

**MULTIMEDIA UNIVERSITY OF KENYA**

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

BACHELOR OF SCIENCE INFORMATION TECHNOLOGY

**PROJECT DOCUMENTATION**

**LIBRARY MANAGEMENT SYSTEM**

**Submitted by:**

REGISTRATION NUMBER: **CIT-221-029/2020**

NAME: **JOHN M WANJEMA**

The project documentation is submitted as a partial fulfillment of the requirement for the award of Bachelors of Science in Information Technology at Multimedia University of Kenya.

# DECLARATION

I hereby declare this Project documentation is my original work and has not been presented for academic purposes in any university, collage or any other institution of higher learning.

**Full Name:**

**Registration number:**

**Signature:**

**Date of submission:**

**Supervisor(s) name:**

**Signature:**

**Date:**

# **ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude to everyone who has contributed to the development of the library management system that we are using today. First and foremost, I would like to thank my mentor who gave me the support needed to make this project possible.

I would also like to extend my appreciation to Shadrack who worked tirelessly to help me design, develop, implement and debug the system. His dedication and hard work have resulted in a system that is user-friendly, efficient, and reliable.

Finally Thank you to my teacher Mr. karis for your contributions to the library management system project. Your efforts are greatly appreciated.

# ABSTRACT

This project utilizes the MERN (MongoDB, Express.js, React, and Node.js) stack with middleware to create a library management system. MongoDB is used as the document database to store data in JSON format, while Express.js serves as the back-end web application framework, running on top of Node.js. React is used as the front-end web app framework, providing a dynamic and responsive user interface. Middleware such as Redux is used for flux architecture and to fetch data rapidly, while Mongoose serves as the ODM (Object Data Model) for MongoDB. The resulting system is a user-friendly, efficient, and reliable library management system that meets the needs of both library staff and patrons.

# LIST OF ABBREVIATIONS

**MERN**: (MongoDB, Express.js, React, and Node.js)

**JSON:** Javascript Object Notation

**CSS:** Cascading style sheet.

**HTML:** Hypertext Mark-up Language.

**React js:** javascript framework

**ODM** :(Object Data Model)

Table of Contents

[DECLARATION 2](#_Toc63074)

[ACKNOWLEDGEMENTS 3](#_Toc63075)

[ABSTRACT 4](#_Toc63076)

[LIST OF ABBREVIATIONS 5](#_Toc63077)

[CHAPTER 1: INTRODUCTION 10](#_Toc63078)

[BACKGROUND 10](#_Toc63079)

[PROBLEM STATEMENT 10](#_Toc63080)

[PROPOSED SOLUTION 10](#_Toc63081)

[PROJECT DESCRIPTION 10](#_Toc63082)

[RESEARCH OBJECTIVES 11](#_Toc63083)

[PROJECT GOAL AND OBJECTIVES 11](#_Toc63084)

[SIGNIFICANCE OF THE STUDY 12](#_Toc63085)

[THE SCOPE OF THE STUDY 12](#_Toc63086)

[ASSUMPTIONS 13](#_Toc63087)

[CHAPTER 2: LITERATURE REVIEW 13](#_Toc63088)

[INTRODUCTION 13](#_Toc63089)

[LIMITATIONS OF THESE SYSTEMS 14](#_Toc63090)

[SOLUTION TO THESE WEAKNESSES 14](#_Toc63091)

[ASSUMPTIONS 15](#_Toc63092)

[LIMITATIONS OF THESE SYSTEMS 15](#_Toc63093)

[SOLUTION TO THESE WEAKNESSES 16](#_Toc63094)

[CHAPTER 3: METHODOLOGY 17](#_Toc63095)

[INTRODUCTION 17](#_Toc63096)

[METHODOLOGY 17](#_Toc63097)

[**Description** 17](#_Toc63098)

[Justification for Using This Methodology 19](#_Toc63099)

[DATA COLLECTION METHODS AND TOOLS 20](#_Toc63100)

[Data Collection 20](#_Toc63101)

[Data Collection Tools 21](#_Toc63102)

[Data Collection Methods 22](#_Toc63103)

[Project Budget and Resources 22](#_Toc63104)

[Project Resources 22](#_Toc63105)

[Project Hardware requirements 22](#_Toc63106)

[Project Software Requirements 22](#_Toc63107)

[Project Schedule 23](#_Toc63108)

[Project Plan 23](#_Toc63109)

[Time Schedule 24](#_Toc63110)

[CHAPTER 4: SYSTEM ANALYSIS 26](#_Toc63111)

[SYSTEM REQUIREMENTS 26](#_Toc63112)

[User Requirements 26](#_Toc63113)

[FUNCTIONAL REQUIREMENTS 27](#_Toc63114)

[NON-FUNCTIONAL REQUIREMENTS 28](#_Toc63115)

[CHAPTER 5: SYSTEM DESIGN 29](#_Toc63116)

[Design objectives 29](#_Toc63117)

[Design Analysis 29](#_Toc63118)

[Architectural Design 29](#_Toc63119)

[Logical Design 30](#_Toc63120)

[Logical Data Design 30](#_Toc63121)

[Physical Design Description 31](#_Toc63122)

[Physical design 31](#_Toc63123)

[Layout design 31](#_Toc63124)

[Data dictionary 35](#_Toc63125)

[Table 3 tenants 35](#_Toc63126)

[Table landlord 36](#_Toc63127)

[Table 5 house manager 36](#_Toc63128)

[Database Design 37](#_Toc63129)

[Database Tables 37](#_Toc63130)

[Admin Database Table 38](#_Toc63131)

[Structure of tenants Table 39](#_Toc63132)

[Structure of Admin Table 39](#_Toc63133)

[Input Design 40](#_Toc63134)

[The admin 40](#_Toc63135)

[The tenant 40](#_Toc63136)

[Landlord 40](#_Toc63137)

[The house manger 40](#_Toc63138)

[Login User Interface 40](#_Toc63139)

[User Log In 41](#_Toc63140)

[Output Design 41](#_Toc63141)

[Tenant Report 41](#_Toc63142)

[Tenant Report 41](#_Toc63143)

[System flowchart figure 3 flowchart 43](#_Toc63144)

[Process Design 43](#_Toc63145)

[User Password Flowchart 45](#_Toc63146)

[Figure 5 Admin log in 46](#_Toc63147)

[Figure 6 tenants log in 46](#_Toc63148)

[Data Flow Diagram [DFD] 50](#_Toc63149)

[Entity Relationship Diagram 51](#_Toc63150)

[CHAPTER 6: IMPLEMENTATION AND TESTING 52](#_Toc63151)

[DEVELOPMENT ENVIRONMENT 52](#_Toc63152)

[SYSTEM COMPONENTS 53](#_Toc63153)

[System user Components 53](#_Toc63154)

[System Administrator 53](#_Toc63155)

[Tenants 53](#_Toc63156)

[TEST PLAN 53](#_Toc63157)

[INTRODDUCTION 53](#_Toc63158)

[SCOPE 54](#_Toc63159)

[QUALITY OBJECTIVES 55](#_Toc63160)

[TYPES OF TESTING 55](#_Toc63161)

[TEST METHODOLOGY- 55](#_Toc63162)

[SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 55](#_Toc63163)

[TEST DATA 56](#_Toc63164)

[TEST COMPLETENESS 58](#_Toc63165)

[TESTING TOOLS 58](#_Toc63166)

[REASONS FOR TESTING 58](#_Toc63167)

[TEST RESULTS 58](#_Toc63168)

[CHAPTER7: CONCLUSION 60](#_Toc63169)

[ACHIEVEMENTS AND LESSONS 60](#_Toc63170)

[ACHIEVEMENTS 60](#_Toc63171)

[LESSONS LEARNT 60](#_Toc63172)

[CONCLUSION 60](#_Toc63173)

[RECOMMENDATIONS 60](#_Toc63174)

[APPENDICES 63](#_Toc63175)

[APPENDIX A 63](#_Toc63176)

[PROJECT SCHEDULE 63](#_Toc63177)

## CHAPTER 1: INTRODUCTION

## BACKGROUND

Libraries have been an important part of society for centuries, providing access to knowledge and information to people of all ages and backgrounds. With the advent of the digital age, libraries have evolved to include not only physical books, but also e-books, audiobooks, and other digital resources. However, managing these resources and providing access to patrons can be a complex and time-consuming process. Traditional library management systems often involve manual record-keeping and time-consuming administrative tasks, which can lead to errors, inefficiencies, and a subpar user experience for patrons. In order to address these challenges, many libraries have turned to technology to streamline their operations and enhance the user experience.

The library management system developed is designed to meet the needs of modern libraries, utilizing the MERN stack with middleware to create a user-friendly, efficient, and reliable system for managing library resources and providing access to patrons. The system aims to simplify administrative tasks, automate record-keeping, and provide a seamless user experience for patrons, ultimately enhancing the overall effectiveness and impact of the library.

## PROBLEM STATEMENT

The current library management system is time-consuming and error-prone, with manual record-keeping and administrative tasks that can lead to data redundancy and inconsistency. This not only creates inefficiencies for library staff, but also results in a subpar user experience for the management..

To address these challenges, a new library management system is needed that is cost-effective, efficient, and environmentally friendly. The system should automate administrative tasks, reduce paper usage, and provide a seamless user experience for patrons, ultimately enhancing the overall effectiveness and impact of the library.

## PROPOSED SOLUTION

The library management system will be developed using the MERN stack with middleware to create a user-friendly, efficient, and reliable system for managing library resources and providing access to management. The resulting system will be a comprehensive library management solution that meets the needs of both students and admins, providing a seamless user experience and streamlining administrative tasks.

## PROJECT DESCRIPTION

For students, the proposed solution includes the following features:

* Registration: Students can register themselves on the app by providing their details and creating a profile.
* Profile management: Students can view and edit their profile, change their password, and keep their information up-to-date.
* Book search: Students can search for books and view their availability in the library.
* Issue history: Students can view their issue history, including the books they have borrowed in the past.

For admins, the proposed solution includes the following features:

* Profile management: Admins can view and edit their profile and keep their information up-to-date.
* Book management: Admins can search for books, view their availability, and add, edit, or delete existing books in the library.
* Issue and return management: Admins can issue books to students and return books that were issued earlier.
* Library statistics: Admins can view all stats of the library, including the number of books, the number of students registered, and the number of books issued and returned.
* Student and admin management: Admins can view the profile of all students and admins, and manage their accounts as needed.

## RESEARCH OBJECTIVES

My objectives for the research are:

* Explore the impact of implementing a new library management system.
* Investigate how the new system affects administrative tasks, record-keeping, and user experience for both library staff and patrons.
* Examine the potential cost savings and environmental benefits of the new system.
* Identify potential challenges and barriers to implementation.
* Provide insights and recommendations for libraries considering implementing a new management system.

.

## PROJECT GOAL AND OBJECTIVES

This study will investigate the implementation of a new library management system to improve the efficiency and effectiveness of library operations. The study will focus on reducing manual record-keeping, enhancing the user experience, and promoting environmental sustainability by reducing paper usage.

The following parties will benefit from this study:

**Project Objectives:**

1. Design and develop a library management system using the MERN stack with middleware.
2. Implement a user-friendly interface for patrons to search for and request books, view their borrowing history, and manage their account information.
3. Automate administrative tasks such as book cataloging, inventory management, and overdue book tracking.
4. Provide real-time book availability status for patrons and staff to minimize wait times and optimize book circulation.
5. Implement a secure authentication and authorization system with different levels of access for library staff and patrons.
6. Provide comprehensive reporting and analytics capabilities for library staff to track book circulation, patron activity, and system performance.
7. Conduct user testing and gather feedback to continuously improve the system and enhance the user experience.
8. Ensure the system is scalable, maintainable, and adaptable to accommodate future changes and expansions.

By achieving these objectives, the resulting library management system will improve operational efficiency, enhance the user experience, and ultimately make the library a more effective and valuable resource for the community.

## SIGNIFICANCE OF THE STUDY

The following parties will benefit from this study:

Library staff:

* The new system will streamline administrative tasks, automating record-keeping and reducing the need for manual data entry, saving time and resources.
* The system will provide real-time information on book availability, enabling faster and more efficient book circulation.
* The system will allow for comprehensive reporting and analytics capabilities, providing insights on library performance and patron activity.

Patrons:

* The new system will provide a user-friendly interface for searching and requesting books, viewing borrowing history, and managing account information.
* The system will enable faster and more efficient book circulation, reducing wait times and enhancing the overall user experience.

Environmental sustainability:

* The new system will reduce paper usage, promoting environmental sustainability and reducing the library's ecological footprint.

Ultimately, the implementation of the new library management system will improve operational efficiency, enhance the user experience, and promote environmental sustainability, making the library a more effective and valuable resource for the community.

## THE SCOPE OF THE STUDY

The study will focus on creating a modern and efficient library management system that streamlines administrative tasks, automates record-keeping, and enhances the user experience for both library staff and patrons. The system will cover the daily activities involved in managing a library, including book cataloging, inventory management, and book circulation. The proposed system will be a web-based application accessible via desktop and mobile devices.

## ASSUMPTIONS

The following assumptions are made in the development and implementation of the library management system:

* All library staff and patrons are registered and entered into the system.
* The library staff and administrators are available to provide support and assistance during the implementation process.
* The system will be deployed on a reliable and stable hosting platform with adequate resources to support the expected usage.
* The system will be developed using industry-standard programming languages and frameworks to ensure compatibility and scalability.
* The system will comply with all relevant data privacy and security regulations to protect sensitive information.

CHAPTER 2

## LIMITATIONS OF THESE SYSTEMS

The response time for transactions can be slow, and the system may not provide adequate security for sensitive patron and library information.

Furthermore, the existing system may not allow for real-time book availability status, leading to wait times and subpar user experience for patrons. The system may not be scalable and may not accommodate future changes and expansions.

Therefore, there is a need for a modern and efficient library management system that streamlines administrative tasks, automates record-keeping, and enhances the user experience for both library staff and patrons. The new system should provide a user-friendly interface, real-time book availability status, and comprehensive reporting and analytics capabilities, while also promoting environmental sustainability and data privacy and security.

## SOLUTION TO THESE WEAKNESSES

Creating a system that allows the landowners and property managers to do their work directly gives the tenants a platform to get their concerns solved faster since they are in direct communication through the system. This system will also reduce paper work by at least 50% by having data stored in databases and having backups for security purposes. This will also reduce data loss and redundancy. The system will increase transparency among all the parties involved and make the management processes a lot easier. Report generation through the system will save a lot of time.

# CHAPTER 3: METHODOLOGY

The methodology I choose for this project is agile methodology, which uses the following steps:

 Agile methodology:

1. Project Planning:

* Define the project goals and objectives.
* Identify the stakeholders and their requirements.
* Define the scope of the project and create a project roadmap.
* Break down the project into user stories and epics.
* Develop a project backlog.

1. Sprint Planning:

* Select the highest priority user stories from the project backlog.
* Break down the selected user stories into tasks.
* Estimate the time and resources required to complete each task.
* Assign tasks to team members based on their skills and availability.
* Create a sprint backlog.

1. Sprint Development:

* Develop the features and functionalities identified in the sprint backlog.
* Conduct daily stand-up meetings to track progress, identify blockers, and adjust course as needed.
* Test and validate the developed features and functionalities against the acceptance criteria.

1. Sprint Review:

* Review the developed features and functionalities with the stakeholders.
* Gather feedback and identify areas for improvement.
* Update the project backlog with new user stories or epics as needed.

1. Sprint Retrospective:

* Evaluate the development process and identify areas for improvement.
* Discuss and implement process improvements for the next sprint.

1. Release Planning:

* Plan for the release of the developed features and functionalities.
* Coordinate testing, documentation, and training efforts.
* Release the new features and functionalities to production.

1. Production:

* Monitor and maintain the system in production.
* Address any issues or bugs that arise.
* Continuously gather feedback and identify areas for improvement.

By using agile methodology, the development team can respond to changing requirements, prioritize customer needs, and deliver a high-quality product that meets customer expectations. The iterative and incremental nature of agile development allows for continuous feedback and improvement throughout the development process, resulting in a more efficient and effective library management system.

## DATA COLLECTION METHODS AND TOOLS

During the development of the library management system, various data collection tools were used to gather feedback and insights from library staff and patrons. These tools included surveys, focus groups, interviews, observation, usage data, and user testing.

* Surveys were conducted both online and in paper format to gather feedback and opinions from library staff and patrons. Focus groups were used to gather more detailed feedback and opinions from a group of participants who were asked specific questions about their experiences with the library management system.
* Interviews were conducted with library staff and patrons to gather more personalized feedback and opinions. Observation was also used to observe how library staff and patrons were using the current library management system.
* Usage data was collected to gain insights into how the current library management system was being used and which features were being used the most.
* User testing was also conducted with library staff and patrons to gather feedback on the usability and functionality of the new library management system.

Overall, using a combination of these data collection tools provided a comprehensive understanding of the needs and preferences of library staff and patrons, and informed the development of a more efficient and effective library management system.

## Project Schedule

## Time Schedule

Table 2 Project Time Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** |  |  | **Completion** |
| Writing and submitting | the | Project |  |
| Documentation |  |  | Week 1 |
| Requirement analysis |  |  | Week 2 |
| System design |  |  | Week 3 |
| Coding |  |  | Week 4 |
| Prototype updating |  |  | Week 5 |
| System validation and verification | |  | Week 6 |
| System testing | |  | Week 7 |
| Project Documentation | |  | Week 8 |
| Project presentation and demonstration | |  | Week 9 |

Project Budget

Table 3 Project Hardware cost

The budget for the Project must be economical. The table below shows the total budget of the Project.

|  |  |
| --- | --- |
| **Resource** | **Cost(Ksh)** |
| Computer system i.e. Laptop | Ksh: 70,000 |
| Research expenses i.e. Transport cost | Ksh: 29000 |
| Internet | Ksh:13500 |
| Documentation printing | Ksh:2000 |
| Documentation Binding | Ksh:1000 |
| Total | Ksh:153500 |

# CHAPTER 4: SYSTEM ANALYSIS

## SYSTEM REQUIREMENTS

1. Information Management:

* The system should provide efficient and effective information management capabilities, including book cataloging, inventory management, and book circulation.
* The system should allow for easy and organized storage and retrieval of information.

1. User Interface:

* The system should have a user-friendly interface that is easy to learn and use.
* The system should allow for customization of the interface to fit the preferences of individual users.

1. Performance:

* The system should be fast and efficient in processing transactions, such as book checkouts and returns.
* The system should be able to handle a high volume of transactions without slowing down.

1. Flexibility:

* The system should be flexible and convenient, allowing for customization and adaptation to meet the changing needs of the library.
* The system should allow for integration with other library systems and technologies.

1. Security:

* The system should be safe and secure, ensuring the protection of sensitive patron and library information.
* The system should comply with all relevant data privacy and security regulations.

By meeting these system requirements, the library management system can provide efficient and effective information management capabilities, a user-friendly interface, fast performance, flexibility, and security, ultimately enhancing the user experience for both library staff and patrons.

**User Permissions**

**Student**

A student can

* register himself on the app
* view and edit his profile
* change his password
* search for books and view availabilty
* view his issue history

**Admin**

An admin can

* view and edit his profile
* search for books and view availability
* view, Edit or Delete existing books
* add new books
* issue a book to a student
* return a book issued earlier
* view all stats of the library
* view issue log and the profile of all the students
* view the profile of all admins

# CHAPTER 5: SYSTEM DESIGN

The design objectives of the library management system are to provide efficient and effective information management capabilities, enhance the user experience for both library staff and patrons, and ensure the security and privacy of sensitive patron and library information. The system should have a user-friendly interface that is easy to learn and use, and should be able to handle a high volume of transactions without slowing down. The system should also be flexible and adaptable to meet the changing needs of the library.

3.2 Design Analysis  
The design analysis phase involves identifying the components and functionalities needed for the library management system, and determining how they will interact with each other. The analysis phase also involves identifying any constraints or limitations that may impact the design of the system, such as budget, time, and technology resources.

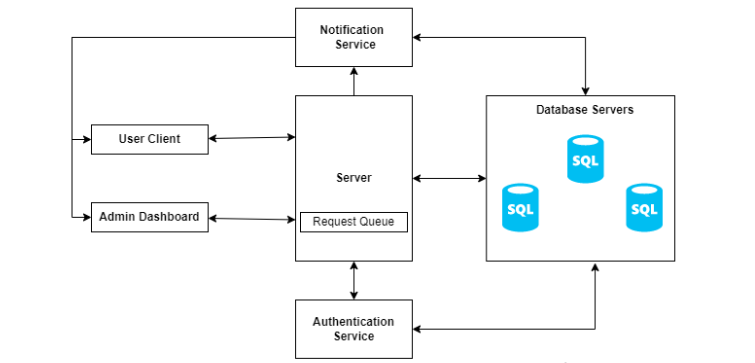
During the design analysis phase, the following components and functionalities were identified:

1. Book Cataloging: The system should allow for efficient and effective book cataloging, including book title, author, publisher, ISBN, and other relevant information.
2. Inventory Management: The system should provide inventory management capabilities, including tracking book availability, location, and status.
3. Book Circulation: The system should allow for efficient and effective book circulation, including book checkouts, returns, and renewals.
4. User Management: The system should allow for user management capabilities, including creating and managing user accounts, and tracking user information and activities.
5. Reporting and Analytics: The system should provide comprehensive reporting and analytics capabilities, including generating reports on book circulation, user activities, and other relevant data.

3.3 Architectural Design  
The architectural design phase involves creating a high-level overview of the system, including the components, their interactions, and how they will be implemented. During the architectural design phase, the following system architecture was proposed:

1. Presentation Layer: The presentation layer will provide a user-friendly interface for library staff and patrons to interact with the system. It will include a graphical user interface (GUI) that is easy to navigate and use.
2. Application Layer: The application layer will provide the core functionalities of the system, including book cataloging, inventory management, book circulation, user management, and reporting and analytics. It will be implemented using a combination of programming languages and frameworks, including Java, Spring, and Hibernate.
3. Data Layer: The data layer will provide the storage and retrieval of data from the system. It will include a relational database management system (RDBMS) such as MySQL or PostgreSQL, which will store all relevant data, including book information, user information, and transaction data.
4. Security Layer: The security layer will ensure the protection of sensitive patron and library information. It will include measures to prevent unauthorized access, data breaches, and data loss, including data encryption and access control.

By implementing this architectural design, the library management system can provide efficient and effective information management capabilities, a user-friendly interface, fast performance, flexibility, and security, ultimately enhancing the user experience for both library staff and patrons.

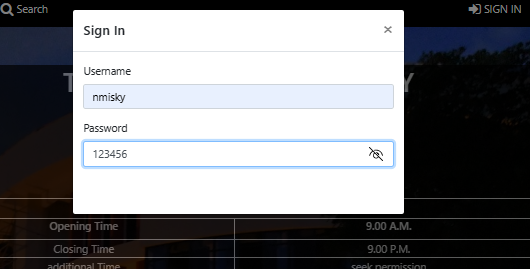


## Physical Design Description

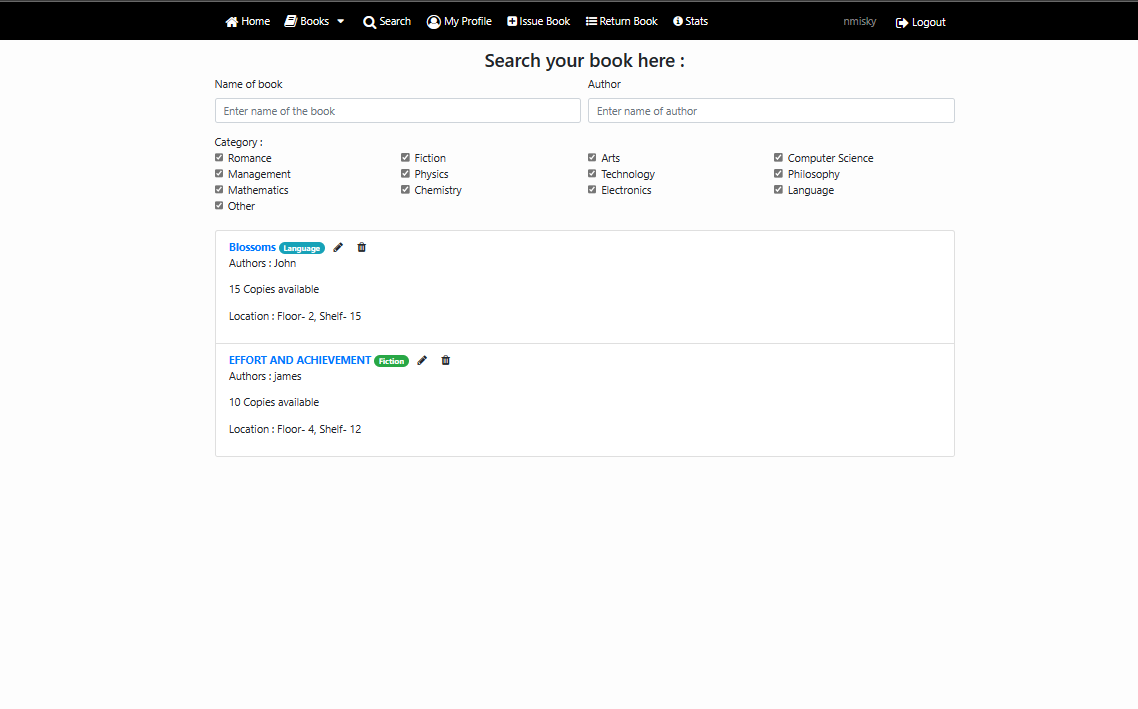
#### Physical design

**HOMEPAGE**

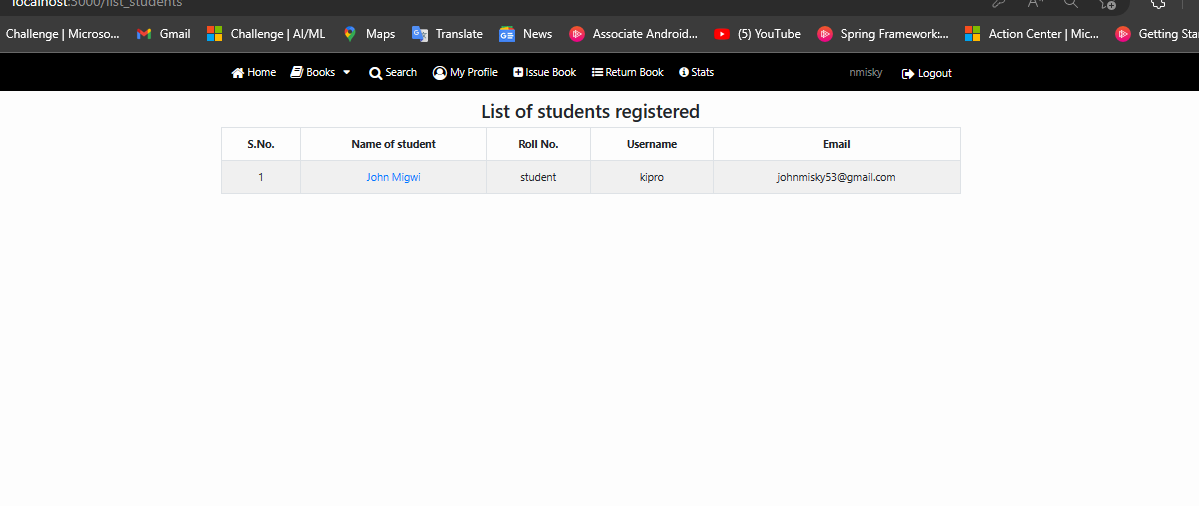




**SEARCHING FOR BOOKS**



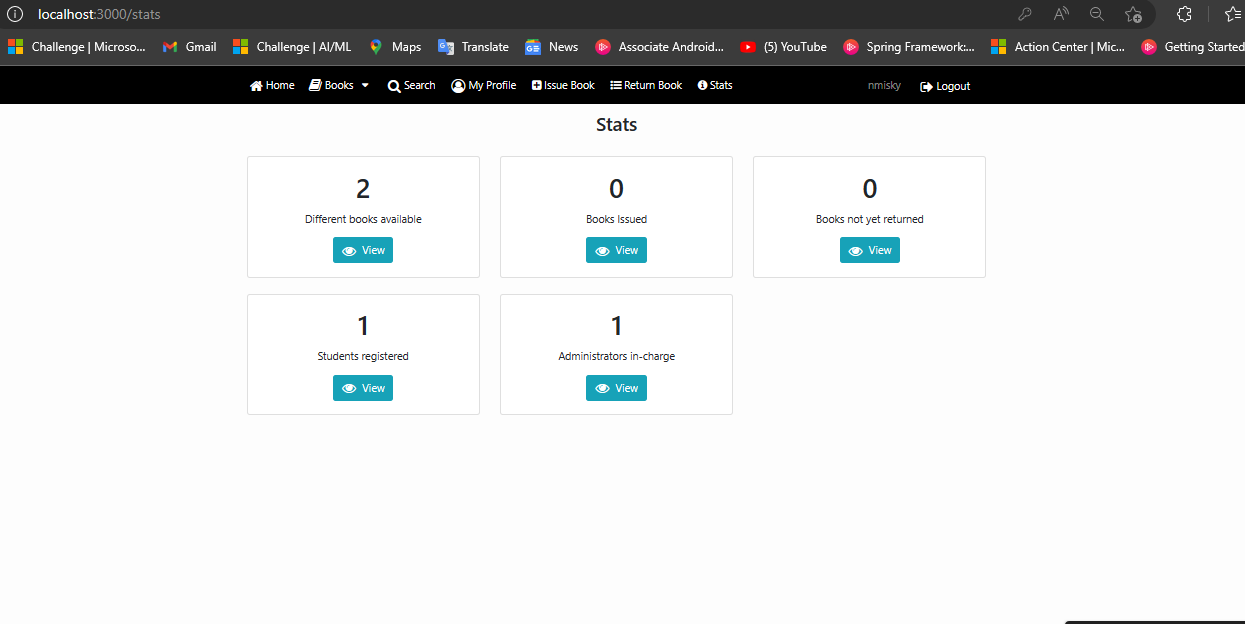
LIST OF STUDENTS REGISTERED



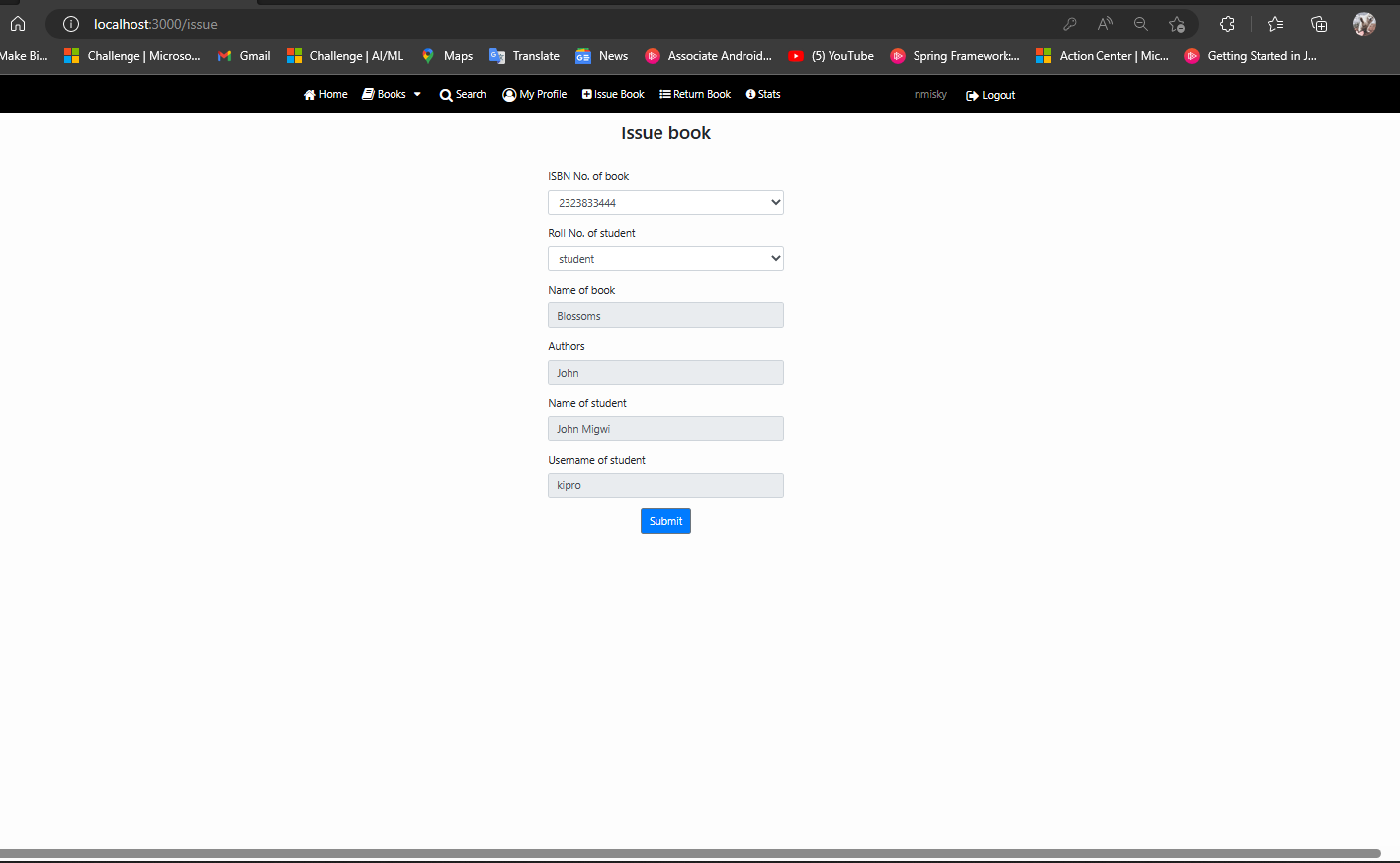
LIST OF ADMINS INCHARGE



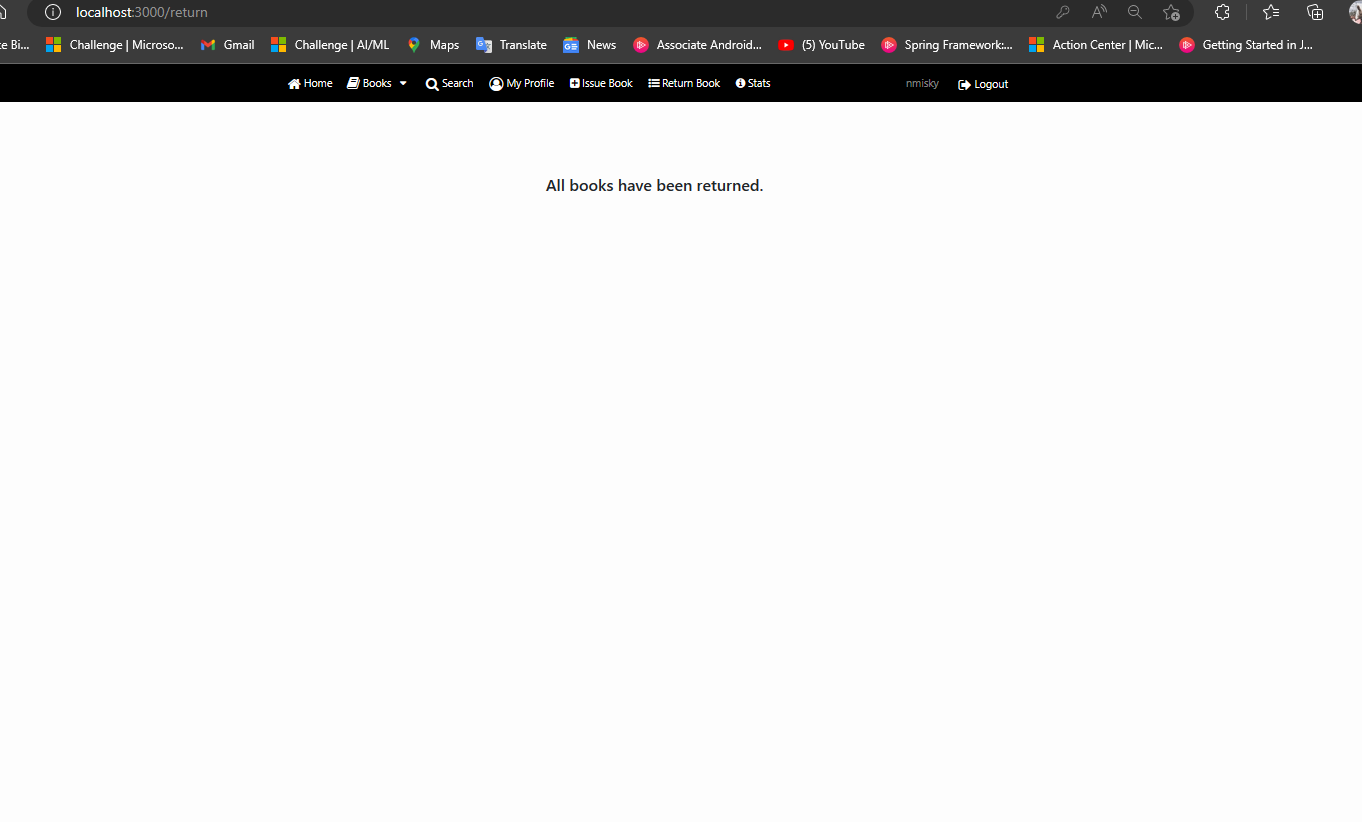
STATS OF THE LIBRARY



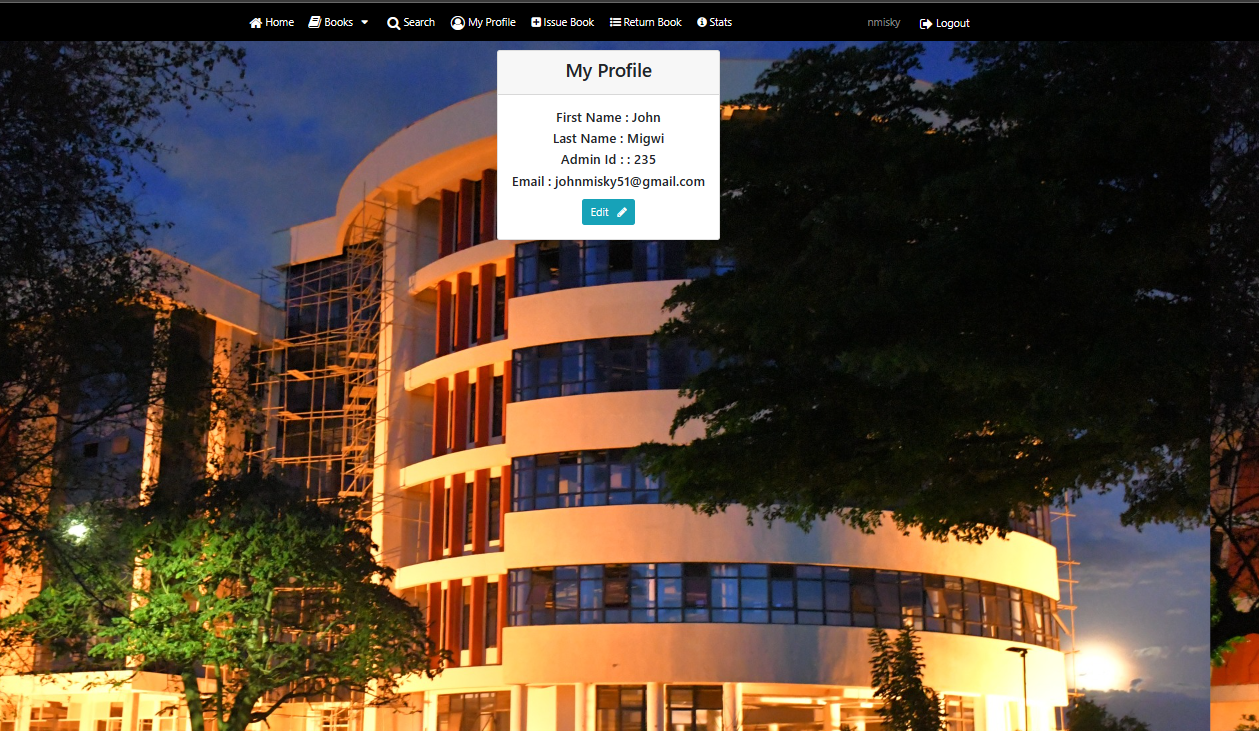
ISSUE BOOK



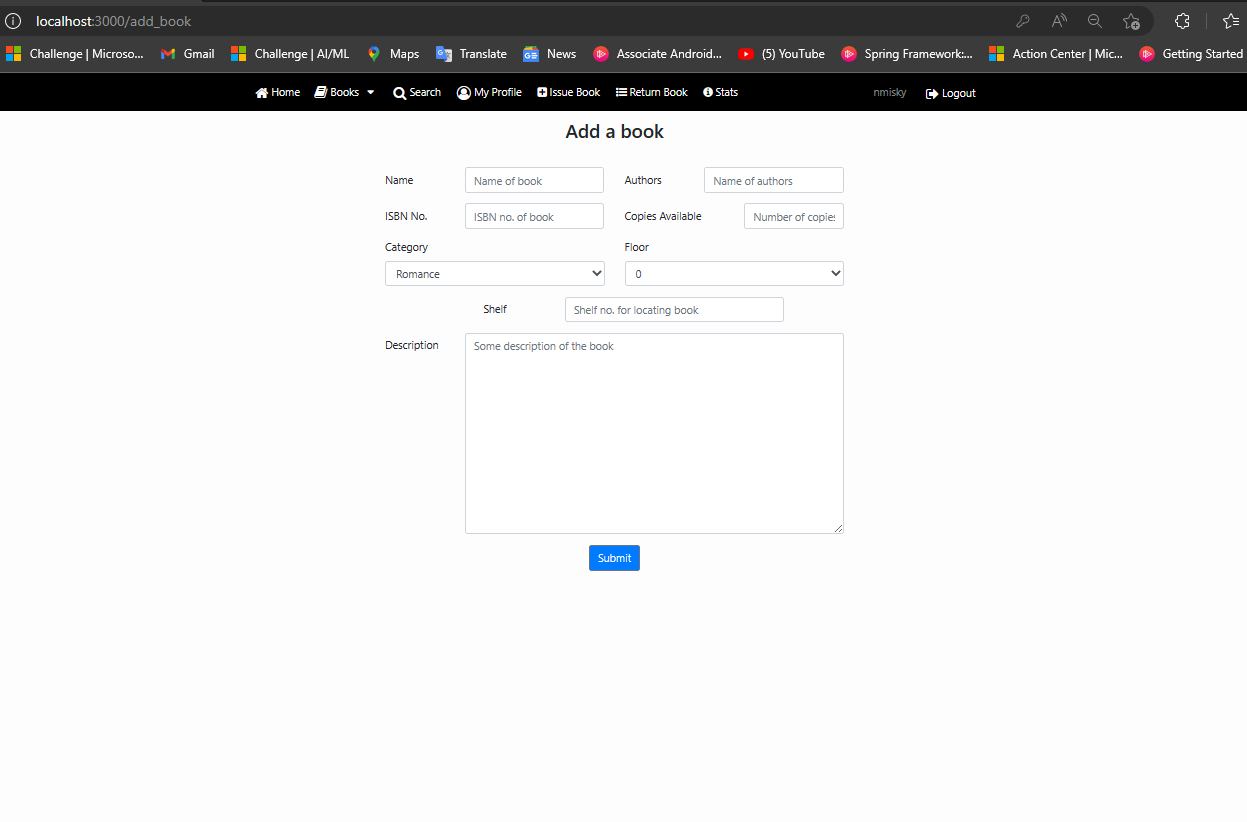
BOOKS RETURNED



**USER PROFILE**

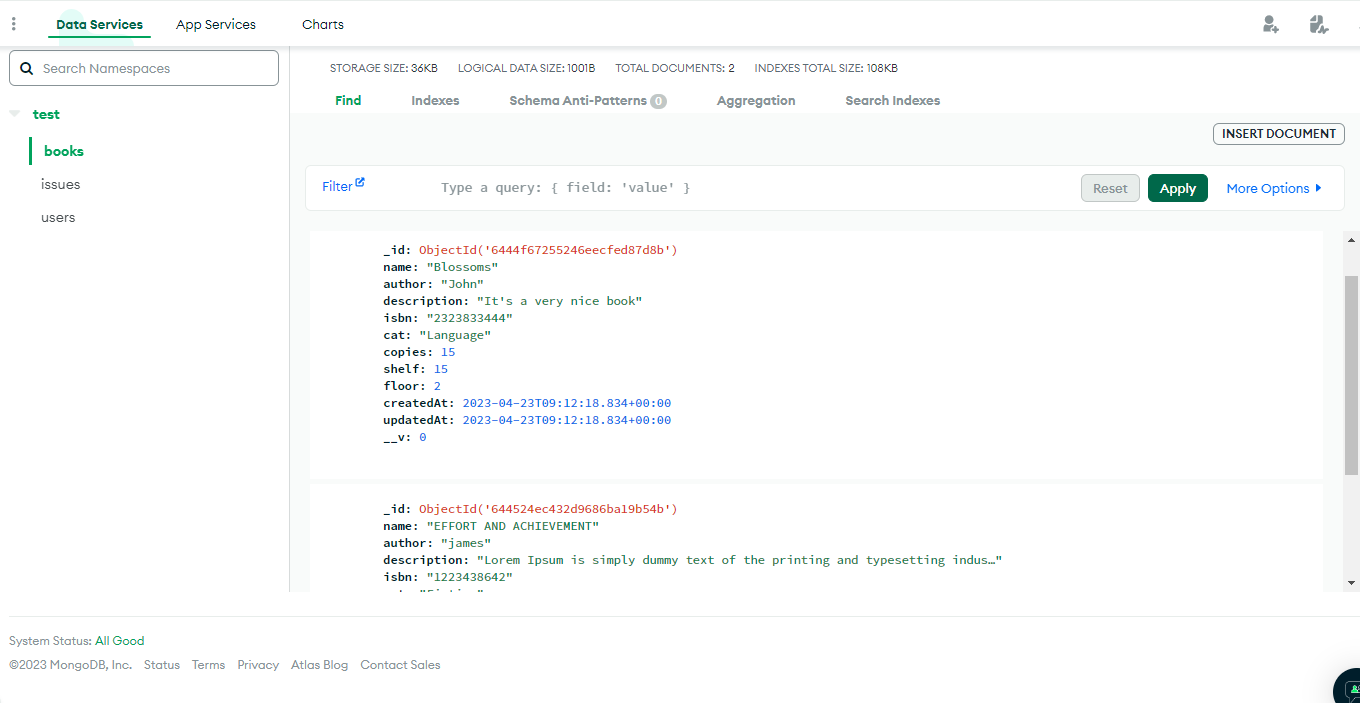
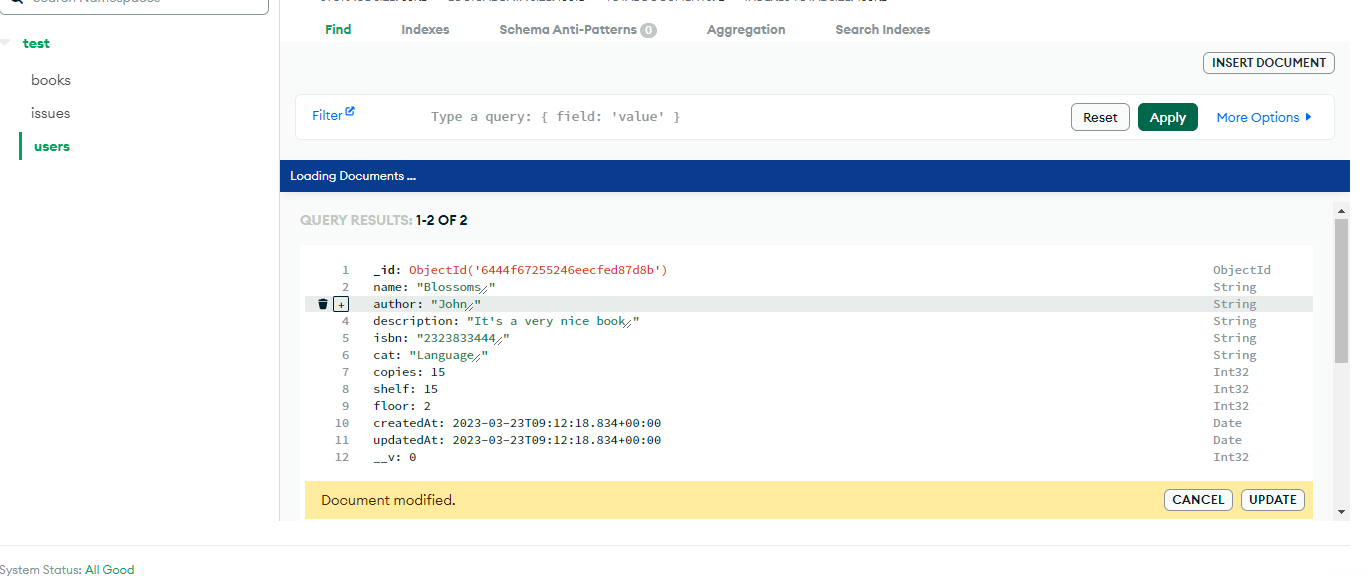
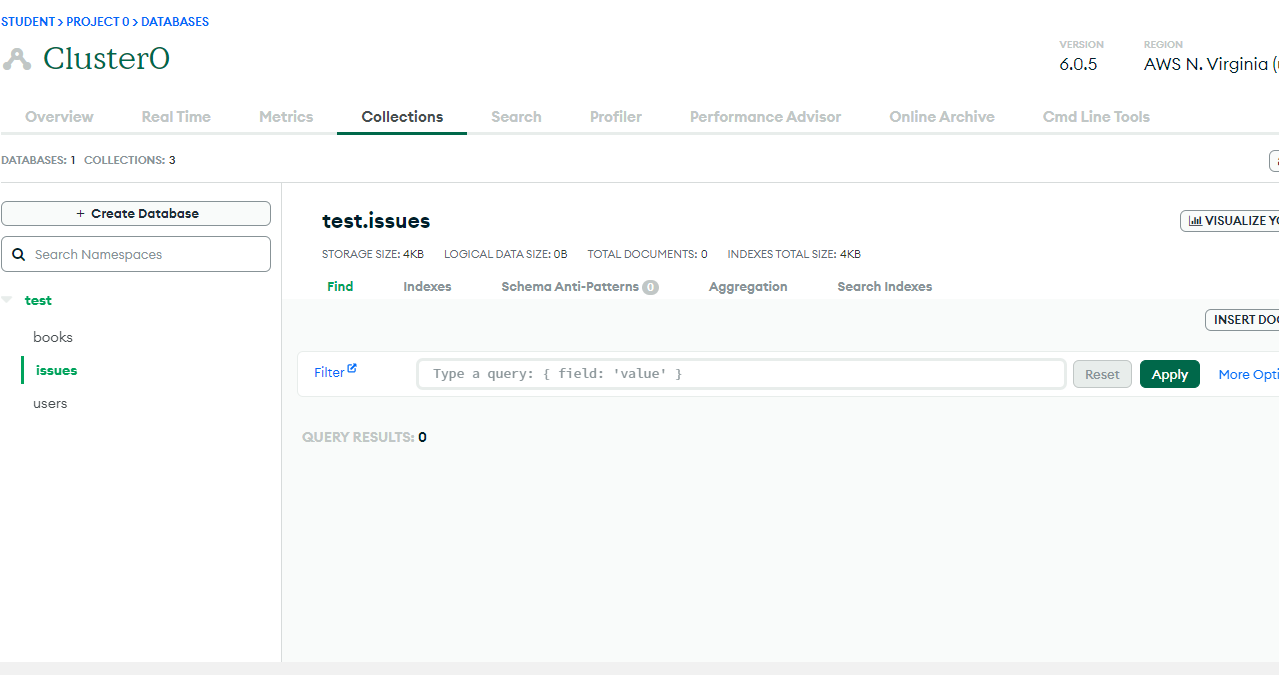


**ADDING A BOOK**



## Data dictionary

BELOW ARE SCREENSHOTS OF MONGO DB JSON FILES USED IN DEVELOPING THE SYSTEM :



#### User/admin Password Flowchart

Figure 4 password flowchart





Start



Enter name and password



Correct



Admin Home page



STUDENT



Start



Log in



Username &



Password correct?



Deadline date due?



Check and update details



Send complaints



View and print rent payment report



Log Out



Stop



Display Error



Message



#### Figure 6 Student Registration

Figure 7 Admin Flow Chart

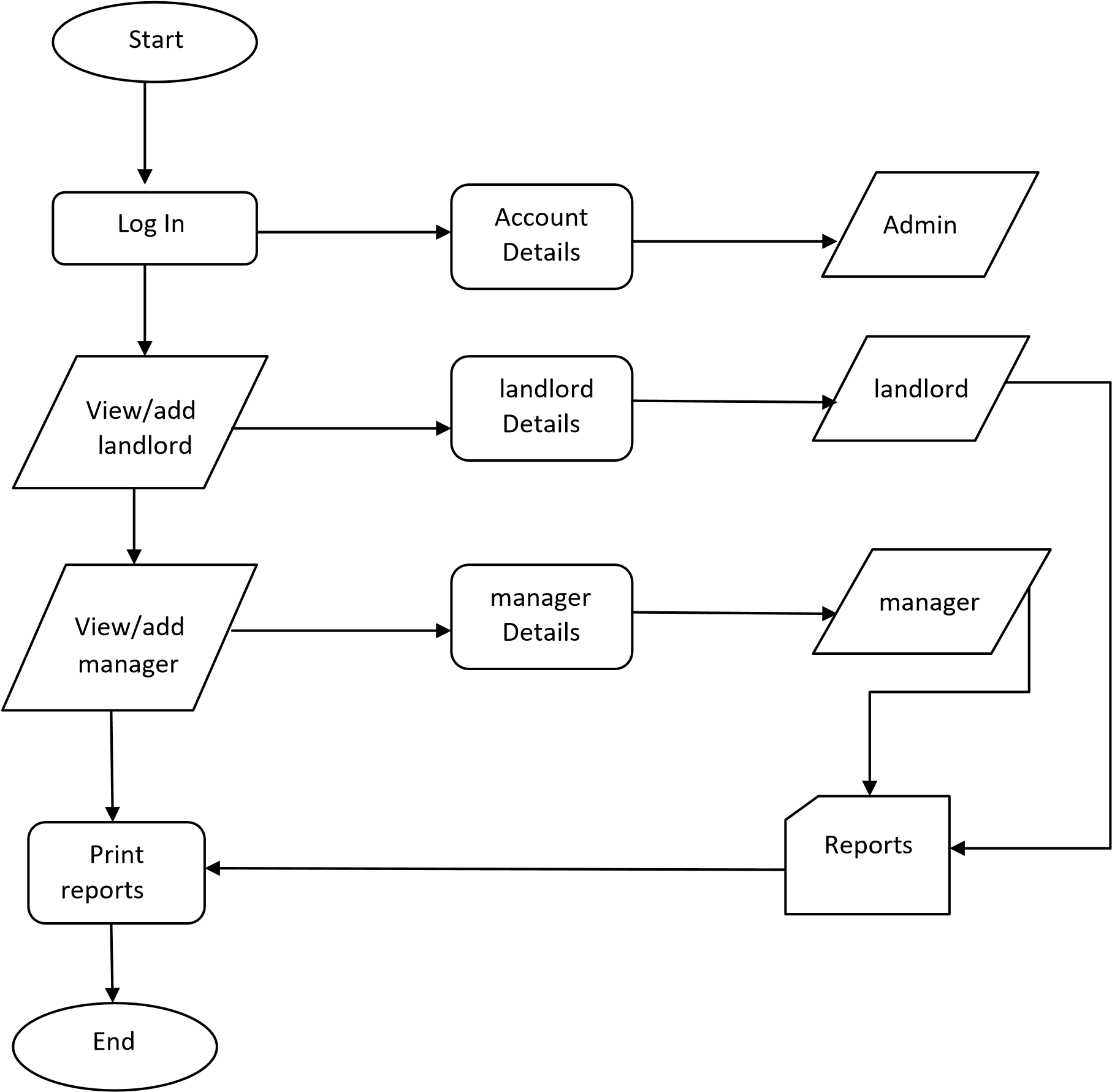


Figure 8 manager Flow Chart



End



Start



Log in



Add tenant



manage



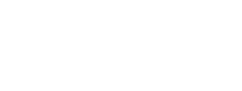
tenants



tenant



Admin



Assign

tenant

house



Figure 9 Sequence diagram

**ADMIN SYSTEM**

Open the application



Request username and password



Provide feedback on login



### Data Flow Diagram [DFD]

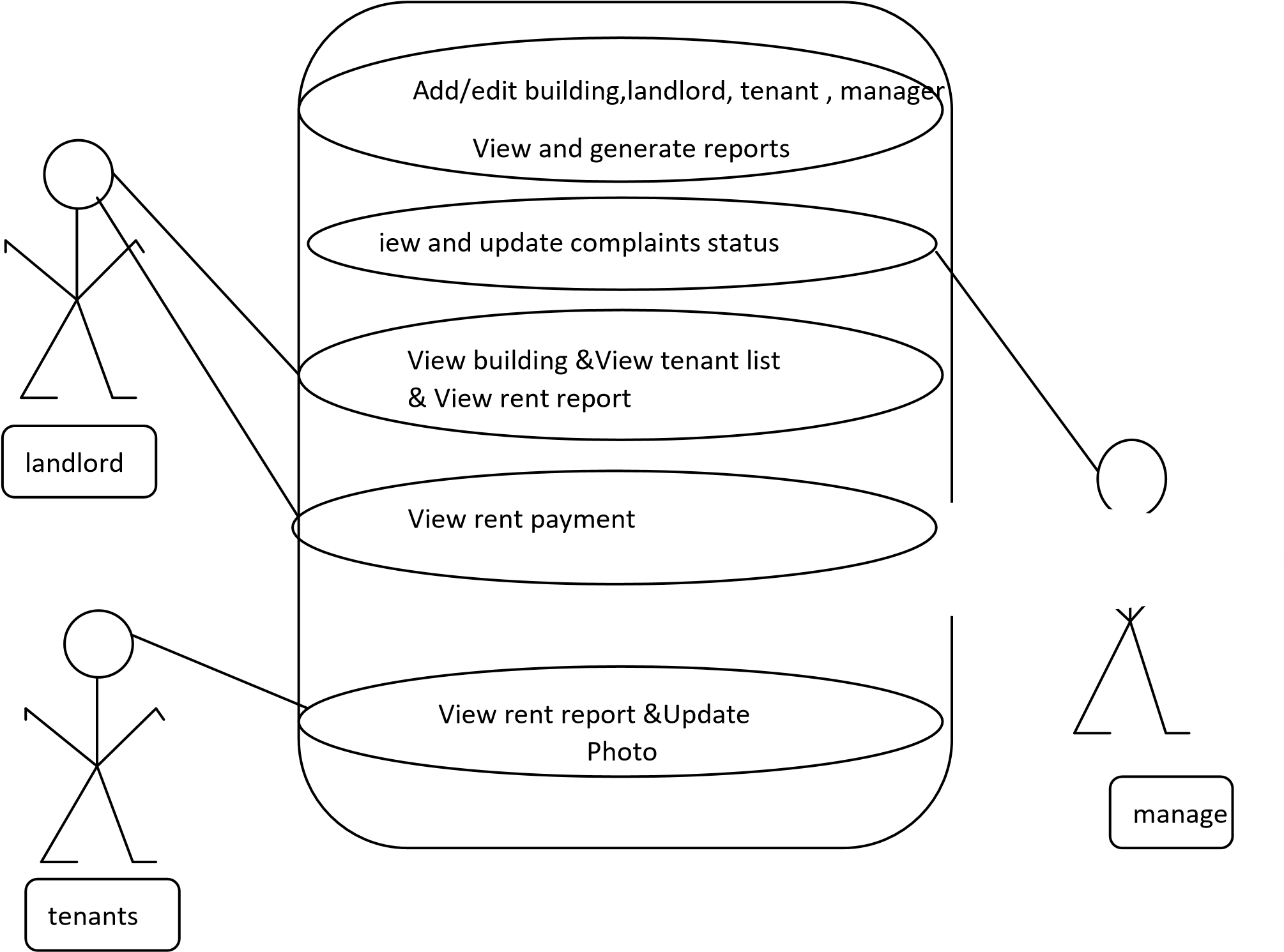
To model the process in the system, Data Flow Diagram which is a graphical representation of the flow of data through an information system, modeling its process aspects was adopted.

These data flow diagrams represent how information used by the users of the system.

The context data flow diagram shows the entire process rental management system. Tenant, landlord house manager and Admin serve as the external entity while the center part represents the incoming and the ongoing process.

The initial Data Flow Diagram which identified the major areas and data flows which are within the terms of reference of the system Project Documentation is as illustrated below.

Figure 10 use case diagrams

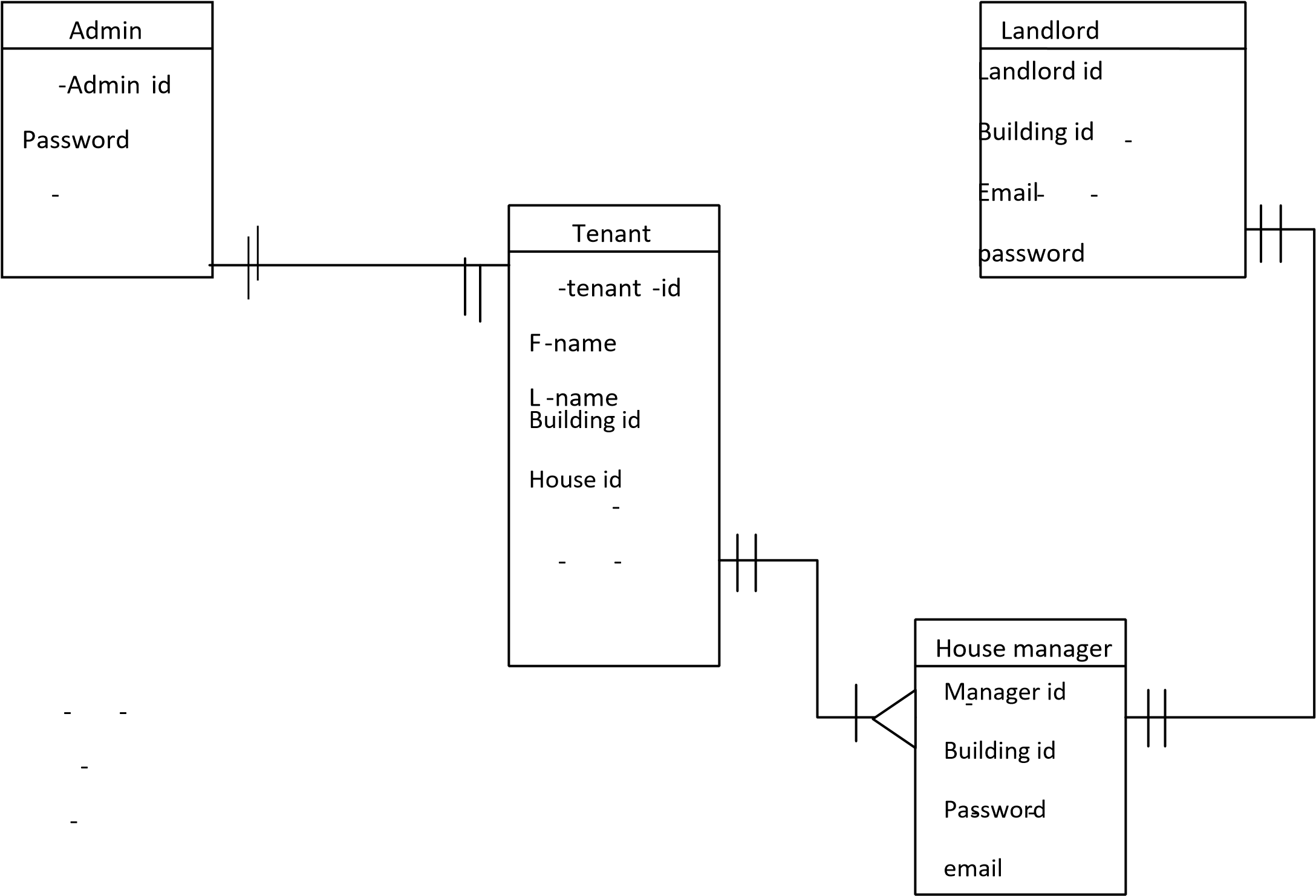


## Entity Relationship Diagram

The figure below shows the relationship of each entity on the table in the database of the

System. The relationship in the below tables is one to-one, one-to-many and many-to-many relationships.

Figure 11 Entity Relationship



# CHAPTER 6: IMPLEMENTATION AND TESTING

.

## SYSTEM COMPONENTS

## System user Components

The Rental Management system has several user components that are involved and are discussed below.

#### System Administrator

The admin has several functions to monitor and ensure the smooth running of the System.

* Database access: The Admin user has access to database and all its information for maintenance purposes only. They can’t view or change passwords of users whatsoever.  Adding and managing other System users.
* Adding and managing the houses.
* Generating reports for various instances.

#### Tenants

 Tenants can check their details in the system.  The tenants can change their passwords.

## TEST PLAN

#### INTRODDUCTION

A test plan is a document describing software testing scope activities. It is the base for formally testing any software product in a project.in simpler terms it can be described as a document describing the scope approach, resources and schedule of intended activities.

The plan identifies the items to be tested, the features to be tested, the types of testing to be performed, the resources required to complete the testing and the risks associated with the plan.

#### SCOPE

*IN-SCOPE*

The most important features of this rental management system were tested.

Table in scope

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Applicable Roles** | **Description** |
| Report Generation | Admin | The administrator is able to generate reports for different areas in the system, such as the rent collection report and employee attendance. |
| Password | Admin      Tenant | The admin can change and access the password of only his account, he cannot change the tenants or access their passwords.  The tenants can access and change passwords of their accounts only. |
| Data Retrieval | Admin | The admin should be able to retrieve the rental data easily from the system. |
| Data Update | Admin | The admin should be in a position to update any data whenever necessary. |
| New Tenant | Admin | The admin should be able to add any new tenants into the system. |

*OUT- SCOPE*

The following features were not to be tested because they were not included in the software requirement specifications:

* User- interfaces
* Hardware interfaces
* Software interfaces
* Database logic

#### QUALITY OBJECTIVES

The test objective was to verify the functionality of this rental management system. The project should focus on managing the rent paid by the various tenants in the houses in managed by the agency.

#### TYPES OF TESTING

##### **i.** Module/ Unit testing

After a module has been coded, the code is thoroughly reviewed and then tested with predesigned test data to determine if the modules are fit for use. I tried to test each module separately, i.e. the search module, the add module, the report generation module. All the codes were fit to run the system.

##### ii. Integration Testing

In integration testing, the separate modules will be tested together to expose faults (if there are any) in the interfaces and in the interaction between integrated components. Testing is black box since the code is not directly checked for errors.

##### iii. System Testing

The system testing will compare the system specifications against the actual. The product met almost all the requirements except the pulling of images from the database which I could not implement due to short period of time.

##### iv. Acceptance Testing

It is the phase of testing used to determine whether a system satisfies the user requirements specified in the requirement analysis phase.

#### TEST METHODOLOGY-

#### SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

This testing criteria required me to stop and fix any area that was 40% and above faulty. I would then precede to my next task after that was complete.

## TEST DATA

The following list includes the steps that should be taken by the user, the conditions that should be met for the successful execution of the test case, and the end result that should be met for the test cases to pass.

1. TC01: To test the Login Authentication interface
   * + Input: Email address and Password
     + Output: Valid Destination Page
     + Valid Range: User Name-Alphanumeric, Password-Alphanumeric
     + Result
     + If (User == Valid User), an order form appears to complete the checkout process
     + If (User! = Valid User), an error message is displayed on the Login interface. i.e. Username or password is Wrong

1. TC02: To test, the users can view the functionalities in their portals in the system.
   * + Description of Purpose: The system shows all the available functionalities in each portal. The user can choose make a sale or go back to continue with other operations.
     + Input: The user clicks on the intended function and the system should perform as expected. An example is a tenant should be able to view their bills.
     + Output: the system should perform as expected. In the case of the tenant checking their bills, on clicking on the bills tab, they should be able to vie the same.
2. TC03: To test, the Admin can add new cars.
   * + Description: The Admin can add new buildings into the system. The Admin can also modify the details
     + Input

i. User=Admin ii. Selection=buildings module iii. Selection= Add New building

* + - Output: New added buildings are added into the car list.
    - Result

New building added to vehicle list in the system

1. TC04: To test, the Admin can view all the users registered in the system.
   * + Description: The Admin can view all the users who are registered in the system in the database.
     + Input

i. User Name - Alphanumeric, Password - Alphanumeric ii. User==Admin iii. Selection==Manage Employee

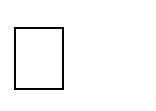
* + - Output: User List
    - Result

i. If( login type == “Admin” & Database.clicked = ‘true’ and list.clicked=true and ii. userlist.exists=true), then display users.

iii. If (login type == “Admin” &Database.clicked = ‘true’ and list.clicked=true and iv. userlist.exists=false), then display the empty database.

1. TC05: To test that users who are landlords cannot delete tenants or add tenants in the system.
   * + Description: The landlord users cannot be able to delete or add tenants in the system..
     + Input

i. User Name-Alphanumeric, Password-Alphanumeric ii. User==Users iii. Selection==delete/ add

 Output: User successfully or unsuccessfully deletes car from the system.

Result

Unable to access the delete buttons to complete delete or add buttons..

6.4 Test results

This section lists the results that were produced by running the test cases. The table lists the test cases that were used while testing the interface along with the expected result and the actual results for each test case**.**

|  |  |  |
| --- | --- | --- |
| **Test case Number** | **Expected Result** | **Actual Result** |
| TC01 | Pass | Pass |
| TC02 | Pass | Pass |
| TC03 | Pass | Pass |
| TC04 | Pass | Pass |

#### TEST COMPLETENESS

The test completeness specifies the criteria that denotes a successful completion of a test phase.

For the test that I ran, I had my test completeness as;

1. The test run was mandatory to be 100% unless a clear reason is given.
2. Pass rate is 80% achieving pass rate was mandatory.

#### TESTING TOOLS

|  |  |
| --- | --- |
| **RESOURCES** | **DESCRIPTIONS** |
| Server | Need a database server the install MySQL server web server then install Apache server. |
| Test Tool | Develop a test tool which can auto generate the test results to the predefined form and automated test results. |
| Network | Set up a LAN Gigabit and 1 internet line with the speed at least 5mb/s. |
| Computer | At least 1 computer run windows10, Ram 4GB, CPU 5.7GHZ. |

#### REASONS FOR TESTING

1. Testing helps to identify potential threats that could interfere with the functions of the system.
2. It ensures that the quality desired is achieved.
3. It also ensures that all the functionalities of the system are met and are working.
4. It increases the possibilities of finding ways to improve the system functionalities.

#### TEST RESULTS

1. I managed to obtain the following results from the tests I ran;
2. The admin can only change the password to only his account. iii. The tenants can change passwords of only their accounts. iv. The admin could update data in the system whenever necessary.
3. The system could generate the reports required by the admin.
4. The admin is able to add a new tenant and remove a tenant from the system whenever necessary.

# CHAPTER7: CONCLUSION

## ACHIEVEMENTS AND LESSONS

#### ACHIEVEMENTS

I am proud to state that managed to create a rental management system that:

1. Stores tenants, employees’ and properties data in databases.
2. Data is easily retrievable by the admin. iii. Data can be updated only by the admin whenever necessary.

#### LESSONS LEARNT

Through the development of this system I have learnt quite a number of things, the most important being;

1. Consult widely before embarking on anything, asking people who know better will help you improve on your creativity and also prevents underestimation or overestimation of the task at hand.
2. Take time to understand before starting to develop it.
3. Be consist while developing it, this helps to ensure quality uniformity throughout the system.

## CONCLUSION

My system is a rental management system. I intended to upgrade the current system used in most rental management agencies. I endeavored to create a system were all the data was safely kept and stored in database. The data would be easily retrievable and safe in these databases.

I wanted to make the rental management process easier, simpler and safe. I am happy and mostly grateful for the success I have managed in creating this system. I have managed to create what I intended in the beginning. I am sincerely grateful to everyone who helped in one way or another.

## RECOMMENDATIONS

* I recommend that the system to be checked for maintenance purposes yearly.
* I recommend that the admin to the system to thoroughly understand its functions before using.
* I recommend the data in the system to be backed up on a regular basis.

REFERENCES.

Data Collection and Analysis Tools. (n.d.). Retrieved November 2, 2019, from

https://asq.org/quality-resources/data-collection-analysis-tools.

LunalovegoodCheck. (2018, April 9). Software Engineering: Requirements Elicitation. Retrieved November 2, 2019, from https://www.geeksforgeeks.org/software-engineeringrequirements-elicitation/.

Sherrell, L. (1970, January 1). Retrieved November 2, 2019, from https://link.springer.com/referenceworkentry/10.1007/978-1-4020-8265-8\_201039.

Sridhar, J. (2018, February 12). What Is Data Analysis and Why Is It Important? Retrieved November 2, 2019, from https://www.makeuseof.com/tag/what-is-dataanalysis/##targetText=Data analysis is the process,business intelligence and data visualization.

System Analysis & Design - System Design. (n.d.). Retrieved November 2, 2019, from https://www.tutorialspoint.com/system\_analysis\_and\_design/system\_design.htm.

Unified Modeling Language (UML): Sequence Diagrams. (2018, February 12). Retrieved November 2, 2019, from https://www.geeksforgeeks.org/unified-modeling-languageuml-sequence-diagrams/.

What is a Functional Requirement? Specification, Types, EXAMPLES. (n.d.). Retrieved November 2, 2019, from https://www.guru99.com/functional-requirement-specificationexample.html##targetText=In software engineering, a functional,functions a software

must perform.&targetText=It can be a calculation,system is likely to perform.

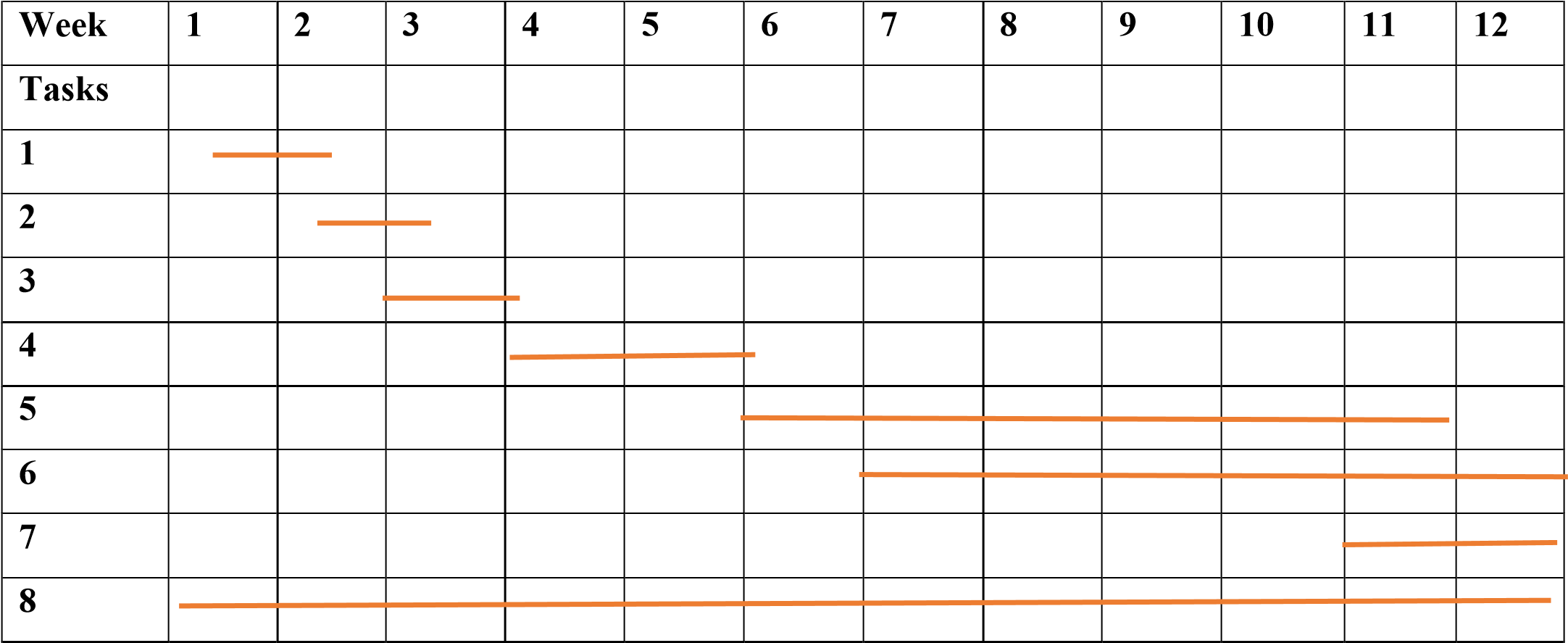
# APPENDICES.

#### APPENDIX A

## PROJECT SCHEDULE

The Gantt chart showing how the project was carried out is shown below:

Gantt chart



**TABLE PROJECT SCHEDULE**

Week 1 – Introduction to the system

Week 2 – Requirements gathering

Week 3 – System Design

Week 4 – Implementation, coding or development

Week 5 – Implementation and testing

Week 6 – Testing and Deployment

Week 7 – Maintenance Activities

**Project Budget**

|  |  |  |
| --- | --- | --- |
| **NO** | **ITEM** | **COST** |
| 1 | **Hardware** |  |
| Backup (CD’s, Flash Disk) | **1,500** |
| Router | **6,000** |
| Laptop | **30,000** |
| **2** | **Software** |  |
| Webserver; apache | **2,000** |
| IDE &Tools :aptana studio,phpmyadmin | **1,300** |
| **3** | **Other Requirements** |  |
| Internet Costs | **2,000** |
| Printing Costs | **1,500** |
|  | **Total** | **44,000** |