

Design and Implementation of To-Let Management System

by

Md. Rakib Chowdhury

ID: CSE1901016168

Md. Hasan Ali

ID: CSE1902017085

Md. Taiabur Rahman

ID: CSE1902017075

Md. Al-Amin Mollah

ID: CSE1702011015

Sabbir Hossain

ID: CSE1902017083

Supervised by

Arifur Rahaman

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in
Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

September 2023

Design and Implementation of To-Let Management System

by

Md. Rakib Chowdhury

ID: CSE1901016168

Md. Hasan Ali

ID: CSE1902017085

Md. Taiabur Rahman

ID: CSE1902017075

Md. Al-Amin Mollah

ID: CSE1702011015

Sabbir Hossain

ID: CSE1902017083

Supervised by

Arifur Rahaman

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in
Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

September 2023

APPROVAL

The project titled “**Design and Implementation of To-Let Management System**” submitted by Md. Rakib Chowdhury (CSE1901016168), Md. Hasan Ali (CSE1902017085), Md. Taiabur Rahman (CSE1902017075), Md Al-Amin Mollah (CSE1702011015) and Sabbir Hossain (CSE1902017083) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Arifur Rahaman

Assistant Professor

Department of Computer Science and Engineering

Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)

Department of Computer Science and Engineering

Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)

Department of Computer Science and Engineering

Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)

Department of Computer Science and Engineering

Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Arifur Rahaman, Assistant Professor**, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

(Arifur Rahaman)
Supervisor

Md. Rakib Chowdhury
ID: CSE1901016168

Md. Hasan Ali
ID: CSE1902017085

Md. Taiabur Rahman
ID: CSE1902017075

Md. Al-Amin Mollah
ID: CSE1702011015

Sabbir Hossain
ID: CSE1902017083

ABSTRACT

The To-Let project aims to develop a user-friendly and efficient online platform for property seekers to find rental accommodations and for property owners to list their available properties. The platform will serve as a bridge connecting tenants and landlords, facilitating the rental process and streamlining property management. The website will offer a robust and intuitive interface to allow users to search, filter, and view available rental properties based on their preferences such as location, property type, price range, and amenities. Landlords, on the other hand, will be able to create accounts, list their properties, and manage their listings through a dedicated dashboard. To ensure the credibility of the listings and build trust among users, the To-Let website will implement a verification process for both tenants and landlords. Tenants will need to provide necessary identification and rental history, while landlords will have to verify property ownership and other relevant details. In addition to the property listing and search features, the website will also incorporate a messaging system that enables direct communication between tenants and landlords. This will simplify the inquiry and negotiation process, enhancing the overall user experience. In conclusion, the To-Let website project aims to create a reliable, user-friendly, and secure online platform that simplifies the process of finding rental accommodations for tenants and property management for landlords. By offering a comprehensive range of features and prioritizing user experience, the website seeks to become a go-to destination for individuals seeking rental properties and landlords looking to connect with potential tenants.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Arifur Rahaman**, Assistant Professor, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

CPU	Central Processing Unit
HTML	Hyper Text Markup Language
PC	Personal Computer
RAM	Random Access Memory
ROM	Read Only Memory
SDLC	Software Development Life Cycle
UI	User Interface

TABLE OF CONTENTS

Title	Page No.
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
LIST OF ABBREVIATION	vi
CHAPTESR 1	1 - 2
INTRODUCTION TO TO-LET MANAGEMENT SYSTEM	
1.1 Introduction	1
1.2 Objectives	1
1.3 Encountered Difficulties.....	1
1.4 Goal of this project	1
1.5 Background	1
1.5.1 How To-Let Services Work.....	1
1.5.2 Benefits of Online To-Let Services.....	2
CHAPTER 2	3 - 9
SURVEY OF TECHNOLOGIES, WORKING PROCEDURE	
2.1 About of ReactJS	3
2.1.1 Concepts of ReactJS.....	3
2.1.2 Working of ReactJS.....	4
2.2 Introduction to Tailwind CSS.....	4
2.3 Introduction to Firebase.....	4 - 5
2.4 Introduction to MongoDB	5 - 6
2.5 Connecting react Application to MongoDB.....	6
2.6 Introduction Node.js.....	6
2.7 Introduction Express.js.....	6
2.8 Testing Method.....	6 - 7
2.8.1 Testing Method we have used (Unit Testing).....	7
2.9 Working Procedure.....	7
2.10 Description of Procedure.....	8
2.10.1 System Design and Development.....	8
2.10.2 Implementation.....	8
2.10.3 Testing.....	8
2.10.4 Maintenance.....	8 - 9
CHAPTER 3	10 - 11
SYSTEM REQUIREMENTS AND ANALYSIS	
3.1 System requirements.....	10
3.2 Hardware and Software Requirement.....	10
3.2.1 Hardware	10
3.2.2 Software	10

3.3	Functional Requirements.....	10 - 11
3.4	Non-Functional Requirements.....	11
CHAPTER 4		12 - 14
SYSTEM DESIGN		
4.1	Flow-chart.....	12
4.1.1	Data Flow Diagram (level 0).....	12 - 13
4.1.2	Data Flow Diagram (level 1).....	13
4.1.3	Use Case Diagram.....	14
CHAPTER 5		15 - 20
IMPLEMENTATION AND OUTCOME		
5.1	User Section.....	15
5.1.1	Registration Module.....	15
5.1.2	Login Module.....	15 - 16
5.1.3	User Dashboard.....	16
5.1.4	Find Home/Seat/Apartment.....	16 - 17
5.1.5	Book Home/Seat/Apartment.....	17
5.1.6	Contact and Feedback.....	17 - 18
5.2	Seller Section.....	18
5.2.1	Seller Request.....	18
5.2.2	Add Home/Seat/Apartment.....	18 - 19
5.2.3	Booking Information.....	19
5.3	Admin Section.....	19
5.3.1	Admin Dashboard.....	19 - 20
5.3.2	Approve Seller Request.....	20
CHAPTER 6		21 - 21
CONCLUSION AND FUTURE WORKS		
6.1	Conclusion.....	21
6.2	Limitations and Future Work.....	
6.2.1	Limitation.....	21
6.2.2	Future Work.....	21
REFERENCES		22

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page No.</u>
Fig.2.1	SDLC model	7
Fig.4.1	Flow Chart	12
Fig.4.2	Data flow diagram (level 0)	13
Fig.4.3	Data flow diagram (level 1)	13
Fig.4.4	Use case diagram	14
Fig.5.1	User and Seller Registration	15
Fig.5.2	User and Seller login	16
Fig.5.3	User dashboard	16
Fig.5.4	Find room/seat/home	17
Fig.5.5	Booking room/seat/home	17
Fig.5.6	Contact and Feedback	18
Fig.5.7	Seller request	18
Fig.5.8	Add home/seat/apartment	19
Fig.5.9	Booking Information	19
Fig.5.10	Admin dashboard	20
Fig5.11	Seller approve dashboard	20

CHAPTER 1

INTRODUCTION TO TO-LET MANAGEMENT SYSTEM

1.1 Introduction

A large number of people rent houses in our country. House rent is required especially in Dhaka city. Finding a home is very time-consuming and painful. There are many complications and problems to face while looking for a house. If these problems can be solved, their life will be much easier. This is where a "Bari Bhara" website can play a vital role and make their daily work easier.

1.2 Objectives

- To transform the manual process of renting a house to an online and computerized system.
- Making it easier for customers to find a home.
- To ease customer's task whenever they need to rent a house.
- Those who want to build a house can easily afford it.
- To validate the house rental system using user feedback and testimonies.

1.3 Encountered Difficulties

We have created this website keeping in mind the various inconveniences like, no need to talk about house searching, no wasting time and it is very boring to search for a house elsewhere. So, we created this website by thinking about all these problems. Users use it very quickly can find nests and they are very conscientious in using it.

1.4 Goal of this Project

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies and their customers of which the Real Estate/House Rentals industry is not left out [1]. A tool through which customers can book available Rooms/House/Apartment online prior to their date of using the house instead of walking around and asking for a vacant house. Project goal is saving time and simplifying their regular life.

1.5 Background

1.5.1 How To-Let Services Work

A House rental is a house that can be used temporarily for a period of time with a fee. Renting a house assists people to live in a comfortable house when they do not have access to build their own personal homes/houses or. The individual who wants to rent a house/room/apartment/home must first contact the House rental company for the desired House/room/apartment. This can be done online. At this point, this person must supply some information such as dates of rental, and

type of house. After these details are worked out, the individual renting the house must present a valid Identification Card [1].

1.5.2 Benefits of Online To-Let Services

- This online house rental solution is fully functional and flexible.
- It is very easy to use.
- It saves a lot of time, money and labor.
- It increases the efficiency of the management at offering quality services to the customers [1].
- It provides custom features development and support with the software.

CHAPTER 2

SURVEY OF TECHNOLOGIES, WORKING PROCEDURE

2.1 About of ReactJS

At its core, React focuses on enabling developers to build complex UIs by breaking them down into smaller, reusable components. This modular approach simplifies development, maintenance, and debugging, as each component can be developed independently and then combined to create a cohesive user interface.

2.1.1 Concepts of ReactJS

- **Component-Based Architecture:** React applications are structured as a tree of components. A component encapsulates a portion of the UI and its behavior, making it easier to manage and maintain large and complex applications.
- **Virtual DOM:** React introduces a virtual representation of the actual DOM (Document Object Model) called the Virtual DOM. When there's a change in the data or state of a component, React calculates the minimal set of changes needed to update the Virtual DOM. This allows for efficient updates to the actual DOM, minimizing performance bottlenecks.
- **Declarative Syntax:** React uses a declarative approach to define the UI. Developers specify what the UI should look like in response to changes in data or state, and React takes care of updating the actual DOM to match that description. This leads to more predictable code and easier debugging.
- **Unidirectional Data Flow:** React enforces a one-way data flow. Data is passed down from parent components to child components, preventing unexpected changes and making it easier to track how data is used throughout the application.
- **JSX (JavaScript XML):** JSX is a syntax extension that allows developers to write UI components using a syntax like HTML within JavaScript code. This makes the component structure more intuitive and readable.
- **State Management:** React allows components to have their own local state, which represents data that can change over time. For more complex applications, external state management libraries like Redux or MobX can be integrated to handle the global state of the application.
- **Reusable Components:** React's component-based architecture promotes reusability. Developers can create generic components that can be used across different parts of the application, improving maintainability and reducing redundancy.
- **Community and Ecosystem:** React has a vast and active community, which has led to the creation of numerous libraries, tools, and resources that enhance the development experience. This includes tools like React Router for handling navigation, and various UI component libraries like Material-UI and Ant Design.

2.1.2 Working of ReactJS

ReactJS works by breaking down the user interface into components, creating a virtual representation of the DOM (Virtual DOM), efficiently updating the actual DOM based on changes, and enforcing a unidirectional flow of data. This combination of concepts results in a highly efficient, modular, and predictable approach to building interactive and dynamic user interfaces for web applications.

2.2 Introduction to Tailwind CSS

Tailwind CSS is a popular utility-first CSS framework that simplifies and streamlines the process of designing and styling websites and web applications. It provides a set of pre-designed CSS classes that you can apply directly to your HTML elements, making it easier to create responsive and visually appealing user interfaces.

- **Utility-First Approach:** Tailwind CSS follows a utility-first approach, where you build your UI by applying small, single-purpose utility classes directly to your HTML elements. These classes control various aspects of styling, such as margins, padding, typography, colors, and more.
- **Responsive Design:** The framework includes responsive design classes that allow you to easily create layouts that adapt to different screen sizes and devices. This is crucial for ensuring your web content looks good on a wide range of devices, from mobile phones to large desktop screens.
- **Customization:** Tailwind CSS is highly customizable. You can configure various aspects of the framework, including colors, spacing, typography, and more, to match your project's specific design requirements. This is done through configuration files.
- **Components and Plugins:** While Tailwind CSS primarily provides utility classes, it also has a collection of pre-built components and plugins that can speed up development. These components cover common UI elements such as navigation bars, cards, modals, and more.
- **Optimized for Production:** Tailwind CSS provides tools to help you purge (remove) any unused CSS classes, which can significantly reduce the size of your CSS files for production environments. This results in faster loading times and better performance.

2.3 Introduction to Firebase

Firebase Authentication is a service provided by Google's Firebase platform that enables you to easily add authentication and identity management to your web and mobile applications. It provides a set of tools and libraries that make it straightforward to implement various authentication methods and features without the need to build everything from scratch.

- **User Authentication:** Firebase Authentication supports various authentication methods, including email/password, Google, Facebook, Twitter, GitHub, phone number, and more. This means you can allow users to sign in using their preferred authentication provider.

- **User Management:** Firebase Authentication provides features to manage user accounts, such as creating, updating, and deleting accounts. You can also associate additional information with user profiles.
- **Security:** Firebase Authentication handles security concerns for you, including secure password storage, Auth-based authentication with popular providers, and token-based verification.
- **Authentication State:** The Firebase Authentication SDK helps manage the authentication state of users in your app. This allows you to show different content or features to authenticated and unauthenticated users.
- **Authorization:** While Firebase Authentication focuses on user identity, Firebase also offers tools like Firebase Realtime Database and Firestore to manage authorization and user-specific data.
- **Session Management:** Firebase Authentication handles user sessions and token management, ensuring users remain authenticated across different devices and app sessions.

2.4 Introduction to MongoDB

MongoDB is a widely used open-source, NoSQL database management system that offers a flexible and scalable approach to storing and managing data. Unlike traditional relational databases, MongoDB follows a document-oriented model, where data is stored in flexible, JSON-like documents instead of tables with rows and columns.

MongoDB is commonly used in modern web applications, content management systems, e-commerce platforms, analytics applications, and other projects that require flexible data storage and retrieval capabilities. Its document-oriented approach, scalability options, and diverse querying capabilities make it suitable for a variety of use cases.

- **Document-Oriented Data Model:** MongoDB stores data in documents, which are composed of field-value pairs similar to JSON objects. These documents are organized into collections, which are analogous to tables in a relational database.
- **Flexible Schema:** Unlike traditional relational databases that enforce a rigid schema, MongoDB offers a flexible schema model. Documents within a collection can have varying fields, and each document can evolve independently as your application's requirements change.
- **Scalability:** MongoDB is designed to scale horizontally, which means you can distribute data across multiple servers to handle large amounts of data and high traffic loads. This is particularly beneficial for applications that need to handle massive data growth.
- **Querying:** MongoDB provides powerful querying capabilities using a query language that is similar to JSON syntax. You can perform queries to retrieve specific documents or subsets of data based on various conditions.
- **Indexes:** Indexes improve query performance by allowing MongoDB to quickly locate the relevant documents. You can create indexes on specific fields to speed up common queries.

- **Aggregation Framework:** MongoDB offers a versatile aggregation framework for performing complex data transformations and analytics tasks, such as grouping, filtering, sorting, and more.
- **Replication:** MongoDB supports data replication, allowing you to create multiple copies of your data across different servers. This enhances data availability and provides fault tolerance.
- **JSON-Like Documents:** MongoDB documents are stored in BSON (Binary JSON) format, which is efficient for storage and transmission while still maintaining the flexibility of JSON-like structures.
- **Geospatial Data Support:** MongoDB includes features for working with geospatial data, allowing you to store and query location-based information.

2.5 Connecting React Application to MongoDB

Connecting a React application to MongoDB involves several steps to establish a communication link between your front-end application and the MongoDB database on the back end. Create a backend server using a technology like Node.js and a framework like Express.js. Set up a MongoDB database using a cloud service or host it locally. Install necessary packages, including MongoDB drivers, in your Node.js project. Create API routes on your backend server to handle CRUD (Create, Read, Update, and Delete) operations with the MongoDB database. These routes will handle requests from the React application and interact with the MongoDB database using the MongoDB driver.

2.6 Introduction Node.js

Node.js has revolutionized server-side development by allowing developers to use the same programming language, JavaScript, on both the client and server sides. This alignment simplifies the development process and facilitates seamless communication between the front end and the back end, resulting in faster development cycles and improved overall performance.

2.7 Introduction Express.js

Express.js is widely used in various types of web applications, including APIs, content management systems, e-commerce platforms, and more. Its simplicity, flexibility, and extensive ecosystem make it a preferred choice for many developers working with Node.js.

2.8 Testing Method

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance. In general, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. Software testing identifies bugs and issues in the development process, so they're fixed prior to product launch. This approach

ensures that only quality products are distributed to consumers, which in turn elevates customer satisfaction and trust.

There are many different types of testing

- **Unit testing:** Unit testing is part of the software development process in which small parts of an application, called units, are individually tested for proper operation. Unit testing can be done manually but is often automated in Agile and DevOps projects.
- **Integration/System Testing:** Integration testing is the part in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before functional testing.
- **Functional Testing:** After the integration tests are performed, more complex levels of tests are used. The functional test process in which software is tested to ensure that it conforms to all business requirements and to ensure that it has all the required functionality for the software to be used by the end user without issues.
- **Acceptance Testing:** Acceptance Testing is where a system is tested for acceptability by the end user. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery into production.

2.8.1 Testing Method we have used (Unit Testing)

Unit testing ensures that all code meets quality standards before its deployed. This ensures a reliable engineering environment where quality is paramount. Over the course of the product development life cycle, unit testing saves time and money, more efficiently. Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.

2.9 Working Procedure

Working is mainly the body of Working Procedure, rules, and a particular procedure or set of procedures for achieving proposed model's objectives [4]. We use SDLC model to complete our project. This project work has completed by following some steps.

Diagrammatic representation of these steps of waterfall model is given below:

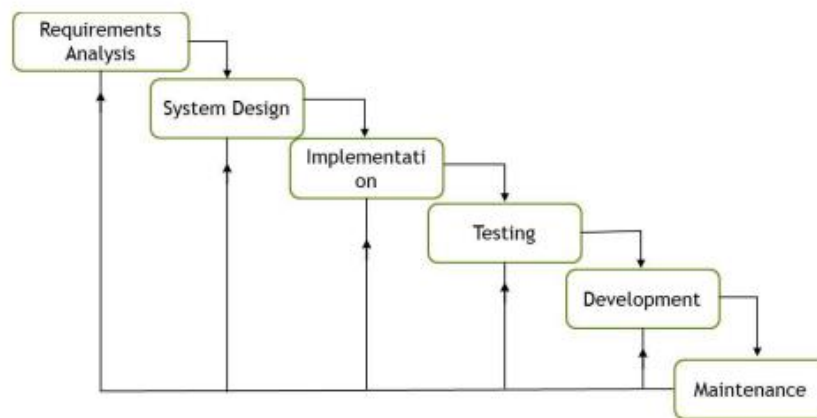


Fig: 2.1: SDLC model

2.10 Description of Procedure

A Description of Procedure is a type of diagram that represents a workflow or process. A Description of Procedure can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The Description of Methodology shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

2.10.1 System Design and Development

System design and development refers to the process of creating and implementing a new computer-based system or improving an existing system to meet specific requirements and solve a particular problem. It involves designing the architecture, components, and functionalities of the system while considering factors like performance, scalability, security, and usability. The process of system design and development typically includes the following steps: System design, Implementation, Testing and quality assurance, Deployment and Maintenance.

2.10.2 Implementation

With regards to our analysis and system design you will implement your code on this stage. You will apply all system logic and service integration to give your To-Let Management System its face.

2.10.3 Testing

All the programs for all possible defects, by running test cases either manually or by automation. The client is involved in the testing phase as well, to ensure all requirements are met.

2.2.5 Maintenance

Installation and maintenance include making the appropriate modification to the product or system or enhancing, changing, or modifying attributes related to performance issues related to the system. Its main role is to improve the performance of the system with the maximum accuracy result of our blink apps output. These progressions raised during the support stage are significantly connected with changes started to be finished by the client or clients after the establishment and testing stage, which includes bugs like imperfections revealed during live employments of the framework or requests raised by the customers. So the customer is furnished with convenient upkeep and backing for the created item. You will really be amazed to know that the effort made in the design and development phase of the product is only 60% effort as compared to the efforts made in the maintenance phase. Maintenance is the generally significant phase of a software life cycle.

There are 3 types of maintenance:

- **Corrective Maintenance:** This sort of upkeep is done to address blunders that were not found during the item advancement stage.
- **Perfection Maintenance:** This kind of support is done to upgrade the functionalities of the framework in view of the client's solicitation.

- **Adaptive Maintenance:** Versatile upkeep is typically needed for porting the product to work in another climate like work on another PC stage or with another working framework.

CHAPTER 3

SYSTEM REQUIREMENTS AND ANALYSIS

3.1 System requirements

System technical requirements is a process that identifies the functionality that is needed by a system to satisfy the user's requirements [2]. System requirements are a broad and narrow subject that could be implemented to many items. Without system requirements, the project will be a failure and the company will fail. If the customer has system requirements that will satisfy their needs, then that company will have a higher percentage of success. Not only will they have success, but they will also have a better chance of saving money and time. Without system requirements, the project will be a failure and the company will fail. If the customer has system requirements that will satisfy their needs, then that company will have a higher percentage of success. Not only will they have success, but they will also have a better chance of saving money and time.

3.2 Hardware and Software Requirement

3.2.1 Hardware:

Processor	Pentium-II or higher
Processor Speed	533 MHZ
Hard Disk Space	20 GB (min.)
Ram Memory	32 MB (64 MB recommended)

3.2.2 Software:

Operating System	Windows 95/98/NT/2000 /10/7/8/11
Database Server	MongoDB
Front-end	ReactJS
Text Editor	Visual Studio Code

3.2 Functional Requirements

Requirement analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product, taking into consideration the possible conflicting requirements of the various users [1]. Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold and the interfaces with the user.

- **Registration:** The system allows new users to register online.
- **Login:** After completing registration process then Login.
- **Booking room/apartment/seat:** Customers should be able to use the system to make bookings online.
- **Seller Request:** Any user can request a seller to be supervised by an administrator. They can add their home if the admin approves them.
- **Feedback to customers:** It should provide a means for customers to leave feedback.

3.3 Non-Functional Requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional Epics, Capabilities, Features, and Stories. They ensure the usability and effectiveness of the entire system [2].

- **Security:** The subsystem should provide a high level of security and integrity of the data held by the system, only users with valid password and username can login to.
- **Performance and Response time:** The system should have high performance rate when executing user's input and should be able to provide feedback or response within a short timespan usually 50 seconds for highly complicated task and 20 to 25 seconds for less complicated task. [1]
- **Error handling:** Validation of user's input is highly essential. Also, the standard time taken to recover from an error should be 15 to 20 seconds.

CHAPTER 4

SYSTEM DESIGN

4.1 Flow-chart

A flowchart is a diagram that shows an overview of a program. Flowcharts normally use standard symbols to represent the different types of instructions. These symbols are used to construct the flowchart and show the step-by-step solution to the problem. Flowcharts are sometimes known as flow diagrams.

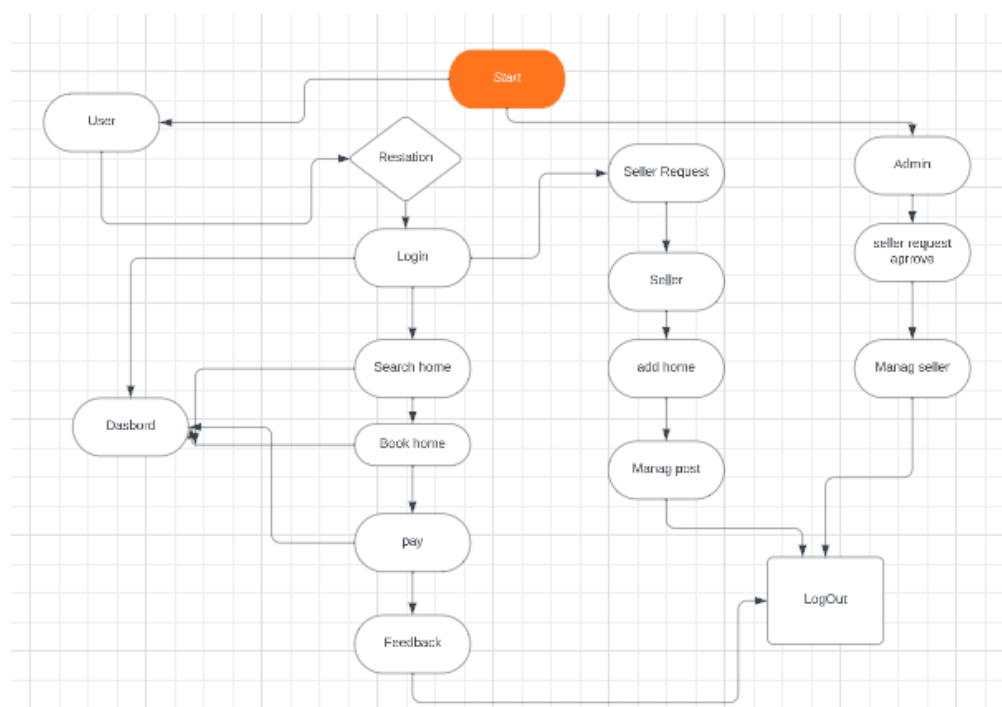


Fig: 4.1 Flow Chart

4.1.1 Data Flow Diagram (level 0)

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination [3].

4.1.3 Use Case Diagram

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented [3].

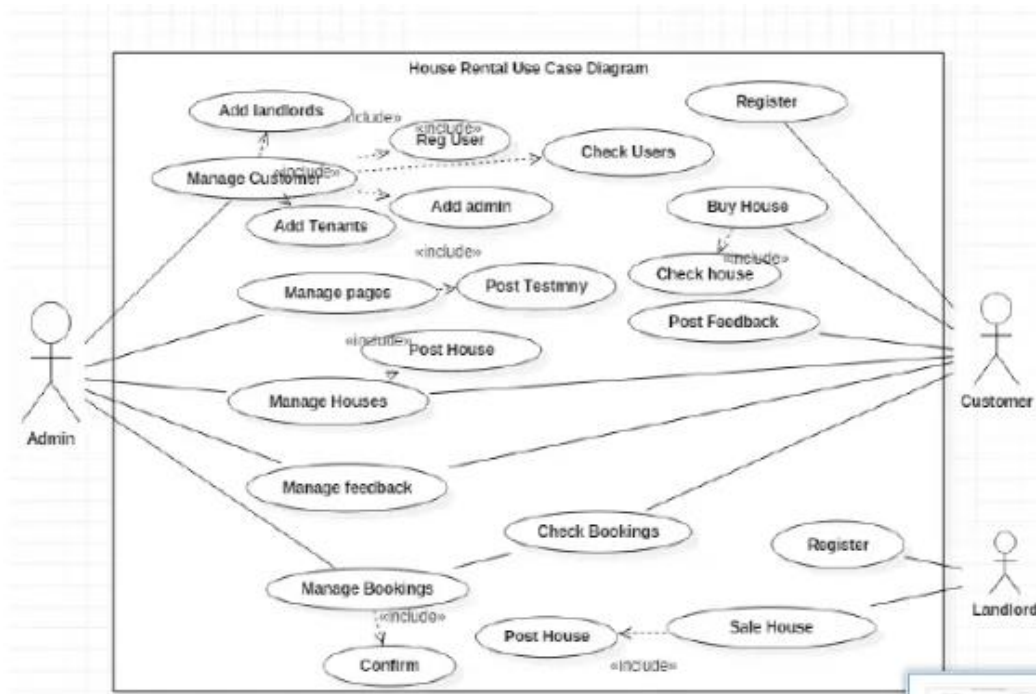


Fig: 4.4 use case diagram

CHAPTER 5

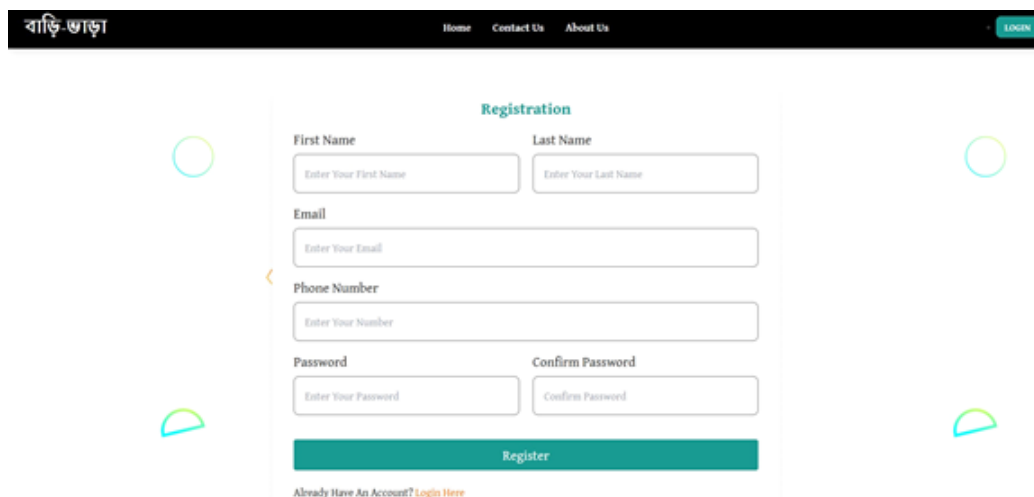
IMPLEMENTATION AND OUTCOME

5.1 User Section

In this project we have we 3 sections (1) Admin section, (2) Seller section and, (3) User section or member section.

5.1.1 Registration Module

This module allows Tenants or the system users to register and log in in case they want to make any inquiries in the system.



The screenshot shows a web application interface for registration. At the top, there is a black navigation bar with the text 'বাড়ি ভাড়া' on the left, 'Home Contact Us About Us' in the center, and a 'LOGIN' button on the right. The main content area is titled 'Registration' and contains several input fields: 'First Name' and 'Last Name' (two separate boxes), 'Email' (one box), 'Phone Number' (one box), 'Password' and 'Confirm Password' (two separate boxes). Below these fields is a large green 'Register' button. At the bottom of the form, there is a link that says 'Already Have An Account? Login Here'.

Fig: 5.1 user & seller registration

5.1.2 Login Module

In this a user login dashboard. From here members can also login using Email, Phone and password.

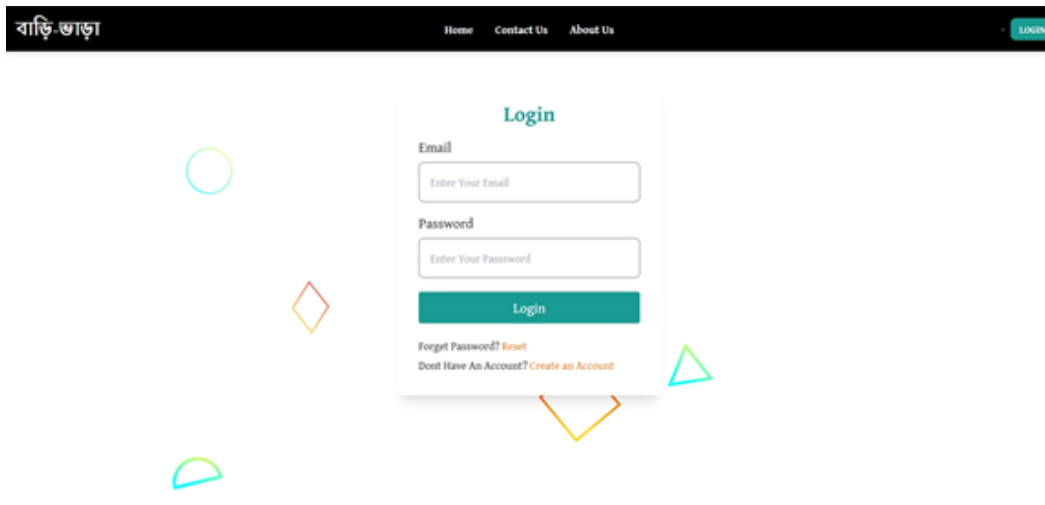


Fig: 5.2 user and seller login

5.1.3 User Dashboard

A user can see all the information about his home on his dashboard.

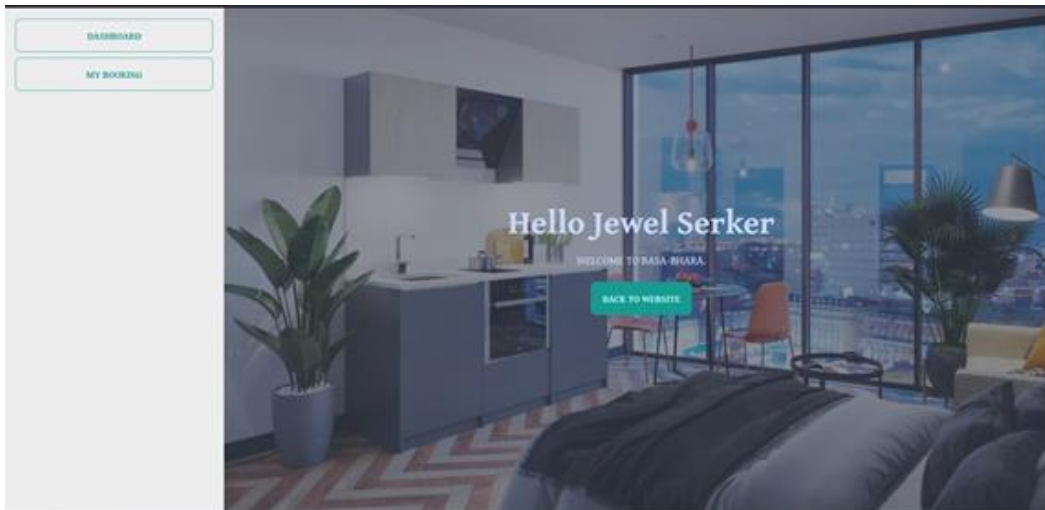


Fig: 5.3 user dashboard

5.1.4 Find Home/Seat/Apartment

When a user needs room/home/apartment for any kind of reason, he can easily find a wide number of seat vacancies from this section of this website.

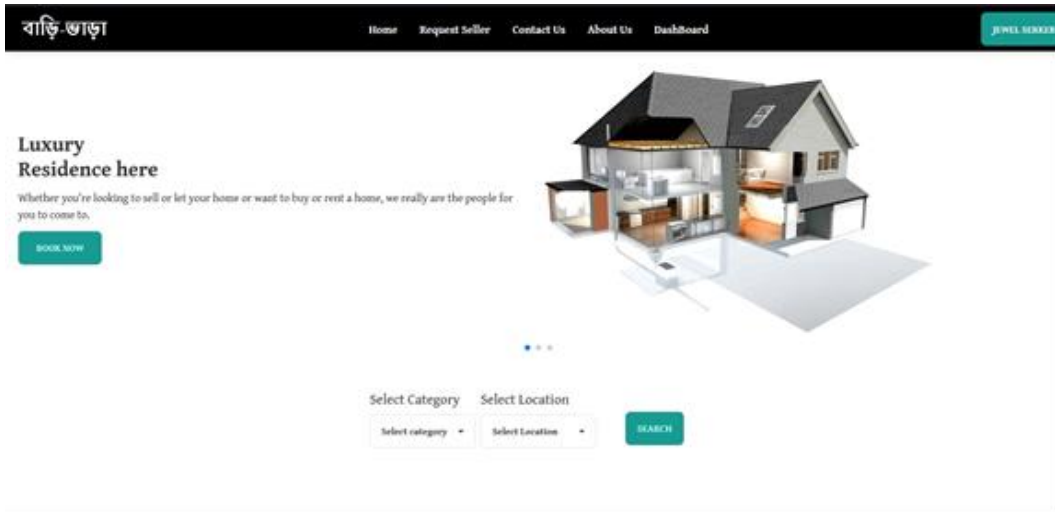


Fig: 5.4 find room/seat/home

5.1.5 Book Home/Seat/Apartment

This module allows Tenant/Customer to Book a house after selecting from available listed houses. To book a house, the customer/Tenant would be required by the system to log in or register to continue with the process.

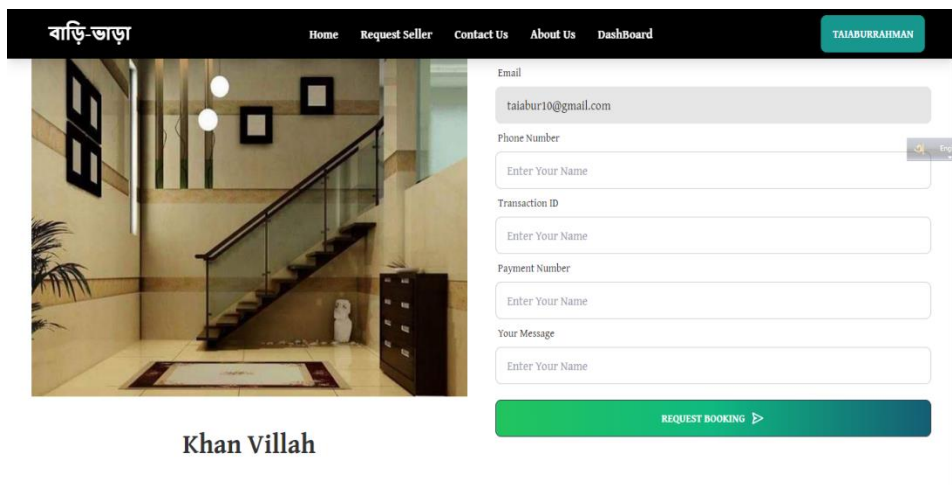


Fig: 5.5 booking room/seat/home

5.1.6 Contact and Feedback

This module allows Customers to send their queries and feedbacks about the system and services enquiries and their experience with the system services.

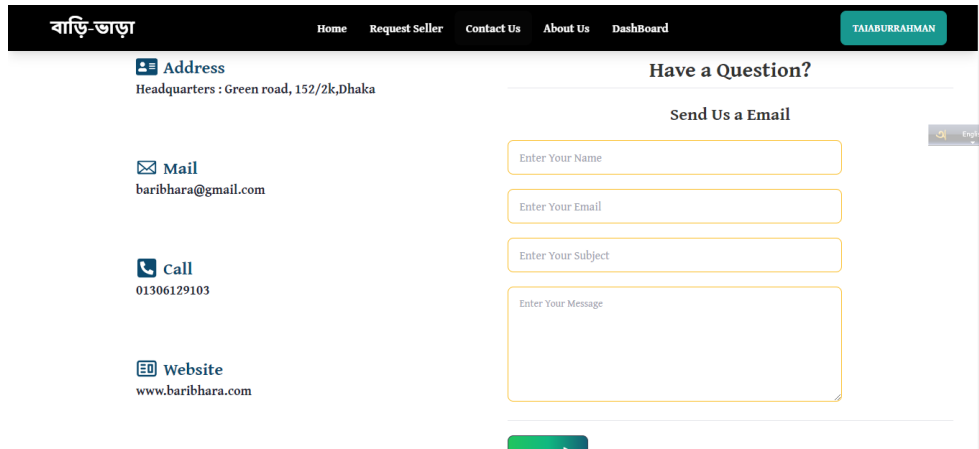


Fig: 5.6 contact and feedback

5.2 Seller Section

5.2.1 Seller Request

Any user can request a seller to be supervised by an administrator. They can add their home if the admin approves them.

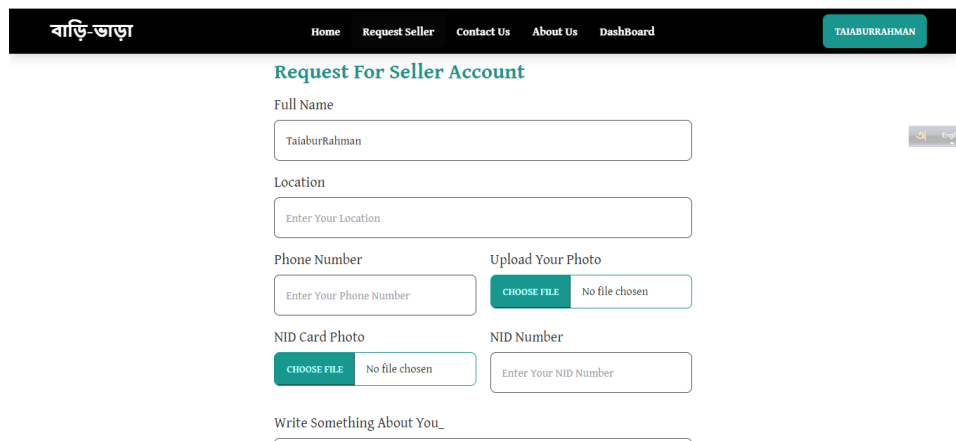


Fig: 5.7 seller request

5.2.2 Add Home/Seat/Apartment

This module allows Landlord to post a property in the system after successful registration or login.

Fig: 5.8 add home/seat/apartment

5.2.3 Booking Information

In the information of the seller's home booking, when a user will book the home

Fig: 5.9 booking Information

5.1 Admin Section

5.3.1 Admin Dashboard

In this an admin dashboard. From here admin can control everything about this application. From here admin can control his every user and seller.

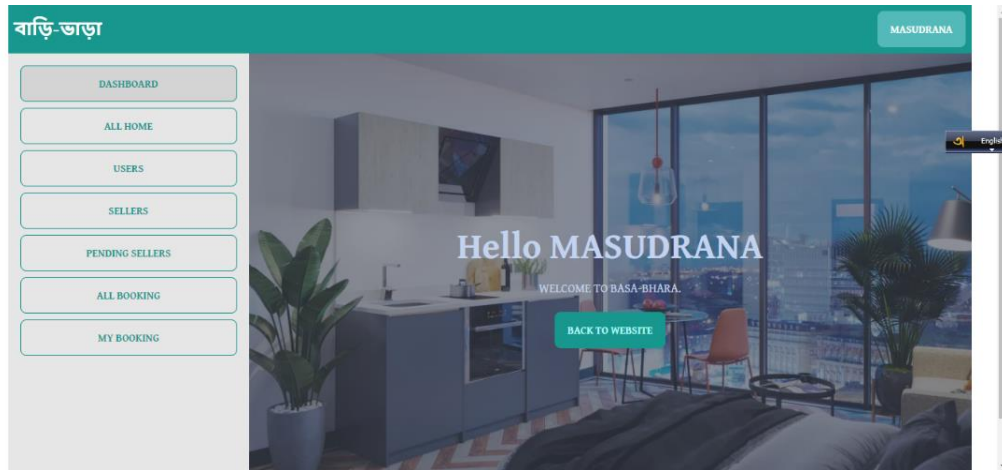


Fig: 5.10 admin dashboard

5.3.2 Approve Seller Request

Any user can request a seller to be supervised by an administrator. They can add their home if the admin approves them.

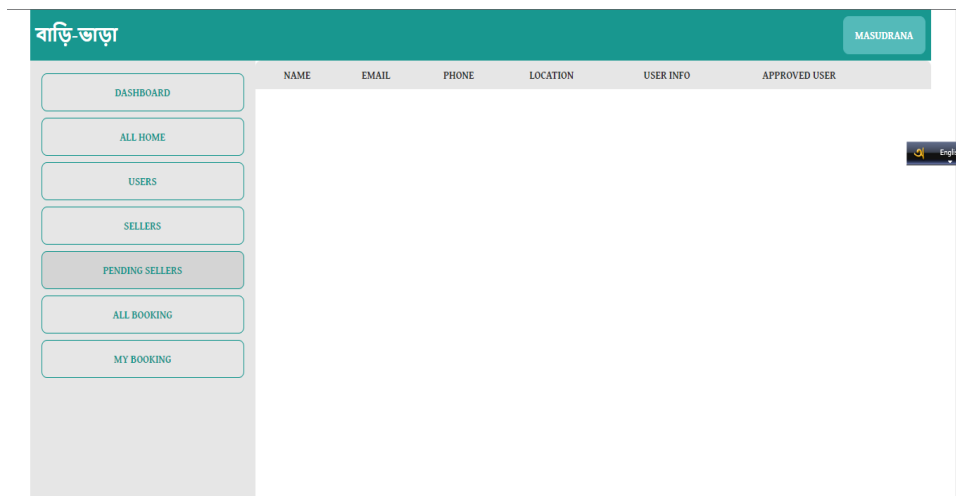


Fig: 5.11 seller approve dashboard

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

The House Rental business has emerged with a new goody compared to the experience where every activity concerning House rental business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of the internet. Nowadays, customers can reserve book/buy/sale House online, rent House online, and have the house contracted successfully without any sweat once the customer is a registered member of the To-Let Management System.

6.2 Limitations and Future Work

6.2.1 Limitation

The system at present does not take care of the money payment methods, as the consolidated constructs need SSL standards and are critically to be initiated in the first face, the application of the credit card transactions is applied as a developmental phase in the coming days. The system needs more elaborative technicality for its inception and evolution.

6.2.2 Future work

- **Payment Getaway:** There will be included different payment gateways like Bkash, Nagad, and Rocket etc. This system will be more helpful for the users and seller.
- **Google Map integration:** Google Map integration is very easy Home can be found.

REFERENCES

- [1] Mary Njeri, “House Rental Management System”, 4, 2014, pp.5-8
- [2] System Requirement Analysis, “System technical requirements is a process that Identifies the functionality that is needed by a system in order to satisfy the users Requirements”, 2, 1993, pp.18-20.
- [3] Lucidchart, “A data flow diagram show the way information” 3, 2015. Online: <https://www.lucidchart.com/>
- [4] Lean Software Development, “Software development life cycle”, because Waterfall model is easy to understand, “SDLC”, vol. 11, no. 5, 2017, pp.21-23.