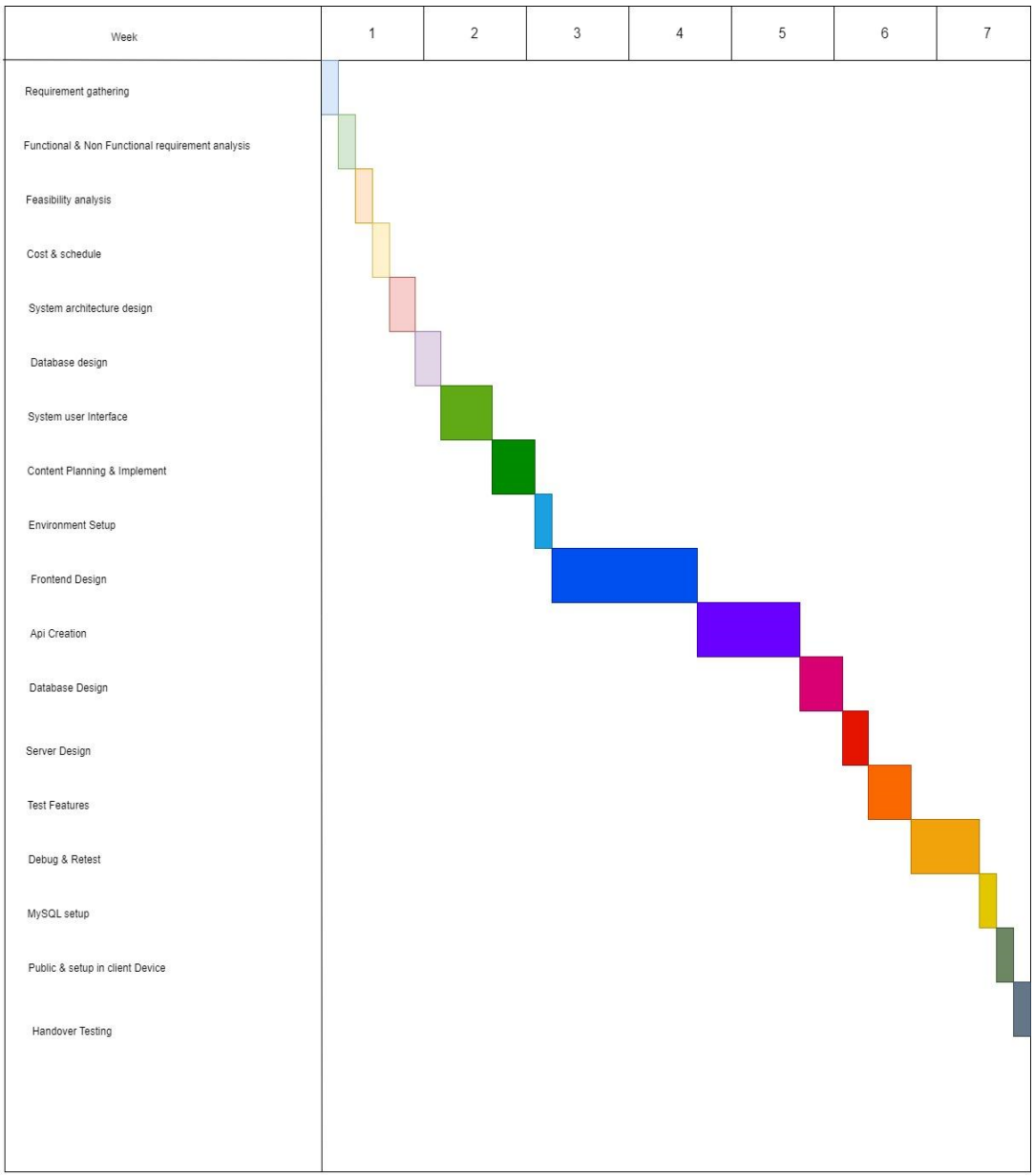


Figure 3.3 Gantt Chart

3.4 Process/Activity wise Resource Allocation

The developers are crucial to this project. Following that are the office Computers and the servers, which are required to complete the project. I completed this job entirely on my own. Each project stage is described in full below.

Requirement Analysis: During the first week, the supervisor and I had in-depth conversations to determine the project's needs. This included selecting the best software/technology stack, describing key functionality, and assessing the resources at our disposal.

Planning: Several planning sessions were held in order to create a complete project roadmap. Throughout this process, I carefully addressed numerous components such as the overall project framework, task breakdown, creating attainable milestones, and establishing clear deadlines to guarantee timely development.

Design: During the design process, we concentrated on making the application pages appear excellent. I was involved in the design of the user interface and experience. I utilized tools like FIGMA to produce both high-level and detailed mockups for the project. This gave us a clear image of what the program might perform.

Implementation: The development process began after the design specifications were finalized. As a developer, my major emphasis was on writing functional code for the application's backend, ensuring the seamless execution of the planned features and functions.

Testing: Testing was an essential component of the entire development process. We initiated testing methods as soon as a new feature was implemented to discover and fix any faults or errors. We did unit testing at the end of the implementation process to ensure the application's stability and reliability.

Deployment: We discovered that the project was running behind time after We completed the testing phase. To guarantee a seamless launch of the application, We had to work extra hard to speed up the deployment process and address any lingering difficulties or errors.

Task	Start Date	End Date	Duration	Complete
System Analysis	Aug 1st, 2023	Aug 6th, 2023	4 Days	20%
System Architecture & Design	Aug 7th, 2023	Aug 22th, 2023	11 Days	20%
System Development	Aug 23th, 2023	Sep 24th, 2023	23 Days	40%
Test & Development	Sep 25th, 2023	Oct 10th, 2023	11 Days	20%
Total			49 Days	100%

Table 3.1 Process/ Activity wise Resource Allocation

3.5 Estimated Costing

Some factors influence the cost of software. They are as follows:

1. Development Costs (50,000tk):

- The software is created by a team of individuals that specialize in various tasks such as development, design, and testing. They are paid by the hour for their services. The overall cost of the app is determined by the number of hours worked and the hourly rate.

2. Infrastructure Fees (25,000tk):

- To function, the app requires servers, hosting services, and other resources. These items are expensive. The more the software does and the more difficult it is to create, the more money it costs.

3. Maintenance Fees (20,000tk):

- When the app is finished, it will still require some maintenance to keep it working well. This include repairing bugs, improving it, and adding new features.

Cost Estimation

Task	Cost Amount in BDT
Development	**50,000tk**
Infrastructure	**25,000tk**
Maintenance	**20,000tk**
Total	**95,000tk**

Table 3.2 Estimated Costing

Chapter 4

Methodology

This chapter describes how we approach our project. We explain what we do and the rules we must follow. Choosing the best way to do things is critical because it allows us to fulfill our project goals while also utilizing the best ways in our sector. The strategy we've chosen allows us to work on the application in an organized manner while also utilizing cutting-edge technologies.

The Software Development Life Cycle (SDLC) is similar to a project management plan. It depicts the various processes involved in developing an information system, from conception to maintenance. There are numerous SDLC approaches, each with its own set of processes and procedures for ensuring the project's success. For this project, I picked the agile methodology. Agile is all about taking incremental steps and communicating frequently with the individuals that want the system. It's not about doing everything all at once, but about improving it gradually.

With Agile, I had to keep discussing and getting feedback all the time, and I had to keep improving things. This way, I could make quick changes if necessary. The goal was to ensure that the solution I designed was exactly what the customers desired.



Figure 4.1 Agile Development

4.1 Why Agile for this project?

We picked the Agile methodology for our project since it provides several advantages in the development of a software. One significant advantage is the staged software release, which allows us to make constant enhancements and obtain feedback. The Agile technique of segmenting the development process helps us to identify and fix problems early on. This iterative technique improves productivity while lowering the likelihood of costly mistakes. When development began, there was a shortage of information in the project's needs section, but these data were eventually supplied as the project progressed. As new features and design upgrades arrived, they were gradually included in accordance with Agile principles. A portion of the coding work was implemented each day, followed by a meeting between the development team and the management committee to confirm that the created functionality matched the standards and requirements, with any proposed improvements applied appropriately.

Chapter 5

Body of the Project

The project chapter's content contains all of the project specifics. Readers will find detailed information on the project here, including a project summary, requirements analysis, system assessment, system design, development, and testing, all of which will be thoroughly documented.

5.1 Work Description

This project's goal is to develop a single data repository to monitor and supervise all rent-related activities. Because they have a large number of offices all around the country. All rent-related actions are managed by the head office. The goal of this initiative is to determine whose offices' leasing agreements need to be renewed ahead of time. Typically, the local office tells the Zonal office, the Zonal office informs the Divisional office, and the Divisional office informs the Head office. When they notify the Head Office, it is sometimes too late to extend the arrangement. The entire point of this is to prevent this from happening. They can use this initiative to see which offices' leasing agreements need to be renewed.

5.2 Requirement Analysis

Rich Picture

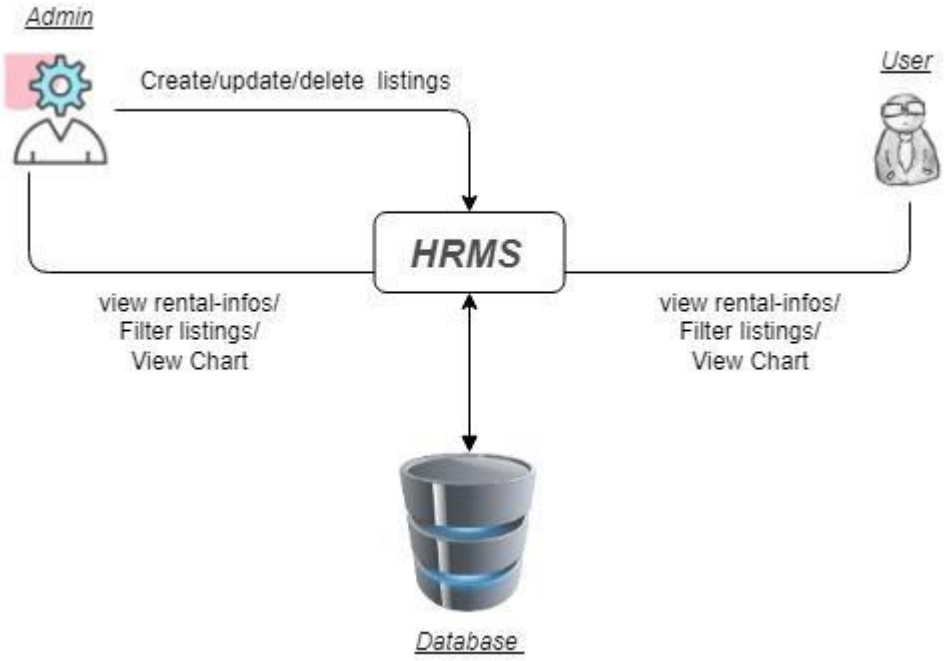


Figure 5.1 Rich Picture

5.2.1 Functional Requirements

<table>
<tr><td colspan="3">Name of the function: View rental-information</td></tr>
<tr><td>Process:
1. Sign into the website
2. Click on the navbar icon on the top right.
3. The drop down will appear.Click on the “Rental Info”. The user will be redirected to /rental-infos end point.</td><td>Output:
1. The user will view all the rental-info at the left side of the page.
2. The right side of the page will load a map with the latitude and longitude of dhaka.
3.If the user click on one of the rental card the map will load the location of that rental-info</td><td>Priority:
High</td></tr>
<tr><td colspan="3">Precondition:
• Users should have proper access to the system.
• Users should be familiar with the system.
• System should be able to request database for data
• System should be able to receive data from the database based on the send request.</td></tr>
<tr><td colspan="3"><u>Post condition:</u>
• The user views list of rental-infos and a map
• The user stays within the interface
If there is no data found, an alternative component will be shown to the user.</td></tr>
</table>

<table>
<tr><td colspan="3">Name of the function: create a listing</td></tr>
<tr><td><u>Process:</u>
1. Sign into the website
2. Click on the navbar icon on the top right.
3. The drop down will appear.Click on the “Rental Info”. The user will be redirected to /listings end point.
4. There will be a table containing all the lists.Each table has an action colum.Click on the 3 dots. New option will appear in the window. Click on the update option.
5. The user will be redirected to listings/[listingId]/edit. listingId will be the id of that listing user going to update.
6. User will view a form filled</td><td><u>Output:</u>
1. After updating a listing, a toast message will be shown.
2. If listing is successfully updated then the user will be redirected to the /listings end point</td><td><u>Priority:</u>
High</td></tr>
</table>

with all the listing values. 7. Users can change field values. 8. After updating the value, the user will click the “Save changes” button to update the listing.		

Precondition:

• Users should have proper access to the system. Only admin have the access to this option.

• Users should be familiar with the system.

• System should be able to request database for data

• System should be able to receive data from the database based on the send request.

Post condition:

• The user views a listing form filled will the value of that specific listing

• The user stays within the interface

Alternative Options:

• The user views a listing form.

• The user stays within the interface.

• If any of the fields is missing the form will not be submitted and an error message will be shown to the user.

Name of the function: update a listing

Process:	**Output:**	**Priority:**
1. Sign into the website 2. Click on the navbar icon on the top right. 3. The drop down will appear.Click on the “Rental Info”. The user will be redirected to /listings end point. 4. The user will view a table with all the listings. Each listing has an action. 5. Click on the 3 dots. Three options will appear on the screen. Click on the update option. Then the user will be redirected to a new page with endpoint /listings/[listingId]/edit. The listingId will be the id of the listing user that wants to	1. After updating a listing a toast message will be shown. 2. If listing is successfully updated then the user will be redirected to the /listings end point	High

update. 6. User will view a form filled with the listing value. Users can change the value. 7. After changing values users have to click on the Save changes button to update the listing.		

Precondition:

• Users should have proper access to the system. Only admin have the access to this option.

• Users should be familiar with the system.

• System should be able to request database for data

• System should be able to receive data from the database based on the send request.

Post condition:

• The user views a listing form filled with the value of a specific listing.

• The user stays within the interface.

Alternative Options:

• If any of the fields is missing the form will not be submitted and an error message will be shown to the user.

• If the listingId is invalid user will not be redirected to the correct page.

Name of the function: Delete a listing		
Process: 1. Sign into the website 2. Click on the navbar icon on the top right. 3. The drop down will appear.Click on the “Rental Info”. The user will be redirected to /listings end point. 4. The user will view a table with all the listings. Each listing has an action. 5. Click on the 3 dots. Three options will appear on the screen. Click on the delete option. A dialog box will appear in front of the user. 6. If the user clicks cancel nothing happens. The dialog box will be closed . 7. If the user clicks on the delete	**Output:** 1. After clicking on delete a toast message will be shown. 2. If listing is successfully deleted then the user will be redirected to the /listings endpoint and the dialog box will be closed	**Priority:** High

button the listing will be deleted.		

Precondition:

• Users should have proper access to the system. Only the admin has the access to delete a listing .

• Users should be familiar with the system.

• System should be able to request database for data

• System should be able to receive data from the database based on the send request.

Post condition:

• The user views a listing table.

• The user stays within the interface.

Alternative Options:

• If the listingId is invalid user will not be redirected to the correct page.

Name of the function: View filtered rental-information

Process:	**Output:**	**Priority:**
1. Sign into the website 2. Click on the navbar icon on the top right. 3. The drop down will appear.Click on the “Rental Info”. The user will be redirected to /rental-infos end point. 4. Rental info list can be filtered by rental_type, division, district, upazila, approval.	1. The user will view filtered rental-info at the left side of the page. 2. The right side of the page will load a map with the latitude and longitude of dhaka. 3.If the user click on one of the rental card the map will load the location of that rental-info	High

Precondition:

• Users should have proper access to the system.

• Users should be familiar with the system.

• System should be able to request database for data

• System should be able to receive data from the database based on the send request.

Post condition:

• The user views list of rental-infos and a map

• The user stays within the interface

If there is no data found, an alternative component will be shown to the user.

Table 5.1 Functional Requirement

5.2.2 Non-Functional Requirements

1. Performance Requirements

The output of the program is critical for evaluating its performance, and having defined requirements is necessary for researching a system. Under heavy workload conditions, HRMS should be able to use quick, optimal responses that are tailored to the user's needs.

2. Safety Requirements

HRMS needs to have a backup of the database with only top-level management and dedicated stakeholders having access. In this system, ineligible users should be denied access. Finally, the system must deal with data entry errors caused by unreliable connection during system setup or data entering.

3. Security Requirements

HRMS must have perfect data integrity, as well as the ability to detect and stop unwanted access to sensitive data (data breach) and system configuration. Any data loss caused by backend services such as database breakdown or unwanted deletion must be avoided and handled by the system.

4. Software Quality Attributes

Availability:HRMS will be able to maintain data integrity, be dependable, and provide secure confidentiality qualities in all security elements. During a system outage, all system data will be accessible as well as data backup functionality.

Interoperability: HRMS will be able to interact and operate in concert with the frontend and backend portions of the system, as well as perform specific analytical and instructive activities. Because data can be transmitted between two or more systems and also understood by each system, HRMS should be able to handle semantic interoperability.

Performance: HRMS must be available to work and must adhere to strict time constraints. Because the software's data, analytical properties, and CRUD activities are all handled by cloud-based servers, the application's overall time, timing bounds and loading issues will be considerably reduced.

Testability: No meaningful remarks can be made at this time because the system is still in the design and development stage. However, software testing specialists should still be able to execute tests based on preset criteria. Under development testing criteria, the system should evaluate how easy it is for software Quality assurance engineers. Engineers should be able to use a variety of techniques in the system, including encapsulation, interfaces, patterns, reduced coupling, and more.

Security: HRMS must have perfect data integrity, as well as the ability to detect and stop unwanted access to sensitive data (data breach) and system configuration. Any data loss caused by backend services such as database breakdown or unwanted deletion must be avoided and handled by the system. It is critical for System Tester specialists to undertake updated security checks on systems on a frequent basis.

Usability: HRMS is meant to be simple to use in order to maximize user usage. All of the system's user task criteria should be met.

Functionality: HRMS is able to achieve the highest level of functional and specification conformance. As it adheres to the critical, requirement, and design phases of modern applications, it achieves the highest desirable functionality for the system during the early stages of software testing.

Finally, because HRMS ensures that all of the above features are available, we are able to maintain and develop a balanced and highly functional web-application for our clients.

5.3 System Analysis

5.3.1 Six Element Analysis

Process	Human	Hardware (Computing)	Software	Database	Network
Login	All	Phone/Tablet/PC	HRMS	retrieve data by email	Internet
Add Listing	Admin	Phone/Tablet/PC	HRMS	store data to database	Internet
Update Listing	Admin	Phone/Tablet/PC	HRMS	update data of the specific listing to database	Internet
Delete Listing	Admin	Phone/Tablet/PC	HRMS	delete data of the specific listing from database	Internet
Show Listing	All	Phone/Tablet/PC	HRMS	retrieve data of the specific listing from database	Internet
Show All Listing	All	Phone/Tablet/PC	HRMS	retrieve all listings from database	Internet

Table 5.2 Six element analysis

5.3.2 Feasibility Analysis

The HRMS (House Rent Management System) project is mostly dependent on feasibility studies to see if it is a good idea and if we can make it operate efficiently. We do many sorts of feasibility studies to examine various aspects of the project to determine if they make sense.

Technical Feasibility:
In this evaluation, we examine the present resources, such as hardware, software, and technology, to establish the technical feasibility of the project. We discuss future enhancements and the expertise of the developers involved. HRMS, being a CRUDprogram, allows for simple scaling based on user numbers. Nonetheless, the program's technological intricacy may provide issues for users who are unfamiliar with technical subjects.

Operational Feasibility:
This evaluation focuses on how well the software does the activities necessary to address business difficulties and fulfill user demands. It considers the people involved, particularly the software development team, and imagines how the program will work and function once it has been produced and implemented. The HRMS application is mostly used by Rupali Bank employees, and it is written in English.

Economic Feasibility:
The goal of this study is to establish whether the software can help the firm's financial growth. It comprises evaluating the costs involved with the software development team, estimating hardware and software costs, and reviewing the costs related with the feasibility study. The aim is to make sure that the HRMS project is a profitable investment for the clients. The time saved and exact computations offered by HRMS will rapidly justify the

product's cost. A rigorous feasibility analysis ensures that HRMS is not only technically viable and operationally ready, but also economically beneficial. It provides the project team with the knowledge they need to make educated decisions, address potential difficulties, and maximize their chances of success.

5.3.3 Problem Solution Analysis

Problem analysis requires recognizing and precisely outlining the difficulties that demand attention, whereas problem resolution focuses on identifying solutions that correspond with the problem's criteria and limits. Multiple issues with the Shaharjo project were identified, and remedies were devised.

5.3.4 Effect and Constraints Analysis

Numerous restrictions have been found that have influenced the development and execution of the solution, affecting numerous project factors such as the budget, timeframe, and scope.

Budget Impact: The financial limitation is crucial in defining the project's resources and length. It has a direct impact on the amount of people who can be employed, as well as the overall development process. The budget determines the size, scope, and availability of necessary resources for the project.

Time Impact: The money and time restrictions are inextricably linked. The project's schedule is mainly dependent on the authorized funding. A two-month schedule was set for the HRMS project, with an October 30th deadline. This time limitation puts additional burden on the development team, mandating effective planning and execution to accomplish the project's objectives within the timeframe specified.

Scope Impact: The scope constraint establishes the project's boundaries and needs. It determines if the project's set requirements can be satisfied. Close monitoring of project requirements is required during the development period. If a deadline approaches, changes to the project's scope, delayed delivery, or prioritizing of certain features may be required. This limitation necessitates continuous conversations, evaluations, and meetings in order to maintain the project's quality.

Implementation Impact: HRMS adoption greatly improves the manual rental management procedure. Initially, not all stakeholders understood the significance and benefits of this digital solution. However, as the system revolutionizes maintenance and general ease, all parties involved stand to benefit. The system has some outstanding characteristics.

Adaptation Constraint: The change from a wholly manual process to a digital HRMS system provides a difficulty in terms of adjusting to the new platform. It takes time for people to become acquainted with the digital interface. Furthermore, original project needs were not completely resolved, resulting in continuous discussions and the introduction of additional requirements. However, these modifications did not demand a total project restart. Management separated the requirements into manageable milestones within the delivery date, ensuring that the project was completed on time.

5.4 System Design

5.4.1 UML Diagrams

Entity Relationship Diagram

UML diagrams are similar to visual maps created with the Unified Modeling Language (UML). They assist us in displaying various components of a software program in a consistent manner. Actors, roles, actions, classes, or objects can all play these roles. UML diagrams are useful for understanding, altering, storing, or writing down information about a system. They're fantastic because they allow software specialists, designers, and project participants to understand how a complicated software system is designed, how it functions, and how different elements are linked. This makes it easy to learn and discuss.

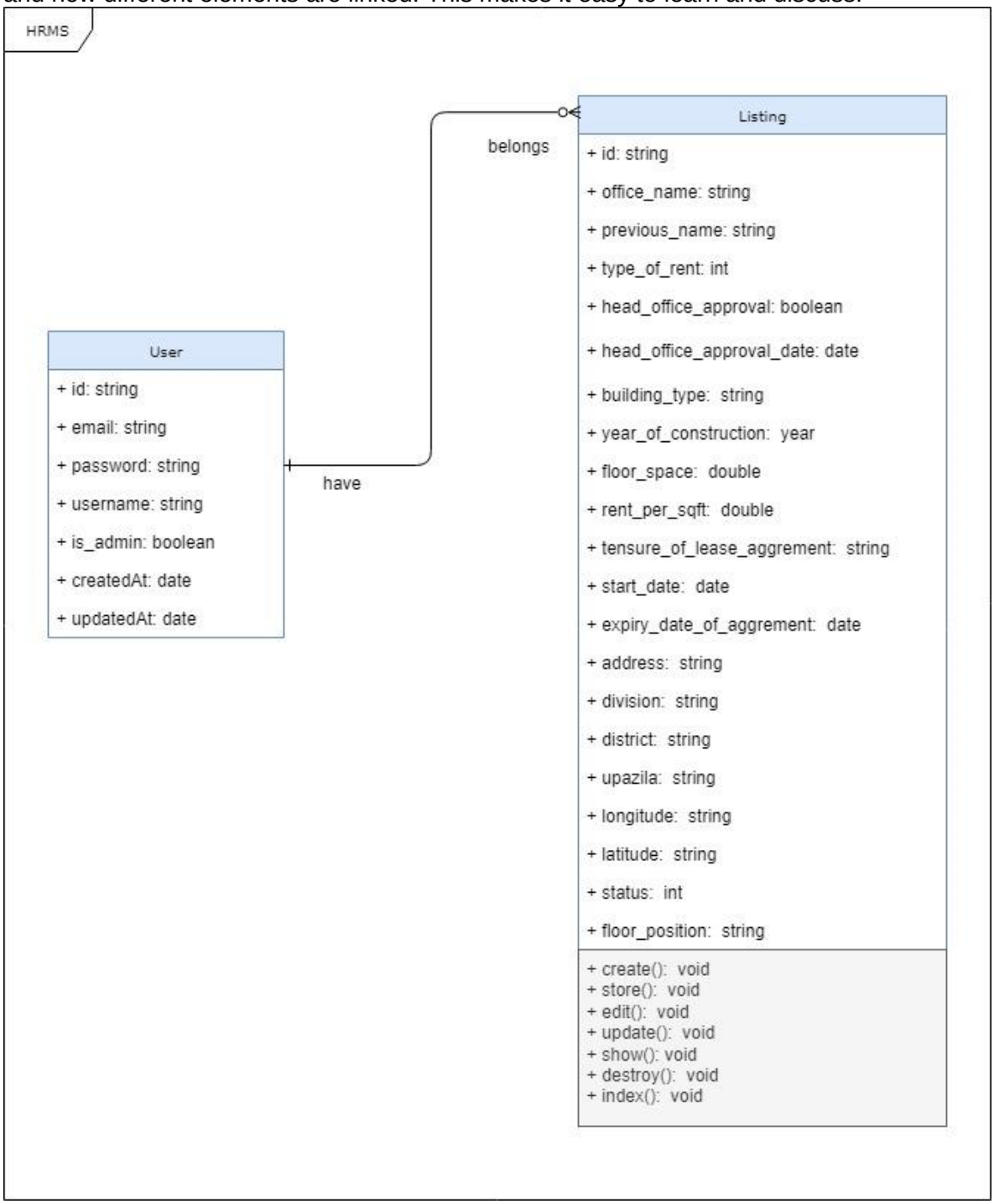


Figure 5.2 Entity Relationship Diagram

Use Case Diagram

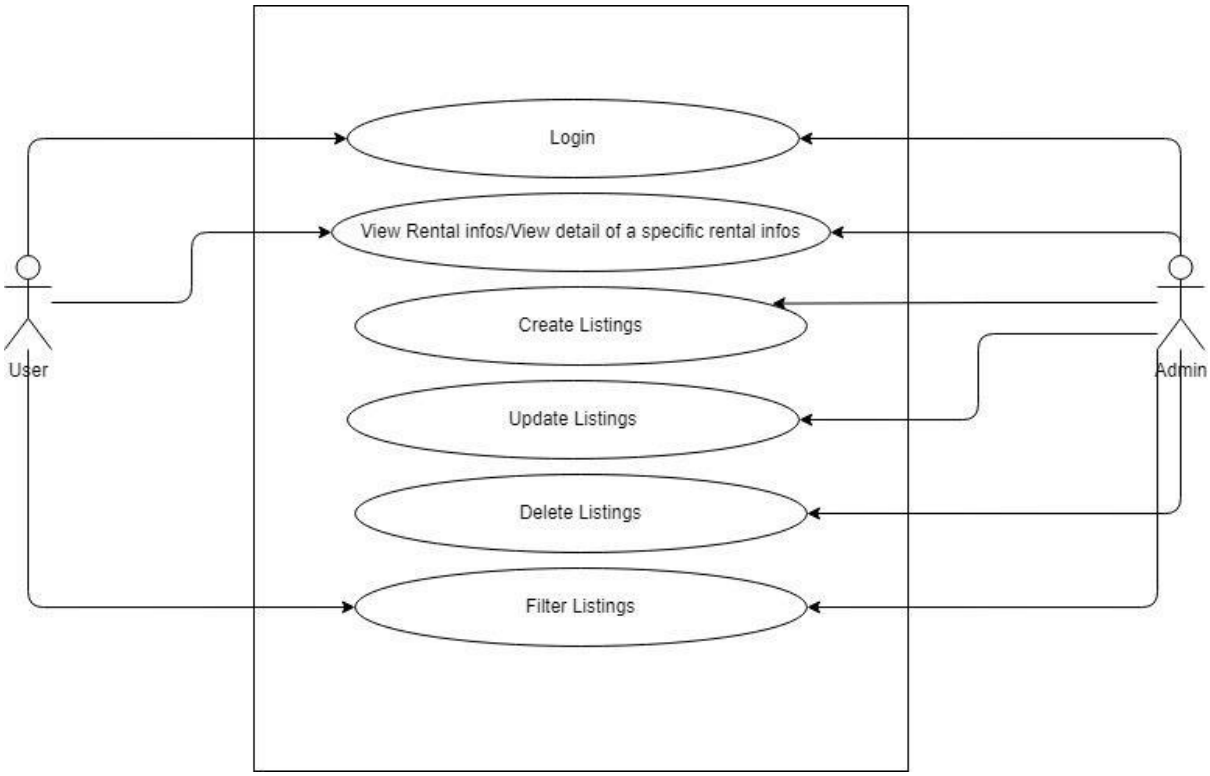


Figure 5.3 Use Case Diagram

Activity Diagram

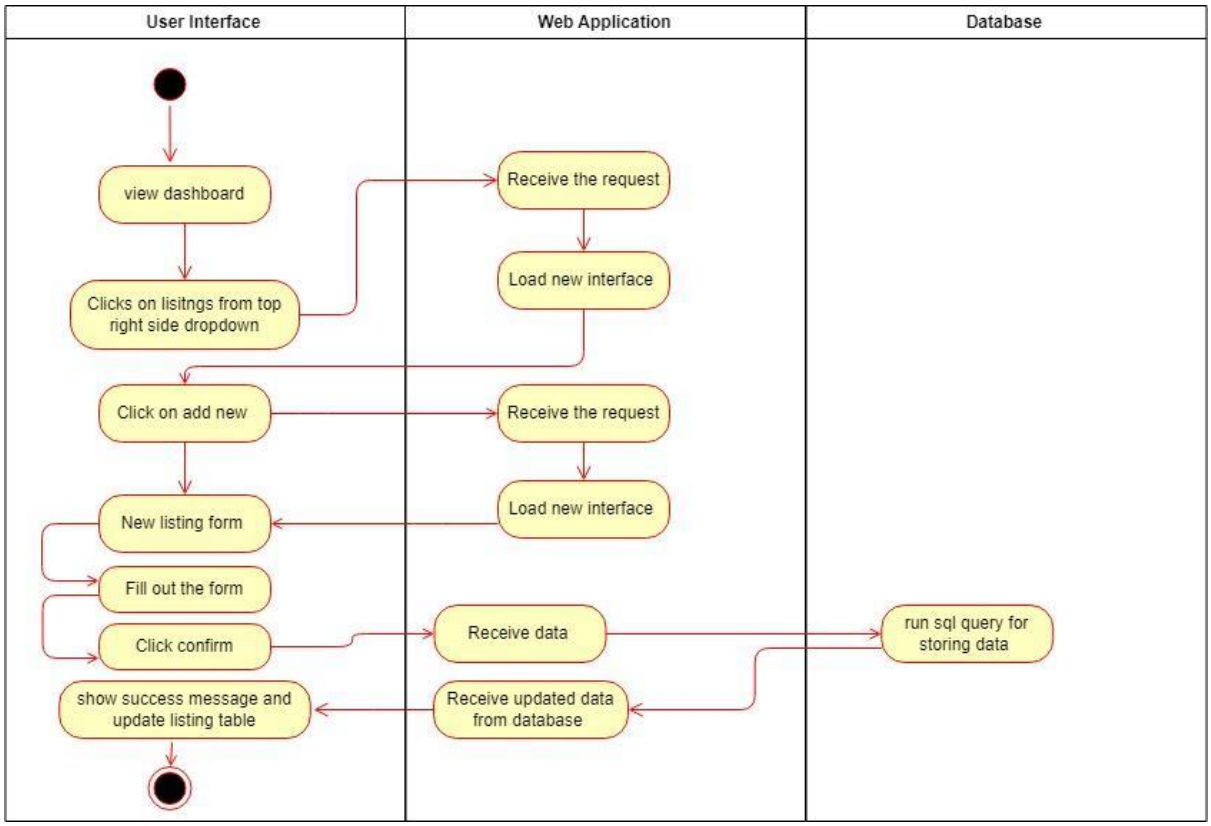


Figure 5.4 Activity Diagram for add listings

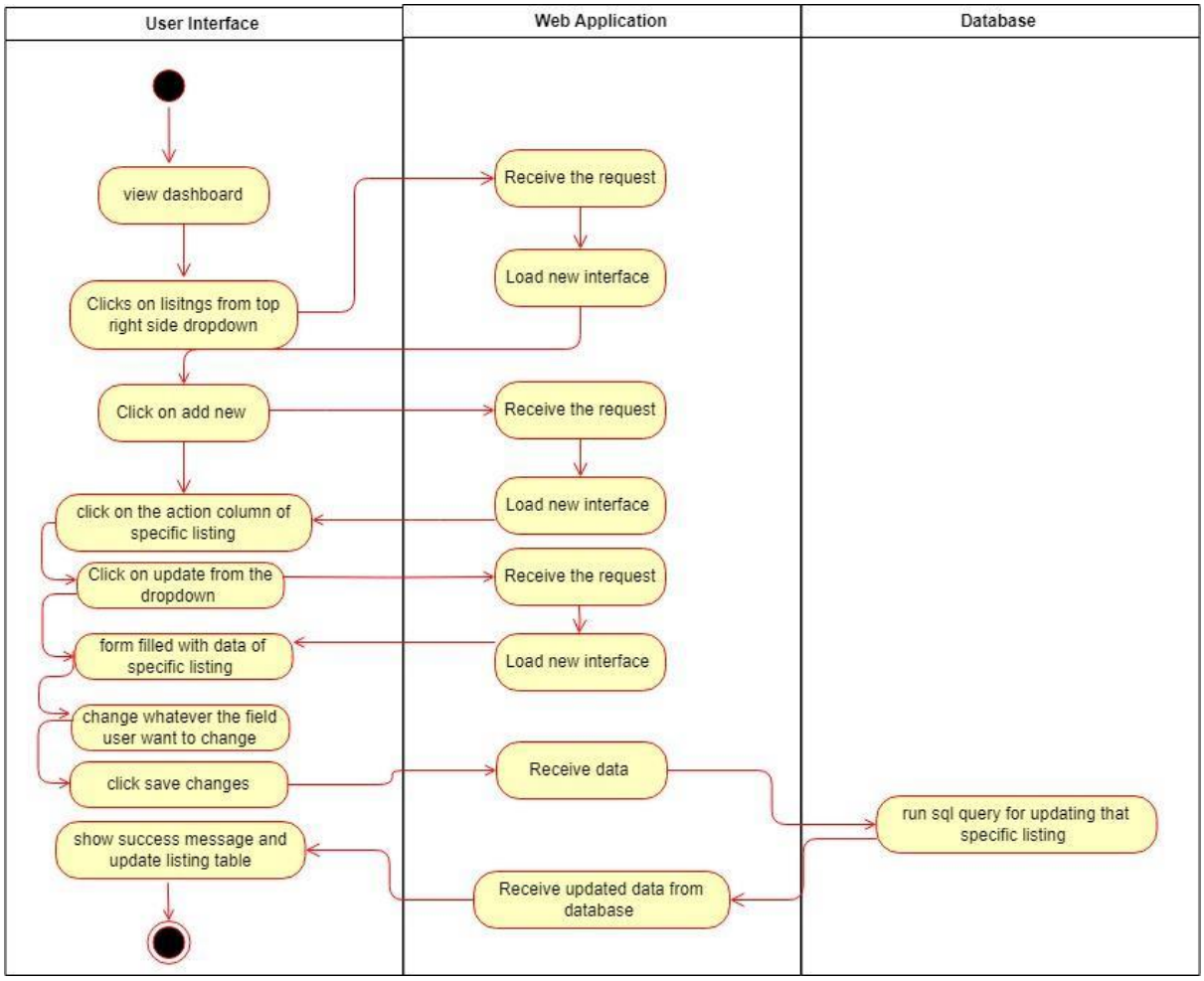


Figure 5.5 Activity Diagram for updating listings

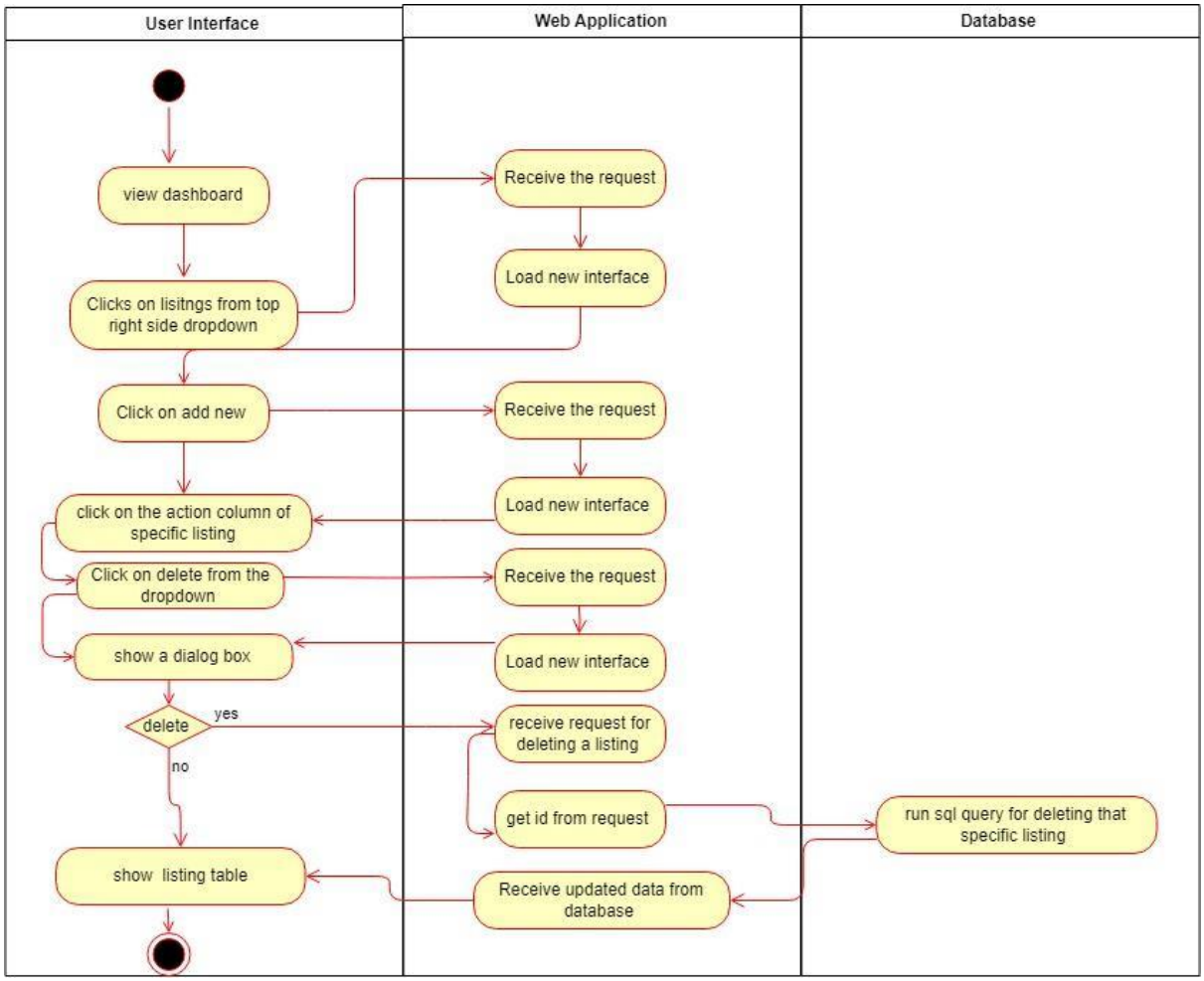


Figure 5.6 Activity Diagram for deleting a listing

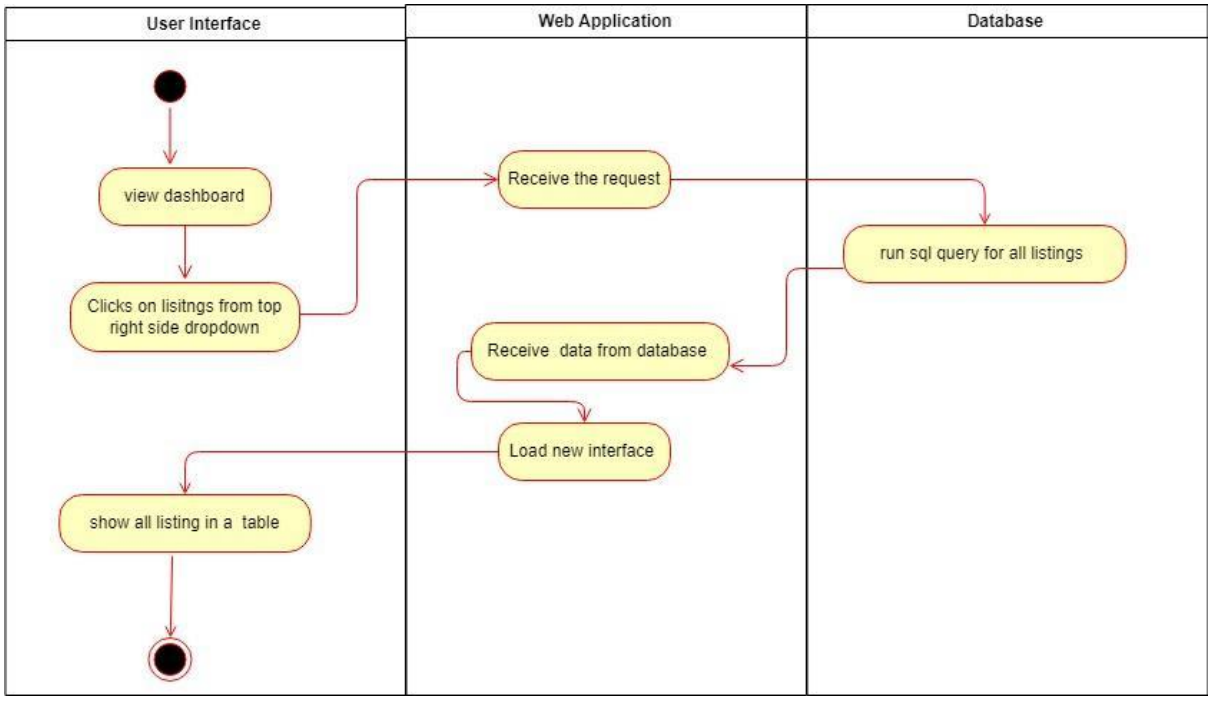


Figure 5.7 Activity Diagram for all listings

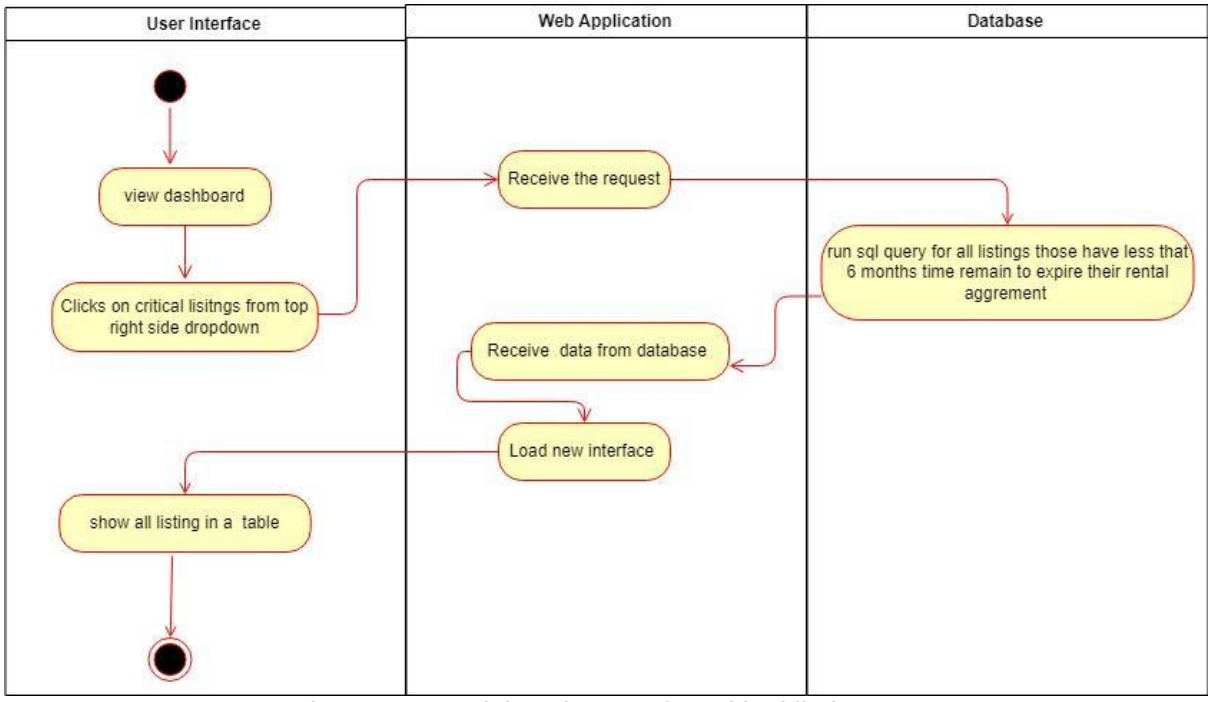


Figure 5.8 Activity Diagram for critical listings

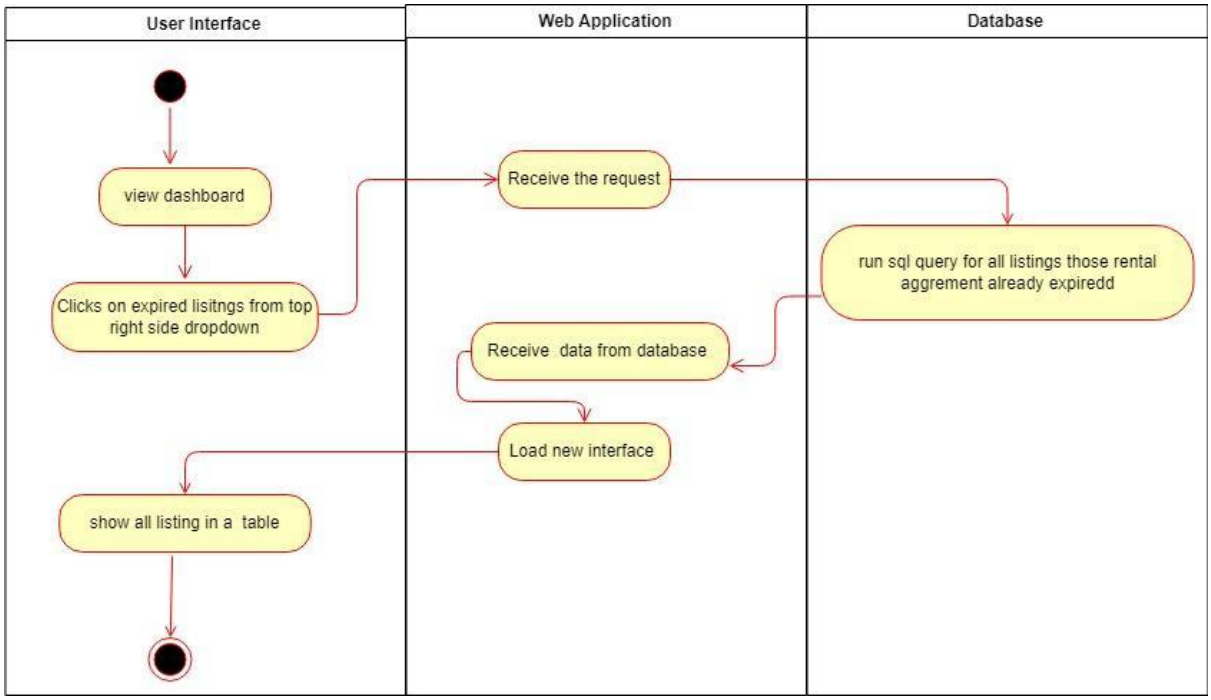


Figure 5.9 Activity Diagram for expired listings

5.4.2 Architecture

Software architecture is a collection of decisions made during the software development process that are impacted by a variety of circumstances. It provides the framework for assuring critical system characteristics such as adaptability, security, performance, traceability, and availability. Furthermore, it serves as a channel of communication among many parties. The software architecture is at the heart of every system, and changing it after it has been established may be costly and time-consuming. As a result, each option has substantial ramifications for completing project needs within specified constraints. For implementation, we used the Model-ViewController (MVC) software architecture paradigm.

The Model-View-Controller (MVC) pattern is a popular software architecture idea that is used in a variety of programming languages and frameworks. It is extensively used in programming languages like as Python, PHP, Ruby, JavaScript, and others, especially in the creation of web and mobile apps. Because the user's input is already presented in an academic fashion, no rewriting is required. The MVC architectural pattern, in essence, separates a program into three distinct and well-defined components:

1. Model.
2. The view.
3. The controller.

Model:
- The Model houses the business rules as well as the application data.
- It is concerned with data structures and database interactions.
- The Model manipulates and computes data.
- Any data updates are communicated to the Views.
- In web applications, the Model communicates with the database to get and save data.

View:
- The View is in charge of displaying data as well as handling user interface interactions.
- It displays user information in a visually pleasing manner.
- User input is sent to the Controller for processing.
- It can monitor Model changes and update its display accordingly.
- HTML templates are frequently utilized by the View in web applications to produce web pages.

Controller:
- It serves as a go-between for the Model and the View.
- It responds to user input.
- The Controller can use Model methods to do specified tasks.
- It updates the View to reflect any changes in the Model.
- The Controller is used in web applications to send HTTP requests, interact with the Model, and pick the appropriate View.

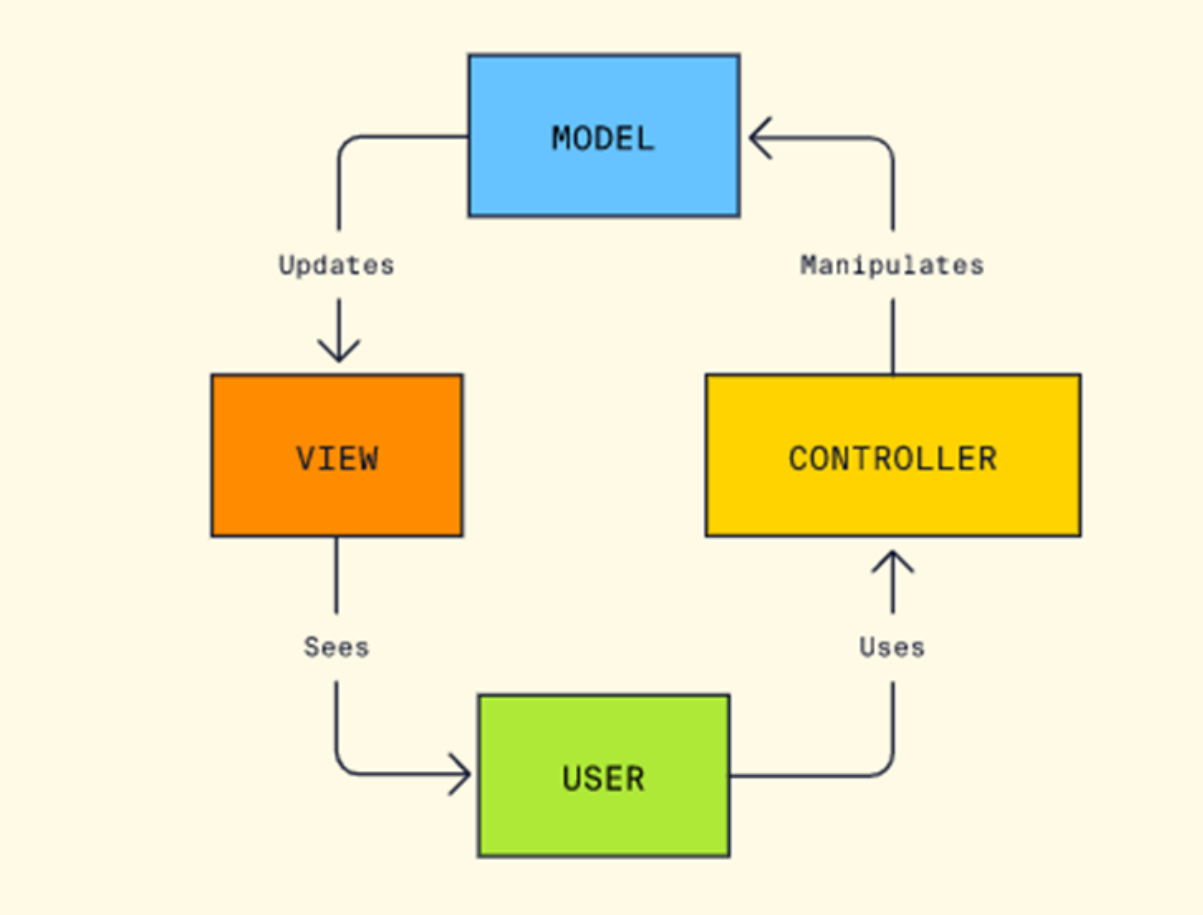


Figure 5.10 Model View Controller Architecture

5.5 Implementation

The technical aspects of designing and deploying the system are highlighted in the HRMS implementation section. This stage involves converting the requirements and design into a fully functional software solution. Here are two examples of screenshots, one for programming and one for database design.

5.5.1 Frontend

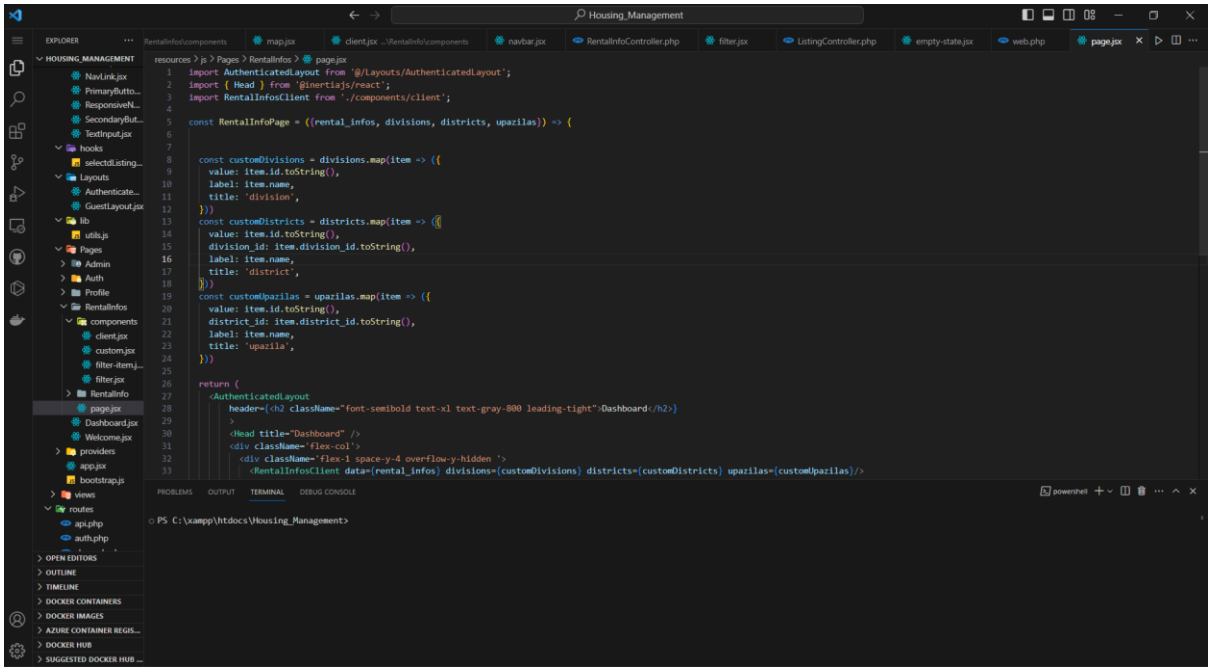


Figure 5.11 Frontend Screenshots

5.5.2 BackEnd

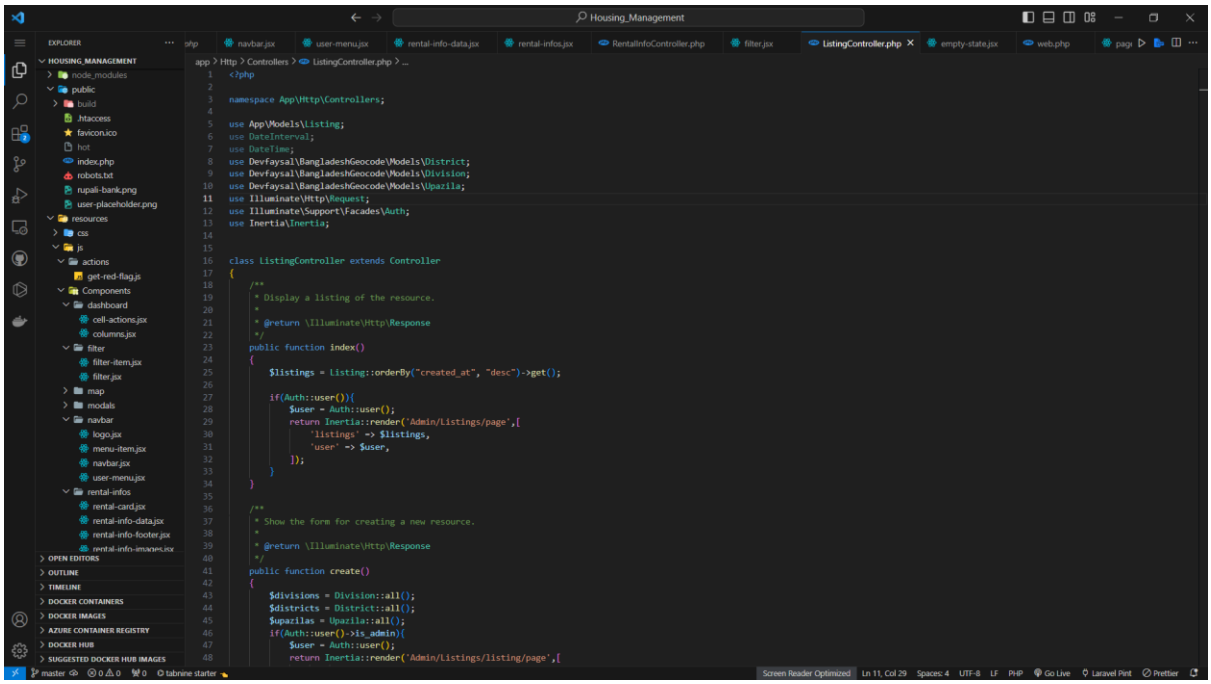


Figure 5.12 Backend Screenshots

5.5.3 Routing

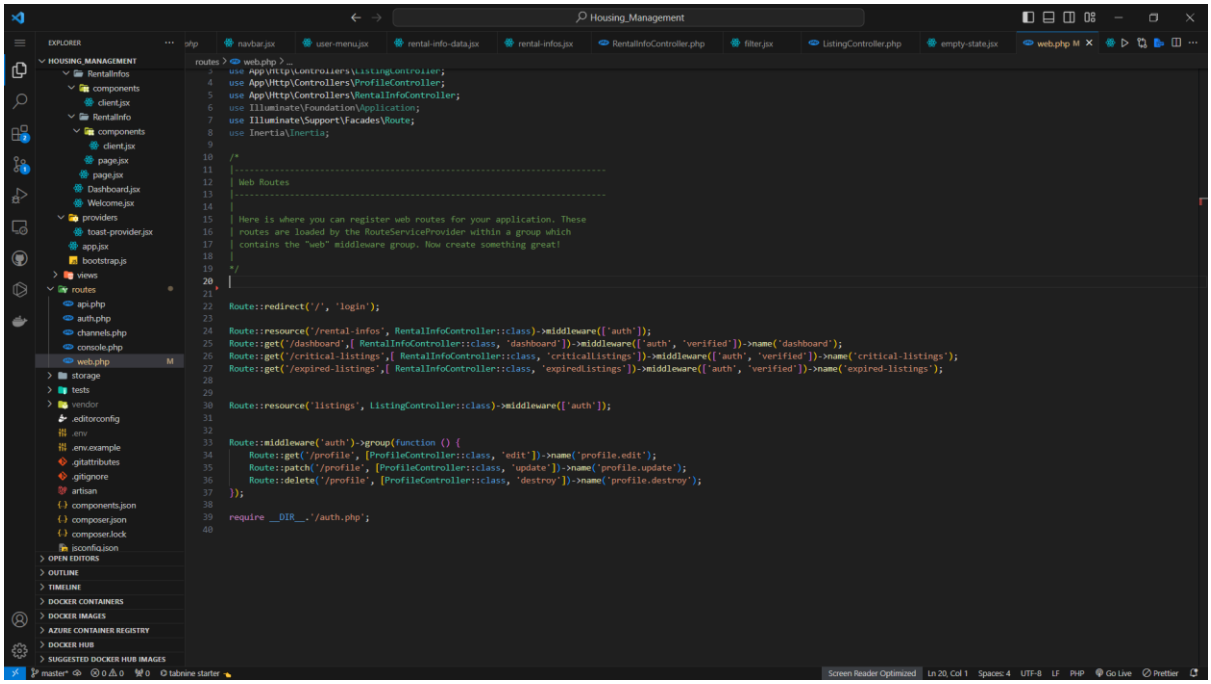


Figure 5.13 Routing Screenshots

5.5.4 Database

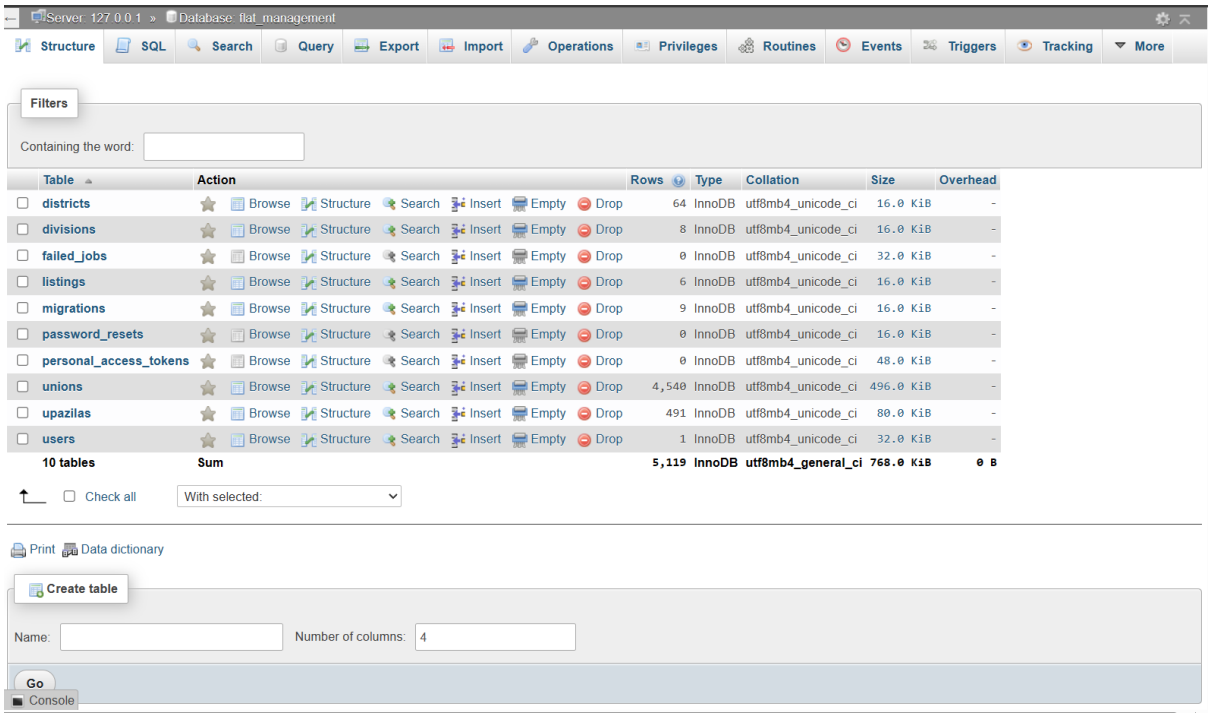


Figure 5.14 Database Screenshots

5.6 Testing

The testing phase of a project is critical for ensuring that the system works properly, fits the requirements, and produces a great experience for users. We employ a variety of testing methodologies to comprehensively examine functionality, quality of service, safety, and usability.

5.6.1 Input

Test Case Id	Test Scenario	Test Input
TC001	User Sign in	User credentials (email, password)
TC002	Add Listing	Listing details(office name, previous name, address, division, district, upazila, rent type, building type, head office approval, rent/sqft, start date, end date, construction date, floor space, floor position, location)

Table 5.2 Input table with Field

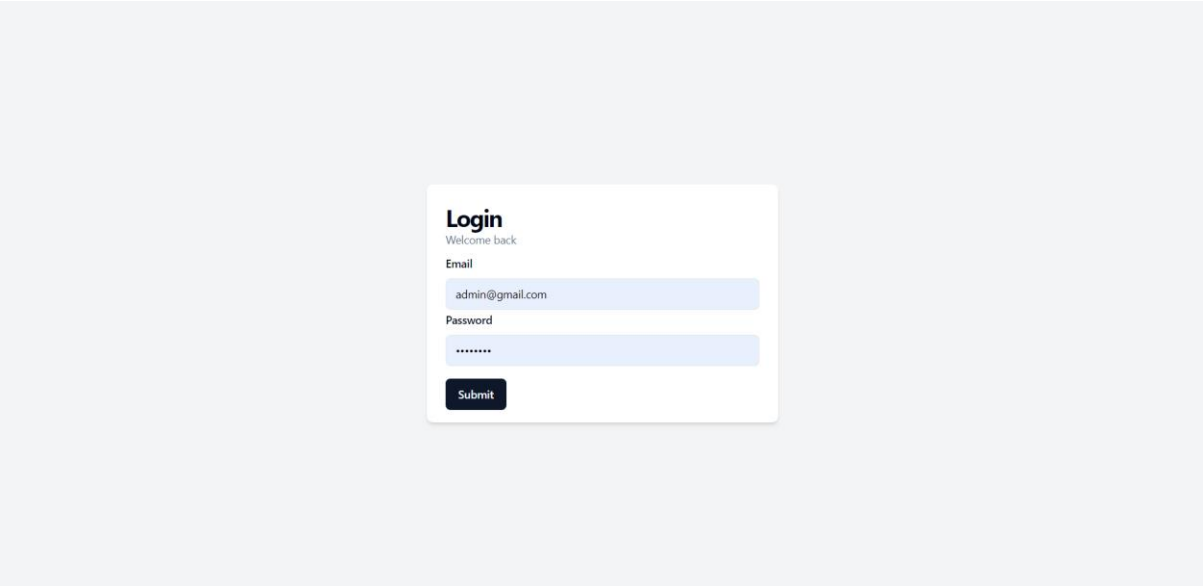


Figure 5.15 Input test case TC001 Screenshoot

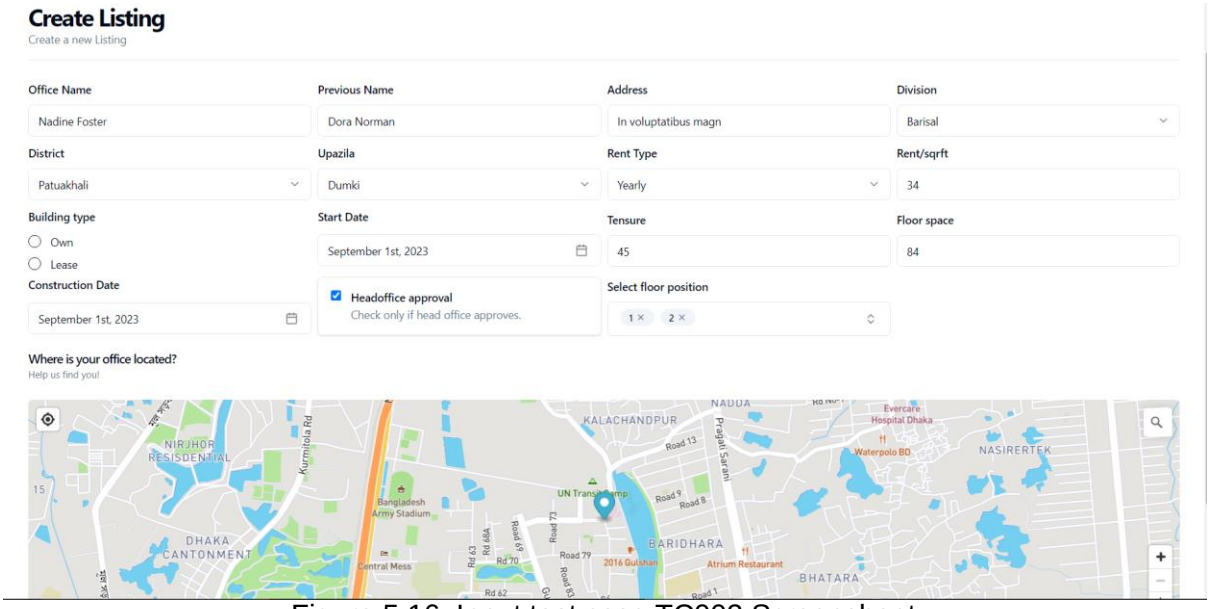


Figure 5.16 Input test case TC002 Screenshoot

5.6.2 Output

Test Case Id	Test Scenario	Test Output
TC001	User Sign in	Success message and redirect to Dashboard.
TC002	Add Listing	Success message and redirect to listings table

Table 5.3 Output table with Field

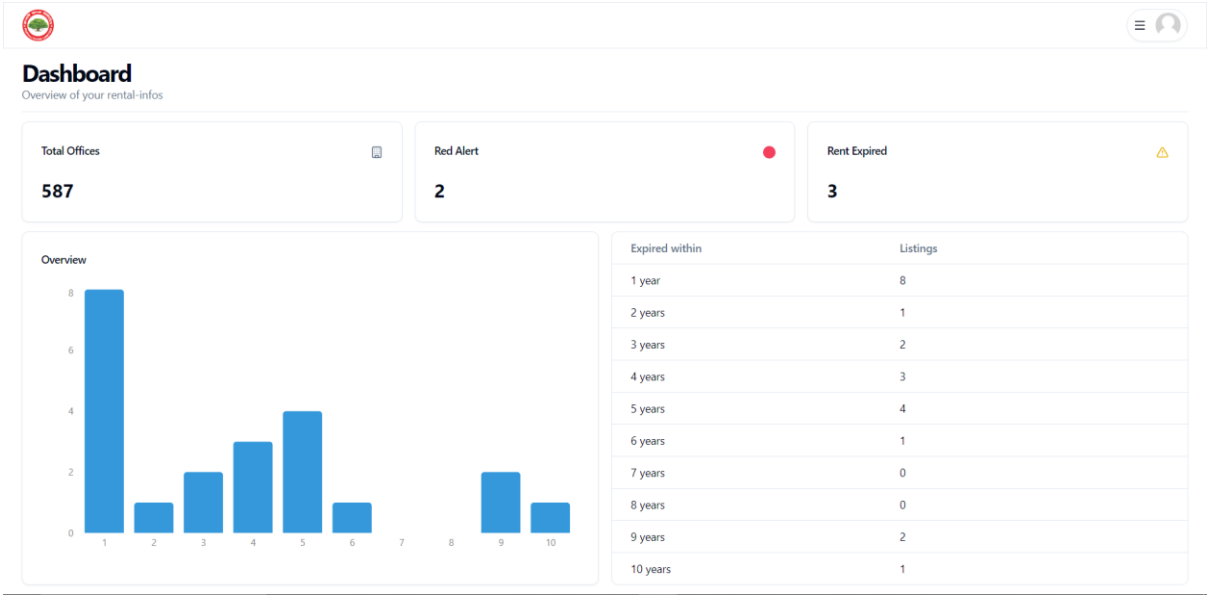


Figure 5.17 Output test case TC001 Screenshoot

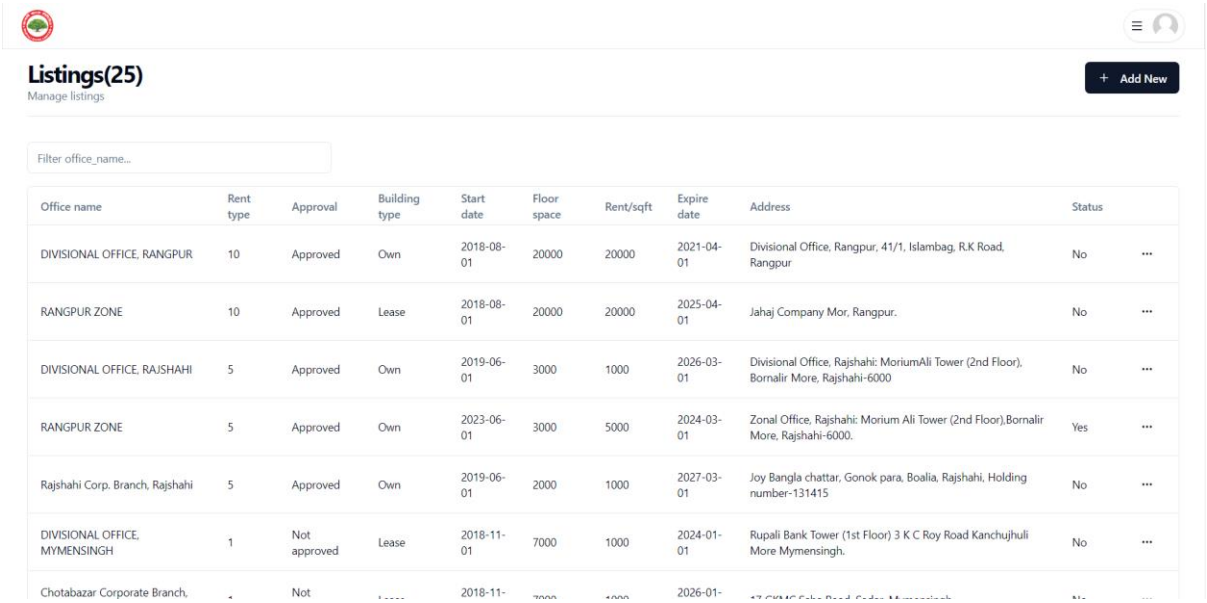


Figure 5.18 Output test case TC001 Screenshoot

5.6.3 Designing Test Cases

Test Case Id	Test Scenario	Test Inputs	Test Output	Pass/Fail
TC001	User Sign in	User Credentials(email, password)	Success Message and redirect to dashboard	pass
TC001	User Sign in	User Credentials(email, password) suppose if the given password or email is incorrect.	These credentials do not match our records.Stays in the same page	fail
TC002	Add Listing	Listing details(office name, previous name, address, division, district, upazila, rent type, building type, head office approval, rent/sqft, start date, end date, construction date, floor space, floor position, location)	Success message and redirect to listings table	pass
TC002	Add Listing	Listing details(office name, previous name, address, division, district, upazila, rent type, building type, head office approval, rent/sqft, start date, end date, construction date, floor space, floor position, location) suppose you missed any of the field	It wouldn't allow you to submit the form and show which field you didn't submit.	fail

Table 5.4 Designing Test case Input/output Pass/Fail table

5.6.4 Test Results

The output or findings of different testing operations conducted on a website to analyze its functionality, performance, security, and general quality are referred to as test results. Website testing is an important component of the software development life cycle since it ensures that the website satisfies its intended requirements and performs properly. From the test case we can see what is the output of a test when it fails or passes.

Chapter 6

Results & Analysis

In this section, we will examine the project/work findings and assess their implications. The achieved results are based on the project's development and implementation. Let us investigate the results and do a detailed analysis.

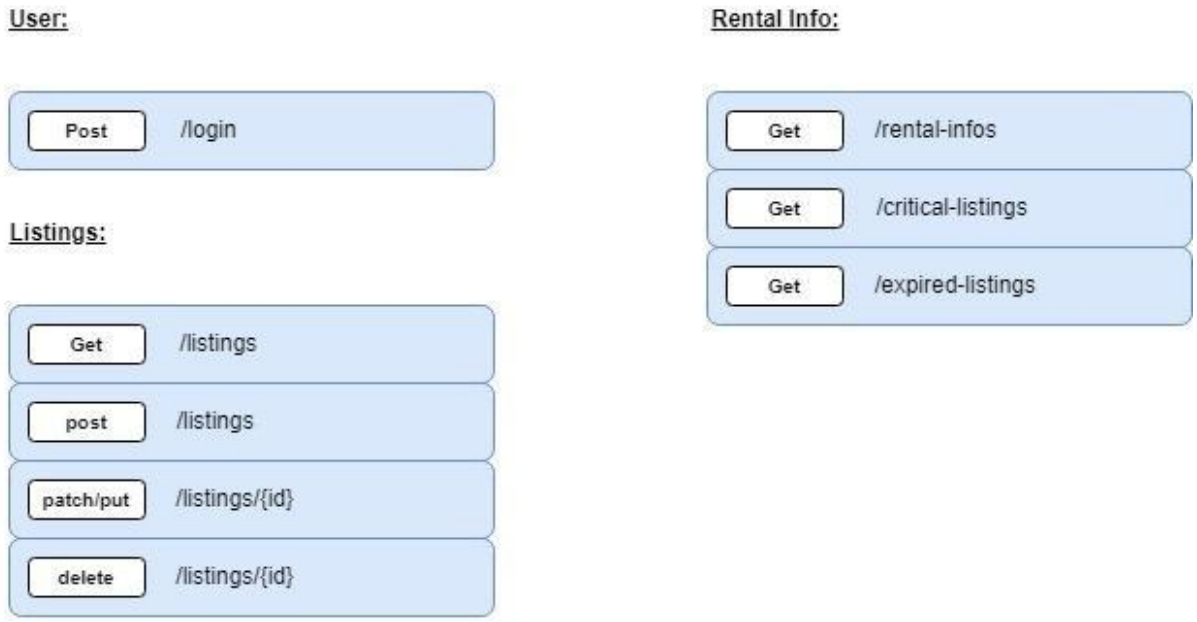


Figure 5.19 API endpoint

Outcome of System Functionality:

- The HRMS (House Resource Management System) was successfully created and installed, offering a digital platform for monitoring rental data.
- The system managed a variety of activities competently, including rent management, alerting for rental agreement expirations, and enabling rent agreement renewals.
- Rent management duties were automated, decreasing manual labor and human participation.

Analysis of Challenges Encountered:

- Throughout the project, several barriers were encountered and successfully overcome.
- The usage of agile development approaches was required to adapt to changing needs and incorporate user feedback.
- Data security and system reliability were carefully considered and handled.

Future Suggestions:

-After examining the results and assessments, we discovered possible areas for system enhancements and extensions.
- We may look at adding other features such as report generation and analytics and interaction with online payment systems.

The successful development and implementation of the HRMS emphasizes the relevance of digital technologies in upgrading Rupali Bank's office rent management.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

Strategic initiatives have been developed by the corporation to ensure the project's long-term viability and continuing expansion. The first aim is to maintain excellent rental management, with plans to expand the application's capabilities in the future. The project's sustainability is divided into two key categories:

Financial viability: It is critical to successfully manage the project's development expenses, which include license fees, database storage costs, and continuous maintenance, to guarantee that the project remains profitable.

Organizational sustainability: The organization expects to continue operations once the application is launched by forming a new staff. The strategy calls for upgrades and the addition of new features to satisfy changing consumer needs and increase the company's product line.

7.2 Social and Environmental Effects and Analysis

The application is created with an eco-friendly approach in mind, eliminating the need for paper to store rental information by digitizing the data. The program runs entirely on the internet and is powered by energy and technological gadgets.

7.3 Addressing Ethics and Ethical Issues

The initiative prioritizes ethics and security. It has been rigorously built to maintain a high degree of security, guaranteeing that no personal information is at risk of being revealed from the database. Access to the program is restricted to authorized users who must go through authentication steps, ensuring that only trustworthy personnel have access to the software. Several ethical considerations have been thoroughly explored and handled in the research, including:

Fraud and Identity Theft: The database is inaccessible to third-party software, guaranteeing that no other programs may access it. This system's data collecting excludes financial information, lowering the danger of fraud and identity theft.

Data Security: Robust security mechanisms, such as login and password protection, carefully restrict and limit server and database access, effectively preventing unwanted data access.

Our initiative prioritizes ethical principles, security, and environmental stewardship. Our goal is to foster trust among users and stakeholders while also ensuring the system's integrity and long-term viability.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

During my internship, I encountered a number of challenges and scenarios that provided valuable learning opportunities. Here are some of the major challenges I encountered:

Technical Difficulties: I frequently faced complicated technical challenges that necessitated in-depth problem resolution. This entailed debugging code, addressing software issues, and adjusting to new technologies and tools. Overcoming these difficulties considerably boosted my technical skills.

Communication Difficulties: Effective communication with team members and managers was critical, although it might be difficult at times. Miscommunication may occur due to language problems, miscommunication, or variations in communication styles. Learning to deal with these difficulties increased my interpersonal abilities.

Project Deadlines: Meeting project deadlines was a regular difficulty. Balancing several duties and ensuring that all work was done on time necessitated effective time management and organizational abilities. It was a chance for me to improve my abilities to operate under pressure.

Adapting to Company Culture: Each organization has its own culture and work environment. Getting used to a new company's culture, policies, and procedures takes some time. It was critical to comprehend the organization's expectations and ideals.

Comments and Improvement: Receiving positive and constructive comments on my work was an important part of the learning process. While favorable feedback was motivating, I needed to learn to accept criticism with grace and utilize it as a tool for personal and professional development.

8.2 Solution of those Problems

Technical problems, communication concerns, tight deadlines, and adjusting to business culture are all regular hurdles for interns. Seeking help, strengthening communication skills, time management, seeking direction, adjusting to the company culture, embracing feedback, establishing remote work habits, and remaining flexible in the face of changing project scopes are some solutions. Surmounting these obstacles promotes personal and professional development and leads to a good internship experience.

Chapter 9

Future Work & Conclusion

9.1 Future Works

The growth of House Rent Management System software is a continuous process, with plans for more upgrades and extensions. Among the possible future initiatives are:

Supply various reports : Supple of various reports toward Finance Ministry, Bangladesh Bank and Internal Authority. The development team might introduce this feature to generate reports in future.

Online payment process : Online based payment system.

HRMS is dedicated to staying current and addressing the changing needs of its customers by constantly improving and expanding the application's capabilities. The key goals are to improve user usability, introduce new services, and provide a seamless and effective digital platform.

9.2 Conclusion

During my internship at Rupali Bank Limited, I got the opportunity to contribute to the development of an application aimed at easing their office’s rent management. This job provided me with invaluable insights and hands-on experience working with technologies like Laravel and React. The system on which I worked was intended to improve efficiency, lower costs, and decrease complications. My internship at Rupali Bank Limited taught me a lot. I learned how web development works in practice and what users expect from software. I learnt how to cope with various situations and figure out the best solutions to them. This internship improved my confidence, problem-solving skills, and awareness of what it takes to be a competent web developer. It also prepared me for the labor and demands of the job. I'm excited to continue studying and use what I've learned to create useful software.

Bibliography

Exchange, S. (n.d.). pp. https://softwareengineering.stackexchange.com/questions/444842/what-is-the-difference-between-these-two-mvc-diagrams.

negociosit. (n.d.). https://negociosit.com/agile-mentoring.html.



An Undergraduate Internship/Project on Topic

House Rent Management System

By

Imran Syam

Student ID: **1921731**

Summer, 2023

Consent from Supervisor

The student modified the internship final report as per the recommendation made by his/her academic supervisor and/panel members during and/or before final viva, and the department can use this version for archiving as well as the OBE course material for CSE499.

This internship report is checked with Turnitin and/or Ithenticate plagiarism checker, and the score is:

Turnitin Score (%) : 15%

Ithenticate Score (%) : —

Ashraful Islam 16.10.23

(Signature of the Supervisor)

Dr. Ashraful Islam, PHD

Assistant Professor

Department of Computer Science & Engineering

Independent University, Bangladesh