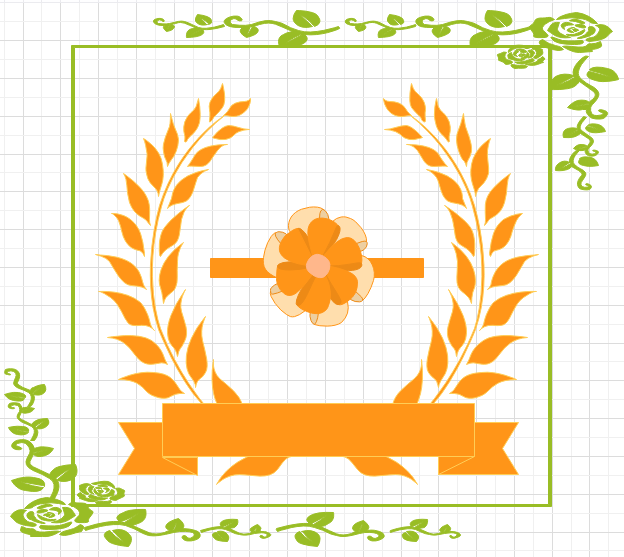
****

**Certificate of Completion**

***This is to certify that***

**THOUSES APARTMENT MANAGEMENT SYSTEM**

***Was Built, Designed and Developed By;***

|  |  |
| --- | --- |
| **Student1305288** | **AGA CALEB MGEMASHIMA** |
| **Student1327860** | **NNAMDI CHRISTUS CHINEMELUM** |
| **4Student1440454** | **NWANKWO IFECHUKWU CHARLES** |

**DATE ISSUED:**

**31stOCTOBER, 2023**

**Table of Contents**

1. **Introduction..............................................................................................................................................3**

1.1. Problem Definition.............................................................................................................................................3

1.2. Purpose....................................................................................................................................................................3

1.3. Scope.........................................................................................................................................................................3

1.4. System Overview.................................................................................................................................................3

1.5. Document Overview...........................................................................................................................................3

**2. System Requirements...............................................................................................................................4**

2.1. Functional Requirements...................................................................................................................................4

2.2. Non-Functional Requirements.........................................................................................................................4

2.3. User Roles..................................................................................................................................................................4

2.4. Hardware Requirements....................................................................................................................................4

2.5. Software Requirements.......................................................................................................................................4

**3. Design and Architecture..........................................................................................................................5**

3.1. Database Design.....................................................................................................................................................5

3.2. User Interface Design...........................................................................................................................................8

3.3. Data Flow Diagrams...........................................................................................................................................14

**4. Conclusion..................................................................................................................................................17**

4.1. Summary.................................................................................................................................................................17

4.2. Future Enhancements.......................................................................................................................................17

4.3. Acknowledgments...............................................................................................................................................17

**1. Introduction**

**1.1. Problem Definition**

The Apartment Management System is designed to address the challenges faced in managing and maintaining apartment complexes, including tenant management, property management and maintenance requests. The system aims to streamline these processes, enhance communication, and improve overall efficiency.

**1.2. Purpose**

The purpose of this documentation is to provide a comprehensive guide to the Apartment Management System, covering its requirements, design, implementation, and usage. It serves as a reference for developers, administrators, and end-users involved in the management of apartment complexes.

**1.3. Scope**

The Apartment Management System will cover the management of multiple apartment complexes, with features for tenant and property management, maintenance requests, payments, and reporting. The system is designed to be user-friendly.

**1.4. System Overview**

The system includes modules for adding tenant and management, property listing and management, a maintenance request system and reporting. It aims to streamline the operations of apartment complexes, making it easier for both residents and administrators.

**1.5. Document Overview**

This document provides a comprehensive guide to the Apartment Management System, including its requirements, design, implementation, and usage. It also covers maintenance and support processes, along with potential future enhancements.

**2. System Requirements**

**2.1. Functional Requirements**

The functional requirements define the features and capabilities of the Apartment Management System. These include tenant management, employee management, building and apartment management, keeping record of payments, apartment change, tenant maintenance requests, tenant complaint functionalities etc.

**2.2. Non-Functional Requirements**

The non-functional requirements outline system performance, security, and scalability aspects. These include response times, data security, and the ability to handle a growing number of users and properties.

**2.3. User Roles**

The system defines various user roles, including landlord, property managers and tenants, each with specific privileges and access rights.

**2.4. Hardware Requirements**

* **CPU** - Intel/AMD: 1.4GHz or higher
* **RAM** - 512 Megabytes of RAM or higher
* **Hard Disk** – 160 GB or higher

**2.5. Software Requirements**

* OS: Windows
* JDK 17
* MySQL Database
* Mysql-connector-j-8.1.0

**3. Design and Architecture**

**3.1. Database Design**

**3.1.1 apartment Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| apartmentNumber | int | 10 | primary key |
| size | varchar | 20 |  |
| aptType | varchar | 50 |  |
| rentalFee | varchar | 50 |  |
| buildingName | varchar | 255 |  |

**3.1.2 apartmentChange Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| apartmentChangeID | int | 10 | primary key |
| tenantID | int | 10 | foreign key |
| apartmentNumber | int | 20 | foreign key |
| newAptNumber | int | 20 |  |
| changeDate | date |  |  |

**3.1.3 apt\_change\_apply Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| applyID | int | 10 | primary key |
| tenantID | int | 10 | foreign key |
| currentApt | int | 10 | foreign key |
| newApt | int | 10 | foreign key |
| reason | text |  |  |
| date | date |  |  |

**3.1.4 building Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| landlordID | int | 10 | foreign key |
| buildingName | varchar | 255 | primary key |
| address | varchar | 255 |  |
| cityStateZip | varchar | 20 |  |

**3.1.5 employee Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| employeeID | int | 10 | primary key |
| firstname | varchar | 50 |  |
| lastname | varchar | 50 |  |
| password | varchar | 20 |  |
| phone | varchar | 20 |  |
| email | varchar | 50 |  |

**3.1.6 landlord Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| landlordID | int | 10 | primary key |
| password | varchar | 20 |  |

**3.1.7 lease Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| leaseID | int | 10 | primary key |
| tenantID | int | 10 | foreign key |
| apartmentNumber | int | 15 | foreign key |
| startDate | date |  |  |
| endDate | date |  |  |
| MonthlyRent | varchar | 50 |  |
| securityDeposit | varchar | 20 |  |
| status | varchar | 20 |  |

**3.1.8 maintenance Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| maintenanceID | int | 10 | primary key |
| apartmentNumber | int | 10 | foreign key |
| maintenanceDate | date |  |  |
| templateName | varchar | 50 | foreign key |

**3.1.9 maintenance\_request Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| maintenance\_requestID | int | 10 | primary key |
| apartmentNo | int | 10 |  |
| description | text |  |  |
| date | date |  |  |

**3.1.10 notificationtemplate Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| templateName | varchar | 50 | primary key |
| description | text |  |  |
| subject | varchar | 100 |  |
| emailAddress | varchar | 50 |  |

**3.1.11 payment Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| payID | int | 10 | primary key |
| payDate | date |  |  |
| payAmount | varchar | 50 |  |
| apartmentNumber | int | 10 | foreign key |

**3.1.12 tenant Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| tenantID | int | 10 | primary key |
| firstname | varchar | 50 |  |
| lastname | varchar | 50 |  |
| password | varchar | 50 |  |
| phone | varchar | 20 |  |
| email | varchar | 50 |  |
| currentAddress | varchar | 255 |  |
| cityStateZip | varchar | 15 |  |

**3.1.13 tenant\_complaint Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** |
| complaintID | int | 10 | primary key |
| buildingName | varchar | 50 |  |
| apartmentNo | int | 10 |  |
| tenant\_email | varchar | 50 |  |
| complaint | text |  |  |

**3.2 User Interface Design**

**3.2.1 Login**

The first login frame is for the tenant’s login, but the admin or in our case the landlord/manager can go ahead and click the **“Landlord/Manager Login >>>”** to login. After filing the fields with the correct data the user can only login then.

**3.2.2. Admin Frame**

The admin frame contains all the functionalities or duties to be performed by the admin, the landlord also has access to this frame, so as to promote transparency of record keeping.

**3.2.3 Apartment Frame**

Here the admin can add, update and delete apartments. The admin also has a list of the apartments in the system.

**3.2.4. Building Frame**

The admin adds, updates and deletes building and there is also a list of buildings in this frame.

**3.2.5. Employee Frame**

Employees can be added, updated and deleted in this frame, a list is also present in this frame.

**3.2.6. Tenant Frame**

Tenant are managed in this frame.

**3.2.7. Lease Frame**

**3.2.8. View Complaint Frame**

Here contains all the tenant complaints.

**3.2.9. Change Apartment Frame**

Here the admin changes apartment for tenants with the help of the tenant apartment change request list.

**3.2.10. Maintenance Frame**

The admin views the tenant maintenance request list and put into record on which apartment to maintain.

**3.2.11. Notification Frame**

In this frame, the admin creates a notification template or feedback templates for the system.

**3.2.12. Payment Frame**

Records about payments are added through this frame.

**3.2.13. Tenant Frame**

This is the tenant’s dashboard, they can access the functionalities they have access to through this frame.

**3.2.14. Make Complaint Frame**

Tenant submits their complaints through this frame.

**3.2.15. Apply For Apt. Change**

**3.2.16. Request Maintenance Frame**

**3.3 Data Flow Diagrams**

**3.3.1. Level 0**

**DFD LEVEL 0**

**3.3.2. Level 1**

**DFD LEVEL 1**

**3.3.2. Level 1**

**DFD LEVEL 1**

**4. Conclusion**

**4.1. Summary**

An apartment management system is a comprehensive software solution designed to streamline the management and operation of residential apartment complexes. This system simplifies tasks such as tenant management, rent collection, maintenance requests, and communication. It provides real-time access to property information for both property managers and tenants, ensuring efficient and transparent processes. Features often include online rent payment, maintenance tracking, lease management, and communication tools. By centralizing data and automating various tasks, apartment management systems enhance the overall efficiency and organization of property management, improving the tenant experience and reducing administrative burdens for property managers.

**4.2. Future Enhancements**

* Available of tenant online rent payment through their dashboard
* Employee Dashboard
* Mobile App
* Real-Time Communications
* Visitor Management. Etc.

**4.3. Acknowledgements**

I would like to extend my heartfelt appreciation to my team, who played a pivotal role in the success of our apartment management project. This endeavor would not have been possible without the dedication, hard work, and collaboration of our talented team members, whose efforts have brought our project to completion.

I also want to express my gratitude to our supervisors and the eProject team, whose guidance, support, and valuable insights have been instrumental in steering this project towards achievement.

In addition, I'd like to acknowledge the cooperation and contributions of my teachers and seniors here in school. Your commitment and involvement have been crucial in overcoming challenges and ensuring its smooth execution.

Together, we have achieved a significant milestone, and I am truly thankful for the unwavering commitment and collective effort that each of you has brought to this project. Your contributions are deeply valued, and I look forward to continued collaboration and future successes.