DOCUMENTATION

Stay Solution BD

Developed by:
Md. Emran
Full Stack Developer

ABSTRACT

This documentation provides a comprehensive guide to StaySolutionBD, a state-of-the-art home rental platform that connects property owners with prospective tenants. Designed to simplify the rental process, StaySolutionBD offers an elegant, user-friendly interface alongside powerful backend services that ensure seamless listing, searching, and booking functionalities.The purpose of StaySolutionBD is to streamline the way renters find and secure rental homes while providing property owners with an effective marketplace to list their properties. This platform addresses the complexities of the rental market, offering tools for price comparison, availability tracking, secure payments, and user verification.

At the heart of the rental industry revolution, StaySolutionBD streamlines the rental process for property owners and tenants alike. This backend repository is a trove of meticulously crafted code that supports the server, database, and APIs - all harmonizing to ensure the frontend application's performance is nothing short of exceptional.

INDEX

INTRODUCTION

1.1 Introduction:

StaySolutionBD is an innovative web portal designed to revolutionize the home rental industry for tenants, property owners, and administrators alike. This comprehensive platform facilitates effortless access to rental listings, appointment scheduling with property owners, secure online payments, and efficient management of rental agreements. By leveraging the power of digital solutions, StaySolutionBD enhances the property rental and management process, making it more accessible, reliable, and user-friendly.

The cornerstone of StaySolutionBD's reliability lies in its capacity to present individual property details and tenant interactions effectively over the internet. The portal enables seamless communication between tenants and property owners, while administrators gain a robust tool for overseeing transactions, managing listings, and ensuring the platform operates smoothly.

1.2 Problem Statement:

In an era where convenience and efficiency are paramount, the property rental sector demands innovation. Traditional methods of managing rentals are often cumbersome, error-prone, and lack the agility needed in a fast-paced world. StaySolutionBD addresses these challenges head-on by digitizing the entire rental journey—from discovering and booking properties to managing tenancies and processing payments online. Our mission is to simplify the complex ecosystem of rental management using advanced, modular technology that offers a streamlined user experience and facilitates expansion and adaptability.

Objective:

The objective of StaySolutionBD is to offer a flawless, secure, and rapid management system for the home rental domain. The portal is engineered to achieve the following goals:

- ❖ To digitize comprehensive details regarding property listings and administrative data.
- ❖ To establish a user-friendly, web-based rental management system.
- ❖ To provide property owners with an efficient platform for managing their listings.
- ❖ To offer tenants a convenient and reliable means of finding and securing rentals.
- ❖ To facilitate the scheduling of property viewings, aligning with the availability of both tenants and owners.
- ❖ To ensure that inventory and property details are updated in real-time upon any transaction.
- ❖ To maintain a historical record of tenancies, providing tenants with access to their rental history and documents.

Scope of the Project:-

StaySolutionBD is envisioned to be the ultimate solution for property rental management, serving diverse housing markets, including apartments, houses, vacation rentals, and more. The portal is designed for:

- Administration oversight,
- Property owner management,
- Tenant engagement,
- Appointment scheduling,
- Online payment processing,

- Rental agreement digitization.

Through StaySolutionBD, we aim to deliver these services efficiently and cost-effectively, significantly reducing the time and resources currently needed for such activities, while enhancing the rental experience for all users.

1.3 MODULES:

The StaySolutionBD platform is architected around several core modules, each designed to facilitate a specific aspect of the home rental process. The system comprises the following key modules

The entire project mainly consists of 7 modules, which are

- ❖ Admin module
- ❖ Doctor module
- ❖ Patient module
- ❖ User

1.3.1 Admin module:

- ❖ Manage Users: Oversee tenant and property owner accounts, ensuring all user interactions adhere to platform guidelines.
- ❖ Property Listings Oversight: Monitor and curate property listings for quality and accuracy.
- ❖ Appointment Coordination: Facilitate the scheduling of property viewings between owners and potential renters.
- ❖ Transaction Oversight: Review and manage financial transactions across the platform.
- ❖ Database Management: Maintain the integrity and performance of the platform's database.
- ❖ Content Moderation: Update informational content and manage platform-wide communications.

1.3.2 Owner module:

- ❖ Property Management: List new properties, update existing listings, and manage rental availability.
- ❖ Appointment Scheduling: Organize and confirm viewings with potential renters.
- ❖ Rental Agreements: Draft and manage digital rental contracts.
- ❖ Financial Overview: Track payments, generate invoices, and review financial reports.
- ❖ Profile Customization: Update personal and property profiles to reflect current offerings and credentials.

1.3.3 Renter Module

- ❖ Property Search: Explore available properties using advanced search and filter tools.
- ❖ Booking Management: Schedule viewings, book properties, and manage rental agreements.
- ❖ Payment Processing: Securely process rental payments and view transaction histories.
- ❖ Favorites and Alerts: Save preferred properties and set alerts for new listings or changes.
- ❖ Profile Management: Update personal information and rental preferences.

1.3.4 User Module

- ❖ Registration: Sign up to join StaySolutionBD as either a property owner or renter.
- ❖ Login/Logout: Secure access to the platform with the ability to log in and log out of the system.
- ❖ Authentication: Robust security measures to protect user accounts and personal information.
- ❖ User Support: Access to help and support resources, including FAQs, contact forms, and live chat.

REQUIREMENT SPECIFICATION

2.1 Introduction

This document outlines the Software Requirements Specification (SRS) for StaySolutionBD, a sophisticated home rental platform. It defines the functional and non-functional requirements, objectives, and features the system is expected to provide. This SRS will serve as a blueprint for the subsequent stages of design and implementation. System requirements are critical to ensuring that the software operates correctly and efficiently on users' systems. They are divided into minimum and recommended requirements, allowing the software to be accessible to a wider range of users while also taking advantage of more powerful hardware to deliver enhanced performance.

As StaySolutionBD is a web-based platform, system requirements mainly pertain to the server infrastructure for hosting the web application and the end-user's device specifications for accessing the platform through a web browser.

2.2 Hardware Requirements

While StaySolutionBD is a web-based application and does not require client-side installation, the server-side infrastructure needs to meet certain specifications to ensure optimal performance and reliability. Additionally, users accessing the platform will have a better experience if their hardware meets certain standards.

Server Hardware Requirements

For the current project, the following server-side hardware configurations are recommended:

- Processor: Intel Xeon or equivalent server-grade CPU
- RAM: 16 GB or higher for handling multiple concurrent sessions
- Hard Disk: 1 TB SSD for storage and quick data retrieval
- Networking: High-speed internet connection with at least 1 Gbps throughput for efficient data handling

Client Hardware Requirements

Users accessing StaySolutionBD will have a smooth experience with the following recommended configurations:

- Processor: Intel Core i3 or higher, or equivalent
- RAM: 4 GB or higher for better multitasking and smooth browsing
- Hard Disk: At least 20 GB of free space
- Display: Capable of running a resolution of 1024x768 for clear visibility of the platform's content
- Internet Connection: Broadband internet connection with minimum speeds of 10 Mbps

Compatibility List

StaySolutionBD is designed to be compatible with the latest web browsers, including:

- Google Chrome (latest version)
- Mozilla Firefox (latest version)
- Safari (latest version)
- Microsoft Edge (latest version)

2.3 SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:

OPERATING SYSTEM: **Windows 7/ XP/8/10/11**

Frontend:

- Next.js: Empowers our server-side rendering for speed and SEO.
- TypeScript: Brings type safety for robust frontend code.
- Ant Design: Delivers a polished UI with a comprehensive component library.
- Redux: Manages global state efficiently.
- Axios: Facilitates API communication.
- React Hook Form & Yup: Streamlines form handling and validation.

Backend:

- Node.js & Express.js: Form our server's backbone for fast, scalable APIs.
- TypeScript: Ensures a maintainable, error-resistant backend.
- Prisma: Interfaces with PostgreSQL, providing a type-safe ORM.
- Zod: Validates data schemas.
- Husky: Maintains code quality via Git hooks.
- Cloudinary & Multer: Manages media uploads seamlessly.
- jsonwebtoken: Secures user sessions and authentication.

Database:

- PostgreSQL: Our choice for a reliable and robust data repository.

SYSTEM ANALYSIS AND DESIGN

3.1 SOFTWARE SYSTEM ATTRIBUTES:

- **3.1.1 Reliability:**

This application is reliable product the produces fast answer verified output all its process.

- **3.1.2 Availability:**

This application will be available to use and help them to carry their operation conveniently.

- **3.1.3 Security:**

This application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

- **3.1.4 Maintainability**

Back-Up: The system shall provide the capability to back-up the Data
Errors: The system shall keep a log of all the errors

3.2 Existing System

In the current property rental market, many operations are managed through a manual, paper-based system. This leads to inefficiencies, such as misplaced documents, data duplication, and inconsistencies across various records. The lack of a centralized data management system often results in incomplete information and complicates the auditing process, leading to a significant waste of time and resources.

3.3 Proposed System

StaySolutionBD is conceived to modernize the property rental sector, catering to tenants, property owners, and administrators. By centralizing control of listings, bookings, and transactions, the platform enhances the ease with which users can engage with rental services. The goal is to offer these services more efficiently and cost-effectively, significantly reducing the time and resources currently required.

3.4 Feasibility Study

This phase evaluates the viability of StaySolutionBD with a preliminary plan and cost estimates. It ensures the platform aligns with the company's capabilities and does not overextend its resources.

3.4.1 Economic Feasibility

This analysis ensures the financial implications of StaySolutionBD are in line with the company's budget. The goal is to utilize open-source technologies wherever possible to minimize costs while ensuring that any necessary investments in proprietary solutions are justified and add significant value.

3.4.2 Technical Feasibility

The technical feasibility assessment confirms that the proposed system does not demand excessive technical resources. StaySolutionBD is designed to have modest requirements, allowing for implementation with minimal system changes and ensuring that the technical demands on users are kept low.

3.4.3 Operational Feasibility

This aspect evaluates user acceptance and the practical use of StaySolutionBD. The platform must be user-friendly, and any training required for users to effectively utilize the system must be straightforward. The user adoption rate is a critical measure of success, and as such, the system is designed to be intuitive and to enhance, rather than disrupt, the user's operational workflow. Users should feel empowered and confident, contributing feedback that leads to continuous improvement.

SYSTEM DESIGN

3.5 SYSTEM DESIGN:

3.5.1 INTRODUCTION TO UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting

Visualizing

Through UML see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise, unambiguous, and complete UML addresses the specification of all the important analysis design and implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language by mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

3.5.2 UML Approach

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. You draw diagram to visualize a system from different perspectives, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

1. Class diagram
2. Use a case diagram
3. Sequence diagram
4. E-R diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Use case diagram Propose Project:

A **use case** is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each **use case** is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

E-R diagram:

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

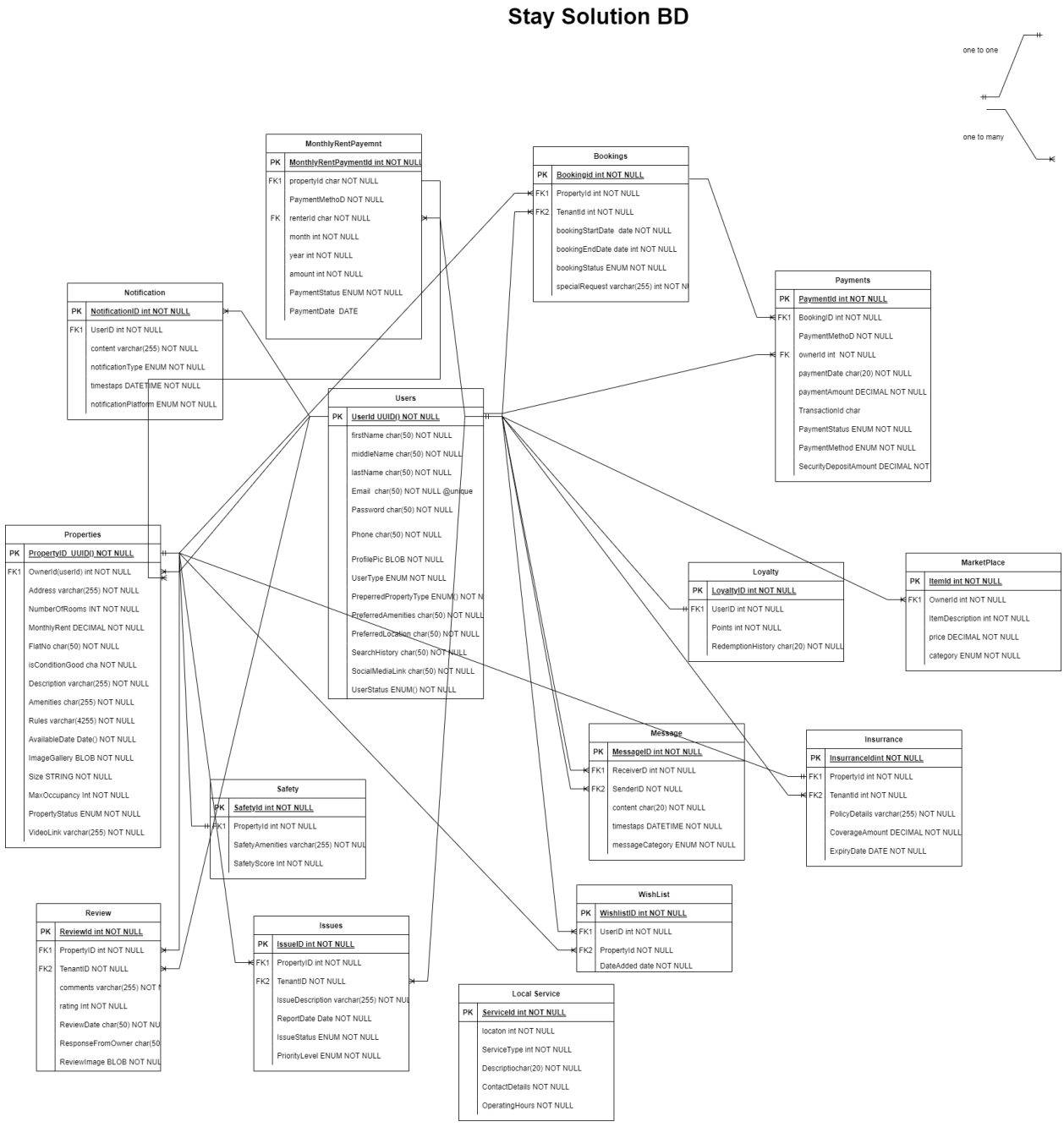


Fig: Easy Doctor Portal (Use-Case)

PROJECT MANAGEMENT

4.1 Project planning and scheduling

Project planning is a part of project management, which relates to the use of schedules such as Gantt chart to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using a activity network diagram that enables identification of the critical path.

4.1.1 Methodology

We have used Iterative and Incremental Development model (IID) for our project development. This development approach is also referred to as Agile Development approach. Iterative and Incremental Development is a software development process developed in response to the more traditional agile model. This model is designed to take care of such big project. The large and complicate project chiefly demand better development and testing procedure. Hence I choose the agile model for developing our proposed software.

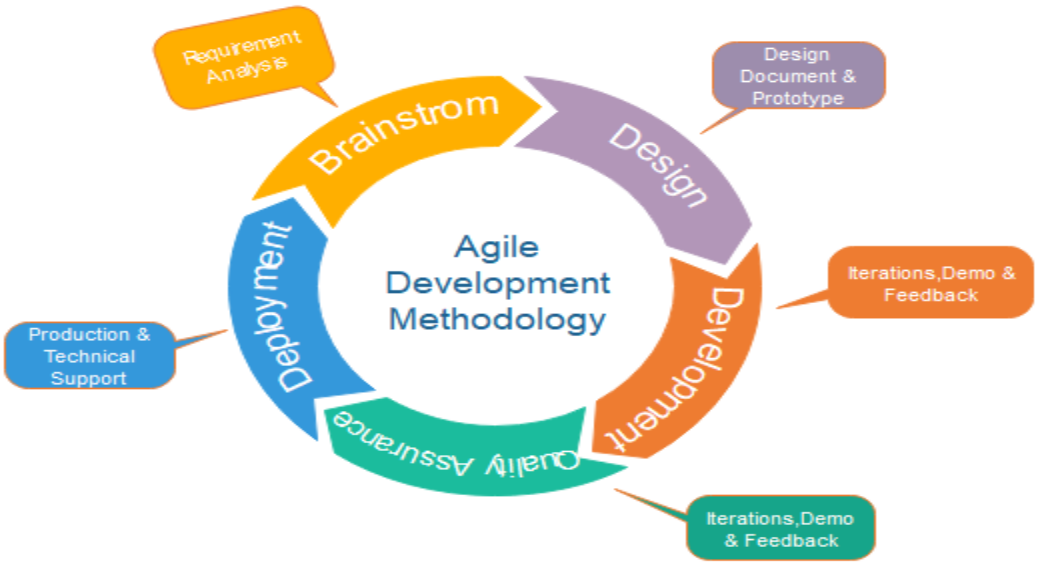


Fig. Agile Model

4.1.2 Project Management Life Cycle

The project management life cycle has four phases. Each project life cycle phase is described along with the tasks need to complete it. The four phases is

1) Requirements gathering
2) Design the requirements
3) Construction/ iteration
4) Testing/ Quality assurance
5) Deployment
6) Feedback

1. Requirements gathering: First step on our proposed project is, must define the requirements. We should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

2. Design the requirements: When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3. Construction/ iteration: When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4. Testing: In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

5. Deployment: In this phase, the team issues a product for the user's work environment.

6. Feedback: After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

4.1.3 Project plan

Once we examine that the project is feasible, I undertake project planning.

Task_ID	Task_Name	Duration	Starting date	Finish date
T1	Registration	6	5/09/2023	10/09/2023
T2	Login	2	11/09/2023	12/09/2023
T3	Logout	2	13/09/2023	14/09/2023
T4	Admin Profile	7	15/09/2023	21/03/2023
T5	Owner Profile	6	22/09/2023	27/03/2023
T6	Renter Profile	9	28/09/2023	5/10/2023
T7	Property list	10	6/10/2023	15/10/2023
T8	Booking	10	16/10/2023	25/10/2023
T9	User	11	26/10/2023	07/11/2023
T10	Unit Test	10	8/11/2023	18/11/2023

5. IMPLEMENTATION:

5.1 Introduction:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.2 Functional Requirement

This system interface is divided into two section

1. Admin interface
2. User interface

5.2.1 Admin interface:

- Admin can add or delete user.
- Admin can make an admin
- Admin can add and delete a doctors

- Admin can verify user account.
- Update profile.

5.2.2 User interface

- User (patient) can see any services (treatment) details and price.
- Users can update or edit their accounts.
- Users can pay using the stripe digital payment gateway to pay their treatment
- Login and logout system.
- If any user forgets her/his password he/she can recover his /her account.

TESTING

6.1 INTRODUCTION TO SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTING:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

- All links should take the user to the correct page.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Decision Table Based Testing:

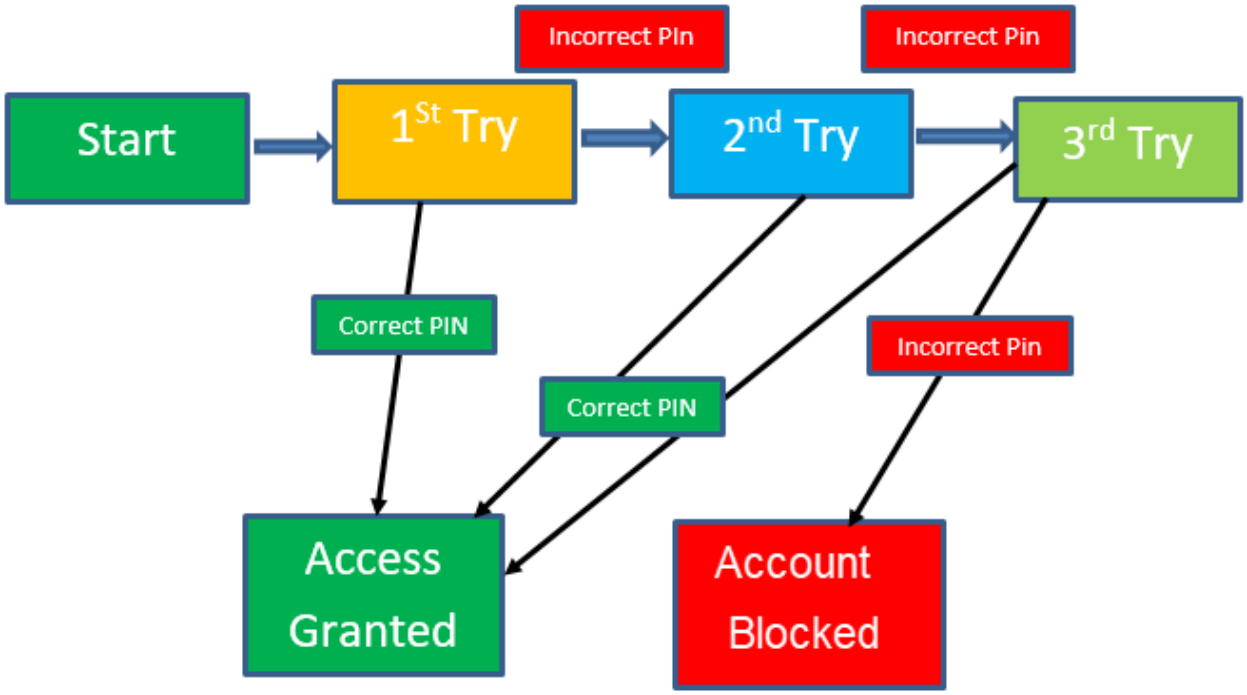
A decision table is also known as to Cause-Effect table. This software testing technique is used for functions which respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Input								
Name	F	T	F	T	F	T	F	T
Email	F	F	T	T	F	F	T	T
Message	F	F	F	F	T	T	T	T
Output								
Submit	F	F	F	F	F	F	F	T

State Transition

In State Transition technique changes in input conditions change the state of the Application Under Test (AUT). This testing technique allows the tester to test the behavior of an AUT. The tester can perform this action by entering various input conditions in a sequence. In State transition technique, the testing team provides positive as well as negative input test values for evaluating the system behavior.

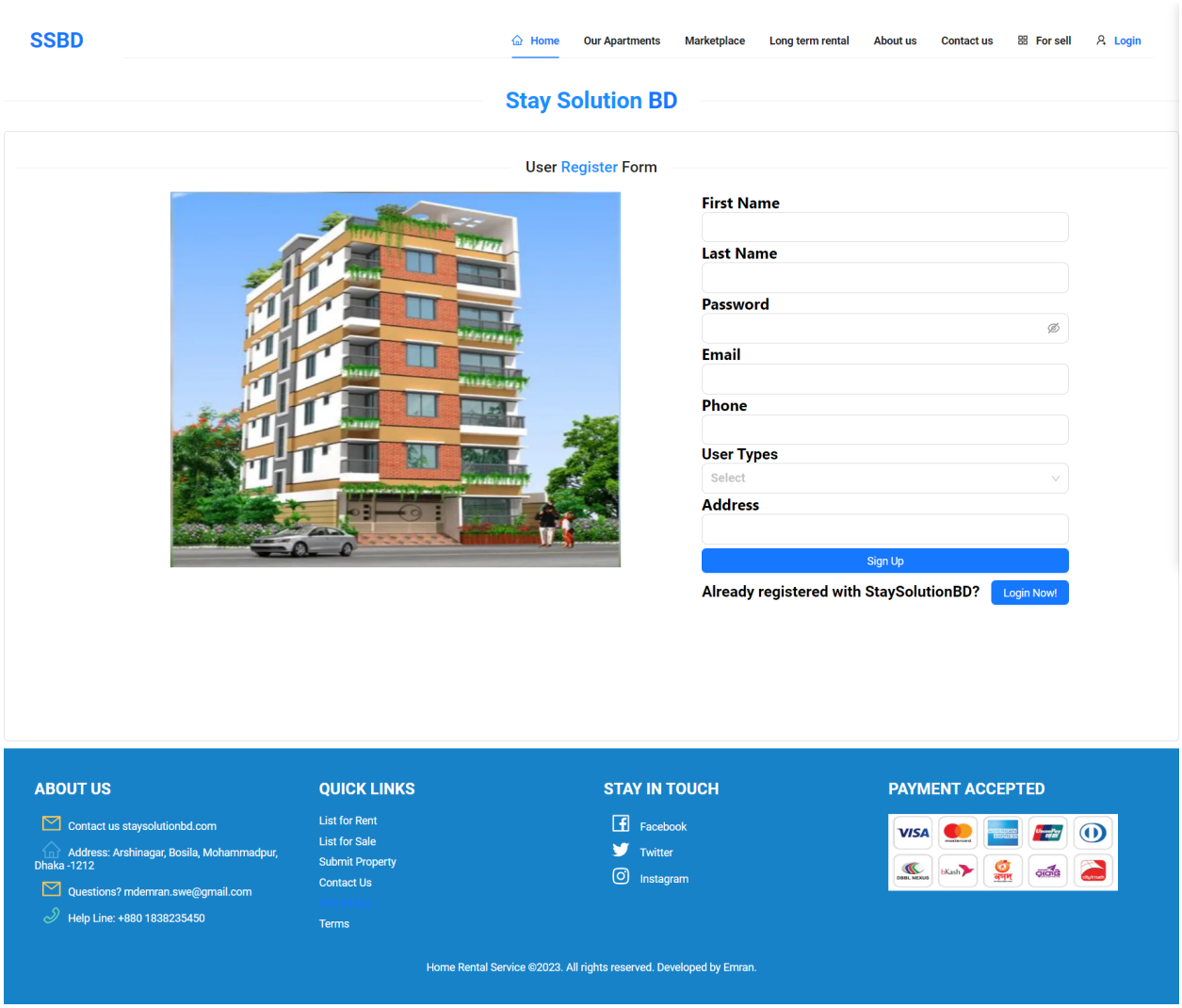
State Transition Table diagram:



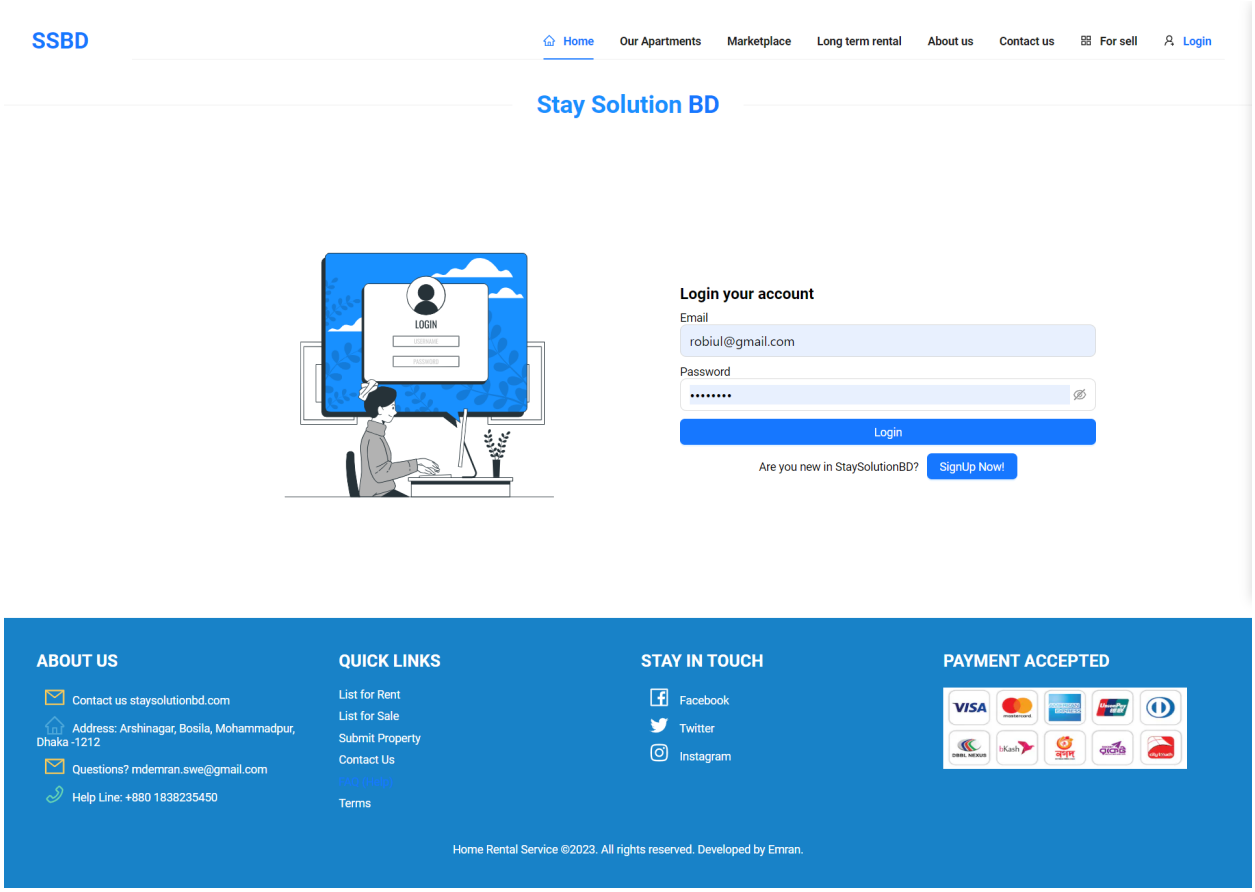
SAMPLE SCREENSHOTS

7.0 UI Screenshot

Registration Page:


SSBD
Home
Our Apartments
Marketplace
Long term rental
About us
Contact us
For sell
Login
Stay Solution BD
User Register Form
First Name
Last Name
Password
Email
Phone
User Types
Select
Address
Sign Up
Already registered with StaySolutionBD?
Login Now!
ABOUT US
Contact us staysolutionbd.com
Address: Arshinagar, Bosila, Mohammadpur, Dhaka -1212
Questions? mdemran.swe@gmail.com
Help Line: +880 1838235450
QUICK LINKS
List for Rent
List for Sale
Submit Property
Contact Us
Terms
STAY IN TOUCH
Facebook
Twitter
Instagram
PAYMENT ACCEPTED
Home Rental Service ©2023. All rights reserved. Developed by Emran.

Login page



Property page

All Selling Property

Sort by relevance

Search by title, ..etc

Monthly Rent

Number of Rooms

Max number of people

4 available $15000

Shekh Home royel class apartme...

Sheikh Home Special Apartment in Arshinagar stands as a testament to luxury and elegance. Nestled in Dhaka's most prestigious area, this apartment is ...

Location: Dhaka

Amenities:Washer/Dryer,Air Conditioning,TV with Cable,Swimming pool,Workspace,BBQ Grill,Garden or backyard

Rules :No smoking,No pets,No outside guests

5 available $15000

Premium Sheikh Home Special F...

Sheikh Home Special Apartment in Arshinagar, Dhaka, is a beacon of modern luxury. This architecturally stunning apartment is a perfect symphony of for...

Location: Dhaka

Amenities:Washer/Dryer,Air Conditioning,Free parking on premises,Dishwasher,Kitchen,Heating

Rules :No smoking,No pets,No loud noises after 10pm,No outside guests,Check-in between 2pm and 8pm,Security deposit required

4 available $10000

Sheikh Home Special Apartment ...

Sheikh Home Special Apartment in Arshinagar, Dhaka, is a beacon of modern luxury. This architecturally stunning apartment is a perfect symphony of fo...

Location: Dhaka

Amenities:Wi-Fi,TV with Cable,Heating,Fireplace

Rules :No smoking,No pets,Check-out by 11am

4 available $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment: A Fusion of Artistry and Elegance in Arshinagar's Skyline. Embrace the zenith of contemporary opulence in Dhaka's illus...

Location: Dhaka

Amenities:Dishwasher,Gym,Heating

Rules :No smoking,No parties or events,Photo ID required upon check-in

4 available $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment in Arshinagar, Dhaka, is a beacon of modern luxury. This architecturally stunning apartment is a perfect symphony of for...

Location: Dhaka

Amenities:Dishwasher,Free parking on premises,Fireplace,Balcony or Patio

Rules :No smoking,No loud noises after 10pm,Minimum age requirement

4 available $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment: A Portrait of Modern Grandeur in Arshinagar. Discover an abode where every detail is a whisper of sophistication, set ...

Location: Dhaka

Amenities:Wi-Fi,Swimming pool,Balcony or Patio

Rules :No smoking,No parties or events

4 booked $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment in Arshinagar stands as a testament to luxury and elegance. Nestled in Dhaka's most prestigious area, this apartment is ...

Location: Dhaka

Amenities:Wi-Fi,TV with Cable,Swimming pool,Hot tub

Rules :No smoking,No pets,No loud noises after 10pm,Check-in between 2pm and 8pm

4 booked $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment in Arshinagar stands as a testament to luxury and elegance. Nestled in Dhaka's most prestigious area, this apartment is ...

Location: Dhaka

Amenities:TV with Cable,Gym,Heating,Kitchen,Balcony or Patio,Child-friendly

Rules :No smoking,No pets,No parties or events,No loud noises after 10pm,Photo ID required upon check-in,Security deposit required

4 booked $10000

Sheikh Home Special Apartment

Sheikh Home Special Apartment in Arshinagar stands as a testament to luxury and elegance. Nestled in Dhaka's most prestigious area, this apartment is ...

Location: Dhaka

Amenities:Wi-Fi,Heating,Kitchen,Child-friendly

Rules :No smoking,No pets,No loud noises after 10pm,No outside guests,Check-in between 2pm and 8pm

4 booked $10000

Sheikh Home Special Apartment

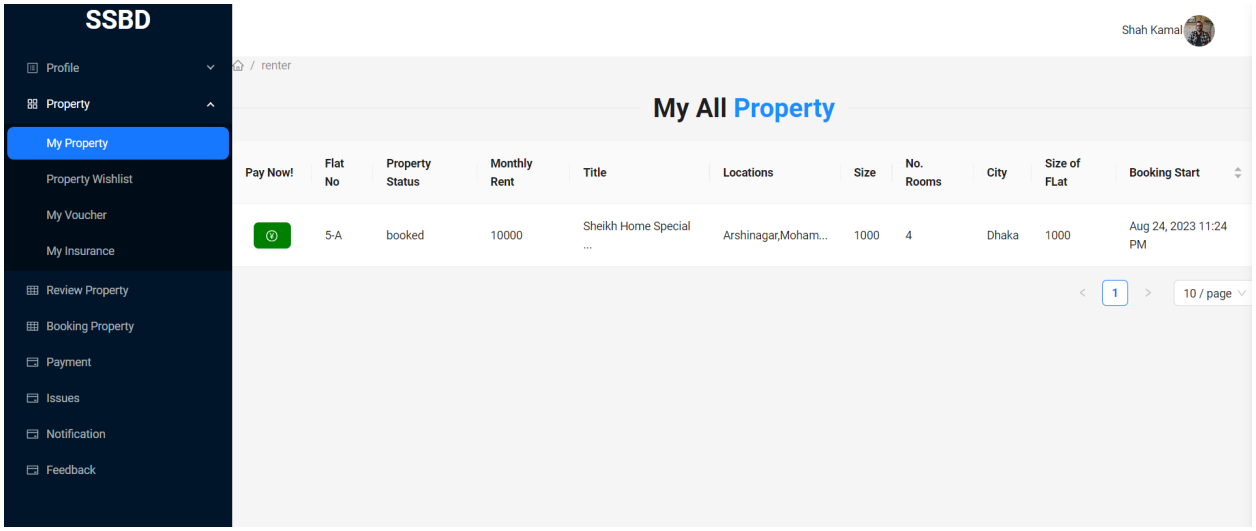
iscover the ultimate in luxury and convenience at Sheikh Home Special Apartment. This beautiful two-bedroom apartment in the heart of Arshinagar, Dhak...

Location: Dhaka

Amenities:Wi-Fi,TV with Cable,Balcony or Patio

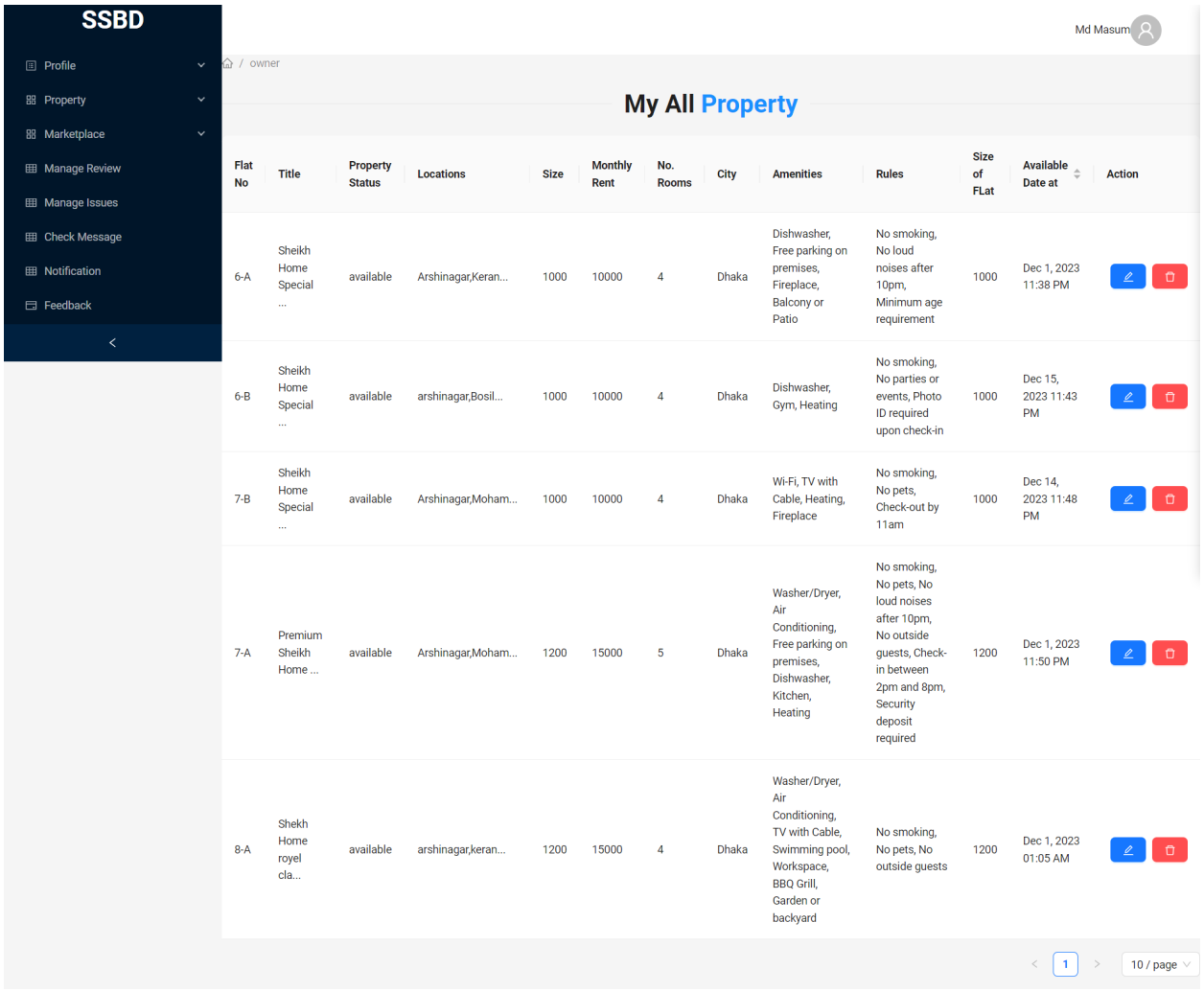
Rules :No smoking,No pets,No loud noises after 10pm,Check-out by 11am,Photo ID required upon check-in

Renter Dashboard

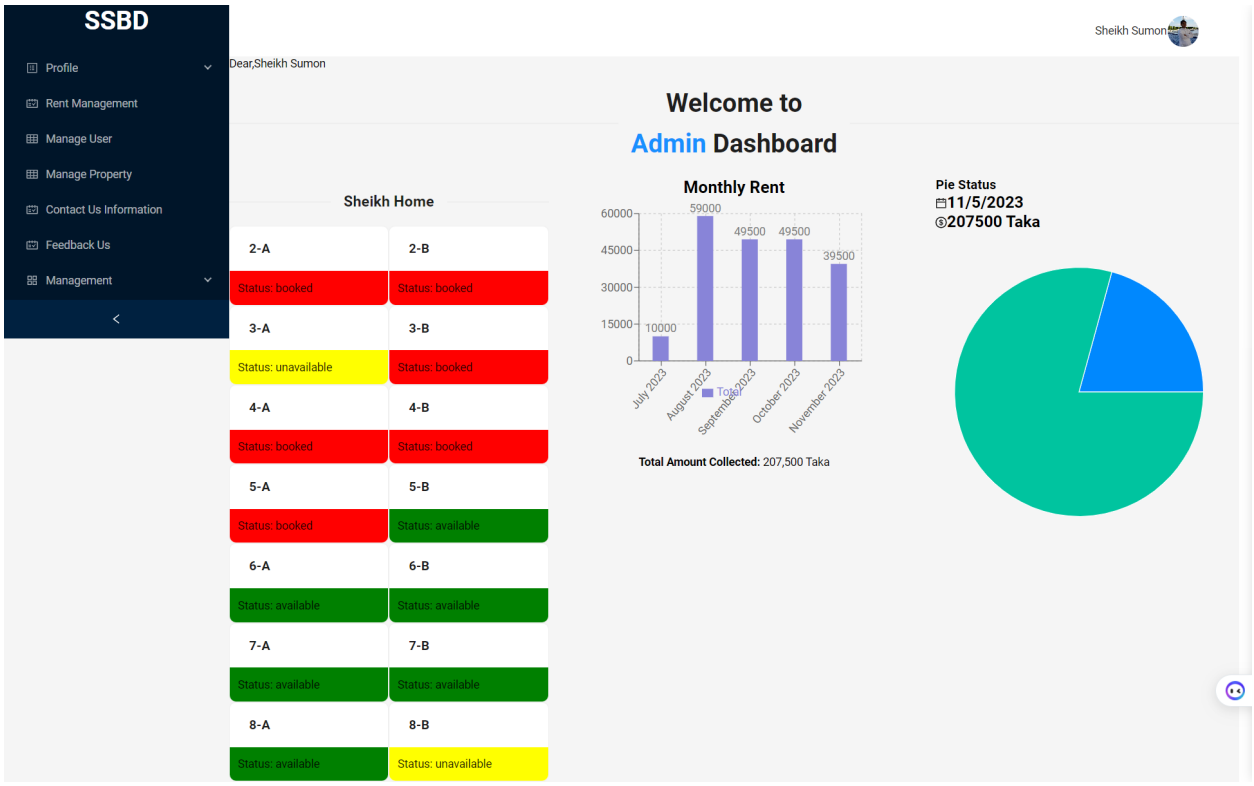


Renter Payment

Owner Dashboard



Admin Dashboard:



8.1 CONCLUSION:

The journey through the development of StaySolutionBD has been illuminating and enriching. The project has afforded us the following insights and skill enhancements:

- We have acquired a profound understanding of the home rental industry, a sector that echoes real-world dynamics and complexities.
- Our proficiency in database design has been significantly bolstered. The necessity for accurate, real-time property and booking information has underscored the importance of a well-structured database.
- Adherence to a development schedule has honed our time management skills, reinforcing the importance of deadlines and milestones in project management.
- Collaboration within our team has fostered a robust sense of teamwork, and through collective effort, our confidence in tackling real-world projects has soared.
- We encountered challenges with input validations initially; however, through persistent teamwork and strategic problem-solving, we implemented a robust validation mechanism.

8.2 Current System Limitations

Despite the successes, we acknowledge certain limitations in the current version of StaySolutionBD:

- Lack of Real-time Communication: The absence of a live chat feature limits real-time interaction between tenants and property owners.
- Risk of Data Loss: Potential for data mismanagement exists, posing a risk to the integrity of user and property information.
- Absence of Direct Owner-Renter Communication: Currently, there is no provision for live communication, which could streamline the negotiation and booking process.
- Historical Data Accessibility: Tenants may find it challenging to access their complete rental history and summaries within the platform.