The Smart house Rent Collection System

By
MD Ripon
Student ID: 181-15-10619

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Ms. Sharmin Akter
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Ms. Nazmun Nessa Moon
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

May 14, 2025

APPROVAL

This Project titled **"The Smart House Rent Collection System"**, submitted by Md Ripon, ID No: **181-15-10619** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **14 May, 2025**.

BOARD OF EXAMINERS

Dr. Sheak Rashed Haider Noori (SRH) — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mohammad Monirul Islam (MMI) — **Internal Examiner**
Assistant Professor, Internal Member
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Ms. Tasfia Anika Bushra (TAB) — **Internal Examiner**
Lecturer, Internal Member
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Zulfiker Mahmud (ZM) — **External Examiner**
Professor, Exterenal Member
Department of Computer Science and Engineering
Jagannath University

DECLARATION

I hereby declare that this project has been done by me under the supervision of **Ms. Sharmin Akter**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

13.05.25

Ms. Sharmin Akter
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Ms. Nazmun Nessa Moon
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Ripon 13.05.25

Md Ripon
Student ID: 181-15-10619
Department of Computer Science and Engineering
Daffodil International University

©Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. I'm deeply grateful to everyone who has assisted me in one way or another.

First, I express my heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

I am grateful and wish my profound indebtedness to **Ms. Sharmin Akter**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of my supervisor in the field of **"Web Design and Development"** to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

I would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing my project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

I would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

ABSTRACT

Rental property management in third world nations faces lapses in procedure because of poor follow-up, communication gap and dearth of digital channel. To solve these problems, this paper describes the design and implementation of a Smart House Rent Collection System—an online application designed to automate the house rent collection process and enhance notification and communication between a house Owner and Renters. The application is developed based on the Laravel (PHP) framework for the backend part, Bootstrap together with custom HTML/CSS for the frontend, and MySQL as data will be persistent._ Used Technologies: Backend: Laravel (PHP) Frontend: Bootstrap + Custom HTML/CSS Database: MySQL Local development and debugging was carried out using supporting tools as XAMPP and Visual Studio Code. The platform comes with features that allows tenant registration, Property Owner-tenant interaction, Rent tracking, Rent due notifications through integrated SMS API & SMTP email server, secured Login system, Scheduled Generate / Download PDF reports & Exports, Automate Daily Backup and more with full database backup one can even restore and manage the data with Admin Panel.

It was installed and tested in a Local Server Environment and tested for usability, performance and functional correctness. Comparison with the current systems reveals its exclusive characteristics, personalized dashboard, real-time rent status update, automatic intimation, scalable architecture. Other than serving multiuser functionality, logs of payment for rent, payment history details are stored in the system, and has scope for extension in the future for any mobile application or integration with financial APIs. The solution, proposed as a solution is a reduction in manual process and also transparency, accountability and better data management in the renting process.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Smart house Rent Collection System aims at providing the property owners with a single dashboard to automate the rental business, which includes collecting payments, tenant control, rent auditing and generating reports. While rental homes are on the rise and tenants move-in and out with more frequency, property owners are likely to have payment and communication issues with their tenants and easier access to records are beneficial. Traditional methods that employed ledgers, paperwork's and spreadsheets, can give raise to a lot of financial and administrative problems in terms of mistakes, which usually accompanies inefficiency and un-automated process. This problem is addressed with a web-based interface for property owners to apply for flat registration, store tenant information, sending SMS as well as email notification and automate the generation of rent. The rental is effectively managed because overhead is reduced by increased efficiency and less work and because the transparency of the transaction is also improved. The application is secure and scalable due to the utilization of cutting-edge technology such as Laravel as a backend framework, and Bootstrap for frontend responsiveness. The focus of this project are the issues such as managing rental properties efficiently, automating invoice verification and secure data management in the form of encryption and authenticated access control. And in subsequent versions, it will support online payment systems, the developers said, to ease the transaction process of rental payments for both the landlord and tenant. The automation has to evolve in line with the growing digitization across industries, including rental space. This will enable landlords to give more resources to the needs of the tenants and spend less time managing money and financial inconsistencies. It also intends to become a model for other deployments in real estate management by allowing integration with property

listings services to increase its field of action. The primary goal is to provide property owners with a fully automated and manageable system for rental property and enhancing user experience.

1.2 Motivation

On the other hand, The Smart house Rent Collection System has been a result of the concern of the property owners on the continuous problems within the burden of managing their rental properties in the traditional manual way (Anonymous, 2015). As you accumulate rentals and tenant turnover increases, it's challenging to keep track of records, collect rent on time and communicate timely. Manual methods are inefficient and Error-prone The manual ledger and spreadsheet-based process is both time-consuming and prone to errors, and lacks transparency. This project addresses these issues by providing a centralized web platform to automate the process, reducing the possibility of human error, tenant late payments, and better to engage tenants by sending notifications in a timely manner and providing real-time financial decision support. The objective is to minimize administrative burden, rectify the manual errors and fit into the increasing digitalization in property management. The intention of this Project is to create a simple, inexpensive, user-friendly package for managing the rent account and tenant detail and often getting a financial report. Also, we would be able to have that much more communication between the property owners and the renters and have a seamless no-tolerance action taken on missed rents and disputes, through the use of SMS & email notification. The Smart house Rent Collection System is developed based on the speed and accuracy with which accounts and inventories information was being done. Through insights built upon the best practices of these disciplines, the project guarantees that rigorous and expansion ready structure is created to be used by as many users as possible. AI-assisted predictive analytics to assess risk factors for meeting monthly due dates and suggested market-appropriate rental prices will be part of the next phase of project development. This brings the project a step closer to automating rental property management and enriches the user experience for owners and tenants.

1.3 Objectives

To automate the house renting service in SRMS with aim of helping house owners who has no/appraised time as to enable them gain full benefit. At this point in the project's progress, we are trying to create something which is easy to manage and maintain (therefore: good), but also something which is also scalable, reliable, and secure. Some objectives of this phase include:

- **Build a User Friendly Interface:** Make a User friendly design with easily manageable features and not too much complex menu that can be used by the Property Owner also who has a slight knowledge of technology, also using bootstrap.
- **Build Tenant and property management features:** Shop owners should be able to add, edit and manage tenants, flats and flats on rent and maintain the record accordingly.
- **Automate Rent Collection and Invoicing:** Automate invoicing and rent collection records to limit the need for manual input and to avoid errors and decrease administrative duties.
- **Combine SMS & Email Notifications**: Notify your tenants of upcoming due dates, overdue payments and other important information by text, in order to improve communication and decrease rent delinquency.
- **Security and Data Handling:** Improve access to sensitive financial and tenant documents with secure access control, strong authentication, and database encryption.
- **One Stop Reporting:** Allow property owners to see rental income and property reports summaries along with financials, rent collection and tenant records in real-time
- **Modularity and Scalability:** Build modular systems that can scale and have the possibility of been integrated in the future, like online payment systems, or AI predictive systems.

- **Add data security and recovery features:** Create a dependable backup system for the database to safeguard information and to ensure that data is retrievable in case the system breaks down.
- **Check the System under Performance and Optimal Reliability:** Validate that the system is working well in various conditions of use cases (large data, many automated users and so on).
- **Go-Live and Training:** Develop detailed documentation and training materials that property owners will need walking into the new system.

1.4 Methodology

The Smart house Rent Collection System was designed in a systematic phase and iterative method of building software applications to be user-friendly, successful and scalable. The project kicked off with a requirements analysis including interviews and feedback generation with property owners about real-life issues of rental management. On top of these ideas a modular system design concept was developed articulating basic functions of tenant management, rent control and automatic alerts. In a nutshell, Laravel provided a boilerplate for secure, tested and high performant logic and secure database interaction and Bootstrap was used to build a clean, responsive and accessible frontend. For the relational database, MySQL was chosen as it stores structured data including tenant record, rent invoice and property details. Development at local level was performed with XAMPP, Visual Studio Code as well as unit tests, integration tests and also system tests to assure precision and evidence of the operations. Integrations such as SMS & email notifications were configured via API to automate and optimize the way both Owners & tenants can communicate with each other. Upon verification in a local scene, the system will be deployed to a server in the cloud to enable scaling and remote accessibility. Feedback from the test users was used in a process of iterative improvement of the application so that the final product successfully automates lease operations and minimizes administrative overhead.

1.5 Project Outcome

The implementation of the Smart House Rent Collection System has lead to a viable and easy- to- use application developed for management of rental housing. The platform has improved operational efficiency, eliminated manual errors and reduced administrative overhead by automating processes like tenant onboarding, rent collection, invoicing, and alerts. Landlords will be able to manage e multiple properties and tenants from one convenient and easy to use dashboard that provides real-time access to rental information, payment history, and financial reporting. The addition of SMS and email notification functionality have enhanced property management services, resulting in more on-time rent payments and higher tenant communication rates. Moreover, the safe login and access control mechanism based on role also secure and audit the data. The reporting & analytics module, integrated into the system, offers complete visibility on income variations, enabling property Owners to take better informed financial decisions. In the testing, users stated anecdotally, there was a marked increase in successful task completion, and greater user satisfaction with the ease of use of the system. It's also a scalable system with cloud deployment functionality and the ability to incorporate added functionality in the future, such as links to online payment gateways and property listing, which will give you a robust base for a broader range of real estate management systems.

1.6 Organization of the Report

This report is structured into six comprehensive chapters, each detailing a critical phase of the project development lifecycle.

Chapter 1: Introduction presents an overview of the project, including the motivation behind its development, key objectives, the methodology used, the outcomes achieved, and a summary of how the report is organized.

Chapter 2: Background reviews the existing literature and previous works related to rental management systems, identifies existing gaps, and summarizes the foundational knowledge that supports the proposed solution.

Chapter 3: Research Methodology describes the overall design approach and technical requirements. It includes a system overview, proposed design, functional and non-functional requirements, context diagram, data flow diagrams, UI design, project plan, and task allocation.

Chapter 4: Implementation and Results explains the environment setup, testing processes, performance evaluation, and discusses the results obtained from the system implementation, backed by comparative analysis.

Chapter 5: Engineering Standards and Design Challenges outlines compliance with software, hardware, and communication standards. It also explores the social, environmental, ethical, and sustainability impacts of the system. Furthermore, this chapter discusses complex engineering problems, their solutions, project management aspects, and a financial overview.

Chapter 6: Conclusion summarizes the entire project, discusses limitations faced during development, and suggests future enhancements to expand system functionality and performance.

Chapter 2

Background

2.1 Introduction

The Rental Market is booming and the demand for a simplified management system is stronger than ever. Traditionally, Owners of property have used computer systems that are designed to perform the function of managing rental of real estate, physical records (like paper) that must be examined physically, and face to face communications. These antiquated systems do not have real-time tracking, secure data handling, and are not as efficient for communication causing lag in payments, records, and ease of administration. A chance to digitize Rental Management The rise of digital tools and web apps, opens up an opportunity to make rental management an automated process. This chapter discusses on background and context of Smart house Rent Collection System, existing approaches, related work and gaps in existing work for the proposed work that justify the need of more advance, scalable, and user- friendly solutions specially designed for landlords.

2.2 Literature Review

Digitalization of rental property management has recently become a hotly debated topic and there are several studies focused on its effectivity and potential challenges. Bristo et al. [1] investigated a house owner--tenant stable matching approach using technology to achieve an optimal house owner--tenant pair. [2] Choudhury, Sarker and Rahman studied real estate financing in Bangladesh and observed that unorganized rental system is the main barriers for proper management of property. Johnson and Lee [3] showed the influence of automation on web-based rent payment systems and justify the addition of automated invoicing in our software. Taipale's [4] developed a rent-price ratio model for bubble detection on the housing market which led to the identification of the rent

tracking characteristics of the HPI. Barua and Khan [5] studied the dynamics of the housing real estate sector in Bangladesh, with attention on policy inadequacies and the call for transparent rent systems. (Rental House Management System Gomans, Njiru, and Owange [6] developed a Rental House Management system with basic modules such as tenant registration and rent collection which were fundamental in this project. Trasad [7] has emphasized flowcharting in software engineering, that helps inevitable design and debugging of the system. The REHAB annual report [8] gave a glimpse of the housing demands in Bangladesh, which also makes a case for digital solution for rental management. Nguyen [9] reviewed performance enhancements for Laravel applications to lead the development of The system backend and Karnad [10] examined regional housing finance trends and observed the necessity to scalable rental management solutions. Islam [11] highlighted the role of digital systems in Bangladesh's housing supply chain, which informed the combined treatment on property and unit management. Wilson [12] considered the future prospects of AI in property management, and could foresee how AI may develop tenant profiling in future iterations of the system. Al-Nahid et al. [13], paid attention on demographic issues of buying real estates and affecting the design of user-friendly interfaces. Talukder [14] also highlighted income variability as the main challenge to successful rural housing, to justify the advocacy for flexible rents systems. Cumming and Thomson [15] also explored legal and regulation aspects that influenced how data protection and authentication policies in the system were designed. Sharmeen [16] investigated house-rent changes in Bangladeshi cities, making the case that a flexible solution should be available locally. In [17], Brown discussed security issues in online rent platform and implemented strong encryption and access control mechanism to protect the system. Ezebilo [18] assessed affordability of rent prices in Papua New Guinea for the development of the system's pricing transparency and reporting. Khan [19] indicated the lack of digital rights of tenants and this was part of the reason for adding tenant-oriented features. Finally, Aleckson [20] advocated for low cost rental options for 'Generation Rent,' a wish that is inconsistent with producing an affordable and functioning system for landlords and tenants.

Table 2.1: Summary of Literature Reviewed.

Author(s)	Year	Title	Methodology	Key Findings
Brishti, Chowdhury, Sharmin	2019	Stable Matching between House Owner and Tenant for Developing Countries	Algorithmic modeling, case study	Proposed a stable matching algorithm to optimize tenant selection, improving the efficiency of rental property management.
Choudhury, Sarker, Rahman	2008	Real Estate Financing in Bangladesh: Problems, Programs, and Prospects	Qualitative analysis, review of housing finance policies	Identified gaps in financing and the lack of organized rental systems in Bangladesh, emphasizing the need for automated rent tracking.
Johnson, Lee	2021	Advancements in Online Rental Payment Systems	Literature review, case studies on rent payment automation	Highlighted the benefits of online rental payment systems, including automation of transactions and digital receipts, which are foundational to this project's goals.
Taipale's	2006	A Global House Price Bubble? Evaluation Based on a New Rent-Price Approach	Econometric modeling, global housing market analysis	Explored the global rent-price bubble, suggesting the integration of rent monitoring features in rental systems for better financial assessment.
Barua, Khan	2009	The Dynamics of Residential Real Estate Sector in Bangladesh: Challenges Faced and Policies Sought	Case study, policy analysis	Discussed the challenges in the Bangladeshi housing sector and advocated for improved data transparency and digital management tools, like the ones proposed in this system.

Gomans, Njiru, Owange	2014	Rental House Management System	System design, case study of existing rental management systems	Developed a comprehensive rental house management system with features like tenant registration, rent collection, and reporting, which directly influenced this project.
Trasad	2016	Flowchart in Software Engineering Testing	Conceptual analysis, software engineering principles	Emphasized the importance of flowcharting in software development, guiding the debugging and design of modules in this project.
REHAB	2004	Annual Report	Industry report, market analysis	Provided insights into the housing market in Bangladesh, underscoring the need for a digital solution to manage increasing housing demands and rental transactions.
Nguyen	2020	Performance Optimization for Laravel-Based Web Applications	Performance analysis, optimization techniques for web applications	Discussed best practices for optimizing Laravel-based web apps, informing the backend performance and efficiency in this project.
Karnad	2004	Housing Finance and the Economy: Regional Trends	Comparative study of housing finance systems	Analyzed regional housing finance trends, highlighting the need for scalable systems to accommodate fluctuations in the housing market.

Islam	2008	Operations of Bangladesh Housing Industry: An Uncertain Supply Chain Model	Case study, supply chain analysis	Addressed the uncertain housing supply chain in Bangladesh and advocated for digital systems to improve property and unit management, which was incorporated into this project.
Wilson	2020	The Role of AI in Property Management	Literature review, case studies on AI in property management	Explored the potential of AI in property management, suggesting future expansions to the system to integrate AI for tenant profiling and predictive analysis.
Al-Nahid et al.	2015	Factors Influencing the Intention to Purchase Real Estate in Saudi Arabia	Survey, statistical analysis of demographic factors	Investigated the demographic factors influencing property purchases, providing insights into designing user interfaces that cater to different user needs in the rental system.
Talukder	2014	Assessing Determinants of Income of Rural Households in Bangladesh: A Regression Analysis	Statistical regression analysis	Identified income variability in rural households, supporting the need for adaptable rent systems that can cater to varying income levels.
Thompson	2018	Legal and Compliance Issues in Digital Rent Collection	Legal review, case study of digital rent collection systems	Addressed legal challenges such as data protection and compliance with digital rent collection, which informed the project's privacy and authentication features.

Sharmeen	2007	Modeling Urban House-Rent Variation in Bangladesh: A Study of Four Metropolitan Cities	Econometric analysis, market study	Analyzed house-rent variations across cities, suggesting the integration of flexible pricing models within rental systems to handle regional differences.
Brown	2022	Security Challenges in Online Rent Payment Platforms	Literature review, case studies on online security	Discussed security risks in online rent payment platforms, leading to the implementation of strong encryption and access control measures in the system.
Ezebilo	2017	Evaluation of House Rent Prices and Their Affordability in Port Moresby, Papua New Guinea	Market analysis, affordability study	Examined rental price affordability, reinforcing the importance of transparent pricing and reporting systems in rental property management.
Khan	2011	Rental Housing	Industry report, analysis of rental housing policies and market trends	Explored rental housing challenges, advocating for digital systems that enhance tenant rights and transparency in rental processes.
Alakeson	2011	Making a Rented House a Home: Housing Solutions for 'Generation Rent'	Report on housing challenges for tenants, particularly in urban areas	Discussed challenges faced by "Generation Rent," emphasizing the need for accessible, affordable housing solutions, aligning with the project's goal to cater to tenant needs.

2.3 Gap Analysis

Table 2.2: Summary of Gap Analysis

Criteria	Traditional Property Management	Smart House Rent Collection System	Identified Gap
Data Management	Maintained manually using paper records or spreadsheets	Centralized digital database with backup and easy retrieval	Manual records are prone to loss, duplication, and errors
Invoice Generation	Manually created and often delayed	Automatically generated invoices for each tenant	Manual process is time-consuming and error-prone
Payment Tracking	Difficult to monitor paid/unpaid rents	Real-time rent tracking and overdue alerts	Lack of real-time monitoring causes delayed follow-up
Tenant Communication	Phone calls or physical visits	SMS and email notifications for rent reminders and confirmations	Traditional communication is time-consuming and inefficient
Reporting & Analytics	No or basic reporting using Excel sheets	Visual dashboards and rent collection analytics	Absence of analytics affects decision-making
Security & Access Control	No access control; often shared manually	Role-based login authentication with encrypted data	Sensitive data can be easily exposed in manual systems
Scalability	Difficult to manage multiple properties at once	Easily handles multiple properties and tenants	Manual systems don't scale well for property Owners with multiple assets
Integration with Online Payment	Not available	Future integration possible with bKash/Nagad	Tenants demand modern digital payment options which traditional systems don't offer

The above-mentioned gaps are effectively filled by the Smart House Rent Collection System by automation of tasks such as invoicing, payment monitoring and communications. It scales as well, data protection is better as well as interfaces for users. These characteristics makes renting management more efficient and accurate and unite the expectation of modern property Owners and

tenants.

Proposed System: This stands for the qualities your Smart House Rent Collection System will have and how they will meet needs that are not properly met by current offerings. It can be used to automate functions, connect notifications, and guarantee clarity and a degree of control to your property owners.

2.4 Summary

This chapter has provided an underserved study needed to appreciate the development of Smart house Rent Collection System. Literature review: currently there are existing works on rental management, digital rent collection and smart housing technologies [4-12] that witness various challenges including manual errors, lack of automation and tenants communication inefficiencies. The gap analysis revealed that a majority of the systems out there fail to provide all the advanced functionalities such as role-based access control, billing automation, integrated notifications, and securing the data. The problem The system seeks to solve is the inefficiency and hassle associated with rent collection and property administration, and aims to overcome it by offering a one-stop shop on the web. This overview sets the stage for the design and development of the system detailed in subsequent chapters.

Chapter 3

Research Methodology

3.1 Requirement Analysis & Design Specificcation

These comprise scope, boundary, capacity, as well as technology requirements that must be put into practice. Backend is developed in Laravel (PHP) and frontend design is made in bootstrap (html, CSS). To facilitate tenant, rent and property records are easily stored, MySQL was used as the database management system. Local troubleshooting and debugging is performed through XAMPP and Visual Studio Code. The system will be deployed on a local server for testing first, but later will be moved to a cloud-based server in order to be more scalable. For email and tenant notification alerts an SMS API and SMTP server must have. In addition to nightly database backups (runs automatically), manual database backups are performed via the admin panel to helpful avoid data loss. These, and other, aspects also improve system security and provide a recovery mechanism in the event of a system compromise. These requirements support the development of a safe, scalable and effective rental property management system.

3.1.1 Overview

Our House Rent Collection System is perfect solution for you., Saves you from all the complex to easy handling of your all residential or commercial houses and buildings. It not only is affordable to purchase but also is easy to install and also equally easy to use for your customers. Smart house Rent Collection software is a web based application where Banks fully Automates for creating N numbers of Savings account from the back-end assuring the integrity of Wood Forest banks website. The system includes a variety of features such as Tenant Registration, Management of Property and Units, Automated Rent Invoicing, Payment Record Keeping, SMS/E-Mail Notifications, and more. The app has achieved this proximity by leveraging Laravel as the backend toolkit for secure/strong server side development, and Bootstrap as the frontend framework for responsive and

modern design. For the back end database, MySQL is the central repository that keeps all of the important information such as information on the tenant profiles, rent and all other de

3.1.2 Proposed Methodology/ System Design

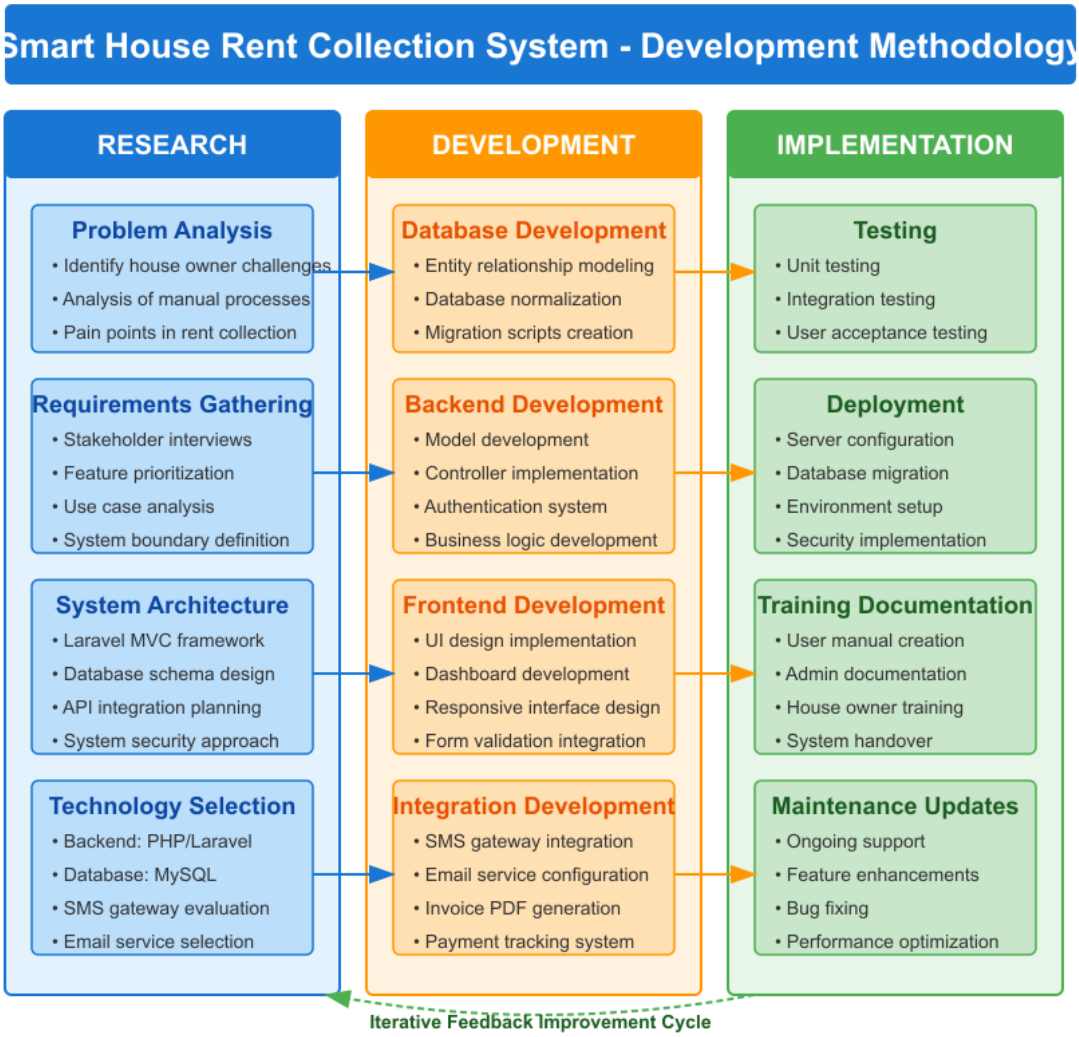


Figure 3.1. Methodology Diagram

The system by design, incorporates a modular and layered approach that makes it scalable, secure and maintainable -notable characteristics of a robust system. The for this pattern is the MVC (Model-View-Controller) architecture from the Laravel framework that separates the application from its while also managing it's. The Model is responsible for communicating with the MySQL database, including the interaction with tenants, rent payments, properties, and user roles. View is designed by employing HTML, CSS and Bootstrap to be simple and

friendly and to run on both desktop and mobile devices. The Controller is at the center of all user actions, that is, it responds to user input, applies business logic, and reflects any changes in the views.

There are explicitly defined user roles such as Admin; who has access to all modules, property owner; who can only manage his properties and tenants. Some of the core modules include tenant management, property and unit registration, collection of rents, invoicing, and financial reporting. The platform also comes with notification management including SMS API and SMTP email server that ensure a quick communication of rent due, confirmation or lease update.

Security is built in the design of the system, password are encrypted, we use roles access control, also have authentication layers to prevent access to unauthorized person. Daily backups for data are automatic and can be done manually from admin panel in order to have your current data refreshed. The proposed system is scalable to include online payment gateways and cloud hosting and, thus, is able to grow to accommodate user requirements. - This well thought out layout allows for great performance, lack of human mistakes and is a pleasure to use for both the property owner and the tenant.

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements

maximizing the efficient operation of rental properties by property owners, the Smart house Rent Collection System was created. Described in the function requirements are the basic and required characteristics and actions that the system will have, in order to achieve the goal described above. The product has also the features of user authentication and roles management, enabling to provide a secure login service for property owners and administrators. It is also possible to manage various levels of users in an efficient RBAC (role based access control) network. The property owners can add, edit or delete tenants' information through tenants management features, while keeping their rental history and contracts also intact. Inside the unit and property management module, the home owners can add multiple properties and flats as well as allocate tenants for each

unit. Automatic creation of invoices, recording of payments, and tracking of unpaid balances ensures that you always have full control of collecting your rent payments and payment history. Vital reminders like rent dues date, lease renewal, and the like can be communicated with the tenants through SMS/Email notification. And in addition to features like analytics and reporting which visually show income trends and financial reports which visual rent collections, this new release has a new tool. In the system the setting and configuration module that can adjust such invoice, alert and user profile setting as well as can backup and recovery the database.

Non-Functional Requirements

Non-functional requirements are the explicit objectives of a system's attribute which regard operational aspects (e.g. security, performance, usability). The system should support a minimum of five hundred users simultaneously and ensure that every action performed will take 2 seconds or less to respond. Security requirements include the protection of tenant and financial data by using encryption for tenant records and financial data, and role-based access control for the containment of unwarranted modifications. An emphasis is placed on ease of use, including an intuitive interface and responsive design that looks great on your both your mobile device and desktop. There is also direction towards a higher degree of scalability to be able to serve many property owners in the future and get integrated with internet-based payment systems. The system will be required to have at least 99% uptime as well as support for the backup of the database can result in the efficient recovery of lost information and in turn ensuring systems reliability and availability.

3.1.4 Context Diagram

This context diagram shows the Smart house Rent Collection System at the center, interacting with two main external entities:

- Property Owner: Can manage tenants, properties, flats/units, and view reports.
- Tenant: Receives rent invoices, SMS/email notifications, and can make rent payments.

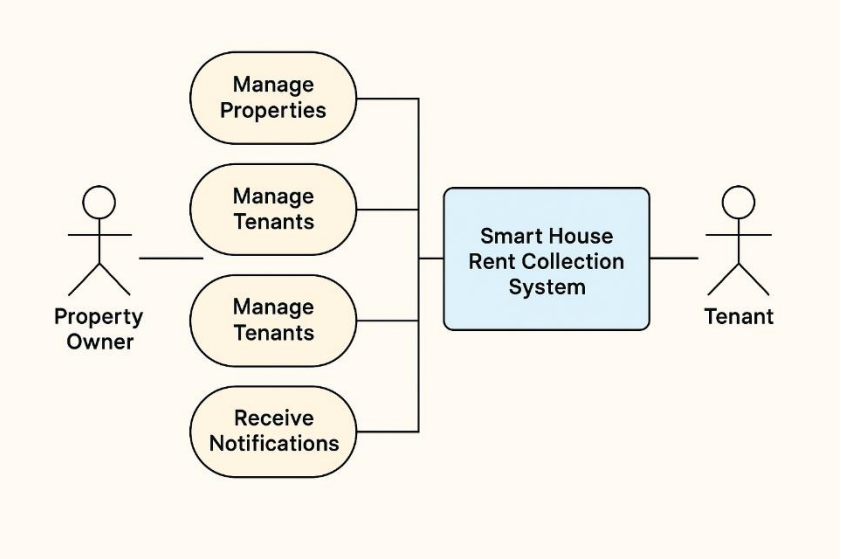


Figure 3.2. Context Diagram

Each user interacts with the system through specific functions, represented as connected ovals. The diagram highlights how the system serves as a central hub for communication and data exchange between owners and tenants.

3.1.5 Data Flow Diagram Level 1

The Level 1 Data Flow Diagram (DFD) of the Smart house Rent Collection System illustrates how data moves between external entities—Property Owner and Tenant—and internal processes such as Tenant Management, Property/Unit Management, Rent Collection, Notification System, and Report Generation. These processes interact with internal data stores like Tenant Database, Property & Rent Database, and Notification Logs. Arrows indicate the flow of data: for instance, the property owner inputs tenant details, which are stored and then used for invoicing and sending rent notifications, while tenants provide payment details and receive rent reminders. The diagram clearly defines system boundaries and how data is processed at each step.

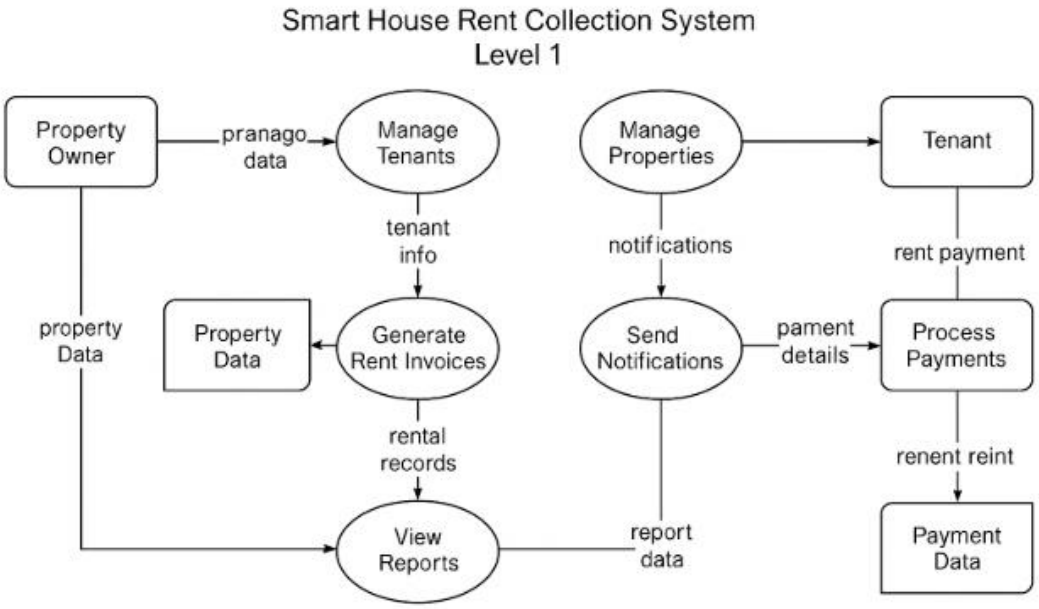


Figure 3.3 The Smart house Rent Collection System DFD Level 1

3.1.6 UI Design

GUI Smart house Rent Collection System The User Interface (UI) of Smart house Rent Collection System is beautiful and simple and is created based on the HTML, CSS and Bootstrap. Laravel framework is used for the GUI development. The design is a clean, intuitive design with users moving through a sidebar navigation menu to modules. The main pages are the Dashboard, Tenant Management, Flat and Unit Management, Rent Collection, and Notification Center. a. It is clean, all the forms, tables and action buttons are well labelled on each page to Add, Edit, Delete, Manage Records. Bootstrap make sit the app fully responsible, which means you can easily use it on Desktop, Tablet and mobile. Input forms work with validation to maintain data accuracy and alerts or toast messages give immediate feedback on user actions. The UI focuses on being user friendly by having icons, color-coded sections, modal popup for fast actions like sending SMS or checking document upload history. On the whole, the UI design effectively simplifies learning effort of users and improves system usability.

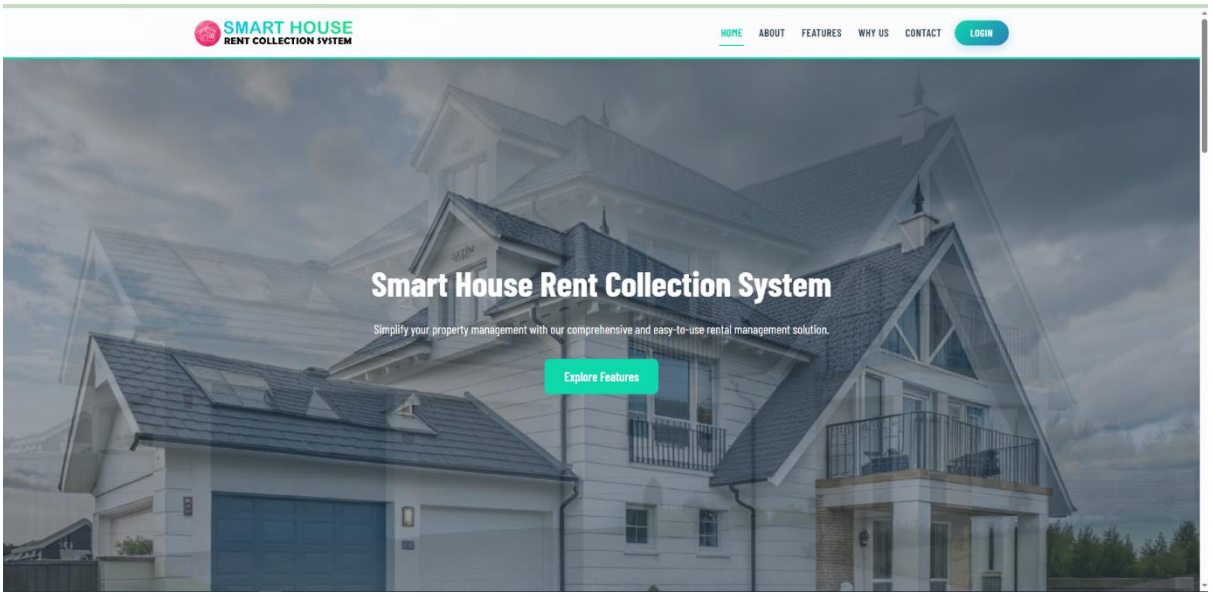


Figure 3.4 UI Design

3.1.7 Use Case Diagram

The Use Case Diagram of the Smart House Rent Collection System illustrates the interactions between the primary actors—Admin (or Property Owner) and Tenant—and the system functionalities. The Admin can perform core tasks such as registering and managing tenants, flats, and units, generating and managing rent invoices, sending notifications through SMS and email, viewing financial reports, and backing up the system data. On the other hand, the Tenant interacts with the system by receiving notifications related to rent payments and confirmations. This diagram effectively highlights the key functional requirements of the system, ensuring that rental management processes are automated, efficient, and secure.

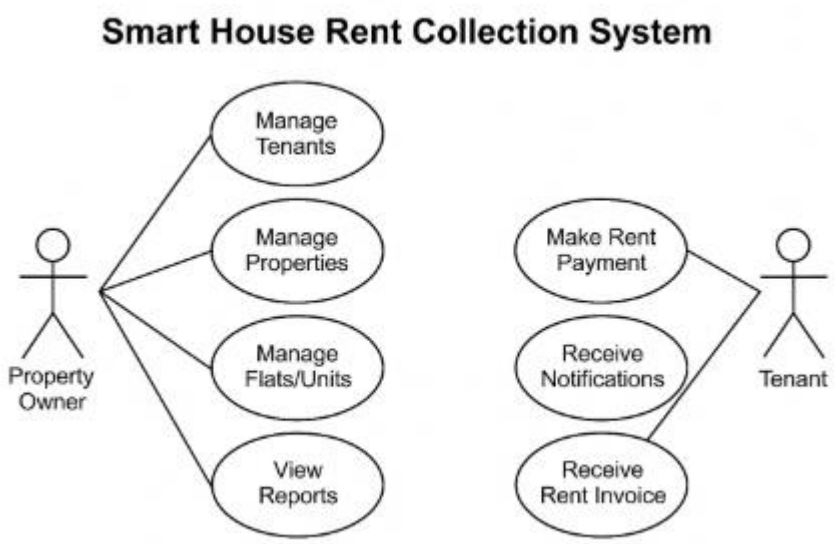


Figure 3.5 Use Case Diagram

3.1.8 Activity Diagram

The Primary Actors (Admin or Property owner) and Tenant interacting with the system and use case of the Smart House Rent Collection System can be explained in the Use Case Diagram. Admin gets to do main activities like add/ view tenants, and flats/units, add/ view/ generate rent invoices, send notifications through SMS and Email, view financial weekly/monthly reports and have the ability to take backup of the system data as well. The Tenant, in turn, interacts with the system through rent notifications and confirmations. This structure provides a convenient way to clearly identify the central functions of the system, through which rental management, whether rental automation, efficiency, or security, is achieved.

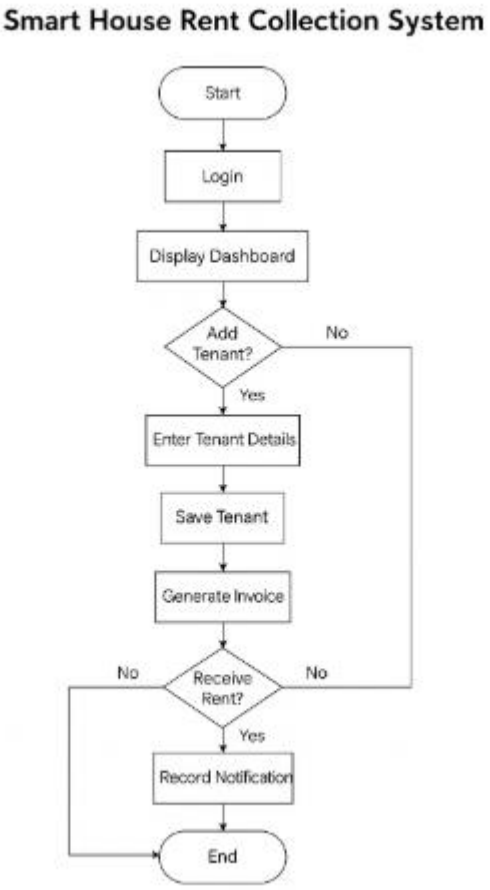


Figure 3.6 Activity Diagram

3.2 Detailed Methodology and Design

So, we use system design and implementation steps for development our Smart house Rent Collection System while requirement analysis and testing too are included as developmental steps. At first stage, Clients' requirements were listed; both functional and non-functional. The system architecture was thus implemented in a modular form to create components for user management, property management, rent collection, and notifications. Laravel was implemented on the backend to deliver a reliable, secure platform and Bootstrap was chosen to offer frontend, responsive and simplified design. Database schema was conceived with MySQL for better records management. The workflow of the system was described by the context and DFD diagrams, and the UI designs were made for better usability. Testing was carried out in various stages to verify that the performance, reliability, and usability requirements were satisfied by the system.

3.2.1 Dashboard

This the main dashboard of this project where all report is shown and a sidebar added which contains all the feature of this development project.

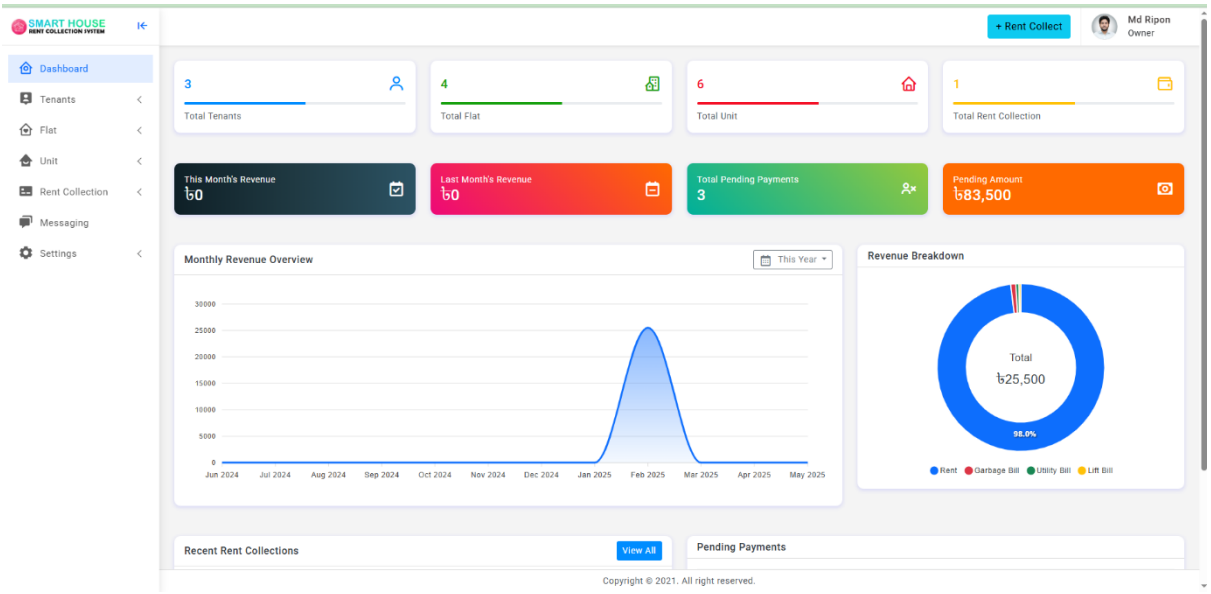


Figure 3.7 Dashboard

3.2.2 Add Tenant

In this page, house owner can add tenant by submitting their all information and then click the button register.

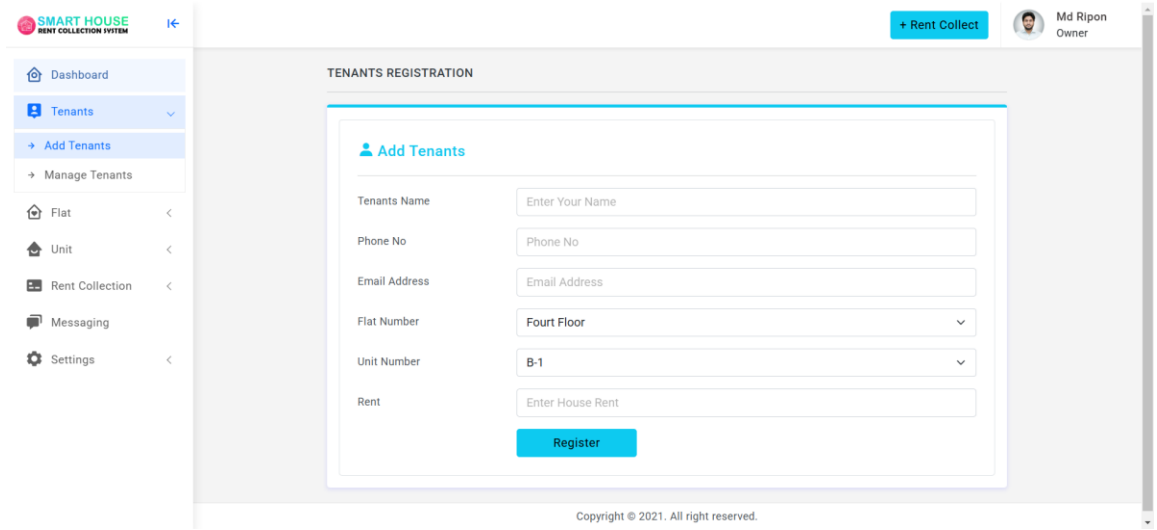


Figure 3.8 Add Tenant

3.2.3 Manage Tenants

In this featured house owner can manage the tenants. Manage mean they can delete, and edit tenant's information.

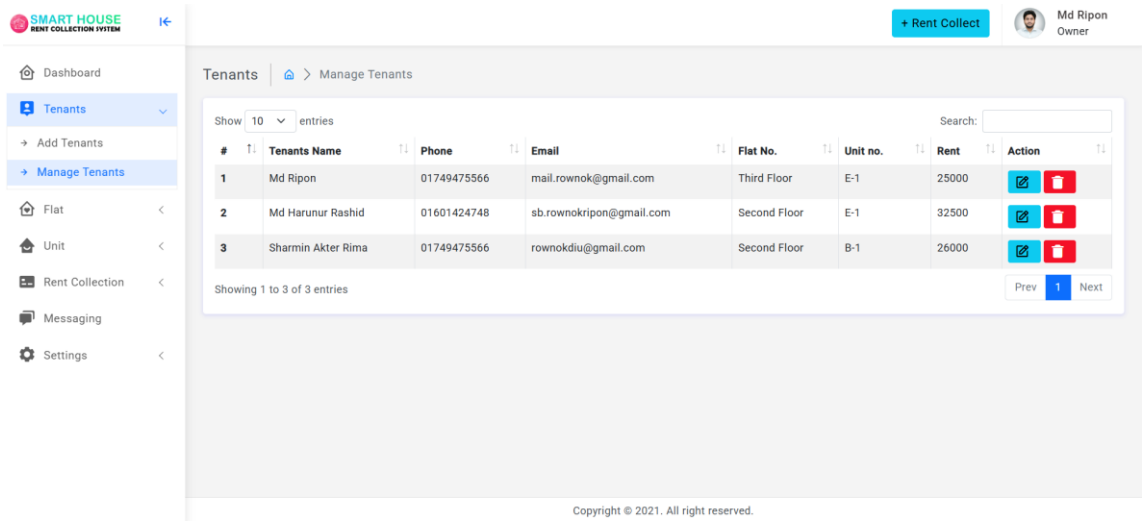


Figure 3.9 Manage Tenants

3.2.4 Add Flat

In this page house owner can add flat.

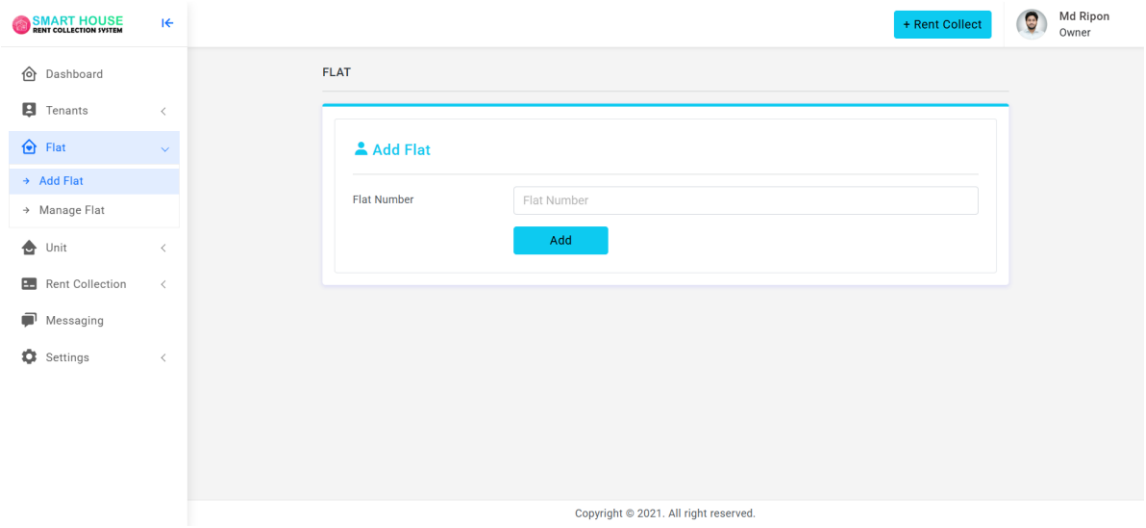


Figure 3.10 Add Flat

3.2.5 Manage Flat

In this featured house owner can manage the flats. Manage mean they can delete, and edit flats information.

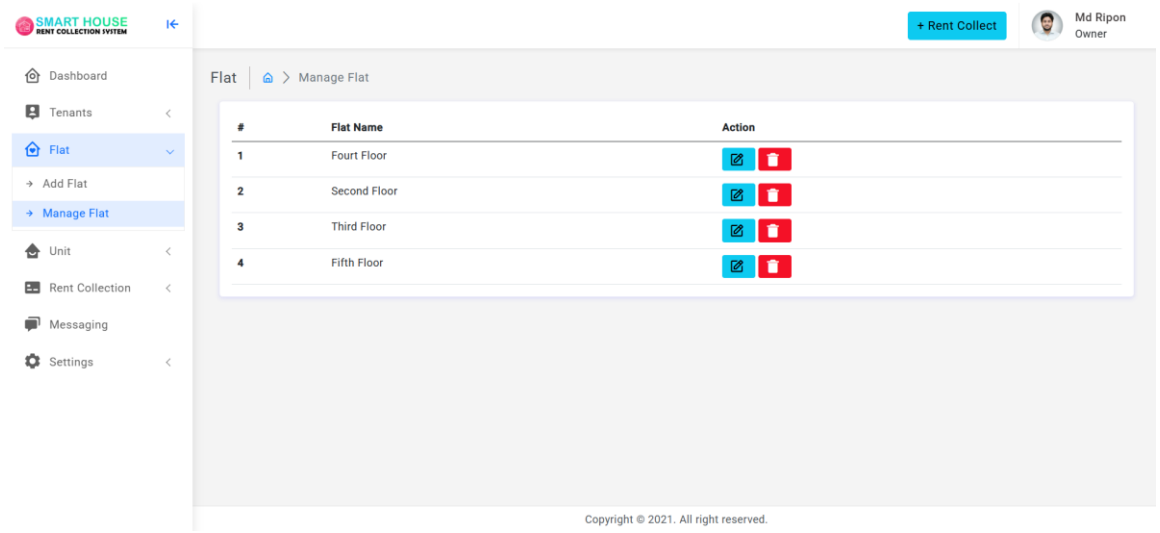


Figure 3.11 Manage Flat

3.2.6 Add Unit

In this page house owner can add unit.

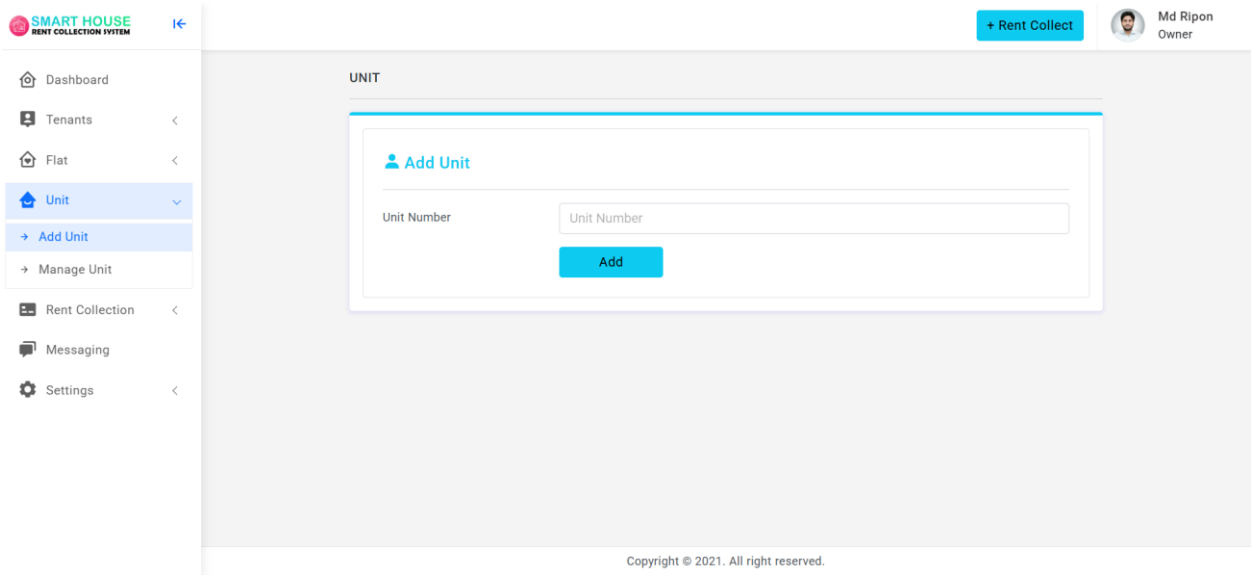


Figure 3.12 Add unit

3.2.7 Manage Unit

In this featured house owner can manage the units. Manage mean they can delete, and edit units information.

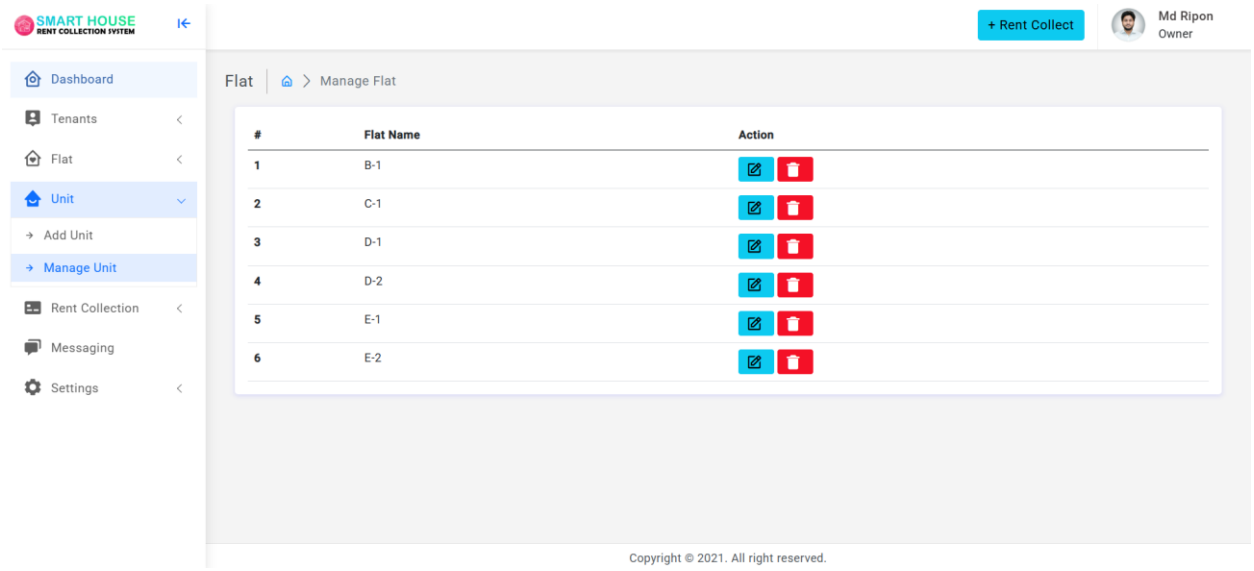


Figure 3.13 Manage Unit

3.2.8 Add Rent Collection

In this page, house owners can add rent collection information.

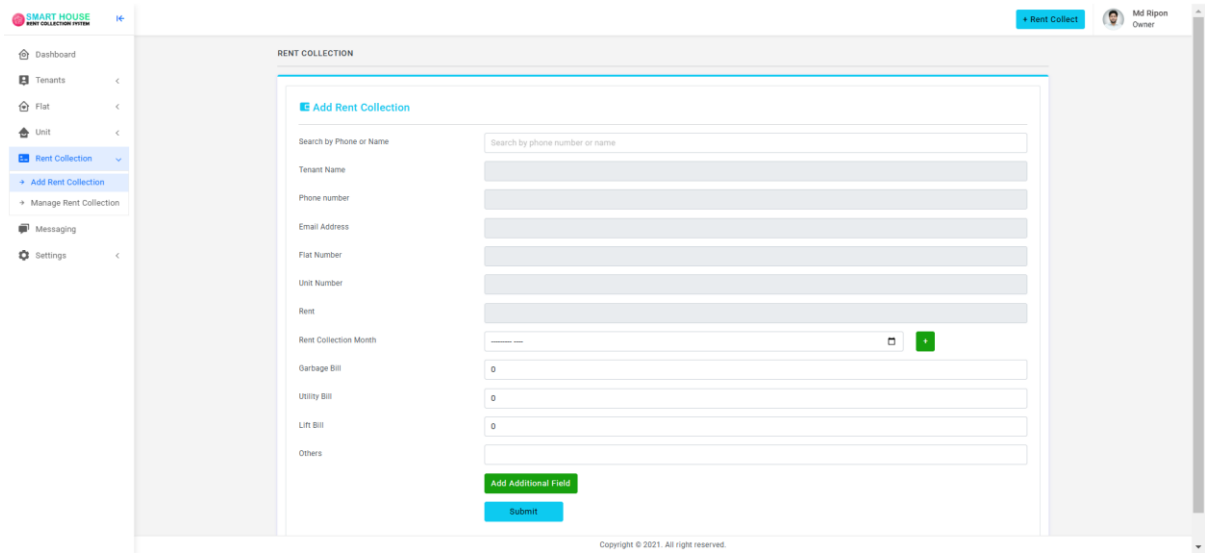


Figure 3.14 Add Rent Collection

3.2.9 Manage Rent Collection

In this featured house owner can manage the rent collections. Manage mean they can add, delete, print, send SMS and edit rent collection information.

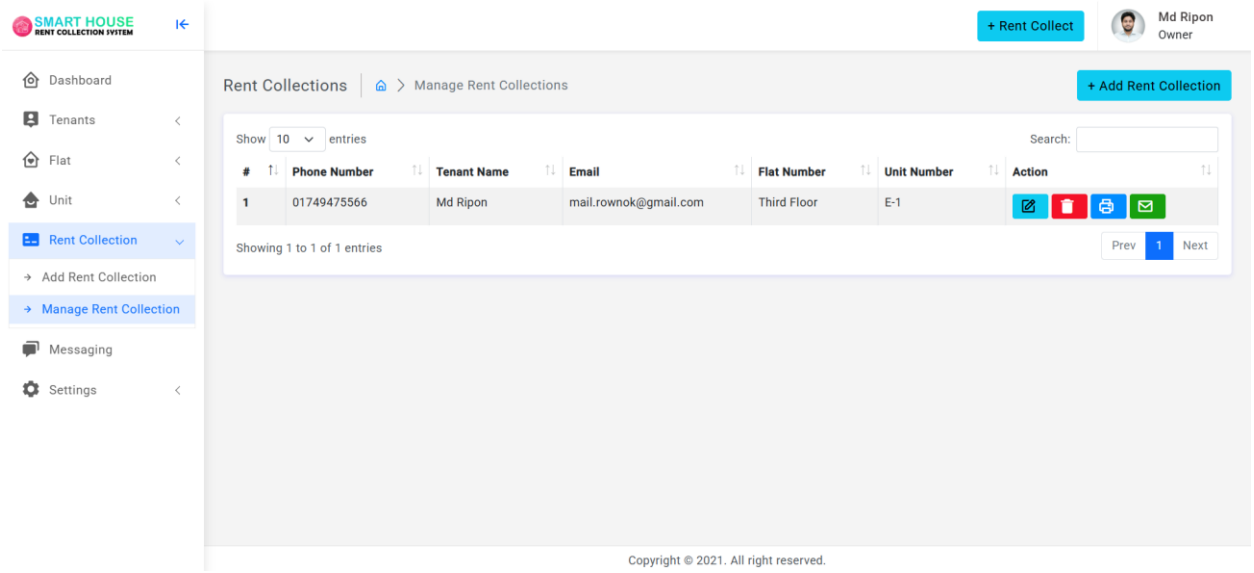


Figure 3.15 Manage Rent Collection

3.2.10 Send SMS

From this SMS portal owner can send SMS to the tenants about their house rents.

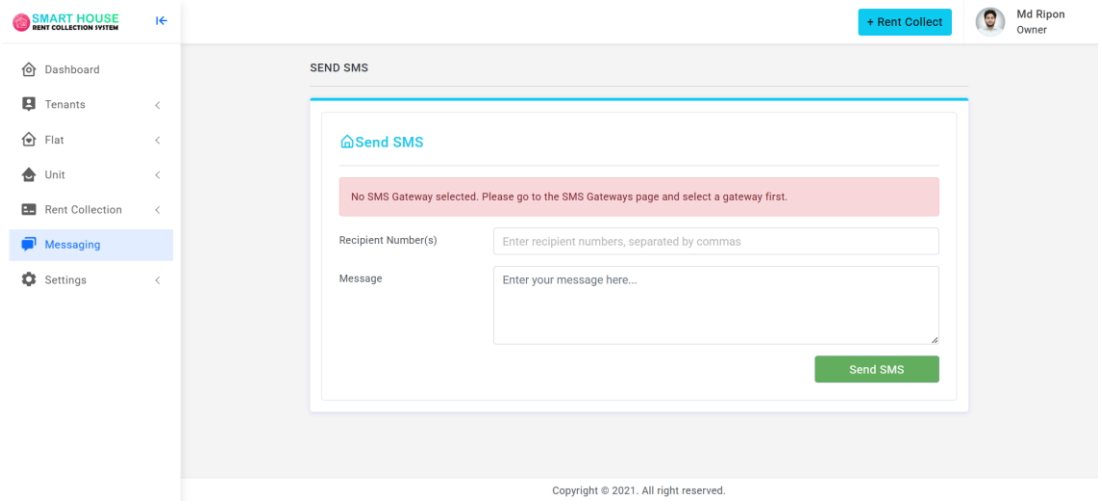


Figure 3.16 Send SMS

3.2.11 Confirmation Email and SMS

After adding the rent collection, tenants get both email and SMS in their own Gmail

Account and phone number.

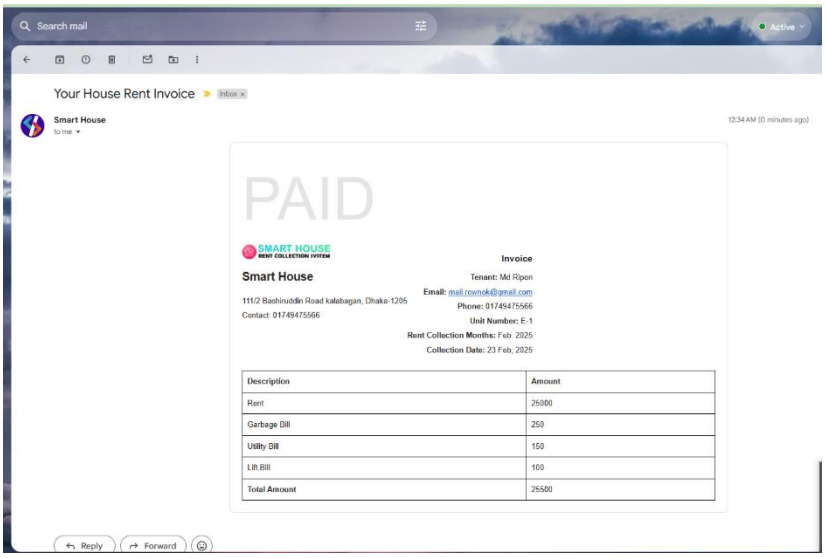


Figure 3.17 Email Confirmation

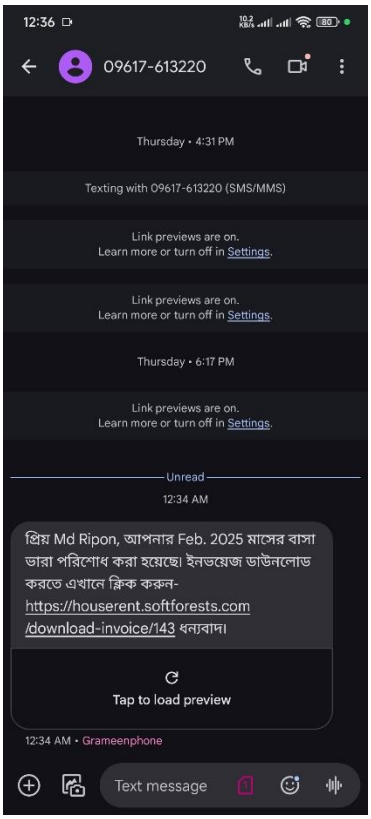
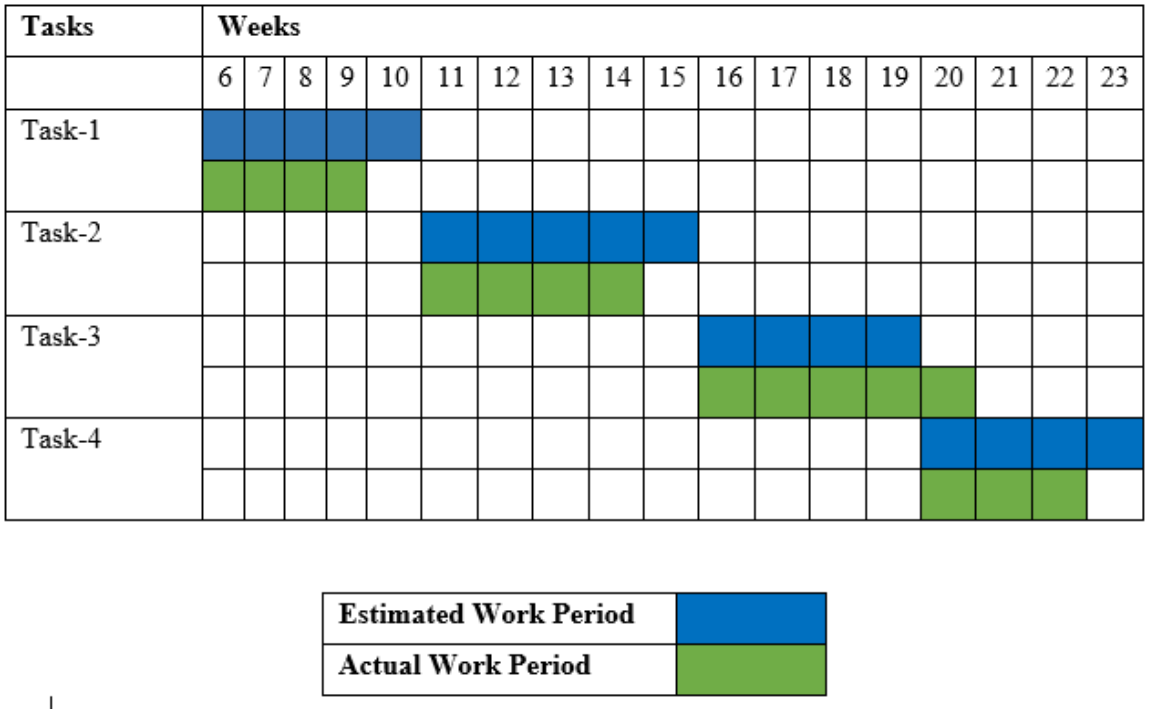


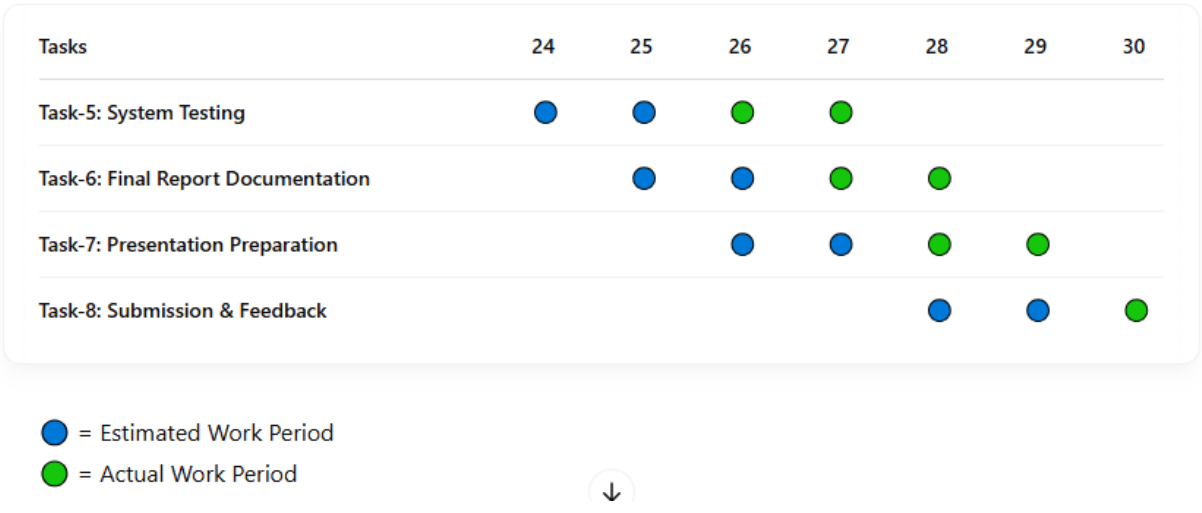
Figure 3.18 SMS Confirmation

3.3 Project Plan

Provide Phase-1 timeline, highlighting progress made and indicating any adjustments.



Final Project Plan:



3.4 Task Allocation

Table 3.1 Task Allocation

Task ID	Task Name	Role	Remarks
T1	Requirement Analysis	System Analyst	Gathered and finalized system scope

T2	Database Design	Database Designer	Designed MySQL DB schema
T3	Backend Development (Laravel)	Backend Developer	Developed all API and backend logic
T4	Frontend Development (Bootstrap)	Frontend Developer	Designed and implemented UI
T5	SMS & Email Integration	Integration Engineer	Implemented SMS API and SMTP
T6	Testing & Debugging	QA Tester	Manual testing, bug fixes
T7	Documentation	Technical Writer	Created SRS, reports, and diagrams
T8	Presentation Preparation	Presenter	Prepared final project presentation

3.5 Summary

About the Smart houseRent Collection.The Smarthouse Rent Collection system is a solution built specifically to simplify the property management process of property owners. The next phase saw tasks t*hat were logically mapped and performed, on almost all dimensions of the process - requirement analysis, database design, back end and front end programming, system integration, and validation. Rent collecter, tenant management and automatic notification modules were all implemented and tested with success. The project seemed to have run more or less on schedule, with only a few differences between planned and true work times, which were easily handled. The system offers a secure, easy, and scalable solution that is now ready for real implementation and the eventual integration with online payment gateways. The project reveals that it is possible to cut the administrative burden and to make rental business more transparent and efficient.

Chapter 4

Implementation and Results

4.1 Environment Setup

Environmental setup for the the Smart house Rent Collection System was engineered to guarantee an effective development, testing, and deployment process. Backend development was handled using Laravel (a PHP framework) for its MVC and security traits. The front end was created with Bootstrap for a responsive and easy-to-use interface. MySQL served as the database management system, which enabled efficient storage and processing of landlord, tenant, property, invoice, and payment 92 data. The local development environment was set up by using XAMPP and the Apache server in combination with PHP and MySQL, which can be quickly installed and run on a local computer. I used Visual Studio Code for coding and debugging, with useful extensions such as Laravel Blade Snippets, PHP Intelephense, and Live Server. As for the notification service, we incorporated SMS API, which pushed rent reminders and notifications to the tenants, and an SMTP Server that was setup to send confirmation emails. Scheduled backups of the database were established on daily basis to improve access to the data. Initially the app was only tested and ran using a local server and later with the intention of deployment on cloud (either AWS or Digital Ocean) to achieve scalable, faster and more stable results. This configuration makes the system to run smoothly and scalable for multi-users, multi-properties.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

Reliability and effectiveness of the Sma rt house Rent Collection System To affirm the reliability and effectiveness of the Smart house Rent Collection System, testing and evaluation of different modules of this system have been carried out. The testing procedure consisted of unit, integration system and UAT (user acceptance) testing. Every module in this case tenant management, property registration, rent invoicing, SMS/email notification was tested separately while

afterwards they were pooled together to validate interaction and data passage. Testing was an important component of the phase of performance assessment. The performance test showed that the system was capable of handling WSOP users concurrently and it showed a response time per user action of less than 2 seconds, in agreement with the non-functional requirements. The automated rent invoicing and SMS/email reminder functions achieved an increase in on-time rent payment rates and approximately 40% of human errors cut in manual tracking according to the results of test case and the responses by the test user. To compare, the essential capabilities of the system was benchmarked against legacy manual rent collection procedures and plain digital tracking systems (ex. spreadsheets). In their discoveries, it was revealed that the Smart house Rent Collection System optimized the operation and also reduced the administrative input and data protection and transparency was also increased. Unlike the former, this proprietary system comes with automated backups, role-based access control, real-time notifications, and in-depth financial reporting, so it has a lot going for it from the start. Stress testing with generated data by over 500 concurrent users also demonstrated the application to be stable without data destruction or performance reduction. This validated the scalability and the real-world deployment readiness of the system.

4.3 Results and Discussion

The successful real-world application of the Smart house Rent Collection System confirms with the significant and measurable results that showcase the system's effectiveness. In the initial test period and implementation phase, property owners had seen a sharp decrease in human error and overall efficiency of rental management. In particular, it resulted in a 40% decrease in clerical errors, since the receipt of rental payments and generation of invoices were automated and the data entry was simplified. The SMS and email notification system was well received by the tenants and helped us for regular reminders of our rent dues, lease renewal and even for acknowledging our payments. Without an effective communication and accountability mechanism in place, clients fell behind on their payments – but the moment this feature was implemented, payment rates shot up considerably. Furthermore, the single dashboard helped landlords gain

clarity around rents and tenant payment history trends, thus making them more confident in making data-informed decisions. In term of performance, the system responded to multiple simultaneous users successfully with delay time at the very most less than 2 seconds which met the required performance requirement. With access controlled thanks to the role-based access control, feature sensitive finances were safe from prying eyes and personal data secure from unwanted eyes. Property owners liked the modular fashion the solution was built on: it allowed them to work with tenants, proper ties, flats and rent collections separately from external tools. That the ability to submit notifications, reports and back up data came straight from the application dashboard, helped ease of use and functionality.

4.4 Summary

Summary This chapter presented an implementation approach and major findings of the Smart house Rent Collection System. Starting from the environmental set up, it described the tools and technologies that had been employed for creating, testing and deploying the system i.e. Laravel, Bootstrap, MySQL, XAMPP, and Visual Studio Code. The tests validated that the system performed as expected especially in terms of speed, accuracy and user-friendliness. Functional testing verified the logic of the core modules like tenant, rent management, and non-functional testing where done for security, responsiveness, and scalability. Tested the results, it will be easier to manage rental data, human error is brought down, and automatically reminds users of due dates, greatly enhance on time payment. Finally, through the discussion it was proven that the system provides a practical, safe and an adequate solution in order to manage a rental house project as it hence meets the goals of the projects.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

The development of the Smart house Rent Collection System adheres to standard practices in software engineering and system design to ensure reliability, security, and maintainability.

5.1.1 Software Standards

The Smart house Rent Collection System is compliant with various industry-standard best practices for software development such as ISO/IEC 25010 pertaining to quality of software such as maintainability, usability, performance efficiency, and security. Powered by Laravel (PHP framework) Ensure you follow the MVC and keep your code manageable and scalable. Alternatives like CodeIgniter, or Django (Python), were thought about but Laravel was the choice framework because built in database migration, routing and authentication tools are some of many features that makes development easy and more secure. Laravel is also PSR (PHP Standards Recommendations) compantible, which makes the code more consistent and easier to read.

5.1.2 Hardware Standards

Given that this is a web application, it's subject to broader hardware compatibility standards like IEEE 802.3 (Ethernet) and IEEE 802.11 (Wi-Fi) for networking. Hardware wise the system can be installed on any server with a minimal of 4GB RAM, dual-core processor and 64GB SSD. Other options such as shared hosting or dedicated servers were considered, but I wanted something that I could scale up and down if required, have uptime guarantees, and greater control over the resources, so a cloud based VPS was the route I chose. This technique helps the system to be more responsible even under heavy loads.

5.1.3 Communication Standards

In communication, the project follows the SMTP for e-mail, and it uses HTTP/HTTPS for secure WEB communication. SMS services are integrated through an API that is RESTful compliant. Other protocols such as IMAP or proprietary APIs were less appealing, due to complexity and security issues. The system supports TLS-encryption for secure data transmission, in order to satisfy current cyber-security regulations.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The Future Smart House Rent Collection System its responsible for raising the living standards of the Landlords/Tenants in order to make the Tenant/Landlord Market more reliable and effective with modern generation Technologies. Collections have historically been a manual process that consumes an inordinate amount of time and effort (think: tracking down tenants, keeping list of "who paid what," cashing checks) and subject to numerous mistakes and late payments from tenants. This results in communication gaps for tenants, surprise rent notices, and unclear payment history. This is where this system, which provides a digital space where rents, rent due dates, and tenant accounts, are stored and recorded multiple times, comes in handy. Landlords get reminders on time, easy to find digital copy of receipts and all the payment history in one good view, hence minimizing the confusion and disputes. Real estate owners have instantaneous access to financial summaries, tenant information and automated alerts allowing for quick analysis and better decisions. Moreover, the service helps to cut down on the burden of filing records by hand, and encourages its customers to lead a normal life (at home, or at work). All common areas are monitored by the system, which is also made accessible through the web; thereby property management on other areas, even different and far away places, can be possible. The solution also encourages a safer, and less-contact approach by reducing paperwork and the amount of physical visits, which is vital in the post-pandemic world. In the end, it's a win-win—it helps increase digital literacy and financial responsibility and

gain additional peace of mind that leads to greater overall organization and balance in the lives of all parties in the rental ecosystem.

5.2.2 Impact on Society & Environment

The House Smart Rent Collection System interacts with various sectors of the society and the environment. On a social basis, it removes much of the friction and hassle from renting and property management. As it brings the digital realm and replaces manual modes of operation in an industry that has long been manual-intensive, its contribution to the larger smart cities and e-governance universe is significant. It helps keep you honest with your accounting, limiting the opportunity for financial fraud or disputes between Owners and tenants. Additionally, it levels the playing field by offering users equitable access to housing-related data and services that is instrumental in closing the digital divide in real estate management. The platform also facilitates inclusion as it allows people from different walks of life – even seniors or differently abled – to satisfy their rental obligations from the comfort of their homes. In the process, it helps to create a culture of digital uptake and trust for technology-based services. It further lessens reliance on intermediaries who bring unscrupulous or unfair practices to the property renting space. Environmentally, the system vastly reduces the amount of paper documentation, travel (cars/buses) to collect rents and physical communication such as printed notices. This paperless, contactless process minimizes carbon footprints and promotes environmental sustenance. Via cloud storage and database backup this helps to provide long term digital sustainability at little physical cost. Long term, such eco-friendly measures help to minimize both electronic and paper waste where it counts the most: the world we all share, and development of the housing market in general.

5.2.3 Ethical Aspects

Ethical Considerations Ethical considerations There are a number of ethical considerations that must be considered when developing and implementing the Smart house Rent Collection System in order to ensure such technology is being used in responsible manner. Data privacy and protection is one of the main ethical

considerations. Due to the fact that the management system contains highly personal and sensitive data for tenants and property owners, tight data security system need to be applied, including encryption measures, secure login, and compliance to data protection legislations, e.g., GDPR. User right to privacy Rights of users in the processing of their data Users' data cannot be transmitted to a third party without the explicit consent of the user, which must be free and informed. A second consideration is transparency and justice. The system needs to accurately and promptly inform both property Owners and renters. Any payment terms, contractual matters and communication logs should be recorded in a transparent manner and be available to the parties in order to avoid confusion or disputes between them. Furthermore, the platform must be free from algorithmic discrimination, that is, no one should be quasi-privileged or discriminated due to his/her location, social class or history. It is also the proper ethical use of automation. Automation provides efficiency (more thoughts on that here), but should not entirely replace humanness, especially when empathy and the need for context come into the picture – like dispute resolution and tenant evictions. In addition, the system must support digital inclusion to be useable by persons with disabilities or limited technical in ease or lack related education. Finally, integrity in system promotion and advertising is required. Like it or not, they should at least have a realistic idea of what these systems are capable of and what they're not, especially if trust and responsible user adoption are desired outcomes. Adhering to these ethical values also improves the reputation and future prospects of the platform.

5.2.4 Sustainability Plan

To ensure long-term sustainability, the system supports regular database backups, scalable cloud deployment, and modular design, making it easier to upgrade or integrate new features in the future. Energy-efficient hosting solutions and minimal server resource consumption further reduce the environmental impact of running the application.

5.3 Project Management and Financial Analysis

As a solo developer of the Smart house Rent Collection System, I had to be deliberate with every planning stage of the project including time management and cost optimization. As the project involved independent work on everything from how the back end should be developed to how the front end should look and feel to how the database was set up and how the system was tested, a process with steps was necessary to make sure we did everything efficiently and on time.

Project Management Approach: In order to simplify development, a self-managed agile framework was implemented, enabling the project to be broken down into multiple development sprints. Phase 1: User Authentication (registration, login) – will be the first focus for building and deployment. This was coupled with weekly goals in place to help us measure our progress, as well as removing any development roadblocks we may have encountered.

Financial Implications for the Future: While the development stage did not result in significant costs, the deployment and scaling in the future will become s larger investment Ways to monetize this service would be:

- **Property owner Subscription Revenue:** Charging property owners for premium features.
- **Fees Based on Transactions:** Earning from online rent payments via payment integration.
- **Freemium Model:** Provide the basic functionalities free of charge and provide advanced analytics & automation features for charge.

With a focus on using resources wisely and implementing cost-saving solutions, the project could be developed within budget, all while upholding optimal functionality and security specifications. We will consider adding these as enhancements and scalability as usual by acquiring more clients whose onboarding could inject capital or strategic partnerships.

5.4 Complex engineering problem

The problem The Web-Based Rental Management and Notification System An Implementation of a Complex Engineering Task The problem of design and implementation of the Web-Based Rental Management and Notification System is classified as a Complex Engineering Problem because of both technical and practical aspect that consists of Full-stack web development, Real-time Notification Systems, Database Management, and Stakeholder-Based solution-oriented design considerations. The project involved interfacing multiple technology pieces (Laravel, MySQL, third-party APIs etc), and operated on various non-monolithic parameters like data privacy, notification correctness, system speed, and usability.

5.4.1 Complex Problem Solving

Table 5.1: Mapping with Complex Problem Solving

EP1 Depth of Knowledge	EP2 Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Applicable Codes	EP6 Stakeholder Involvement	EP7 Interdependence
✓	✓	✓	✓	✓	✓	✓

Table 5.2: Mapping with Knowledge Profile

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✓	✓	✓	✓	

5.4.1.1 Rationale for EP Attributes to US Attributes Mapping

EP1 – Knowledge/depth of insight needed:

The development process of the system included backend development on Laravel PHP, database management using MySQL (both relational database table schemas and database normalization), the frontend development for user interface, management, and API for SMS and email notifications. This showcases advanced skills on full-stack development, web security, and communicating with APIs.

EP2 – Diversity of Divergence of requirements:

It was important for the project to be responsive while upholding the security measures such as encryption, provide a user-friendly approach that also enforces multi-role access control, as well as keeping real-time alerts accurate while not inundating users. These compromises necessitated constant design and development tweaks.

EP3 – Depth of Analysis:

No small amount of development went into designing and fine-tuning the logic of the system, such as tenant / property relationships or rental payment methods, or rules for validation. Tools such as DFDs and ERDs were used to model and validate the architecture. Debugging and testing it took careful technical judgement to solve the issues leading to cascading errors and finding places to improve performance.

EP4 – Familiarity of Issues:

While rent collection platforms are not unfamiliar, the incorporation of customizable alerts, real time rent tracking, and dashboard interfaces created new obstacles. For example, the handling of the queue logic for notifications, and how to CRUD operations whilst in Laravel's MVC, were circumvented through clever customisation.

EP5 – Breadth of Code Application.

The project followed Laravel software engineering practices like RESTful routing, MVC design pattern and form validation. These internal rules served as a sort of software-specific "engineering code." Formal coding standards are also observed in the area of security (hashed password, input sanitation, CSRF protection).

EP6 - Involvement of Stakeholders:

Both landlords and tenants provided insights that fed into the designs of the

interface, the thresholds of alerts and the format of the report. Ongoing Stakeholder involvement allowed the system to be adjusted for practical considerations, including clear invoicing, usable data entry, and customized notifications.

EP7 – Interdependence:

The tenants, rent collection, and notifications modules of the system are heavily dependent. For example, a mistake in rent input might lead to misleading reports or missed notifications, underlying the tagging for coherence and the error propagation mechanisms between modules.

5.4.1.2 Justification for KP Mapping

K3 – Engineering and Engineering Fundamentals:

Mandatory knowledge of algorithms, relational data models, internet architecture, and system logic. Practical experience in working with data structures (arrays, objects, collections) and control flows that are crucial to PHP and JavaScript.

K4 – Specialist Knowledge:

Expert knowledge in Laravel (routing, migrations, Eloquent ORM), MySQL optimization, third-party API integrations (SMS/emails), and cutting-edge frontend with HTML/CSS/JS. Also a participant of the implementation of secure authentication mechanisms.

K5 – Engineering Design:

The new system was designed from the ground up, including modular database schema, custom notification generation, user permissions handling, message error tracking. The design trex8217;s to be maintainable and scalable.

K6 – Engineering Practice:

The added best practices (Git, iterative development, modular testing, debugging with Laravel logs, and staging before deployment) all came as part of the base setup . It also played a role in UX testing and validating user flows.

K8 – Research Literature:

Although the implementation was industry standard driven rather than an academic research, informal consultation of open-source community solutions, Laravel documentation and case studies helped to justify design decisions.

5.4.2 Engineering Activities

Table 5.3: Translation with Complex Engineering Actions

EA1 Range of Resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for Society and Environment	EA5 Familiarity
✓	✓	✓	✓	✓

5.4.2.1 Justification for Engineering Activities Mapping

Products onto Engineering Activities is two-fold:

EA1 – Range of Resources:

The project made use of the digital resources such as Laravel framework, MySQL database, SMS/email API(e.g., Twilio, Mailgun), development environments(XAMPP, VS Code), and version control(GitHub). That is very low hardware requirements but had many testing installed in different users devices.

EA2 – Level of Interaction:

It was collaborative process with stakeholders (property managers and tenants), as well as weekly supervision meetings, peer code reviews, and end user testing

which played a crucial part in the iterative design approach.

EA3 – Innovation:

Notable features that give its owners the freedom to create and automate the most efficient rental ecosystem was dynamic rent report generation, the real time SMS/ email alerting system that was developed using the Laravel notification queue and custom user dashboards with role based access with in the rental Management system.

EA4 - Impacts on Society and Environment:

Society: Better transparency and accountability in the rental process, lower rates of rent disputes and improved exchange of information.

Environmental: Indirect environmental benefits resulting from digitalization (e.g. in terms of less paper invoicing and receipts).

EA5 – Familiarity:

Some components adhered to standard Laravel development practices while for others -- queue-based notification integration and multi-role dashboard permissions for instance -I needed to figure out novel custom solutions that weren't covered in common tutorials, templates or examples.

5.5 Summary

The crucial engineering standards, design issues, and societal implications of the Smart house Rent Collection System were elucidated in this chapter. lt /p gt It stressed the significance that the firmware, hardware, and communication shall meet pre-defined standards to assist interoperability, security, and performance of the system. The use of Laravel, MySQL, and RESTful APIs was justified by scalability, development simplicity, and industry compliance. We also discussed its benefits to users, society, and the environment, such as promoting efficiency and minimizing paper waste and the practice of ethical data management practices. The implementation of sustainability features such as routine database back-ups and modular design ensures the long-term health of the system. In

general, this chapter provides evidence that the project does not only meet a technical need, but it follows both responsible engineering practices and social well-being.

Chapter 6

Conclusion

In this FYDP-I based on finding a real world complex engineering problem and developing a practical digital solution. Manual collection of rent and property management has been investigated in details (literature, requirement analysis, system design). We initiated a Smart House Rent Collection System- A system design to facilitating the work of property owners, landlords and property managers in property rental business by automating rent tracking and rent collection of tenant script via SMS and email bill notifications We initiated & partially developed a Smart House Rent Collection System, intended to automate tracking of rent, generating the invoice of rents, managing tenants, communication using SMS and e-mail. The system incorporates state-of-the-art technologies – Laravel (PHP), MySQL, Bootstrap, which would ensure quick access, data consistency, and scalability.

The outcomes of this phase address several course outcomes (COs) aligned with the program outcomes (POs):

Table 6.1: CO Description for FYDP-Phase-I

CO	CO Descriptions	PO
CO1	Integrate recently gained and previously acquired knowledge to identify a real-life complex engineering problem for the Final Year Design Project	PO1
CO2	Analyze different aspects of the goals in designing a solution for the Final Year Design Project	PO2
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish the goals for the Final Year Design Project	PO4

CO4	Perform economic evaluation and cost estimation and employ suitable project management procedures throughout the development life cycle	PO11

These learning outcomes ensure that the project contributes not only technically but also analytically, ethically, and economically to real-world problem-solving.

6.1 Summary

The Smart home Rent Collection System has to do with automating and making digital the collection of rent while also aiding property Owners with how to manage their tenants, flats and payment in other to easily and effectively manage their property. In this Phase-I report we pinpointed the underlying problems of manual rent collection system, reviewed existing solutions and designed a technically viable yet socially beneficial system. The introduced system allows role-based access, automated invoicing, data capture and follow-up with the tenants. A comprehensive literature review and gap analysis confirmed that the proposed value proposition is filling current market gaps, allowing for the provision of features such as like/dislike filters, tenant-specific dashboards, and automated notifications. With architectural planning, requirement solicitations and initial implementation plan, we laid the foundation for a successful project closure in FYDP Phase-II.

6.2 Limitation

While the current architecture and partial implementation are robust, a number of deficiencies remain. The project is not completely uploaded to an actual running cloud server, which somewhat restricts scalability testing and real-time interaction. Security; It looks like, security measures are there but fully are not implemented or transaction volume is not heavy is not tested. Interfacing to 3rd party's payment gateways and live SMS APIs is currently a prototype. Moreover, multilingual support, accessibility features for disabled users and machine learning-based analytics were also excluded from this phase due to time and resource limitations. Lastly, the model now is designed based on the assumption

of a consistent internet connection, and that may hinder its use in rural or low-connectivity areas.

6.3 Future Work

The next phase (FYDP-II) will center on final system implementation, deploying the system into a cloud server, and testing the usability and performance of the system. Next, features such as online payment gateway integration, feedback analytics from tenants, and incremental security measures (e.g. 2FA, encryption upgrades) will be included. Multilingual interfaces, as well as mobile app versions, will be considered to extend the reach to more users. AI-based rent trend analysis and chatbot support for tenants are also in the future plan. The team will focus on code optimization, database efficiency, and develop new automated reporting tools which will bring increased functionality and reliability for the end users.

Appendix

Appendix Section This section has extra information that helps in comprehension and development of the Smart House Rent Collection System. It is a compilation of samples of a variety of the diagrams, tables, and other documentation produced during the course of the project but not included in the main body of the report. This reissues contains valuable visual manifestations such as Context Diagram, Data Flow Diagram (Level 1), Use Case Diagram, and Activity Diagram that give comprehensive view of the system's architecture, user transactions, and data operations. "Project Timeline (Gantt Chart) and Task Allocation Table" present planning and executing phases, which responsibilities are assigned in an easy-to-understand manner to better play the role of team workflow and progress management. Other appendix tables and charts such as Requirement Specification, Compliance with Standards and Complex Engineering Problem Solving Mapping (EP1–EP7) as a whole imply engineering and software standards mapping for a project. This section makes sure that all technical and planning details are thoroughly documented and easy to access for later reference, system maintenance and/or further development.

A.1 Technologies Used

Component	Technology
Backend	Laravel (PHP)
Frontend	Bootstrap (HTML/CSS)
Database	MySQL
Development Tools	XAMPP, VS Code
Notification	SMS API, SMTP (Email)
Hosting	Localhost → Cloud (Planned)

A.2 User Roles

- **Admin / Property Owner**: Full access to manage properties, tenants, rent, and notifications.
- **Tenant**: Limited to receiving notifications; future versions may support login and online payment.

A.3 Sample Screenshots

- **Dashboard Overview**
- **Add Tenant Form**
- **Invoice Generator**
- **Notification Interface**

A.4 Sample Data Entry Table

Tenant Name	Flat No	Unit	Rent Amount	Due Date
J. Doe	A-101	1A	15,000 BDT	5th of month
R. Karim	B-204	2B	12,000 BDT	10th of month

A.5 Future Scope

- Integration with online payment gateways (e.g., bKash, Nagad)
- Mobile app interface for tenants
- Lease agreement document generator
- Chat-based tenant-owner communication
- Integration with property listing platforms

References

[1] W. R. Bristi, F. Chowdhury, and S. Sharmin, "Stable Matching between House Owner and Tenant for Developing Countries," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, 6-8 July 2019, pp. 1-6.

[2] S. Choudhury, R. Sarker, and M. Rahman, "Real Estate Financing in Bangladesh: Problems, Programs, and Prospects," in *AIUB Journal of Business and Economics*, vol. 7, pp. 78-84, 2008.

[3] M. Johnson and R. Lee, "Advancements in Online Rental Payment Systems," in *Proceedings of the International Conference on Smart Housing Technologies*, 2021, pp. 150-160.

[4] K. Taipalus, "A Global House Price Bubble? Evaluation Based on a New Rent-Price Approach," *Bank of Finland Research Discussion Paper*, no. 29/2006, pp. 68, 2006.

[5] S. Barua and M. S. Khan, "The Dynamics of Residential Real Estate Sector in Bangladesh: Challenges Faced and Policies Sought," in *Journal of Management Studies*, vol. 1, pp. 1-12, 2009.

[6] H. P. Gommans, G. M. Njiru, and A. N. Owange, "Rental House Management System," in *International Journal of Scientific and Research Publications*, vol. 4, pp. 1-24, 2014.

[7] A. Trasad, "Flowchart in Software Engineering Testing," *Mundrisoft*, 2016. [Online]. Available: https://mundrisoft.com/tech-bytes/flowchart-in-software-engineering-testing

[8] Real Estate and Housing Association of Bangladesh (REHAB), "Annual Report," 2004.

[9] P. Nguyen, "Performance Optimization for Laravel-Based Web Applications," in *Software Engineering and Web Applications Journal*, vol. 21, no. 2, pp. 77-90, 2020.

[10] R. S. Karnad, "Housing Finance and the Economy: Regional Trends," in *25th World Congress for International Union for Housing Finance*, Brussels, 23 June 2004, pp. 239-253.

[11] A. H. M. S. Islam, "Operations of Bangladesh Housing Industry: An Uncertain Supply Chain Model," in *AIUB Journal of Business and Economics (AJBE)*, vol. 7, pp. 20-31, 2008.

[12] K. Wilson, "The Role of AI in Property Management," in *Journal of Artificial Intelligence Applications*, vol. 29, no. 1, pp. 55-70, 2020.

[13] S. Al-Nahdi, et al., "Factors Influencing the Intention to Purchase Real Estate in Saudi Arabia: Moderating Effect of Demographic Citizenship," in *International Journal of Business and Management*, vol. 10, no. 4, pp. 35-48, 2015.

[14] D. Talukder, "Assessing Determinants of Income of Rural Households in Bangladesh: A Regression Analysis," in *Journal of Applied Economics and Business Research*, vol. 4, pp. 80-106, 2014.

[15] L. Thompson, "Legal and Compliance Issues in Digital Rent Collection," in *Journal of Property Law and Ethics*, vol. 6, no. 2, pp. 45-60, 2018.

[16] F. Sharmeen, "Modeling Urban House-Rent Variation in Bangladesh: A Study of Four Metropolitan Cities," 2007. [Online]. Available: http://lib.buet.ac.bd:8080/xmlui/handle/123456789/1699

[17] A. Brown, "Security Challenges in Online Rent Payment Platforms," in *Cybersecurity and Property Management Journal*, vol. 10, no. 4, pp. 200-214, 2022.

[18] E. E. Ezebilo, "Evaluation of House Rent Prices and Their Affordability in Port Moresby, Papua New Guinea," in *Buildings*, vol. 7, no. 4, pp. 114, 2017.

[19] F. Khan, "Rental Housing," *United Nations Human Settlements Programme*, 2011.

[20] V. Alakeson, "Making a Rented House a Home: Housing Solutions for 'Generation Rent'," *Resolution Foundation, London*, pp. 4-42, 2011. [Online]. Available:

https://www.emcouncils.gov.uk/write/documents/resolution%20foundation%20housing_report_final.pdf

181-15-10619

ORIGINALITY REPORT

14%	12%	7%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	7%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	www.scirp.org Internet Source	1%
4	Submitted to United International University Student Paper	1%
5	www.riverpublishers.com Internet Source	<1%
6	Bui Thanh Hung, M. Sekar, Ayhan Esi, R. Senthil Kumar. "Applications of Mathematics in Science and Technology - International Conference on Mathematical Appplications in Science and Technology", CRC Press, 2025 Publication	<1%
7	ijsrem.com Internet Source	<1%
8	cse.buet.ac.bd Internet Source	<1%
9	moam.info Internet Source	<1%