WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE

ONLINE CONDOMINIUM HOUSE MANAGEMENT SYSTEM FOR WOLKITE CITY

BY

NAME	ID
1. PHARMACY BANATA	CIR/080/11
2. BELAYNEH LEGAMO	CIR/024/11
3. GEDION ABRHAM	CIR/049/11

PROJECT ADVISOR: MR.SISAY TIGISTU (MSC.)

June 15, 2022

Wolkite University, Wolkite, Ethiopia

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE

ONLINE CONDOMINIUM HOUSE MANAGEMENT SYSTEM FOR WOLKITE CITY

SUBMITED TO DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHLER OF SCIENCE IN COMPUTER SCIENCE

BY

NAME	ID
1. PHARMACY BANATA	CIR/080/11
2. BELAYNEH LEGAMO	CIR/024/11
3. GEDION ABRHAM	CIR/049/11

PROJECT ADVISOR: MR.SISAY TIGISTU (Msc.)

Wolkite University, Wolkite, Ethiopia

June 15, 2022

DECLARATION

This is to declare that this project work which is done under the supervision of Mr.Sisay Tigistu (Msc) and having the title **Online Condominium House Management System for Wolkite City** is the sole contribution of: Pharmacy Banata, Belayneh Legamo and Gedion Abrham

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Members:

Full Name	Signature
1. Pharmacy Banata	_____
2. Belayneh Legamo	_____
3. Gedion Abrham	_____

Approval Form

This is to confirm that the project report entitled **Online Condominium House Management System for Wolkite City** submitted to **Wolkite University, College of Computing and Informatics Department of computer science** by: : Pharmacy Banata, Belayneh Legamo and Gedion Abraham is approved for submission.

---------- ---------- ----------

Advisor Name Signature

Date

---------- ---------- ----------

Department Head Name Signature

Date

---------- ---------- ----------

Examiner 1 Name Signature

Date

---------- ---------- ----------

Examiner 2 Name Signature

Date

---------- ---------- ----------

Examiner 3 Name Signature

Date

Acknowledgement

First of all, we would like to thank our God, who gives us love, patience, healthy, wisdom and ability to walk through all the problems and obstacles during the period of our study. Then we would like to thank our advisor, instructor Sisay Tigistu for his constructive opinion and willingness to participate in each part of our project and his effective direction, assistance and guidance for the accomplishing of this project.

Finally, we would like to express love, thanks, appreciation, and respect to our families and friends. Also we would like to thank the teaching staffs of Computer Science who have contributed wholly to the success of this project.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LISTS OF ABBREVIATIONS

BR	Business Rules
CCI	College of Computing and Informatics
CD	Compact Disc
CHA	Condominium House Administrator
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DBMS	Database Management System
DVD	Digital Versatile Disc
E.C	Ethiopia Calendar
G	Ground
GB	Giga Byte
G.C	Gregorian calendar
HD	Hard Disk
HTML	Hypertext Markup Language
Https	Hypertext Transfer Protocol Secure
ID_No	Identification Number
IQ	Intelligence Quotient
JS	JavaScript
Km	Kilo Meters
MD5	Message Digest 5
MySQLI	My Structured Query Language Improved
OCHMS	Online Condominium House Management System
PHP	Hyper Text Pre Processor
RAM	Random Access Memory
SMS	Short Message Service
UC	Use Case

UC_ID	Use case Id
UI	User Interface
UML	Unified Modeling Language
WKU	Wolkite University

EXECUTIVE SUMMARY

This project document deals with online condominium house management system development specifically project proposal system analysis, design, methodology and partial conclusion of online Condominium house management system. Our propose system maintain necessary information of the Wolkite housing development agency.

Online condominium house management system mainly provides effective and fast data processing, registration, notification and placement. This web based system of managing applicant house information in house development agency setting is expected to help various services keep an updated data on the status of their information. In designing and analyzing such a system, object oriented designing and analysis tool and technique has been employed.

Generally, the main goal of online condominium house management system is to shorten data-processing time, to reduce errors, to improve the accuracy of input and to provide data reliability of the information and to change the manual data handling system into automated system.

CHAPTER ONE

1. INTRODUCTION

Now a day, most people are having familiarity with computer and computer based applications. Many organizations and individuals have their computer based applications for the purpose of running their business, to perform different activity. The aim of this project is to develop an online condominium house management system in Wolkite city [1].

Wolkite city administration has opened an office named Wolkite Housing Development Agency which has Housing Transfer and Administration department. Currently, the office/agency/ process data manually, and this manual processing system has many drawbacks. Hence with a gear of solving these setbacks we are going to develop an online condominium housing management system.

After completion the project is expected to solve the problems that are affecting the condominium housing development agency offices. Since it is an online system it will reduces costs, time to travel to the offices, work overload and it also minimizes the space used to store the data. Besides, it enables applicants' online registration, search, update, applicants' data, and placement and register finished condominium houses.

1.1. Background of the Organization

Wolkite is a town which founds 158 km from Addis Ababa. Wolkite has its own administration structure to organize, control and manage the local communities. In addition to this, in the town there are governmental and non-governmental organizations which facilitate the development of the town and provide services to the community. The Ethiopian government polices introduce a lot of strategies to address problem that citizen are facing. Housing the houseless people is one of the basic strategies which ha were proposed around 1997 E.C. House is one of the basic necessities for human beings. However, it is a critical problem for thousands of poor people in Ethiopia. Ethiopia's urban centers are characterized by poorly developed economic base. Like most urban centers of developing countries, most cities and towns including Wolkite in Ethiopia face an overabundance of problems, including a critical and ever-worsening housing shortage. Wolkite Housing Development office is one of an office which found under Wolkite city administration office. It was established in 1999

E.C. The five (5) areas in which building is constructing are “Sebat bate mazoria”, “G & B site”, “and 20 kuteba”, “70 kuteba” and “stadiyom girge”. All building are “G+2”. The integration of computer technology in condominium management service is necessary for providing housing service effectively and efficiently through a better management at all levels of office. This shifts to automation by the integration of operations using computer technologies to save record time and improving reliability of ordering and obtaining results. The computer system does provide an excellent means of securing data of the agency [3].

- The office has the following missions: -
  - Provision of affordable housing for low and medium income of the city inhabitant and ensuring resident security.
  - Improve the leaving standard of Wolkite city resident by expanding employment creation.
  - Provide premium condominium and commercial units that will be result to enhanced value of the client’s lives resources and overall business
  - Improving small industries to big industries.
- The office has the following Visions
  - The office encourage and support an association of peoples who voluntarily work together to increase property values and promote the spirit of respect, caring, community, family, and friends.
- The office has the following Values:-
  - The offices values are built upon the agencies’ mission and vision, and include: creating long-term value, integrity, community, competency, Justice, Kindness, loyalty, fairness, faith, and growth.

1.2. Statement of the Problem

Currently the office is using manual handling system in its day to day activities which has many problems.

These problems are:-

- **Problem in distributing condominium house:**-Since they do manually it’s difficult to know the occupied class and unoccupied class surely.

❖**Problem in document management:-**since there are a lot of documents in the office it's hard to manage such huge data manually.

❖**Data Redundancy:-**Since there is no organized data base there is problem of giving more than one house for a single person.

❖**Lack of accuracy and loss of document**:-Since registration is handled manually there is chance of getting two houses for single family like husband and wife and also loss of applicant and contractor data.

❖**Lack of security:-**Since the office use manual system, the mechanism of data handling is unsecured.

❖They need large number of human resource to process office job.

❖The manual lottery system is error prone, vulnerable to corruption.

1.3. Objectives of the Project

In this project we can specify the objective of the project as general objective and specific objective.

1.3.1. General Objective

The general objective of this project is to develop Online Condominium house management system for Wolkite city.

1.3.2 Specific Objectives

The specific objectives of our project will be:-

❖Analyze the limitations of the existing system

❖Suggest alternative solutions and selecting the best among the alternative.

❖Analysis a system which is secured, efficient and user friendly

❖Design the system that will solve the current problems and provide reliable functionalities.

❖Implementation of the proposed system in efficient way

❖Giving recommendation on further studies to be conducted on the areas of condominium house management system

1.4. Feasibility Study

Most of information system projects have budgets and deadlines, and analysis of factors for the feasibility forms the business case (analysis of assumptions like resource availability and potential problem and system costs and benefits) that justifies the expenditure on the project.

The feasibility factors of our project will be:-

1.4.1. Economic Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system. So, our industrial project will be economically feasible because its anticipated benefit is greater than the expected cost.

- **Tangible benefit**

Tangible benefits will fit the following classes:

- ✓ Increase speed of activities
- ✓ Error reduction
- ✓ Improvement of management

- **Intangible benefits**

Intangible benefits are as follows:

- ✓ It automate the work load of the Wolkite Condominium House management agency.
- ✓ Work flow will be efficient and effective
- ✓ Increase security.
- ✓ Cost Reduction.

1.4.2. Technical Feasibility

Here you are expected to analysis the project in terms whether the required technology is available or not, whether the required resources are available, Manpower- programmers, testers & debuggers, and Software and hardware. Our proposed system will be evaluated from the technical point of view. The assessment of this feasibility must be based on an

outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Through the technology may become outdated after some period of time, due to the fact that new version of some software supports older versions, the system may still be used. So, there are minimal constraints in this project, and it in turn indicates that the solution is technically feasible.

1.4.3 Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. And also, does management support the project? Are the users not happy with current business practices? Will it reduce the time (operation) considerably?

So, our industrial project will be operationally or timely feasible due to the:-

- It determines how the proposed system will satisfy the need of Wolkite city house development agency, and it also offers Secure, accurate and efficient system to the organization. The system in which we are developing is also compatible to all operating systems and web browsers.
- By conducting operational feasibility study we examined whether the new project will attain its desired objective, we also understand the degree to which the proposed system will likely solve the problems, so in this study we identify that operational feasible that the system is user friendly, easy to access. It can be run in any operating system.

1.5. Scope and Limitation of the Project

This project will focus on the development and implementation of web application based on condominium management system for Wolkite city which solve the existing problem regarding to the Wolkite city Housing management agency.

1.5.1 .Scope of the Project

The system will be used for any online condominium house management system. But, we will develop this system for the Wolkite city online condominium housing management system, so it will be use by others by customizing the system.

1.5.2. Limitation of the Project

Our system is limited only in the process condominium house management of the Wolkite city.

The system does not include the following**:-**

- It does not manage house contractor information,
- It does not provides budget allocation for construction of house
- It does not includes mechanism like digital signature.
- It does not draw lottery automatly
- The system does not include SMS gate way that used to send message from computer to mobile.

1.6. Significance of the Project

Online condominium house management System has significance both in societal and technological for any users.Becuase of its accuracy, timeliness, reliability, security, and relevance and Valuablility data are needed for condominium house management in all dimensions.

❖This project will be reduces work overload of condominium house development office workers.

❖Make the work environment favorable, information dissemination from agency to applicant.

❖The system will helps the Wolkite city condominium house development office workers to handle the applicant effectively and supports the smooth functioning of the business.

❖The system increase document preservation without need of large area.

❖This project will provide a basis for validation and Verification process.

❖Fast accessibility of stored data and saving of resource.

1.7. Beneficiary of Project

The Beneficiary of our industrial project will be:-

- ✓ **Condominium agency manager**s :-it will decrease work load,time,increase security
- ✓ **Users /applicants**:-it will help them to save their time wisely, to access and use data easily
- ✓ **Developers**:-it will use as references

1.8. Methodology of the Project

1.8.1. Data Gathering Techniques

For the collection of data we will use techniques like Observation, Interview and Document analysis.

❖**Observation**:-Observation is common methods of scientific research to collect the data. We used observation to know how the existing system work, to know exactly how different sub offices and how office member are handling the work in the office.

❖**Interview**:-Interview is particularly useful for getting the history behind the participant's experiences. We used interview to get information about the existing system for developing our project. The interview was conducted on the head of city condominium house management office and staff members.

❖**Document Analysis**: - Document analysis is used to understand how the system is working. We used this method to know all about the staff mission, vision, function and overall of their work in short and brief.

1.8.2. System Analysis and Design Techniques

In the project analysis and design phase we will use unified modeling language (UML).The data analysis model applied in this project will be an object-oriented approach. Because it based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods Object oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of

classes, which also determine their types. So, for designing purposes an object-oriented designing will be applied in this project. Since it is easy to maintain if any error is occurred.

We will use some system designing techniques for our project development. Some of them are:-

- Use case diagram
- Sequence diagram
- Activity diagram
- Class diagram
- State chart diagram
- Component diagram
- Deployment diagram
- Package diagram

1.8.3. System Development Model

System development method is a frame work that is used to structure, plan and control the process of developing a system so to develop this project we have select the waterfall model which is a classical model used in system development life cycle to create a system with a linear and sequential approach. The reasons why select this are:-the model develops systematically from one phase to another in a downward fashion, divided into different phases and the output of one phase is used as the input of the next phase and every phase has to be completed before the next phase starts and there is no overlapping of the phases.

1.8.4. System Testing Methodology

Before and after the completion of the system we test the system we do, before deployment of the system. It helps to make sure that the system meets the require goal. So we will use unit testing, integration testing and acceptance testing.

1.8.4.1. Unit Testing

We use unit testing at the source code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness. In this level of testing process we will test as [4].

- Check whether the return type of functions is correct.
- Check how the sub procedures or functions are called correctly.
- Check if the correct output is produced for different inputs.
- Check the efficiency of the code with respect to the memory and CPU time.

1.8.4.2. Integration testing

This test shows the new module integrate or work together with the existing one to achieve the goal of the system.

1.8.4.3. Acceptance testing

The purpose of acceptance testing is purely to ensure that the end user can achieve the goals set in the business requirements. Rather than focus on functionality of specific features, acceptance testing involves reviewing the feature-complete application flow and end-to-end experience.

1.8.5. Deployment Tools of technology

- The deployment hardware/software are:
  - ✓ Desktop
  - ✓ Laptop
  - ✓ Any browser
  - ✓ Network cables

1.8.5.1. Front end Technologies

- ✓ They work on what the user can see.
- ✓ Those technologies are:-
  - Hypertext Markup Language (HTML),

- JavaScript (JS): - It is very interesting language used to validate data and develop different messages. We used to validate our data which we used in html code.
- Cascading Style Sheets (CSS), which we used in this project work.

1.8.5.2. Back end Technologies

✓ They includes server, application and database that work behind the scenes to deliver information to the user.

❖**Back end technologies**

- MYSQLI: - this software was used to create the database of the project.
- Xampp(Cross-platform, Apache, MySQLI, PHP and Perl)server
- PHP for server side coding: The reason why we choose PHP it is object-oriented, free, and compatible with many platforms, compatible with many DBMS and much secured scripting language

1.8.5.3 Documentation and Modeling Tools

➢ We will use the following tools to document ,model and implement our system:-

- Microsoft word 2013
- Microsoft PowerPoint 2013
- Microsoft excel 2013
- Draw io

1.8.5.4 Deployment Environment

✓ This system that will deploy and provide reliable functionalities for Wolkite city condominium housing agency.

1.8.5.5 Hardware requirement

❖ We will use the following different hardware for our works:-

➢ **Any Desktop Computer**

- Highest processor speed and latest CPU, 4GB RAM, Hard disc 500 GB

- **Flash Disk and CD**: -used for the movement of data from one machine to another. We used both CD and flash disk when we move our data from one machine to another. Both devices will around 8GB - 32GB.
- **Laptop**:-This will be essential in case of no light or in any place rather than laboratory.
- **Network cable: -**was used to get the internet access by connecting internet line from internet switch to computer for further reading and search of information from internet.

1.9 Document organization

The document organization of our project is well organized and documented. The project documentation has five chapters. Chapters' organization is based on the WKU CCI industrial project guidelines.

Chapter one contains the introduction part of the project: - It describes the statements of the problems, objectives, significances, beneficiaries, methodologies and approaches that we use are described well as necessarily.

Chapter two contains the description of the existing system: - It describes activities and major functions, users of the system and the business rules or the work flow as well as the drawbacks of the existing system.

Chapter three contains the proposed system: - in this chapter the functional and non-functional requirements of the system are described well. All the security and performance related issues and hardware and software requirements, the error handling mechanisms, and quality related concepts are also discussed in this chapter.

Chapter four contains System Analysis: - in the system analysis section all the system model (use-case model and diagrams, use-case descriptions and scenarios), object model (class diagram and data dictionary) and the dynamic model (sequence, activity and state cart diagrams) of the system are described clearly and precisely.

Chapter five contains system design: - the system design section discussed about the design goal, system architecture, system decompositions, software and hardware mapping and

packages, detailed class and consistent data management also described in this section. The abbreviated words used in the document are presented with their description in a table form for clear understanding.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

After studying the existing system, we also determined the requirement or the feature that must be included in the proposed system. Furthermore by analyzing the current system, we could also estimate how the propose system solve the setbacks of the existing system.

2.2. Users of Existing System

In the existing system, there are four users which give and take informations related to condominium houses.

Those are:-

- **Kebele' Offices:** provide the following service for the resident.-Register resident that need condominium house. It passes all the resident applicant information to the city administrator office of housing development agency.
    - **Wolkite City Municipalities Office**: The condominium housing management offices ask information about the customer to the Wolkite city municipality's offices and the offices responsibilities is checking whether any house is register by the customer name or by his family. If any house is registered by the customer the applicant's will not accepted. If there is no house registered by the applicant the condominium housing management offices permits to get the house of condominium house for the applicant.

- **Wolkite City Condominium House development Office:**

  Collect applicant data that is already registered in all kebele's of the city and receive the constructed condominium houses from Condominium houses construction agency. Then draw the lottery in order to get a condominium houses. When the winner of the lottery pays the money, pre-payment or full payment. After these, this agency in turn distributes the condominium house for the applicant that won the lottery (balance with the number of the constructed house). The office also gives services, maintenance and infrastructure to the applicant.

- **Applicant**: In the existing system, applicant gave their Kebele informations to Wolkite city Condomium agency to apply for house. And after that he/she followed all necessary information from start up to final process by going to agency physically always .And finally, if the offices notify new announcement he/she saw on annocement board physically.

2.3. Major Functions/Activities of the Existing System

Here under we have listed several activities of the current system that are operated Manually:-

- **Registering applicant**: Before applying for a house, the residents of each kebele's; of the city have to register for house in the offices of Wolkite City Condominium House development . People eligible for registration are ones who have been saving money under the 10/90, condominium schemes but still have not received their homes and people who are able to pay cost of the house as an advance payment.
- **Storing applicant's data:** Using paper document files the employees of the organization store the data manually, so it can be easily lost and requires a lot of storage space
- **Announcement**: Public offering statement in the common elements of the condominium. The one who are paying the payment according to the rule can be announced by different ways, i.e. manual notification by using notice board, by SMS send to the Kebele officers and phone calls.

- **Control the status condominium houses.** The condominium board is supposed to give status for applicants according to the rules and regulation by using different techniques.
- **Placement of lottery winners**. The lottery winners can get their placements by checking the status of the organizations time to time and by following the notice board timely.

2.4. Drawbacks of the Existing System

The existing system has many problems. Among others

- Lack of accurate data
- The system has problems related to security.
- It takes long time to search data
- Lack of efficiency, effectiveness
- Since it operable manually it uses large amount of space to store data
- Lack of portability's
- Redundancy of data are the most important one.

2.5. Business Rules of the Existing System

BR 1. The applicant for house should be above 18 years old.

BR 2. The applicant should be from Wolkite city having the renewed Keble ID card or should live for at least 6 month.

BR 3. Women have the opportunity to get the house 20 %(affirmative action).

BR 4. The applicant should have the capacity to pay some amount of money from total Payment according the following Programme type

BR 4.1: if the Programme type was 10/90 he/she must pay pre-payment 10% to get house

BR 4.2: if the Programme type was 20/80 he/she must pay pre-payment 20% to get house

BR 4.3: if the Programme type was 40/60 he/she must pay pre-payment 40% to get house

BR 5. If the applicant has a Condomium house registered by one of the following was not allowed to take it:-

- by him/her wife/husband and
- by their name before.

BR 6. The applicant who are disabled have two priority:-

- Chance to get the condominium house and
- Chance to get the ground floor.

BR 7. Applicant must select one of the following house type according to his/her interest:-

- studio,
- one bed room,
- two bed rooms and
- Three bed rooms.

CHAPTER THREE

3. PROPOSED SYSTEM

The proposed system was designed in order to reduce the problem arise in existing system. This system has users with different privilege.

Applicant in proposed system is any resident of Keble who had a renewed Kebele ID with full information can applicant can register for house by going to the agency first physically .And after that he/she follow up on online all process except lottery drawn which done manually.

The **condominium house manager** performs different activities on this system. Some of the activities that are performed by the condominium house manager is registering houses that are finished and ready to give services for the applicant with their block number and address that the house is constructed also by entering related information about house.

After that by equating the number of houses available and applicants, the system display lists of applicant to draw lottery manually to select winners from the applicants. Then next to this step, winner's notification is send to the winners of house by their e-mail address.

The condominium house manager can view information related to house, sites, blocks and winners of condominium and other information's related to condominium house.

The condominium house manager will have applicant payment status controlling page on the Wolkite OCHMS and on this system, the administrator can view and control who are going to pay for condominium house.

The **system admin** who manage the system at all and had privileges to manage the system's account by creating, deleting and updating.

Purpose of the proposed system provides:-

- ❖**Efficient way of applicant's data management**:-Since the proposed system uses database system there is no loose of data .The applicant information will highly secured, the search and update of applicant will be simple.
- ❖**Efficient distributing method:** - This means our system uses efficient way to distributing Condominium houses for the applicant.
- ❖**Give online information**:-The proposed system gives online information about Condominium house for applicant.
- ❖**Give's Notification information**:-The proposed system will be expected to gives notification through posting in the website and providing the notification information in the applicant user account

Requirements analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional *and* Non-functional requirements

3.1. Functional Requirements

Functional Requirements are those that refer to the functionality of the system, i.e. what services it will provide to the user. It also provides how the system should behave in particular situations.

These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. So, the functional requirements of this system are:-

- ❖ **Manage Accounts:** The system administrator can Create Account, Delete Account and update account to all users.
- ❖ **Make Payment:** Before registration for the condominium house the applicant can process payment by using his/her bank account number on specified bank.
- ❖ **Apply for House:** The system must allow applicants' to search their bank account number to register, update and delete the information.

- **View Notifications:** The system should be able to give notification through the opened user account of the applicant and admin.
- **Register finished house:** The system should allow condominium house manager to register finished Condominium house.
- **Manage Applicants:** The system able the condominium house manager to Create, delete and update the applicant information when it is needed.
- **Evaluate Applicants:** The system should allow condominium house manager to evaluate applicants' status to display list out applicants for lottery.
- **Generate Applicants list for Lottery:** The condominium house manager could display lists of applicants for lottery and give to national lottery to draw lottery manually to get winner applicants.
- **Send Commits:** the system can post news and notice for the people in order to make them informed by the condominium house manager and applicant.

3.2. Non-functional Requirements

Non-functional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately. They are constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc. So, the non-functional requirements of this system are:-

- **User interface**: The system provides PHP user interfaces that are compatible with windows platform
- **Hardware consideration**: The organization should have computers having typical storage capacity and processing speed.
- **Error handling**: Our system handles error by showing the message" invalid input" when the user enter invalid input.
- **Security**: The system should have a security privilege that secures the system. And also there must be a physical security hat secures (especially) the server computer. That means the server computer is only allowed for the server admin. The admin of the system change yearly.

❖**Performance characteristics**: The end user computer should have medium processor and the server computer should have large processor. It's measured by its speed of processor.

❖**Back up:-**The system should have back up using external hard disk. The backup is taken weekly.

3.2.1. User Interface and Human Factors

User interface (UI) is any method or means by which the end-user of a product interacts with, or controls a product, software or hardware device. It is designed to allow humans to control machines effectively and efficiently.

From the good user interface design Criteria we choose the following for this system:-

- **Visibility of system status.** Users should always know where they are and what's going on.
- **Real world - system match.** This system should mirror the real world of the user as much as possible.
- **Control and freedom.** It clearly marked exit, undo, and redo functions. Don't force them into a long linear sequence of operations with no escape.
- **Consistency and standards.** Use objects and phrases consistently. Here is a User Interface Consistency Checklist: - interface, appearance, text characteristics, navigation and etc**.**
- **Recognition not recall.** Provide visual objects, actions, and options to assist the user for navigation and input activities. Don't expect they will memorize commands.
- **Effective error handling.** Assist users to recognize, diagnose, and recover from errors. Don't just tell them there's an error, suggest corrective action whenever possible.
- **Error prevention.** A design that prevents errors from occurring is better than a good error message.

There are different types of user interfaces but the main are:

- **Form-based user interface/ Form-fill interfaces:** Used to enter data into a program or application by offering a limited selection of choices. It also consists on screen forms displaying fields containing data items or parameters that need to be communicated to the user.
- **Graphical user interface:** is a system of interactive visual components for computer software. It has elements like window , menu, dialog box ,check boxes, buttons, label buttons, radio buttons, text boxes and the like.

So, our system uses Form-based user interface or Form-fill interfaces and Graphical user interface.

- The level of expertise of the users
  - The main goal of good User interface designer is to identify user is a beginner, an intermediate user or an expert. This understanding may help designer to design system as per designated user requirements. So, our system users do not belong to similar group of expertise as they are mixed bunch of people. These categories can be decided from their age, IQ level, skills they have and experience.Becuase of that our system is usable for what level of expertise or users. Why, because it is simple to use or interact with system.

3.2.2. Hardware Consideration

- The hardware that is consumed by the new system is basically a computer machine that is used for all the operation of the system. The Software product to be developed should run on existing standard computers. This system is portable that can run on any type of computer and it supports any type of browsers.
- This Website opens in any personal computer supporting different types of Microsoft Windows, Ubuntu (Ubuntu is an operating system that is a distribution of Linux itself) and other smart phone. Because this system is design and implement by PHP server-side scripting Language that support platform independent, the proposed system is compatible for any hardware; it can run on any platform.

3.2.3. Security Issues

This proposed system provides a high level of security by authenticate authorized user and encrypt the user's username and password through MD5 encryption to prevent the readability of username and password in the database. The system also provides restriction in using system functionality and information access by its user.

3.2.4. Performance Consideration

There might be many users accessing to the web server simultaneously. As an online Condominium Housing tool performance shouldn't be affected much and response time for submitted page should be less than a minute.

- The Performance consideration of this system are:-
  - **Response Time**

    Upon request for user inquiry the system under normal condition should display results as quickly as possible.
  - **Processing Time**

    Since the system is developing with efficient programming language and database upon request for user's activities the system under normal condition should process the request as quickly as possible.
  - **Work ability**

    The system should be suitable for all users. It should be accurate in performing its functions and secured enough from attacks by external users.
  - **Correctness**

    The system provides correct response to the correct request.
  - **Concurrent Processing**

    Because the proposed system is web based can execute multiple users.
  - **Concurrency**

    The system should support multiple accesses of users. It should give service to multiple users concurrently.

The above listed performance consideration of this system have been available because this system has developed by using PHP server side scripting language which can increase a system performance and also MySQLI query optimization using indexes the query response time is very fast.

3.2.5. Error Handling and Validation

- Errors could rise from users and from the system. Errors that occurred from the wrong doing of users will be handled by appropriate exception handling mechanisms. Generally, if an error occurs, this system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error and should display an error message if the user inputted the characters that are mismatched to corresponding data types.
    - ✓ **Incorrect input**: this system handles many exceptions like inserting empty string to the database and inserting a duplicated customer id, inserting incorrect customer name and display an appropriate message for each error.
    - ✓ **Login error**: this system shall handle an attempt to login with incorrect username and password and display error message.
- So, in generally when the users of this system interact with the system error may appear. To control these inaccuracies, this system has made validation and display different messages to the user. This system also handle exceptions like input mismatch exception such as interchanging numeric and characters, username and password mismatch by displaying alerts.

3.2.6. Quality Issues

- **Availability:** This system should have to be functional at any given time. There is no delay in the availability of any information, whatever needed, can be captured very quickly and easily by using more than one server. The server should be always on to be available. The system shall support the use of multiple users at a time and understandable by all users.

- **Maintainability**: This System should be easily maintainable in case of problems and gives consistent service at all times without fluctuation until there is internet connection and should be testable.
- **Reliability**: This proposed system will minimize crash during its runtime, since more than one user could use the system simultaneously. The system should be reliable and matured enough in giving its service. It should have a fault tolerance mechanism in which it can recover fast from problems that may occur. It also provides reliable and accurate information which help in increasing efficiency of the process and for ease of management decision. There could be multiple parameters to meet for achieving reliability which depends on server infrastructure, database implementation tool.
- **Robustness**: This proposed system has an ability of tolerating error that may affect the system functionality.

3.2.7. Backup and Recovery

- In order to prevent data loss in case of system failure, the result of houses that were finished are drawed till then have to be saved in the database, for the system to resume the drawing lottery on reboot.
- The another method that reduce data loss and other risk there we use a frequent and full back up mechanism to avoid any information loss using copy of the system to restore when hardware and software failure is occur. In addition to this we use different storage devices like Hard Disk (HD), CD, and DVD to duplicate the data.

3.2.8. Physical Environment

- This condominium management system deployed for Wolkite City Administration.
- The proposed system we are going to develop can be affected by the physical environment when a natural disaster occurs. On the other hand external conditions such as weather condition will not have any effect on the performance of the system. The new system will be affected by internal physical environment like attack by viruses, worms and the like.

To overcome this problems the team member develop the backup (recovery) system to protect the data when the system damage.

3.2.9. Resource Issues

As our system is a web-based or online application it should use less resource.

3.2.10. Documentation

Documentation is any communicable material that is used to describe, explain or instruct regarding some attributes of an object, system or procedure, such as its parts, assembly, installation, maintenance and use. Documentation is often distributed via websites, software products, and other online applications.

There are different types of documentation but in our system we have used:-

- **System documentation** this system has represent documents that describe the system itself and its parts. It also comprised of detailed language, illustrations and photos that help different people understand the software, and it is essential reference material. It includes requirements document, source code document, quality assurance documentation, software architecture documentation, solution instructions and a help guide for advanced users. It further divided into **internal** and **external** documentation.
    - **Internal documentation**: This system contains program source code.
    - **External documentation:** It also includes the outcome of all of the

      Structured diagramming techniques, such as data-flow and entity-relationship diagrams.
- **User documentation** this system also covers manuals that are mainly has prepared for end-users of the product and system administrators. Such as training manuals, user manuals, release notes and installation guides.

CHAPTER FOUR

4. SYSTEM ANALYSIS

System analysis is the process of collecting and interacting facts, identifying the problems and decomposition of a system into its components [6]. The analysis model contains the functional model, object model and the dynamic model. The functional model can described by use case diagram.

4.1. System Model

System model is the process of developing abstract models a system, with each model presenting a different view and perspective of the system. The system model will help the system analysts to understand the functionality of the system and it also used to communicate with customers [8].

We use the following diagram to describe different model interactions:-

- Activity diagram to show the data proceeding flow.
- Use case diagram to show the interactions between a system and its environment.
- Sequence diagram to show the interaction between actors and the system and between the system components.
- Class diagram to show the object classes in the system and the association between classes.

- State chart diagram to show how the system reacts to internal and external events.

4.1.1. Use Case Model

A use case model is a model of how different types of users interact with the system to solve a problem. It describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in the satisfying goals. A use case model consists of a number of model elements. These important model elements are use cases, Actors and the relationships between them [9].

4.1.1.1. Use Case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system [5].

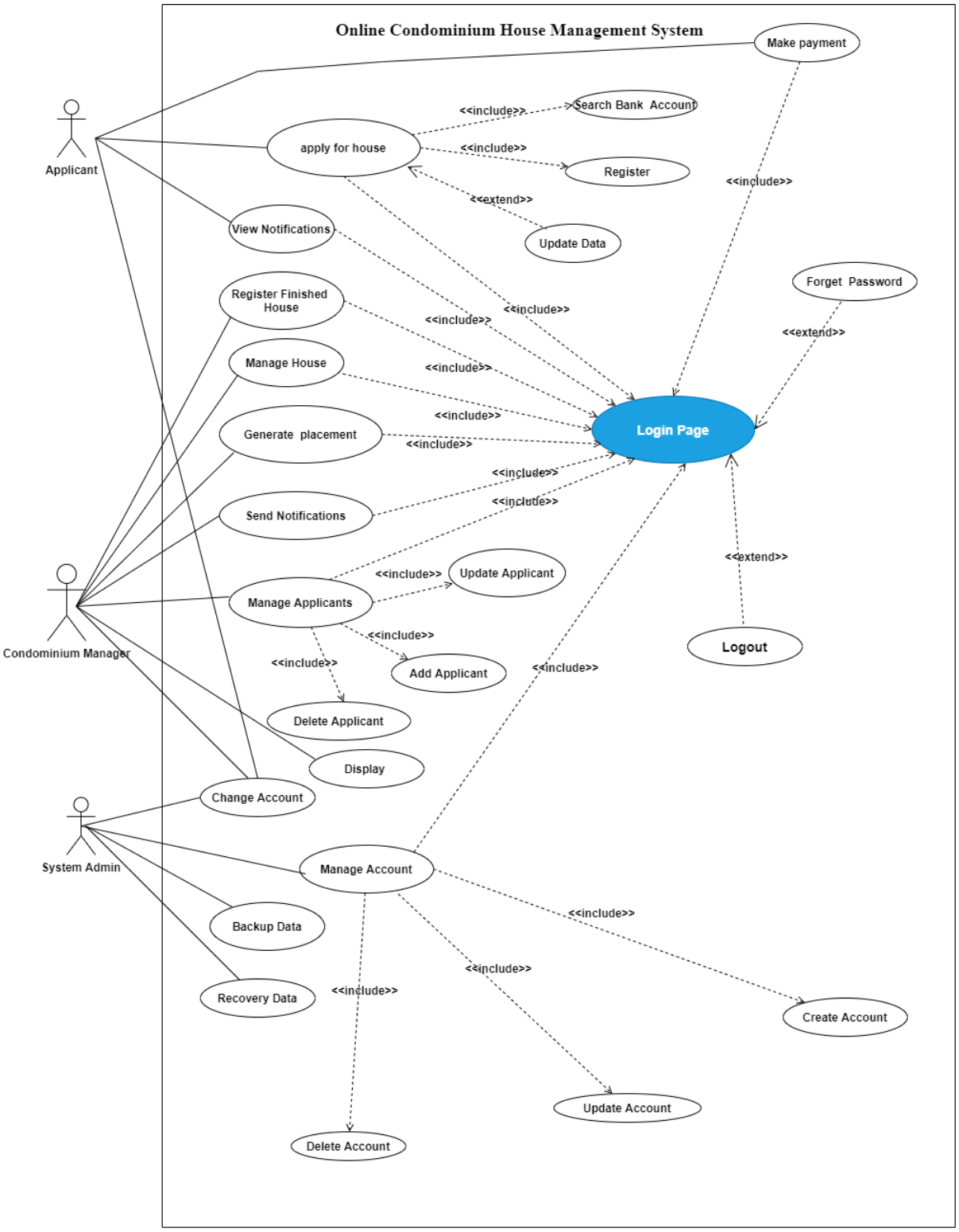


Figure 4. 1 General Use Case Diagram

4.1.1.2. Use Case Description

A use case description is a text-based narrative of a functionality comprised of detailed, step-by-step interaction between the actor and the system. It describes the outcomes of an action taken to accomplish a specific goal. It also details different paths that can be followed by defining primary, alternate, and exception flows [10].

Table 4. 1 Use case description for login use case

Use case Name	**Login**	
UC_ID(Use case id):	**UC_01**	
Actor:	**Users** (condominium manager, System administrator and applicant)	
Description:	It describes how the actors log into the system in order to use the system functionality.	
Pre-condition:	Browser must open and connection must available.	
Normal flow:	**Actor action**	**System response**
	Step 1: The user open Home page **Step 3.** Click login form **Step 5:** The user fills his or her username and password. **Step 6:** click submit button.	 **Step 2:** The System displays the Home page **Step 4:** The System displays the login page form **Step 7** :The system check the validation of username and password **Step 8:** the system displays login successful if the user enter the correct inputs **Step 9:** Use case ends.
Alternative course of action	8.1 If the user enters invalid username or password. i. The system displays invalid username or password. Please enter valid username or password. ii. The Actor repeats step 3 from the normal flow. 8.2 If the user leaves the username or password fields empty. i. The system displays username or password field cannot be empty message. ii. The Actor repeats step 3 from the normal flow.	

Post-condition	The Actor can have an access to his/her account.

Table 4. 2 Use case description for manage account

Use case Name	**Manage account**	
UC_ID:	**UC_02**	
Actor	System admin	
Pre-condition	The user must login into the system**(UC-01)**	
Normal flow:	**Actor action**	**System response**
	Step 1. The actor clicks on account Button **Step 3.** The user enter Account key to search the Account **Step 5**. The actor performs the task he/she want	**Step 2.** The system displays the account page **Step 4.** The system search result accurate to the key and display **Step 6.** The use case ends.
Alternative Course of action	4.1.If the user input the invalid account i. The system displays error message Please enter valid account. ii. The user repeats step 3 from the normal flow. 4.2: If the Actor leaves the account empty. i. The system displays the account cannot be empty message. ii. The Actor repeats step 3 from the normal flow.	
Post-Condition	The account successfully update or delete or create.	

Table 4. 3 Use case description for Make payment

Use case Name	**Make Payment**
UC_ID:	**UC-05**
Actor:	**Applicant**

Description:	Applicant who apply for condominium pay the payment of the house first.	
Precondition:	**a**. The user must login to make payment **(UC_01)** and b.The applicant must have account number on specific bank	
Normal flow:	**Actor action**	**System response**
	Step1: Click payment link from registration form **Step3:** fill payment information. **Step 4:**click pay button	**Step2:** display payment page **Step5:** The System validate inputs and display sent successful message. **Step6:** the use case end
Post condition	The condominium house manager must control payment process.	
Alternative course of action	5.1.If the applicant inputs the incorrect information of payment i. The system displays invalid informations Please enter valid information of payment ii. The user repeats step 3 from the normal flow. 5.2: If the applicant leaves the information of payment empty. i. The system displays the information of payment cannot be empty message. ii. The applicant repeats step 3 from the normal flow.	

Table 4. 4 Use case description for apply for house

Use case Name	**apply for house**	
UC_ID:	**UC_06**	
Actor:	**applicant**	
Description:	This use case done by the applicant in order to register new applicant for getting condominium house.	
Pre-condition:	1.The user must login to fill register forms (**UC_01**) 2. He/she must have an internet connection	
Normal flow:	**Actor action**	**System response**
	Step 1: Click apply for house **Step3:** applicant enter his/her bank account number to search. **Step4:**click search button **Step 6**:fill the registration	**Step 2:** The system displays the dialogue box to search bank account number. **Step 5:** If the account number found the system display registration form

	form **Step 8:** click register button	**Step 7**: the system register successfully **Step 9**: The use case end
Post-condition:	The applicant must be registered successfully.	
Alternative course of action	5.1: If the applicant enter uncreated bank account number i. The system displays not found account number Please enter correct account number ii. The applicant repeats step 3 from the normal flow. 5.2: If the applicant leaves the account number dialog box empty. i. The system displays the account number dialog box cannot be empty message. Please fill account number ii. The applicant repeats step 3 from the normal flow. 5.3: If the applicant enter invalid bank account number i. The system displays invalid bank account number Please enter valid account number ii. The applicant repeats step 3 from the normal flow. 8.1: If the applicant fills the incorrect informations ’ i. The system displays invalid applicant informations Please enter valid informations ii. The applicant repeats step 6 from the normal flow. 8.2: If the applicant leaves the informations empty. i. The system displays the information of applicant cannot be empty message. ii. The applicant repeats step 6 from the normal flow.	

Table 4. 5 Use case description for register finished house

Use case Name	**Register finished house**	
UC_ID:	**UC_07**	
Actor:	**Condominium house manager**	
Description:	Condominium house manager register house that are finished or ready with their full description	
Pre-condition:	The user must login to register the applicant information (**UC_01**), new finished house must be existed.	
Normal flow:	**Actor action**	**System response**

	Step 1: Click house registration button **Step 3:** Register finished condominium house information **Step 4:** click submit button	**Step 2:**display house registration form **Step 5:**display register successful **Step 6:** the use case end
Exception Condition:	If there is no new finished condominium house, the condominium house management administrator stop or pause the use case	
Post condition	The entire new houses that are finished must be registered or recorded with their full description	
Alternative course of action	5.1: If the manager inputs invalid informations 'of house i. The system displays invalid informations Please enter valid informations of house ii. The admin repeats step 3 from the normal flow. 5.2: If the manager leaves the informations empty. i. The system displays the information of house cannot be empty message. ii. The manager repeats step 3 from the normal flow.	

Table 4. 6 Use case description of Evaluate applicant

Use case Name	**Check applicants' status**	
UC_ID:	**UC_08**	
Actor:	CH manger	
Description:	This use case done condominium house manager evaluate all applicants' status including the payment process of applicant.	
Precondition:	The condominium house manger must login to the system (**UC_01)**	
Normal flow:	**Actor action**	**System response**
	Step 1:CHA manger click evaluate applicants' link **Step 3:** CHA manger selects applicants to evaluate. **Step 4**: the manager overview the status and prepare list of applicant for lottery based on	**Step2:** the system displays the evaluation applicants' page **Step5:**the use case end

	business rules	
Post condition	The CHM manager should view all the applicant data.	

Table 4. 7 Use case description for Generate placements

Use case Name	**Display lists of applicants for lottery**	
UC_ID:	**UC_09**	
Actor:	**Condominium house manager**	
Description:	Condominium house manager display lists of applicants for lottery, by considering the finished new house	
Pre-condition:	1.The condominium house manager must login to the system (**UC_01)** 2.The status of all applicant must be checked**(UC_08)** 3.There must existence of finished house	
Normal flow:	**Actor action**	**System response**
	Step1: Click display lists link **Step 3**:Print lists of applicant and gave to national lottery to draw lottery manually .	**Step 2:** The system display lists of applicants for lottery **Step 4:** finish task and the use case end
Exception Condition:		
Post-condition	The system must draw lottery effectively for the applicant.	
Alternative course of action	**-**	

Table 4. 8 Use case description for manage applicant

Use case Name	**Manage applicant**	
UC_ID:	**UC_10**	
Actor:	**condominium office manager**	
Description:	This use case is done by the CHM manager when they need update, search and delete applicant information	
Precondition:	The manger must login to the system to manage applicant (**UC_01)**.	
Normal flow:	**Actor action**	**System response**
	Step 1: the manger click manage applicant link	**Step 2:** the system display the manage applicant page

	Step 3: the manger can delete applicant and confirmation of resident **Step 4**:click submit button	**Step 5**: the system display applicant information **Step 6**: Submitted successfully
Alternative Course of action		1. If the applicant withdraw his or her saved money or finished its tasks. i. Admin will delete the applicant from the system. ii. The system will display applicant is successfully deleted message.

Table 4. 9Use case description for send Notifications

Use case Name	**Send news**	
UC_ID:	**UC_11**	
Actor:	**Condominium house manager**	
Description:		
Precondition:	The user must login into the system**(UC-01)**	
Normal flow:	**Actor action**	**System response**
	Step1: The user click notification link **Step3:** fill the notification **Step 4:**repeat information the process end **Step5:** click send button	**Step2:** display notification page **Step6:**display sent successful message **Step7:** finish notification and the use case end
Post-condition	The entire information could be notified	
Alternative Course of action	Not available.	

Table 4. 10 Use case description for View notification.

Use case Name	**View Notification**	
UC-ID	**UC-12**	
Actor:	**applicant**	
Pre-condition	The user must login into the system**(UC-01)**	
Normal flow:	**Actor action**	**System response**

	Step 1:The user click notification link	**Step 2**: The system shows notifications. **Step 3**:Use case end
Alternative Course of action	Not available.	
Post-Condition	The user can view the new notification	

Table 4. 11 Use case description for Log out

Use case	**Logout**	
UC-ID	**UC-13**	
Actor	**All system users**	
description	After finishing their task each user must be logout of the system	
Pre-condition	User login to the system(**UC-01)**	
Normal flow:	**Actor action**	**System response**
	1. If the user wants to logout. 2. Click logout button.	3. System logout the user. 4. End of the use case.
Alternative course of action	Not available.	
Post -condition	User Logged out from the system	

4.1.1.3. Use case Scenario

Use case scenario describes the real world examples how the user of the system interact with the system. It describes the steps and action events takes place to interact with the system [7].

It can be very detailed and indicates exact work flow within the user interface as well it describes the high level business actions of the system.

Scenario name: User Login

Participating Actor: Mr. Pharmacy

Initial assumption: browser must open and connection must available.

Normal flow of events:

- The Mr. Pharmacy open Home page
- The System displays the Home page
- Mr. Phramacy Click login form
- The System displays the login page form
- Mr. Phramacy fills his or username and password.
- Mr. Phramacy click submit button.
- The system check the validation of username and password entered by Mr. Phramacy.
- the system displays login successful if the Mr. Phramacy enter the correct inputs display
- Use case ends.

Alternative flow of events:

- If Mr. Phramacy enters invalid username or password.

  i. The system displays invalid username or password.

  Please enter valid username or password.

  ii.Then Mr. Phramacy repeats step 3 from the normal flow.

- If Mr. Phramacy leaves the username or password fields empty.

  i. The system displays username or password field cannot be empty message.

  ii.Then Mr. Phramacy repeats step 3 from the normal flow.

Scenario name: House registration

Participating Actor: Mr. Belay

Initial assumption: the Home Page must be displayed first.

Normal flow of events:

- Mr. Belay click on register link on the Home page.
- The system displays the registration form.
- Mr. Belay enter information on the registration form.
- Mr. Belay click register button to send/submit information.
- The system validates the information entered by Mr. Belay.
- Mr. Belay click on yes to submit information.
- The system displays registration is successful message.
- The system returns Mr. Belay to the Home Page.
- Use case end.

Alternative flow of events:

- If Mr. Belay enters invalid data type the system displays invalid data type message.
- If Mr. Belay leaves a field empty the system displays a field cannot be empty message.

Scenario name: House registration

Participating Actor: Mr. Gedion

Initial assumption: the Home Page must be displayed first and Make payment.

Normal flow of events:

- Mr. Gedion click apply for house
- The system displays the dialogue box to search bank account number.
- Mr. Gedion enter his bank account number to search.
- Mr. Gedion click search button
- If the account number of Mr. Gedion found the system display registration form
- Mr. Gedion fill the registration form
- The system register successfully
- Mr. Gedion click register button
- The use case end

Alternative flow of events:

If Mr. Gedion enter invalid bank account number

i. The system displays not found account number of Mr. Gedion

He enter correct account number

ii. Then Mr. Gedion repeats step from the normal flow.

4.2. Object Model

An object model is a logical interface, software or system that is modeled through the use of object- oriented techniques. It enables the creation of an architectural software or system model prior to development or programming. An object model is part of the object-oriented programming life cycle.

It also refers to a visual representation of software or systems' objects, attributes, actions, and relationships. The basic factors of an object model are classes and objects .Each object in the system has data structure and behavior. This factors share the same features as a grouped to a class.

4.2.1. Class Diagram

Class diagram, one of the most commonly used diagrams in object-oriented system, models the static design view for a system. The static view mainly supports the functional requirements of a system – the services the system should provide to the end users.

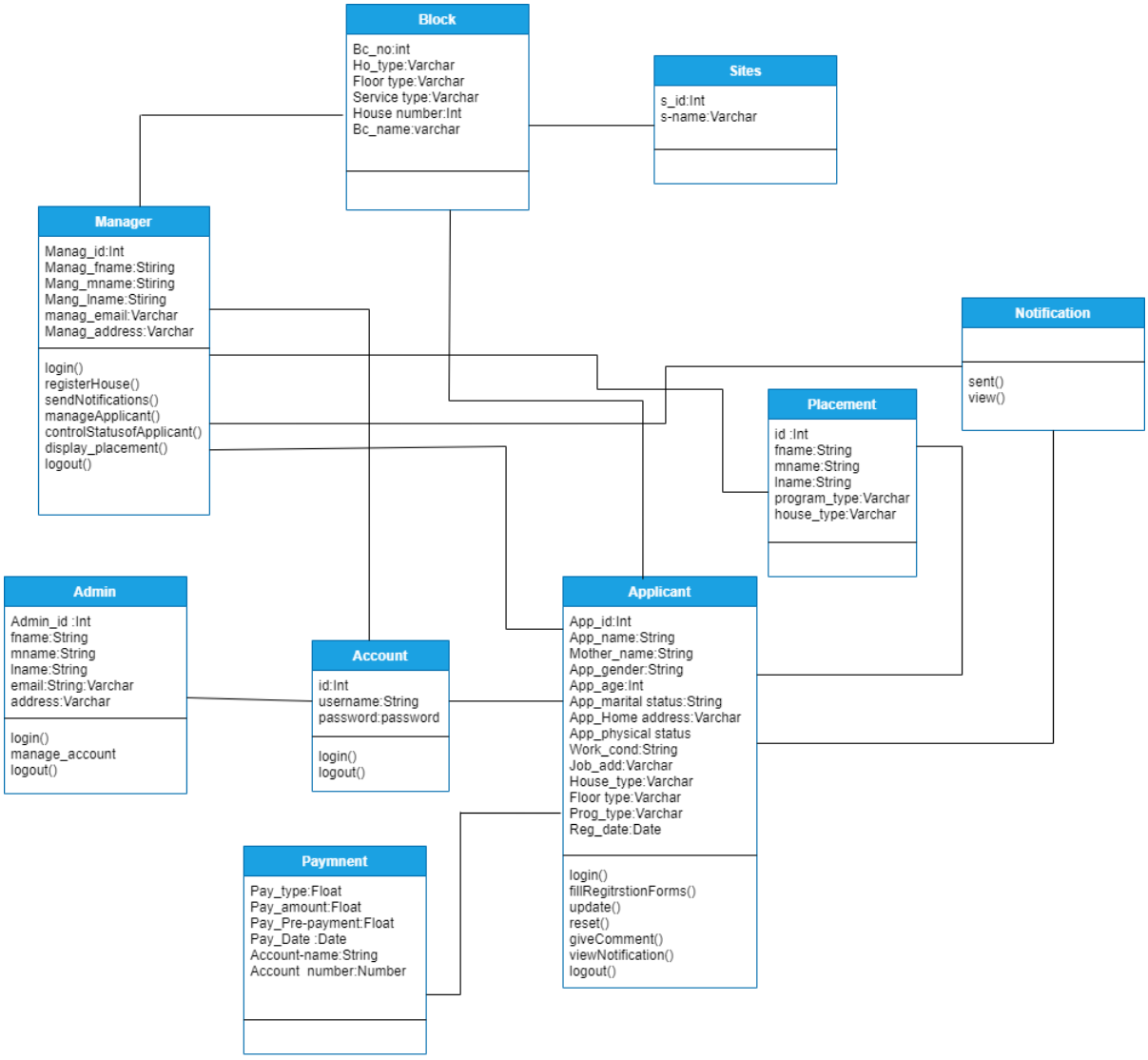


Figure 4. 2 Class Diagram

4.2.2. Data Dictionary

Table 4. 12 CHM manager

Attributes	Description	Data type	Data size	Key constraint
Manag_id	Manager identification number	Integer	10	Primary key
Manag _fname	Manager first name	String	25	Not Null
Manag _mname	Manager middle name	String	25	Not Null

Manag _lname	Manager last name	String	25	Not Null
Manag _email	Manager email address	Variable character	15	Not Null
Manag _address	Manager address	Variable character	30	Not Null

Table 4.13 Applicant

Attributes	**Description**	**Data type**	**Data size**	**Key constraint**
App_id	Applicant id	integer	10	Primary key
App_age	Applicant age	integer	7	Not Null
App_name	Applicant name	String	10	Not Null
Mother_name	Applicant mother	String	20	Not Null
App_gender	Applicant gender	String	10	Not Null
App_marital status	Applicant marital status	String	30	Not Null
App_Home address	Applicant Home address	Variable character	35	Not Null
App_physical status	Applicant physical status	String	40	Not Null
Work_cond	Applicant work condition	String	40	Not Null
Job_add	Applicant job address	Variable character	35	Not Null
Prog_type	Programme type	Variable character r	15	Not Null
Reg_date	Registration date	Date	20	Not Null

Table 4. 14 Payment

Attributes	**Description**	**Data type**	**Data size**	**Key constraint**
Pay_type	Payment type	Float	10	Not Null
Pay_amount	Payment amount	Float	20	Not Null

Pay_Pre-payment	Pre-payment	Float	10	Not Null
Pay_Date	Payment date	Date	10	Not Null
Pay_Account-name	Account name	String	25	Not Null
Pay_Account number	Account number	Number	15	unique

Table 4. 15 Account

Attributes	description's	Data type	Data size	Key constraint
id	Account id	Integer	10	Primary key
username	User name	String	20	Not Null
password	Account password	password	10	Not Null

Table 4. 16 Sites

Attributes	Descriptions	Data type	Data size	Key constraint
S_id	Site id	Integer	10	Primary key
S_name	Site name	Variable character	20	Not Null

Table 4. 17 Block

Attributes	Descriptions	Data type	Data size	Key constraint
Bc_no	Block number	Integer	10	Primary key
Bc_name	Block name	Variable character	20	Not null
Ho_type	House type	Variable character	20	Not Null
Ho_Floor type	House floor type	Variable character	20	Not Null
Ho_Service type	House Service type	Variable character	25	Not Null
Ho_number	House	Integer	10	Primary key

	number			

Table 4. 18 Placement

Attributes	Description	Data type	Data size	Key constraint
Pl_id	Placement id	Integer	10	Primary key
f_name	First name	String	15	Not Null
m_name	Middle name	String	15	
l_name	Last name	String	15	
Programme type	Programme	Variable character	40	Not Null

4.3. Dynamic Model

4.3.1. Sequence Diagrams

Sequence diagrams, commonly used to model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

In simpler words, a sequence diagram shows different parts of a system work in a ‘sequence’ to get something done [11].

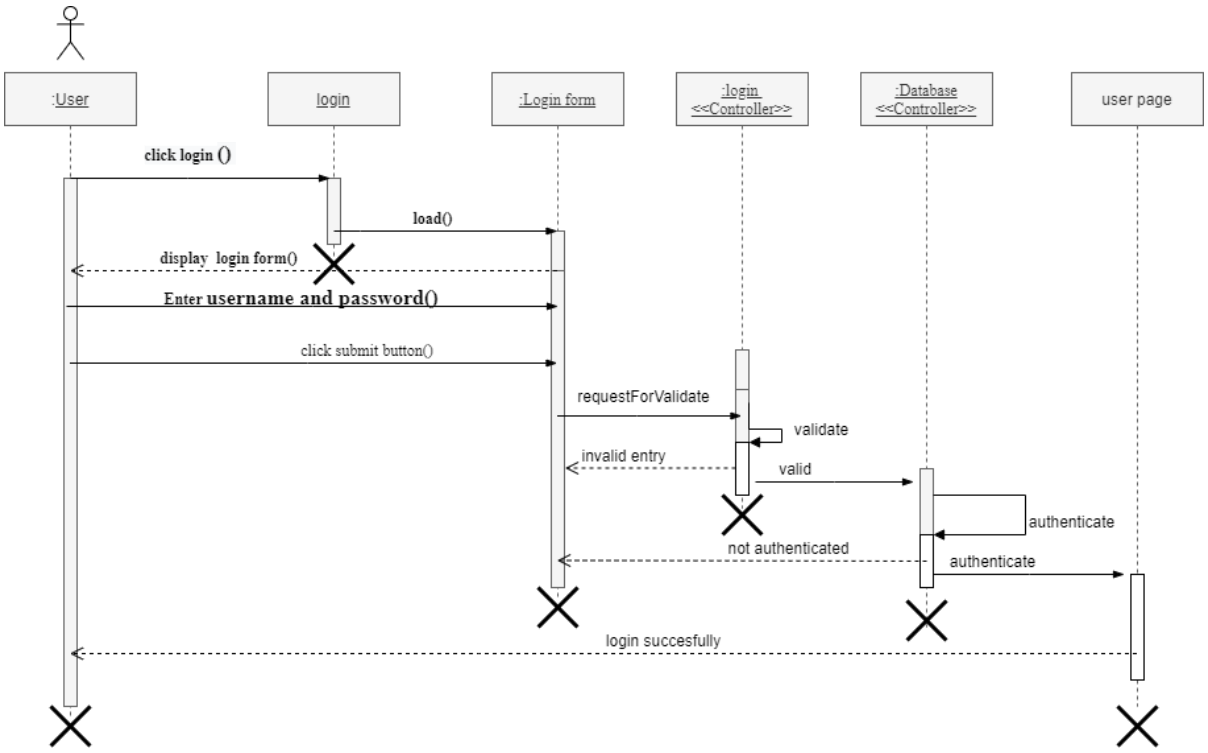


Figure 4. 3 sequence diagram for login

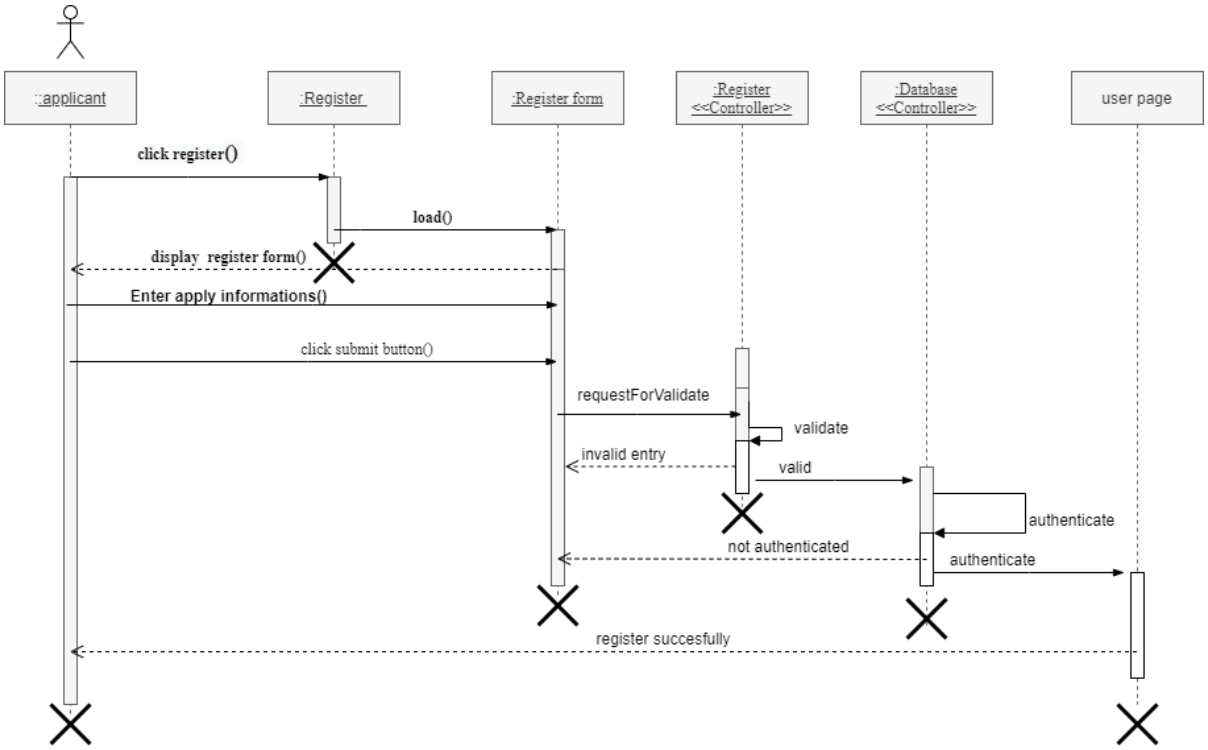


Figure 4. 4 sequence diagram for applying for house

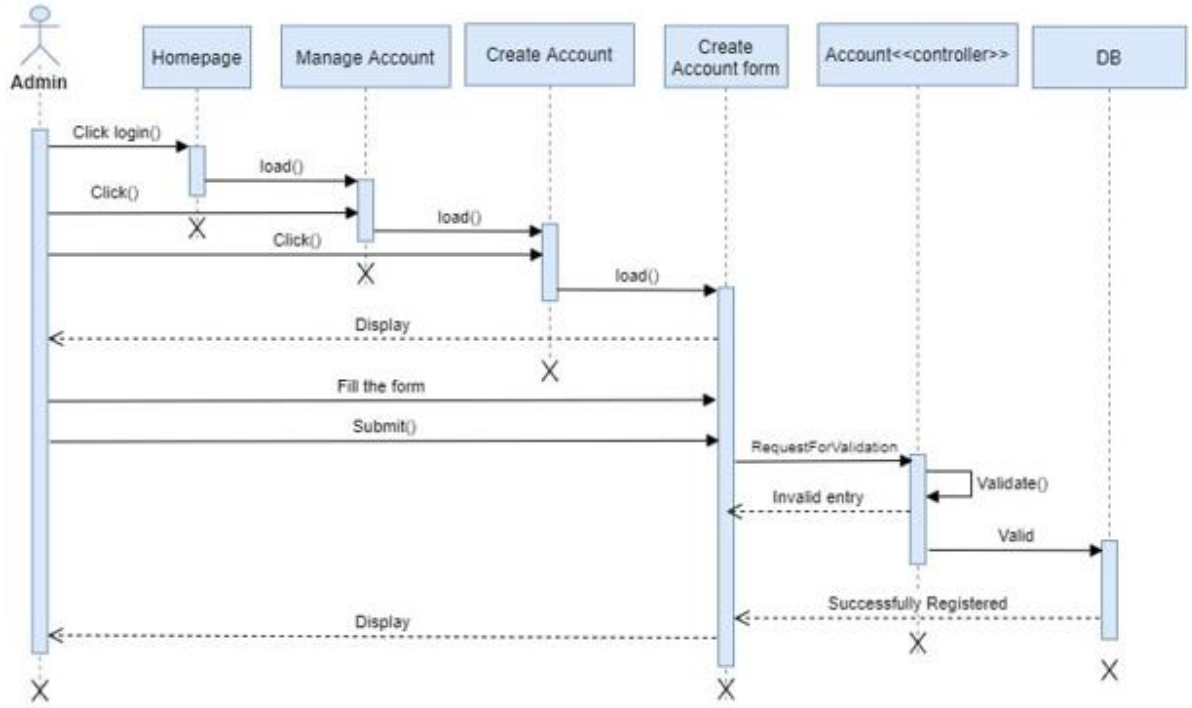


Figure 4. 5 sequence diagram for register finished house

4.3.2. Activity Diagram

It is one of UML Diagrams which illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. So, it basically depict workflows. It also focuses on condition of flow and the sequence in which it happens [13].

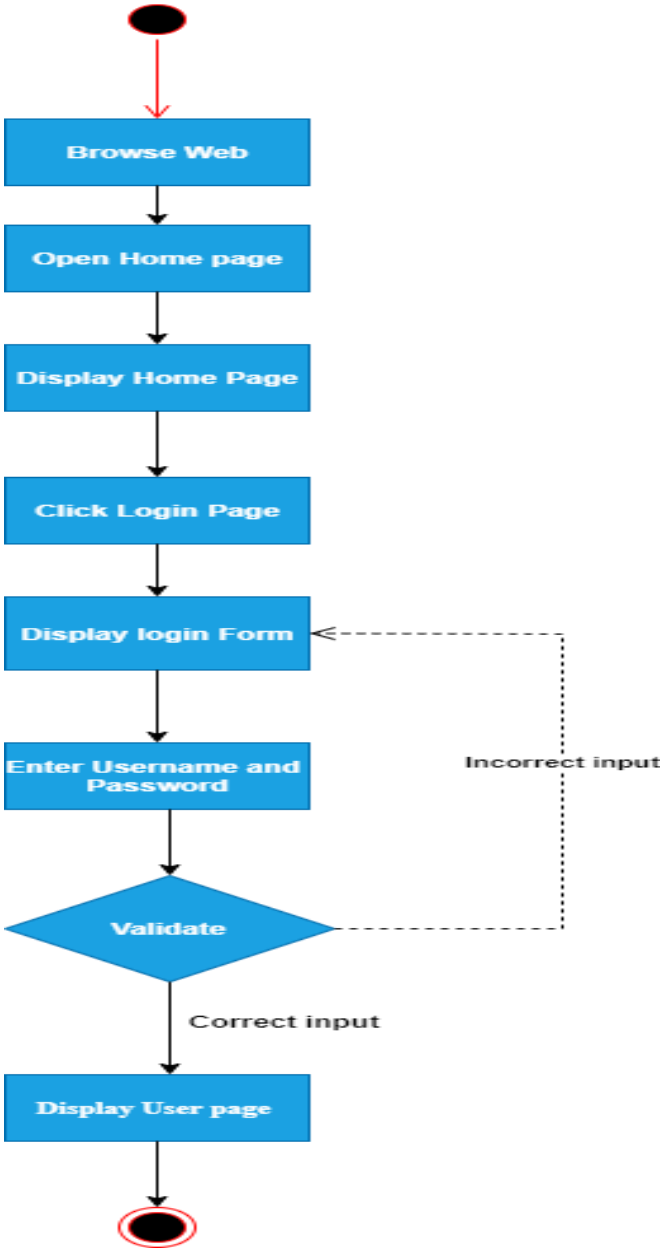


Figure 4. 4 Activity diagrams for login

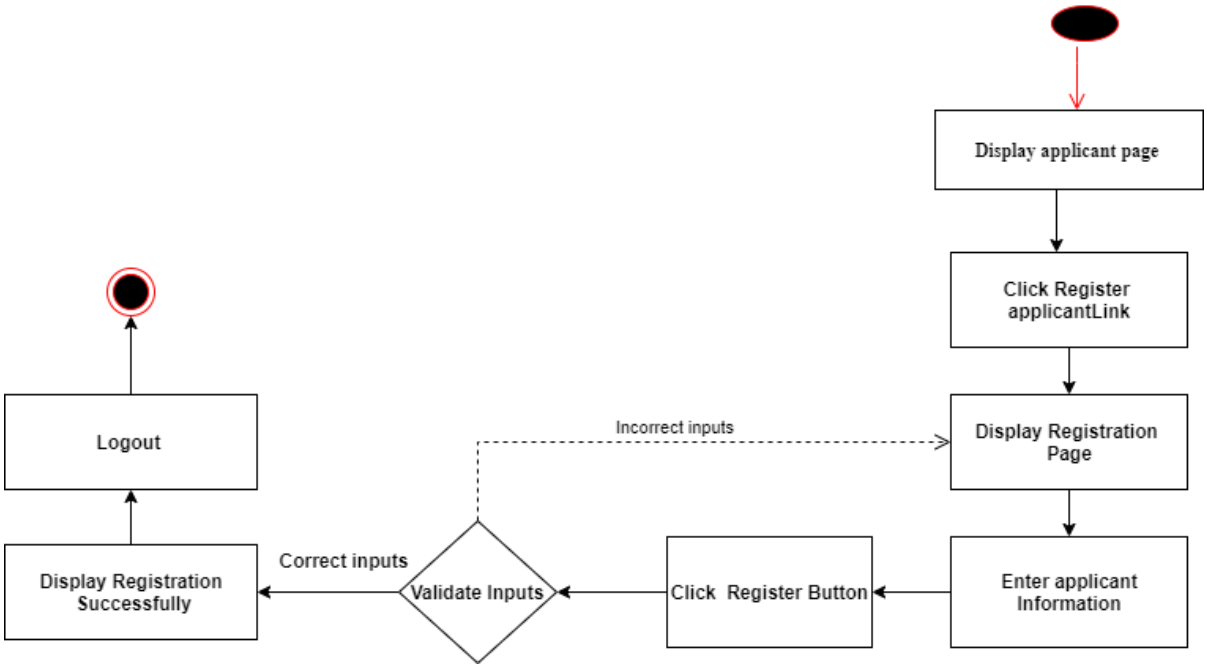


Figure 4. 5 Activity diagrams for registering applicant

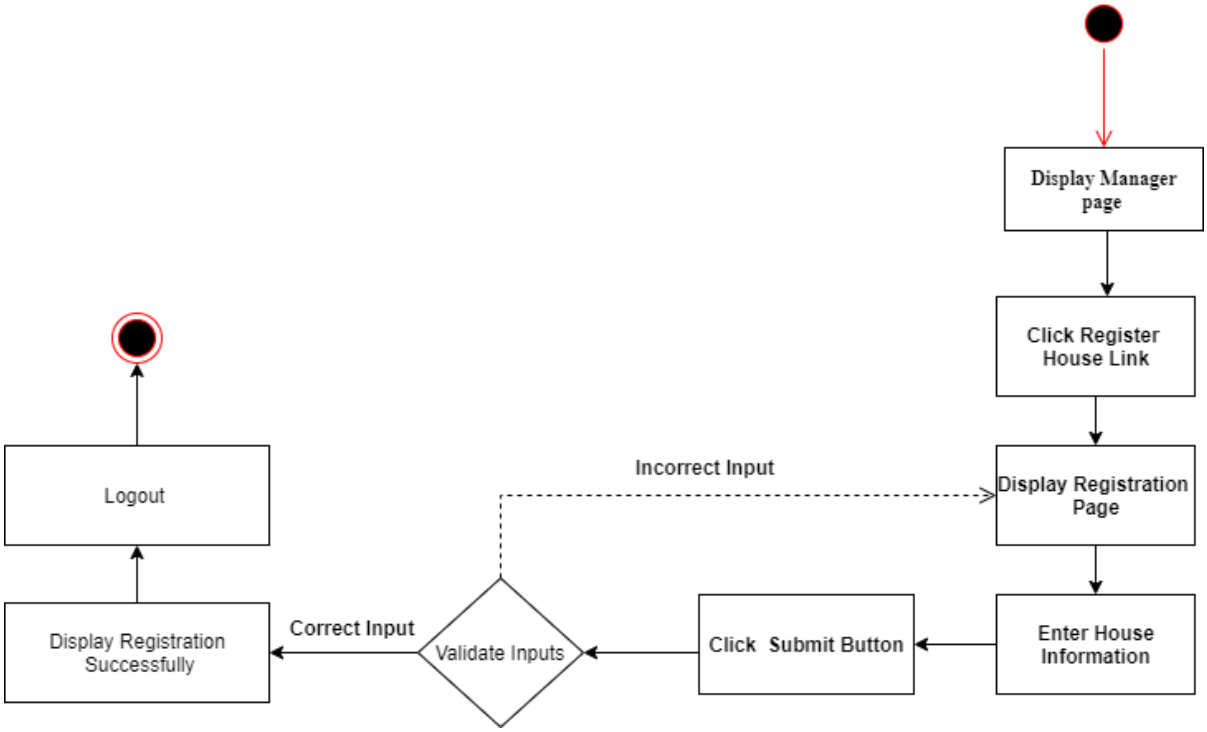


Figure 4. 6 Activity diagrams for registering house

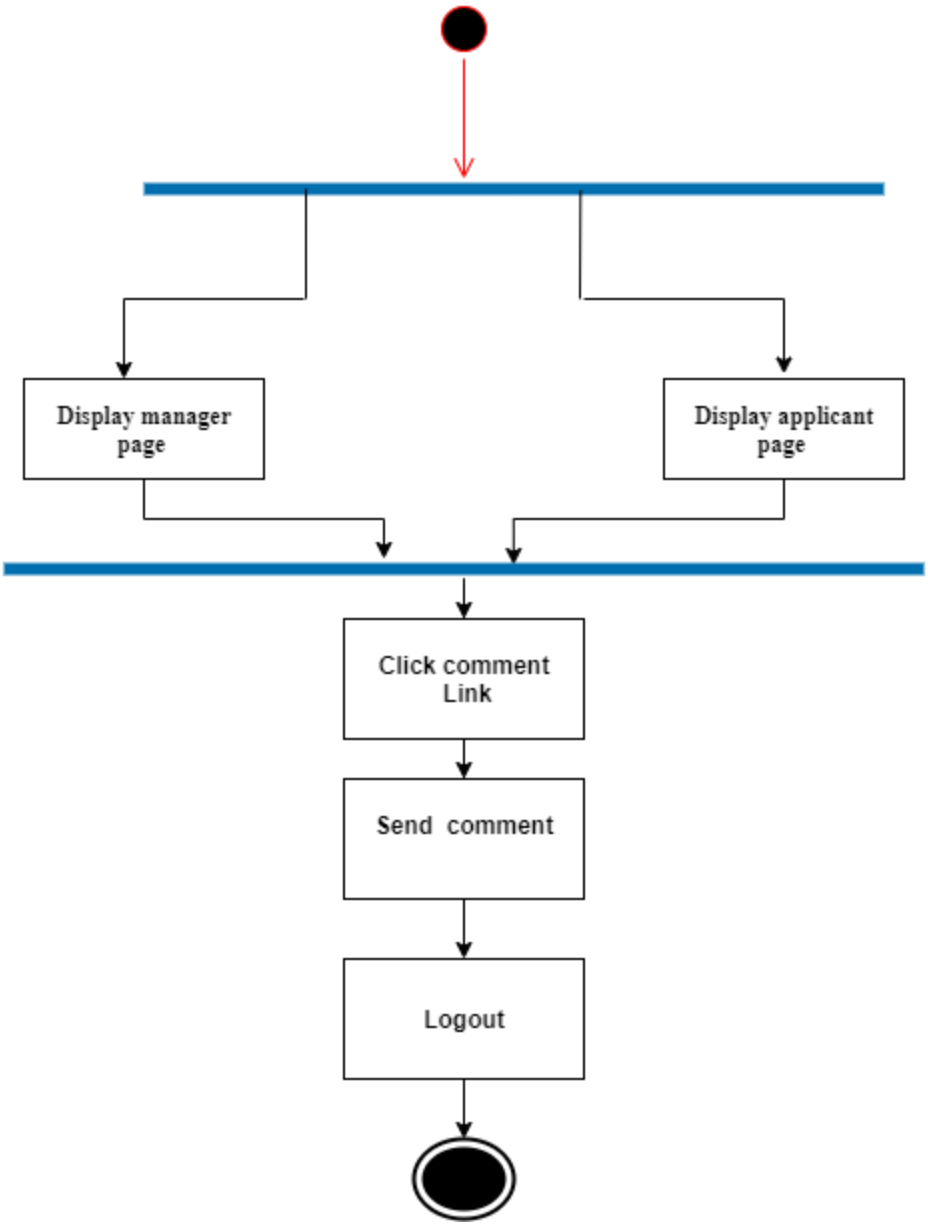


Figure 4. 7 ***Activity diagrams for Sending comments***

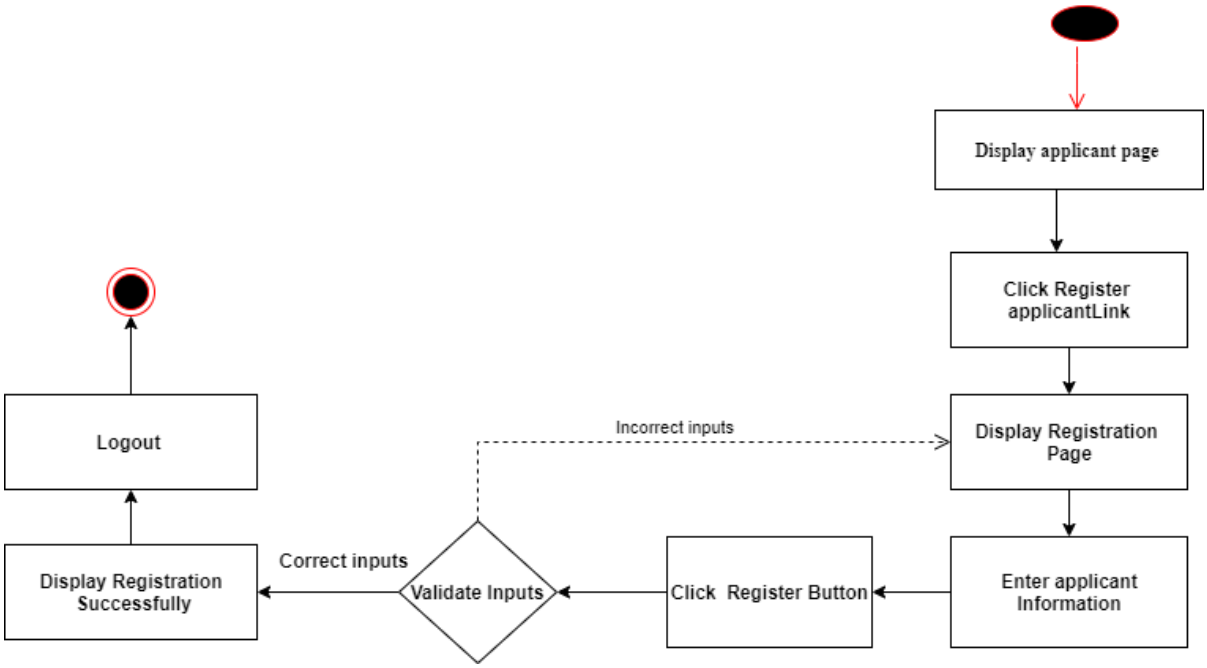


Figure 4. 8 Activity diagram for deleting applicant

4.3.3. State Chart Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions [12].

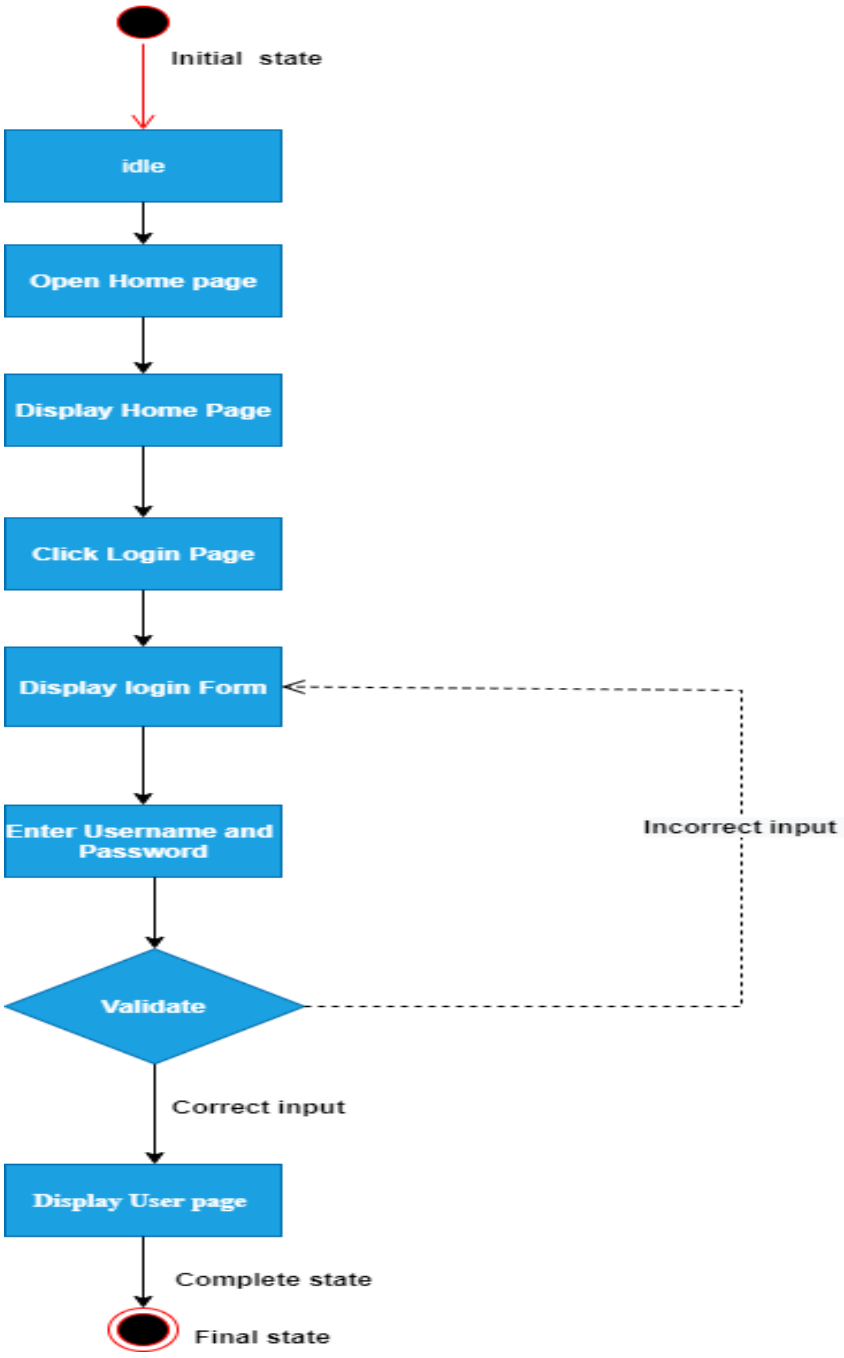


Figure 4. 9 ***State chart diagram for login***

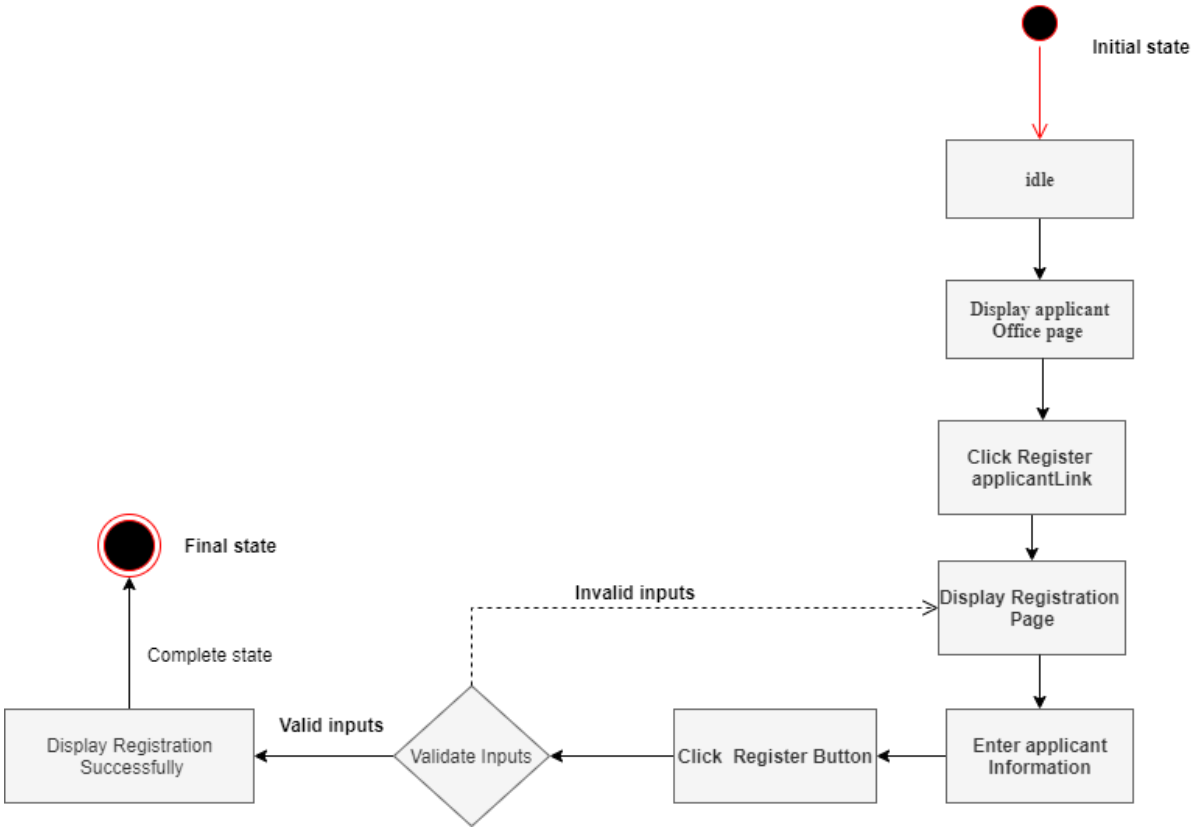


Figure 4. 10 State chart diagram for register applicant

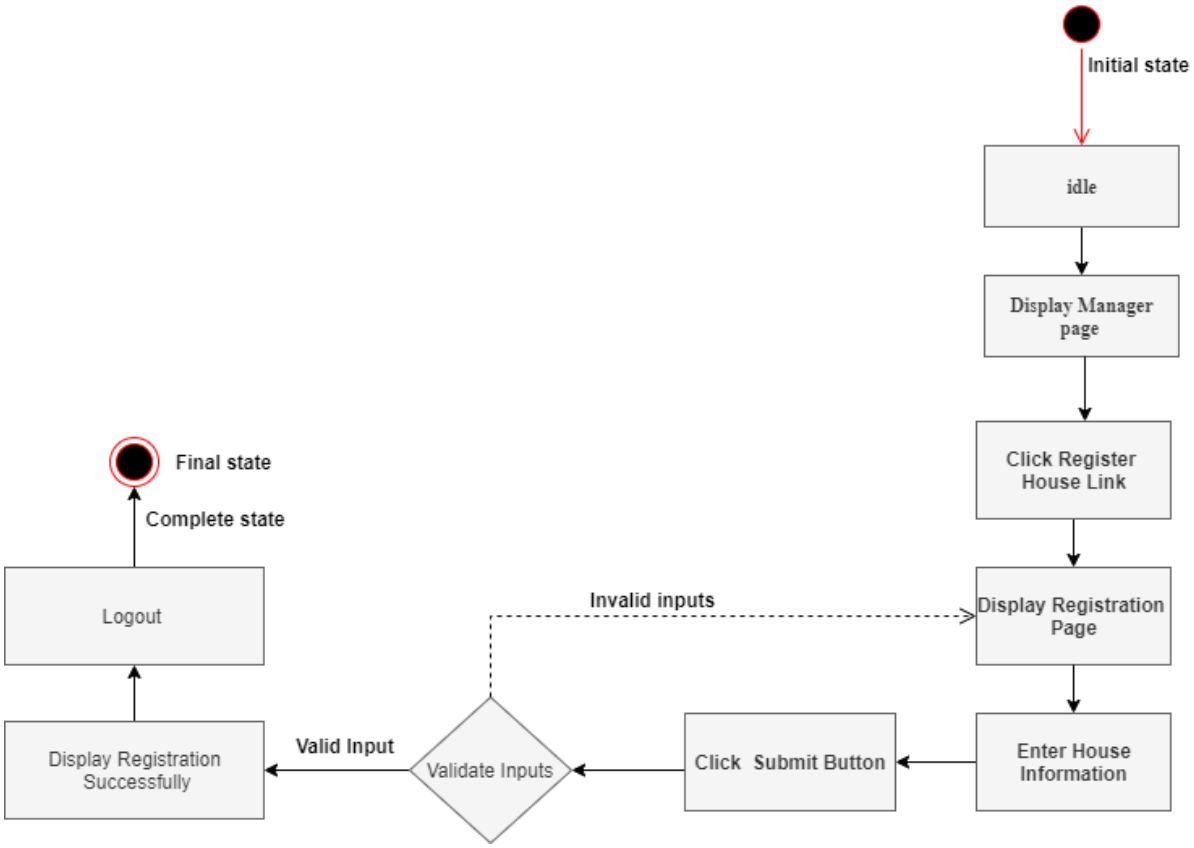


Figure 4. 11 State chart diagram for apply for House

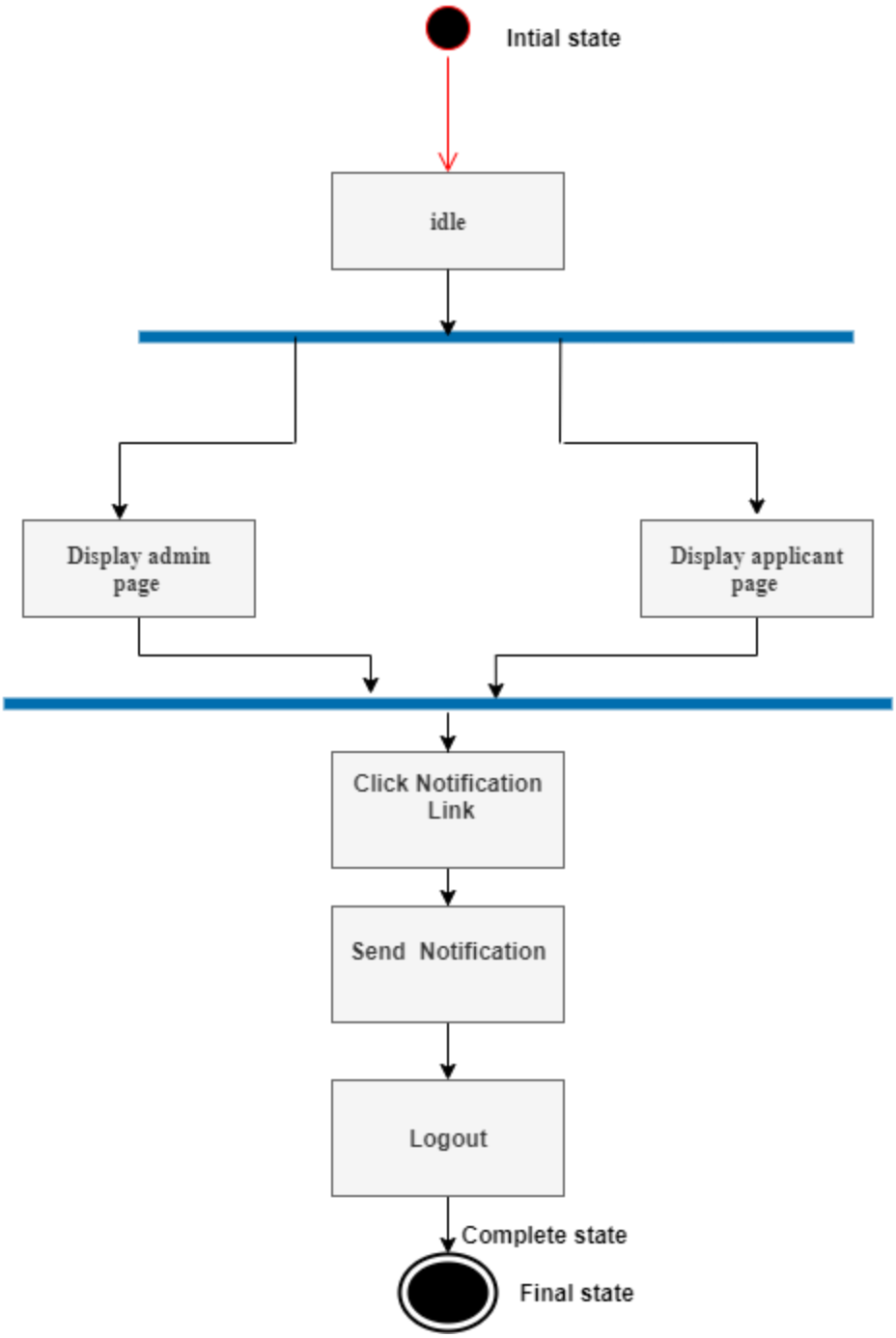


Figure 4. 12 State chart diagrams for sending comments

CHAPTER FIVE

SYSTEM DESIGN

The purpose of designing is to show the direction how the Application is built and to obtain clear and enough information needed to drive the actual implementation of application. It is based on understanding of the model the Application built on system design also focuses on decomposing the system in to manageable parts.

5.1 Design Goals

Design goal primarily emerged from non-functional requirement of the system and the objectives of the design goal are to model a system with high quality that should achieved, and addressed during the design of the system. The designer creates the nature of the design and it is more important for the programmer to implement a high quality and error free system.

5.1.1 Performance

The system should meet the following performance criteria's; Response time: The speed imposed on the system. The system should responsive maximum number of tasks with minimum times; Throughput: number of tasks accomplished in a fixed period of times; Memory: memory space available for speed optimizations should use efficiently. These performance issues should have to be meeting in our system.

5.1.2 Dependability

The condominium management system should achieve the following dependability characteristics in order to resist crash and be available and reliable.

- **Robustness:** Since the system is a web based system that mainly uses a menu driven entry there wouldn't be an input problem by the user side. But for the server side there might be an error during the process of entering a data. In this time the system will provide an error page and the system will continue without failure or crush.
- **Security**: the system should be secured, i.e., not allow unauthorized users to access the database system.

- **Reliability**: the information provided by the system is as reliable as it is presented on the web page interface, and this is maintained by the persistent database.

5.1.3 Maintenance

In time of failure or need modification the system need to be maintained. To be maintainable the system should meet the following maintenance criteria

- **Extensibility**: if it is needed to add new functionality to the system, this must be achieved by only making a separate page and integrate this page with the existing system.
- **Modifiability**: if in the system, some functionality requires to be modified, this modification must be done specifically to that function or page without affecting the overall system organization.

5.1.4 Priorities of design goal

Priorities of design goal are a design goal which should be given priority when we compare one design goal of the system with the other we have putted the priority of the system design goal.

- **Security Vs Performance**

Security is the degree of the system to be hard for accessing of unauthorized user and performance is the quality of the system to do what it should do. System with high performance of processing and performing but with less security control are exposed to vulnerabilities and different attacks.so we select security factor in case of our system should be secured and only be accessible by authorized users.

5.2. Current system architecture

The current system is manual system. Since it is a manual system there is no any architecture for the current system**.**

5.3. Proposed System Architecture

The architecture chosen for this system is three tier. The first layer runs on the client side, the second layer at the middle layer and the third layer will be the database system. The system will run using web technology.

This architecture provides greater application scalability, high flexibility, high efficiency, lower maintenance, and reusability of components. Since each tier runs on a separate machine, it improves systems performance. The system uses dynamic web technology, i.e., adding and retrieving data to and from the data store whenever requested is possible. It requires a client side program which is accessed by the Condomium administrator, by the applicant and also an interface that communicate with the external system. It needs server side functions that implement the functional requirements and the database system that stores data.

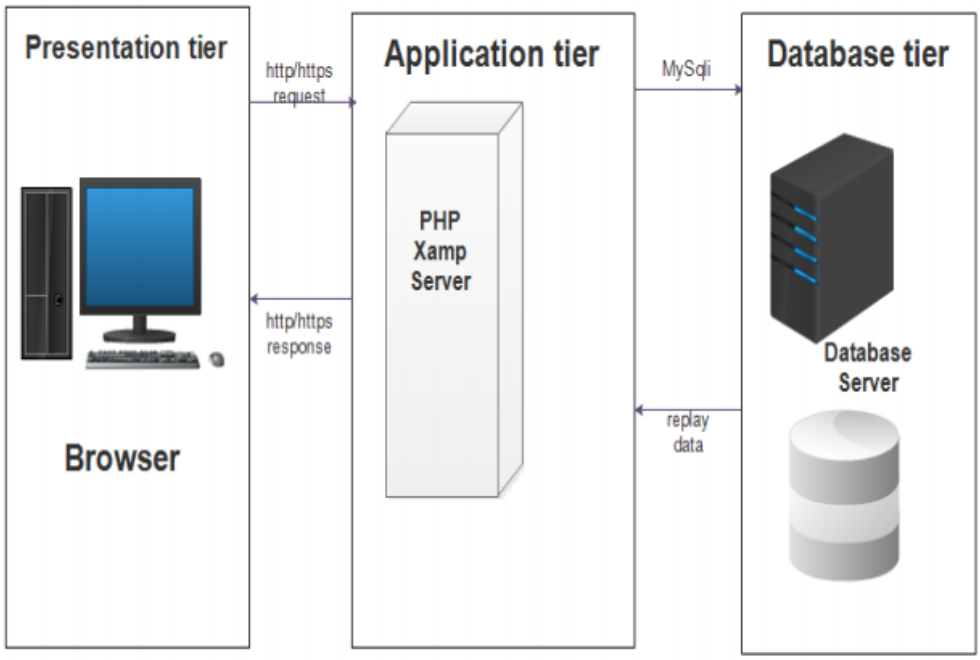


Figure 5. 1 Proposed System Architecture diagram

- **Client Tier/ Presentation tier**

  End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database could provide by the application. As described above at the client side there are three kinds of users such as the system admin, the Kebele office and the applicant.

- **Middle Tier/Application tier**

At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

- **Data Tier/Database tier**

  At this tier, the database resides along with its query processing languages. A database server is a server which uses a database application that provides database services to other computer programs or to computer as defined by the client-server model. User access a database through a front end running on the user's computer which displays requested data. The system uses one database; this database is the repository consisting of the application data. And finally, all the database tables will be stored.

5.3.1. Subsystem Decomposition and Description

The purpose of this activity is to divide the system in self-contained components that can be managed by individuals. Subsystem composition will help reduce the complexity of the system.

The sub systems that we take the classes that our system contain and to distribute the class of the system in to large scale and cohesive components. The following are the major subsystems.

- **Manager subsystem**
  - login
  - Register House
  - Manage applicants
  - Check applicants status
  - Display lists of applicant for lottery
  - Send Notifications
  - Control payment

Admin subsystem

- login
- Manage accounts

❖**Applicant subsystem**

- Login
- Make payment
- Search bank account
- Register
- Update
- Reset
- View notifications

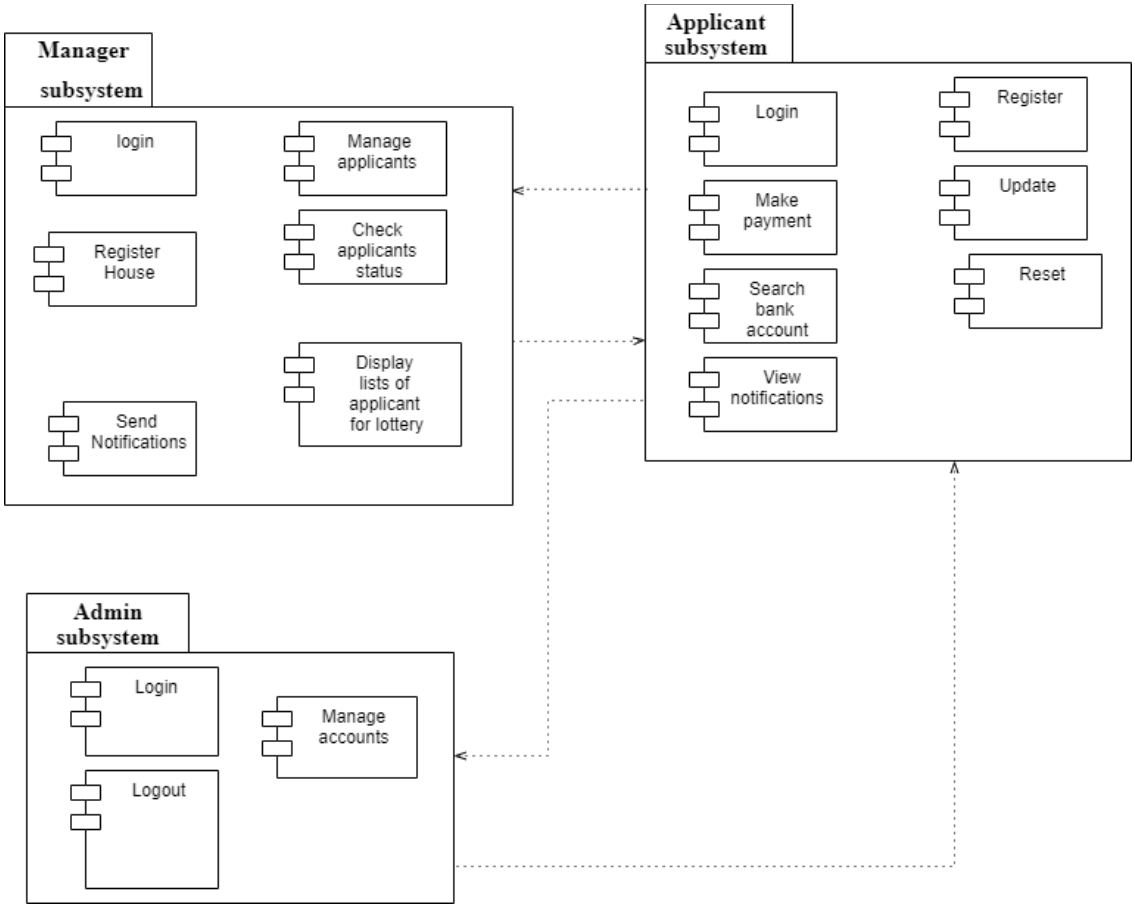


Figure 5. 2 Subsystem Decomposition and Description diagram

5.3.2. Hardware/Software Mapping

The hardware software mapping is described to indicate the various hardware devices and equipment used in the system and its interaction with the software components. It address dependencies and distribution issues of UML components during system design. The deployment diagram shows the hardware for our system, the software that is installed on that hardware, and the middleware administrator. Hardware software mapping shows the software and the hardware components work together. It describes the interconnection of hardware and software how they work and communicate to achieve the desired goals.

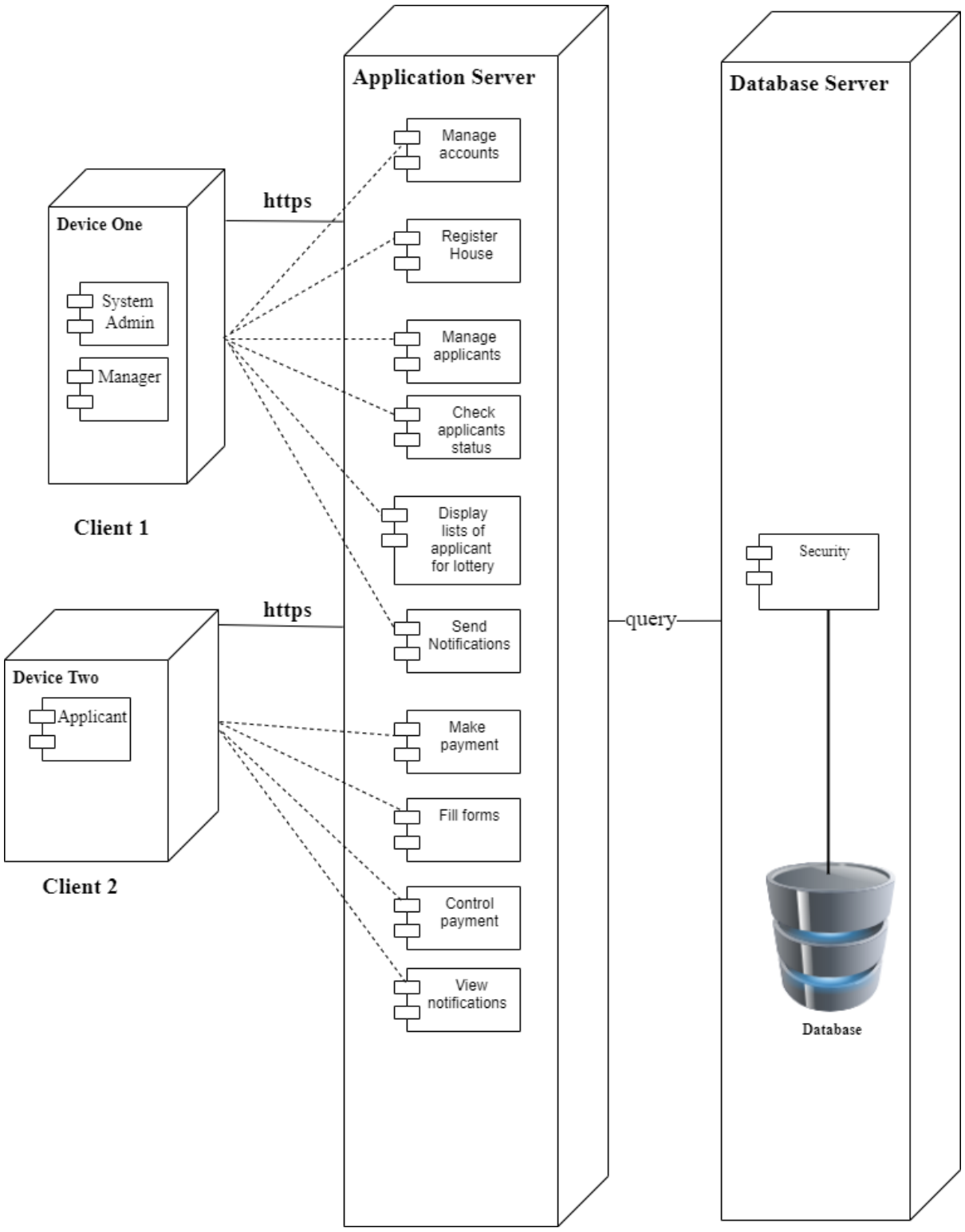


Figure 5. 3 Deployment diagram

5.3.3. Detailed Class Diagram

A class diagram in the Unified Modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (methods), and the relationships among objects. It is made up of a sets of classes and sets of relationships between classes. The purposes of the detailed class diagram are:-

- To shows static structure of classifiers in a system.
- To provide a basic notation for other structure diagrams prescribed by UML.
- It is helpful for developers and team members too.

Business Analysts can use class diagrams to model systems from a business perspectives.

This detailed class diagram shows classes, attributes, methods (operations), data types, visibility ((public (+), private (-), protected (#)) of the attributes and methods), inheritances, associations, aggregation, composition, dependencies and multiplicities. Below we use the detailed class UML diagram to show how the aforementioned concepts are organized and designed for better understandings of the system's detailed class diagram.

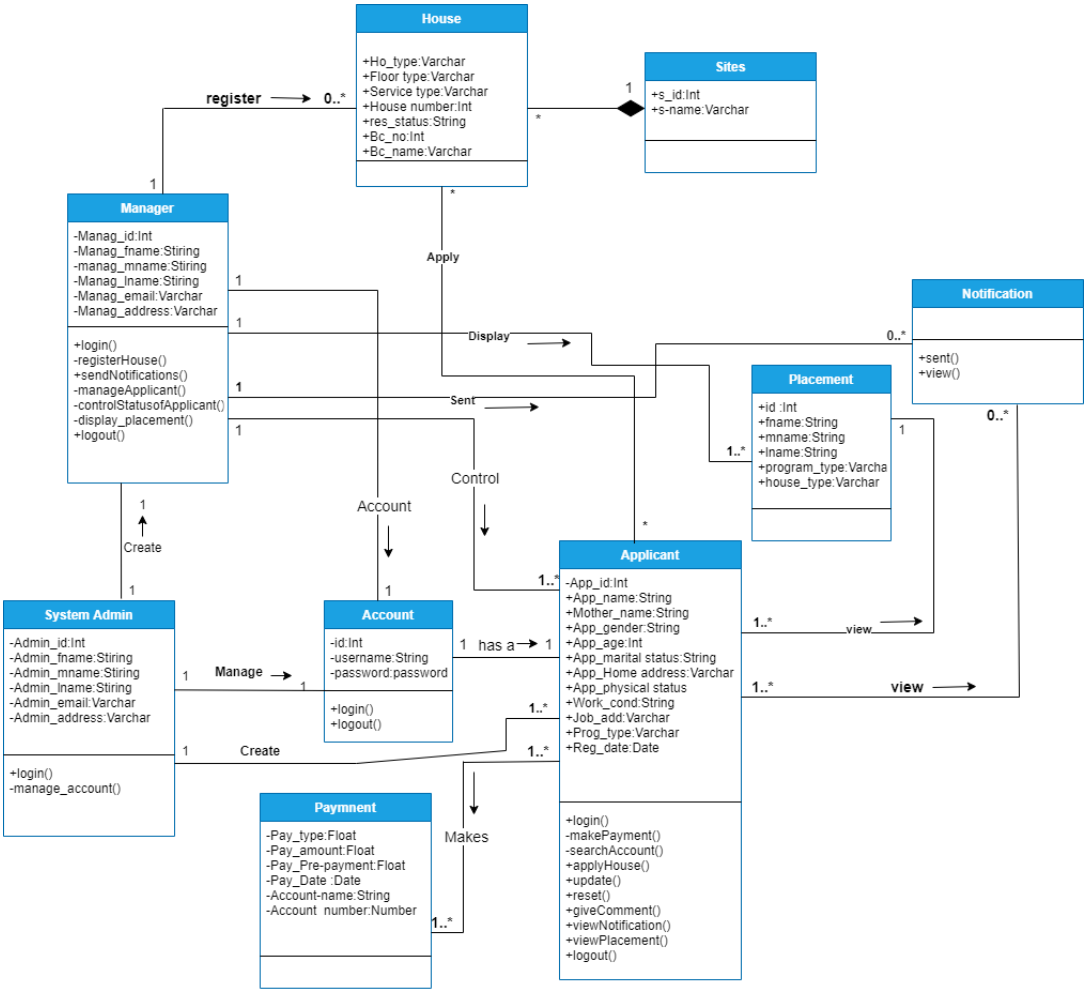
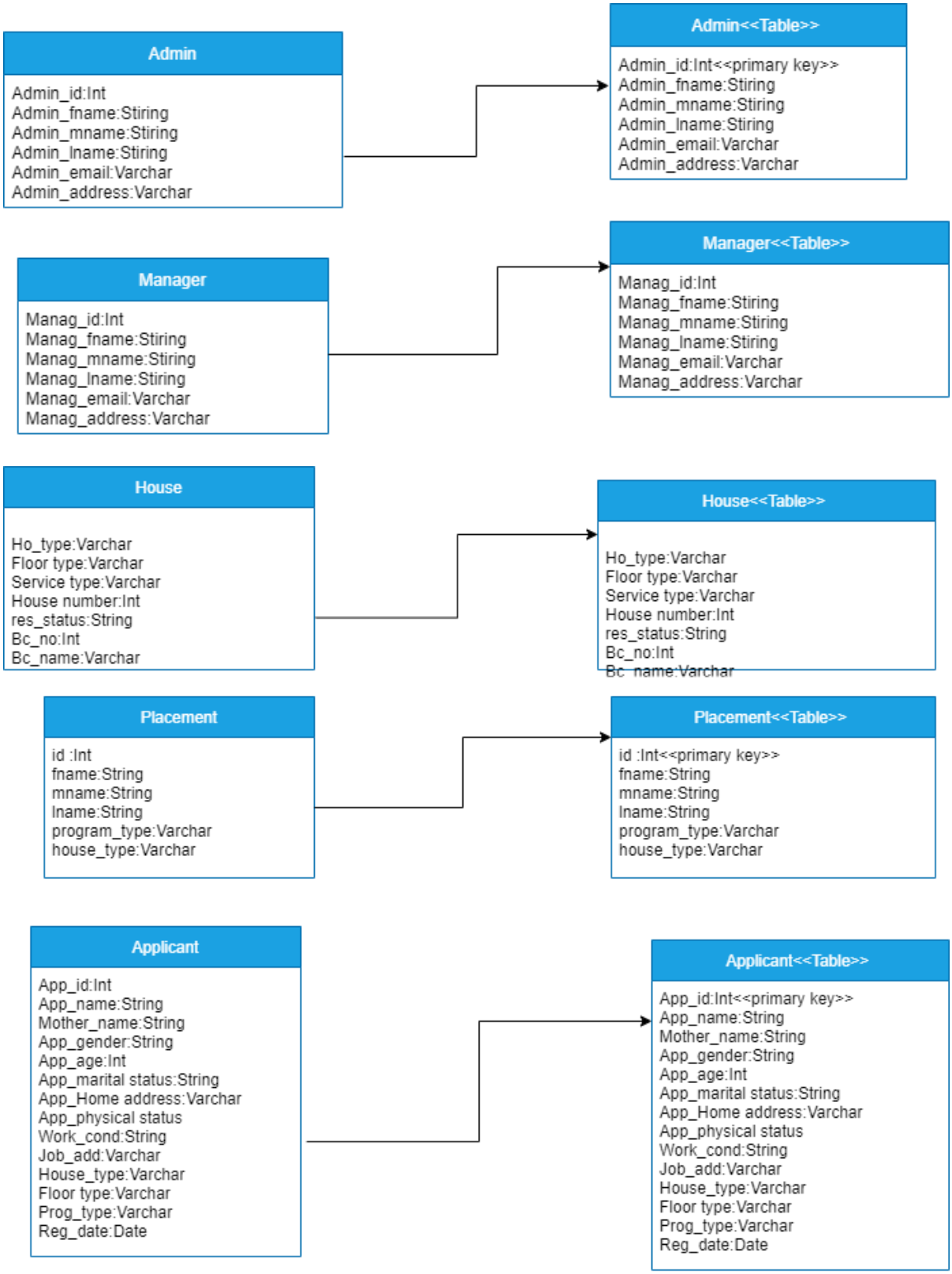


Figure 5. 4 Detail class diagram

5.3.4. Persistent Data Management

Persistence data encapsulate the capability to store, retrieve, and delete objects/data permanently.

In the current database system we have used different tables as object and each object is related to each other. This schema enables as data manipulation activity such as select, search, delete, update on the database. The following tables indicate the persistence data management of the system.


Admin
Admin_id:Int
Admin_fname:Stiring
Admin_mname:Stiring
Admin_lname:Stiring
Admin_email:Varchar
Admin_address:Varchar
Admin<<Table>>
Admin_id:Int<<primary key>>
Admin_fname:Stiring
Admin_mname:Stiring
Admin_lname:Stiring
Admin_email:Varchar
Admin_address:Varchar
Manager
Manag_id:Int
Manag_fname:Stiring
Manag_mname:Stiring
Manag_lname:Stiring
Manag_email:Varchar
Manag_address:Varchar
Manager<<Table>>
Manag_id:Int
Manag_fname:Stiring
Manag_mname:Stiring
Manag_lname:Stiring
Manag_email:Varchar
Manag_address:Varchar
House
Ho_type:Varchar
Floor type:Varchar
Service type:Varchar
House number:Int
res_status:String
Bc_no:Int
Bc_name:Varchar
House<<Table>>
Ho_type:Varchar
Floor type:Varchar
Service type:Varchar
House number:Int
res_status:String
Bc_no:Int
Bc_name:Varchar
Placement
id :Int
fname:String
mname:String
lname:String
program_type:Varchar
house_type:Varchar
Placement<<Table>>
id :Int<<primary key>>
fname:String
mname:String
lname:String
program_type:Varchar
house_type:Varchar
Applicant
App_id:Int
App_name:String
Mother_name:String
App_gender:String
App_age:Int
App_marital status:String
App_Home address:Varchar
App_physical status
Work_cond:String
Job_add:Varchar
House_type:Varchar
Floor type:Varchar
Prog_type:Varchar
Reg_date:Date
Applicant<<Table>>
App_id:Int<<primary key>>
App_name:String
Mother_name:String
App_gender:String
App_age:Int
App_marital status:String
App_Home address:Varchar
App_physical status
Work_cond:String
Job_add:Varchar
House_type:Varchar
Floor type:Varchar
Prog_type:Varchar
Reg_date:Date

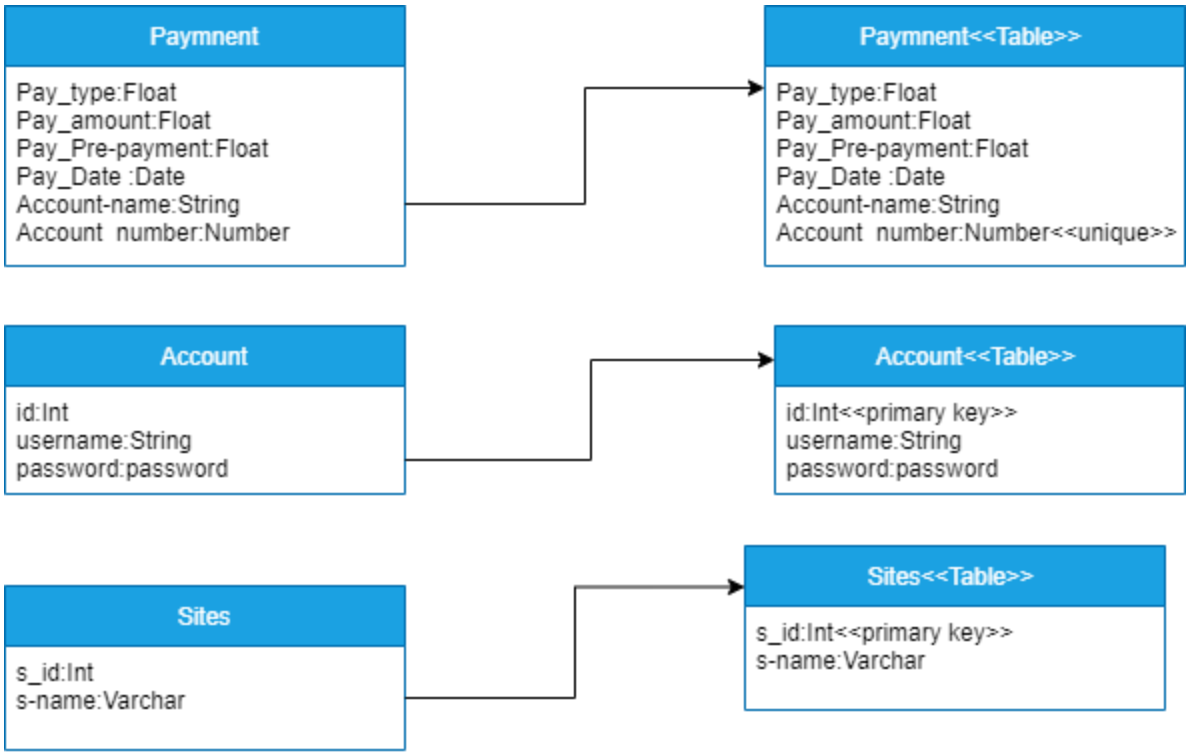


Figure 5. 5 Persistent Data Management

5.3.5. Access Control and Security

In our system, different actors have access to different information and data. Access control and security specify what the user can access or what cannot perform by some users. This access control is verified by a username and password. The system admin represents an authenticated user. The system follows multi-user system. In multi-user system, different actors have access to different functionality and data. Then it must be having: -

Confidentiality: - Only authorized persons can see the information. Private data is kept private, personal privacy is respect

Table 5. 1 Access Control and Security table

Operations	Actors		
	Manager	**Administrator**	**Applicant**
Manage account		✓	
Manage applicant	✓		
Control payment	✓		✓
Registration to apply			✓
Display applicant for lottery	✓		
Evaluate applicant	✓		
Register Finished House	✓		
Notification	✓		✓
Login	✓	✓	✓
logout	✓	✓	✓

5.4. Packages

Package diagram is a kind of structural diagram that shows the arrangement and organization of model elements in the system. It can show both structure and dependencies between sub-systems or modules and shows the different views of a system. We use package diagram to structure high level system elements and organized the large system into sub-modules. We use packages in the system:-

- To simplify complex class diagram into classes package groups.
- To group logically related UML elements.
- To depict file folders and used on any of the UML diagrams.

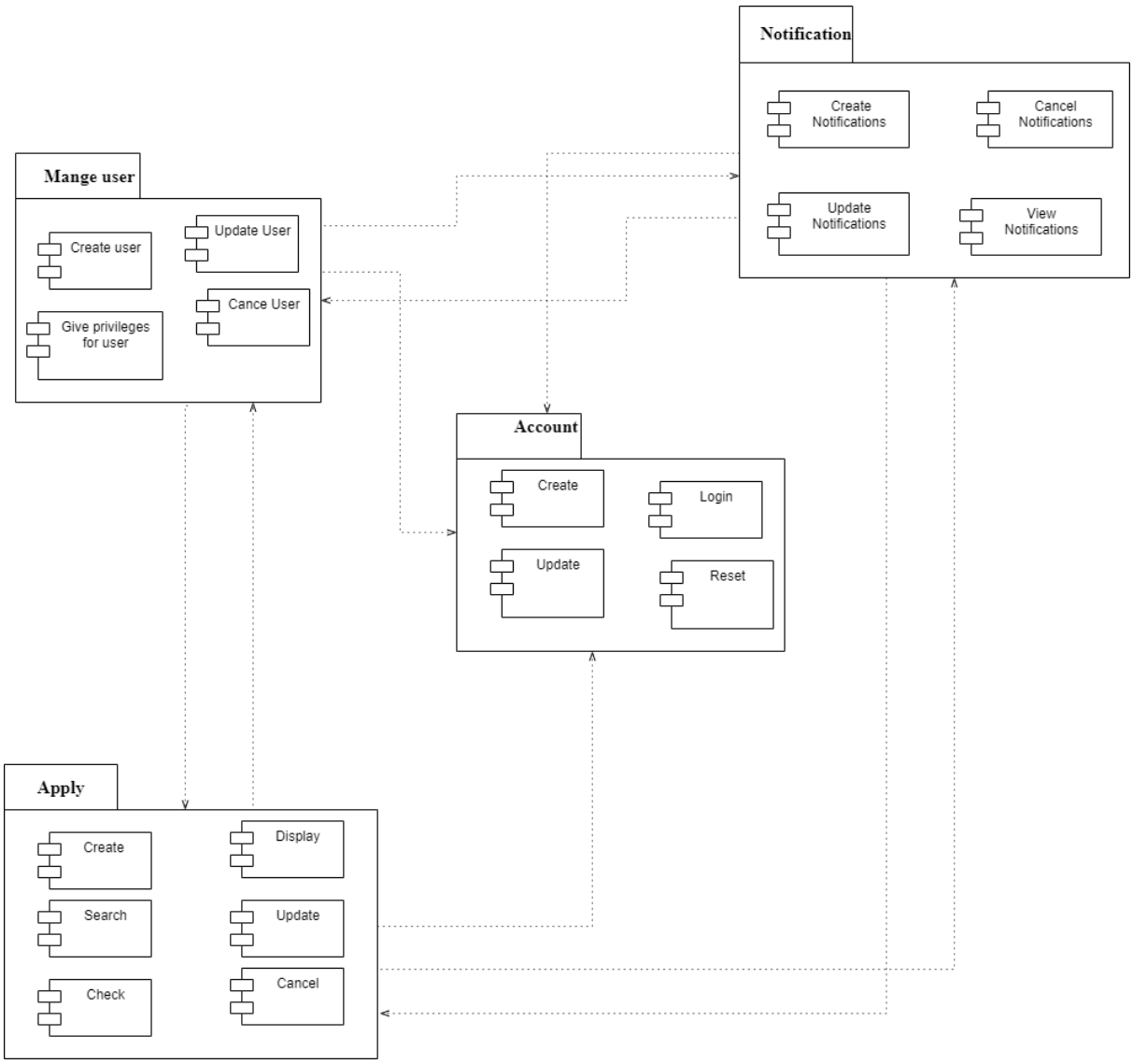


Figure 5. 6 Packages diagram

5.5. Algorithm Design

Pseudo code is a compact and informal high-level description of a computer programming algorithm that uses the structural conventions of a programming language but is intended for human reading rather than machine reading. The purpose of using pseudocode is that it is easier for humans to understand than conventional programming language code and that it is a compact and environment independent description of the key principles of an algorithm.

Pseudo code algorithm for login

```
Begin
    If (user click the Login button)
        Fill Login Form
         If (valid)
             Redirect to authenticated page
             Allow to preform privileged task
          Else
            Display error message.
            Redirect to login page
           Ask the user to refill the form
        End if
    End if
End
```

Pseudo code algorithm for applying for House

```
Begin
       If user click on register link
            If (user must be Kebele resident and not apply before)
            If make payment
            If user click on Search account
                 Fill registration forms
                  If valid
                    Register successfully
                  Else
                    Display error massage
                    Redirect to registration forms
```

End if

End if

End if

End if

End if

End

Pseudo code algorithm for registering finished House

Begin

If user click on register link

Fill registration forms

If valid

Register successfully

Else

Display error massage

Redirect to registration forms

End if

End if

End

Pseudo code algorithm for registering applicant

Begin

If user click on register link

Fill registration forms

If user click on submit button

If valid

Register successfully

Else

Display error massage

Redirect to registration forms

End if

End if

End

5.6. User Interface Design

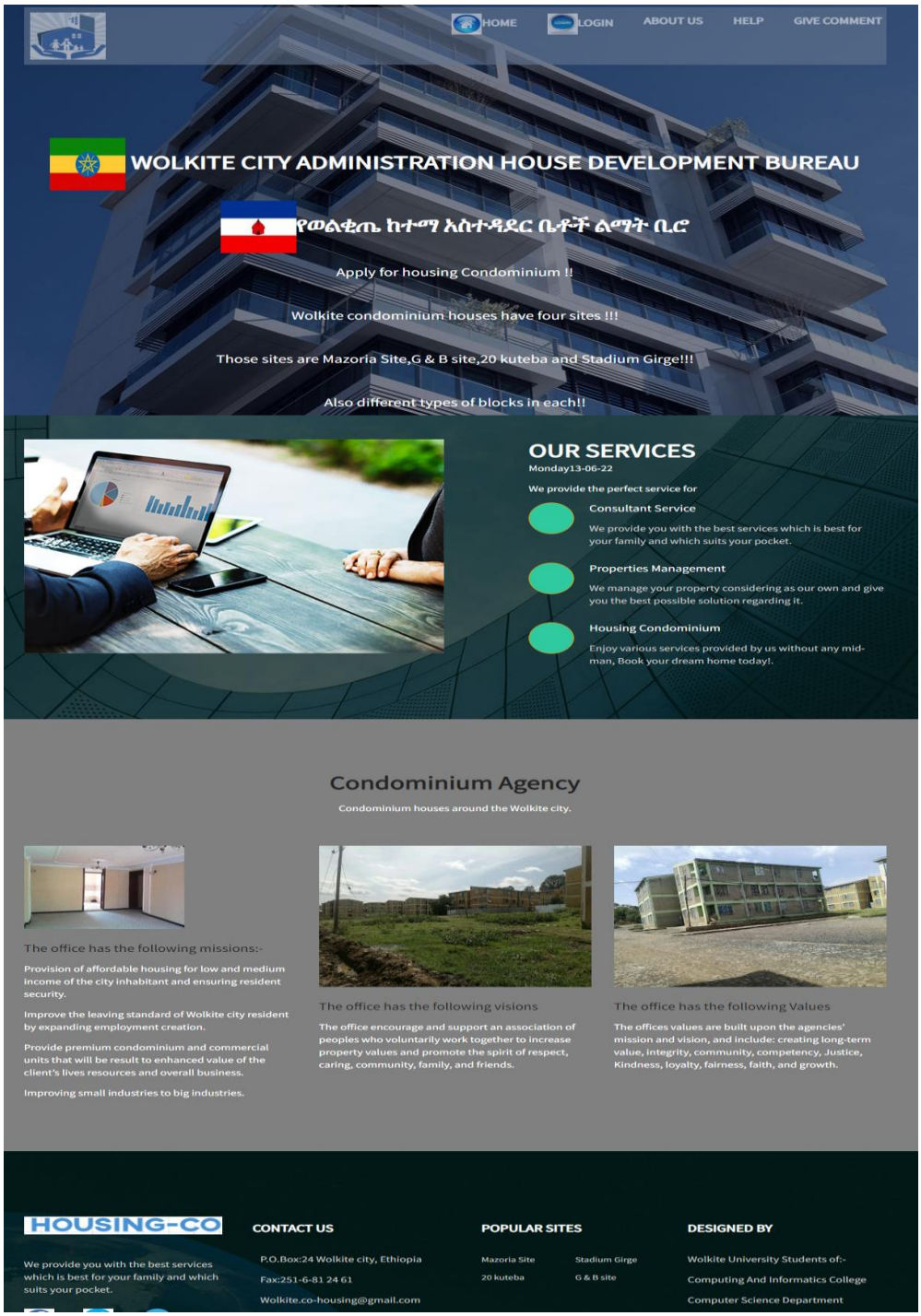


Figure 5. 7 Home Page User Interface Design

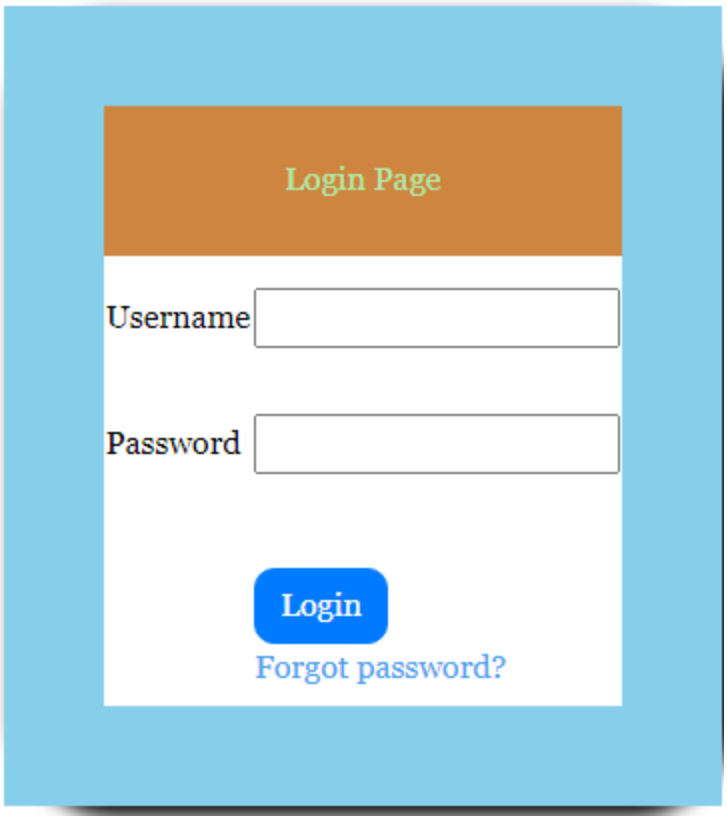


Figure 5. 8 Login Page User Interface Design

CHAPTER SIX

6. IMPLEMENTATİON AND TESTING

This chapter basically highlights the issues deal with the implementation phases. Implementation is the phase where objectives of physical operations of the system turned into reality i.e. real working model. In this phase the coding convention has made it possible as it's the real phase of objectivity to reality. Then the code is tested until most of the errors have been detected and corrected. The goal of implementation is to introduce our system for the users in real sense that how they use this new system which is developed for their intended objectives.

6.1. Implementation of the Database

In order to implement the main application language is MYSQLI that used to store database values and used to store it for long time. We have used MYSQLI database because that can run any operating or browser application, Installation Is Streamlined, Security Features Are Better, Enhanced Performance, It's Important to Maintain an Environment That's Standardized. MYSQLI is easy fast and can used for any type of database weather it is relational or simple database, large or small database.so MYSQL server for implementation our system or used to store our data.

6.2. Implementation of the Class Diagram

In this class diagram implementation we have implement the class diagram structure.
That implements the view of an application, visualizing, describing and documenting different aspect of the system. And also implements the attribute and operation of the class.

6.3. Configuration of the Application Server

We use XAMPP SERVER application server because is simple and Lightweight. Apache distribution it's extremely easy to create a local webserver for testing and deployment purposes. Everything we needed is to setup a web server-server application (Apache), MYSQL and scripting language (PHP). The configuration server follows the version and loaded extension etc.

6.4. Configuration of Application Security

Our system called Wolkite city online condominium management system is a reservation issue validates all the inputs by returning error message and suggesting to try again when invalid input occurred. We implement encryption for user password by MD5 when the system admin creates a user account for manager, receptionist and others create their account or the user changes their password system encryption the password. The system has its security principles that control unauthorized authentication.
The configuration of application security server is well handled for the security purposes. Security issues are handled properly.

- ✓ All input validations are implemented properly.
- ✓ Encryption and decryption algorithms are implemented properly for sensitive data.
- ✓ User's roles are handled properly for accessing the system.
- ✓ User's accounts are assigned with necessary access privileges.
- ✓ Functional requirements are described well and checked.

- ✓ Session and cookies are also implemented in this system properly

6.5. Implementation of User Interface

The user interface of this system is implemented based on the design. It is clear and interactive; as much as possible complexity is handled in this user interface.

- ✓ It reduces memory loads of the users.
- ✓ It is consistent and interactive.
- ✓ It is easy and clear to use

6.6. Testing

Testing evaluates a software product to ensure that it satisfies its planned purpose. A test that is modified to and consistent with development methodologies provides an observable and structured approach to verifying requirements and quantifiable performance.

Criteria are standards by which we evaluate our systems that help us to determine whether a test case passes or fail.

- ➢ Fail Criteria: when the system does not meet the all specific requirements of the system and if the test case is said to fail the expected result is not satisfied by the system that relates with its functionality.
- ➢ Pass Criteria: when the system meets the all specific requirements of the system and if the test case is said to pass the expected result is satisfied by the system that relates with its functionality.

6.6.1. Test Case

Is a set of actions executed to verify a particular feature or functionality of your software applications?

- ➢ Test Case 1: Check result on entering Valid User ID & Password
- ➢ Test Case 2: Check result on entering Invalid User ID& Password
- ➢ Test Case 3: Check response when a User ID is empty & Login Button is pressed, and many more

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

6.6.2. Testing Tools and Environment

In this project different testing tools are used for testing the system development process progress and for debugging purposes.

Software testing tools

- ✓ Window 10 operating system.
- ✓ Xampp Server
- ✓ Web browsers (Chrome, Microsoft edge).
- ✓ Visual studio

Hardware testing tools

- ✓ Computer(Dell desktop, HP laptop)
- ✓ Internet cable

6.6.3. Unit Testing

The main objective of unit testing to ensure that each individual part is working well and as it's supposed to work. The entire system will only be able to work well if the individual part are working well. Unit testing is performed by software developers themselves. In this level of testing process, the OHR system developers test different sub procedures, functions and tested by applying the black and white box testing.

- ✓ Cheek whether the return type of the function is correct.
- ✓ Cheek how the sub procedure or function are call correctly.
- ✓ Cheek if the correct output is produced for different inputs

6.6.4. System Testing

The test performs on the whole programs to ensure the whole functional and non-functional requirement specified on the system and also to decide the completeness of the system as a system that full fill perfectly the whole functional and non-functional requirements of the system

Since we have tested the overall features of our systems and we have conducted testing activities to check if there are errors as well as to correct them as not to occur after or later. The goal of the system testing phase is to evaluate all what we have been implemented in the previous development phase and to find errors in order to make corrections on the errors and these cannot occur again

6.6.5. Integration Testing

As the name of the test indicates that the test performs to ensure the integration of the user interface module that interact the user with the system and the database module that store the transaction of the hotel and then in order to make a system as complete system the two module must be integrate to each other

6.6.6. Acceptance Testing

The test performs to ensure the security of the system as a whole and also the test will be done by using black box testing methods that examines the functionalities of the application by focus only on input and expected output.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. Conclusion

Currently, Wolkite condominium management system is manual based information management system. The system that will provide more efficiency, and accuracy than the manual based system. The new computerized and management system perform many operations performed manually. It prevents the loss of paper documents. This system solves many problems that the organization faces. It brings effectiveness to the applicant and reduce work load of employee. The new system can retrieve data in real time. From this we conclude that the project is very important in work environment of the condominium organization after implementation. We could accomplish some of the objectives. But now there is a clear and base idea how the system can be developed and integrated, this project can be seen as an initiation for students who are doing project on the same field for nationalization/generalization condominium management system.

7.2. Recommendation

While doing this system the team has faced different types of challenges. But by the cooperation of all the group members and the advisor, the team is now able to reach to the final result. We would like to recommend some of the extra features add to this System which we could not do due to time limitation and unavoidable reasons and challenges including the following features. We strongly recommend that one who under goes through this project can succeed, if he or she (they) pay(s) attention for expanding the system national wise by adding email conformation, Real online payment and Verification with SMS gate.

The system that we are designed to implement does not include or perform other functionality, so other project team who are interested can build similar system by including those functionality like managing house contactor information and budget allocation. Full construction management functionality, security mechanism like digital signature

8. REFERENCES

[1]https://www.fig.net/resources/proceedings/fig_proceedings/fig2012/papers/ts02c/TS02C_adamu_5531.pdf

[2]https://chilot.me/https://chilot.me/wp-content/uploads/2021/03/-1-2013.pdf

[3] "Wolkite University Library and information serves" [Online]. Available: http://10.194.40.20/jspui/handle/123456789/9592

[4] "System testing methodology," [Online]. Available: https://www.guru99.com/unit-testing- guide.html, https://www.toptal.com/qa/how-to-write-testable-code-and-why-it-matters.

[5] "Use CaseDiagram,"Online].Available:https://www.uml-diagrams.org/use-case-diagrams.html

[6]"SystemsAnalysis."[Online].Available:https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overi ew.htm.

[7] E. Birhanu, C. Science, W. Muluye, S. Engineering, Z. Mulugeta, and C. Science, "WOLKITE UNIVERSITY COLLEGE OF COMPUTING AND INFORMATICS Industrial Project Guideline
Version 2," 2019.

[8]"SystemModeling,"[Online].Available:https://cs.ccsu.edu/~stan/classes/CS410/Notes16/05-SystemModeling.html.

[9] "Use-Case Model," [Online]. Available: https://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_case_model_D18 AF9.html.

[10]" use case description," [Online].Available: https://thebadoc.com/ba-techniques/f/use-case-description-basics

[11] "Sequence diagram," [Online].Available: https://creately.com/blog/diagrams/sequence-diagram-tutorial/

[12] "State diagram" [Online].Available:https://www.geeksforgeeks.org/unified-modeling-language-uml-statediagrams/?ref=lbp

9. APPENDICES

Appendix I: Existing System Forms and Reports

የወልቂጤ ከተማ አስተዳደር ቤቶች ልማትና አስተዳደር ቢሮ

በመኖሪያ ቤት የህብረት ስራ ማህበራት ለማደራጀት የአባልነት ማመልከቻ ቅጽ

1. የህብረት ስራ ማህበሩ መጠሪያ፡----------

---------- የመኖሪያ ቤት ኃ/የተ/የህ/ስ/ማ (**በቢሮው የሚሞላ)**

2. የአመልካች ሁኔታ:-

2.1. ስም ---------- የአባት ስም ---------- አያት ስም----------

ፆታ --------- እድሜ----------

2.2 የአመልካች እናት ስም ---------- የእናት አባት ስም ----------

የእናት አያት ስም ----------

2.3 የትዳር ሁኔታ:-

☐ ያገባ/ች ☐ ያላገባ/ች ☐ በፍቺ የተለየ/ች ☐ በሞት የተለየ/ች

2.4. ያገባ/ች ከሆነ የትዳር አጋር ስም ከነአያት ----------

2.5. የአመልካች መኖሪያ አድራሻ፡-

ከተማ----------ክ/ከተማ----------ወረዳ ------የቤት ቁጥር----------

የቤት ስልክ ቁጥር ----------የሞባይል ስልክ ቁጥር ----------

ኢ-ሜል አድራሻ----------

መታወቂያቁጥር፡----------

ቀበሌ……የመ/ቤት……(የፓስፖርት)ቁጥር----------

2.6. የአመልካች የእንቅስቃሴ አካል ጉዳት ካለ ሁኔታው ይገለጽ----------

---

3. የስራና የገቢ ሁኔታ

3.1. የስራ ሁኔታ፤-

☐ የመንግሥት ተቀጣሪ ☐ መንግስታዊ ያልሆነ/ች ☐ የግል ተቀጣሪ ☐ በግል ስራ

3.2. የስራ አድራሻ

የመ/ቤቱ ስም----------የሚገኝበት ከተማ----------ክ/ከ**ተማ**-----

ወረዳ----------ስልክ ቁጥር----------ኢ-ሜል አድራሻ----------ፖ.ሣ.ቁ.------

3.3.የአመልካች የወር ገቢ መጠን በብር ----------/----------

3. የሚፈልጉት የከፍል ብዛት

☐ ባለ 1 መኝታ ☐ ባለ 2 መኝታ ☐ ባለ 3 መኝታ

4. የሚመርጡት የህንጻ ከፍታ(ለመጀመሪያ ዙር የቀረበ አማራጭ)

☐ B +G+9 ☐ 2B+G+13

5. የሚኖርበት/ የምትኖርበት ቤት ሁኔታ፡-

☐ የቀበሌ ☐ የኪራይ ቤቶች ☐ ደባል ጥገኛ ☐ ከግለሰብ ኪራይ ☐ በሌላ

6. የምዝገባ ቁጥር የ40/60---------- ወይም የ20/80 ----------

7. እኔ ስሜ ከላይ የተገለጸው አመልካች

7.1. ቀደም ሲል በመንግስት በተዘረጉት በማናቸውም የቤት ልማት ፕሮግራሞች ተጠቃሚ ያልሆንኩኝ፣

3.1. በወልቂጤ ከተማ ውስጥ በራሴ ሆነ በትዳር አጋሬ ስም የተመዘገበ የመኖሪያ ቤት ወይም የቤት መስሪያ ቦታ የሌለኝ ወይም ኖሮኝ ከ1997 ዓ.ም. ወዲህ በሽያጭ ወይም በስጦታ ለሶስተኛ ወገን ያላስተላለፍኩ መሆኑን፤

3.2. ለምዝገባ ብቁ የሚያደርገኝ የሚጠበቅብኝ የቤቱን ግንባታ ዋጋ 10/30/40 በመቶ በተፈቀደው ባንክ በዝግ ሂሳብ የማስቀምጥ መሆኑን፤

3.3. በምረከበው ቤት በጋራ ህንፃ ህግ መሠረት ለመተዳደር ፈቃደኛ መሆኔን፤

3.4. እኔም ሆነ የትዳር አጋሬ የምንኖረው በመንግስት ቤት ከሆነ ቤቱን ለሚያስተዳድረው

3.5. አካል ቤቱን በተረከብኩ በ30 ቀን ውስጥ ለማስረከብ ፈቃደኛ መሆኔን፤

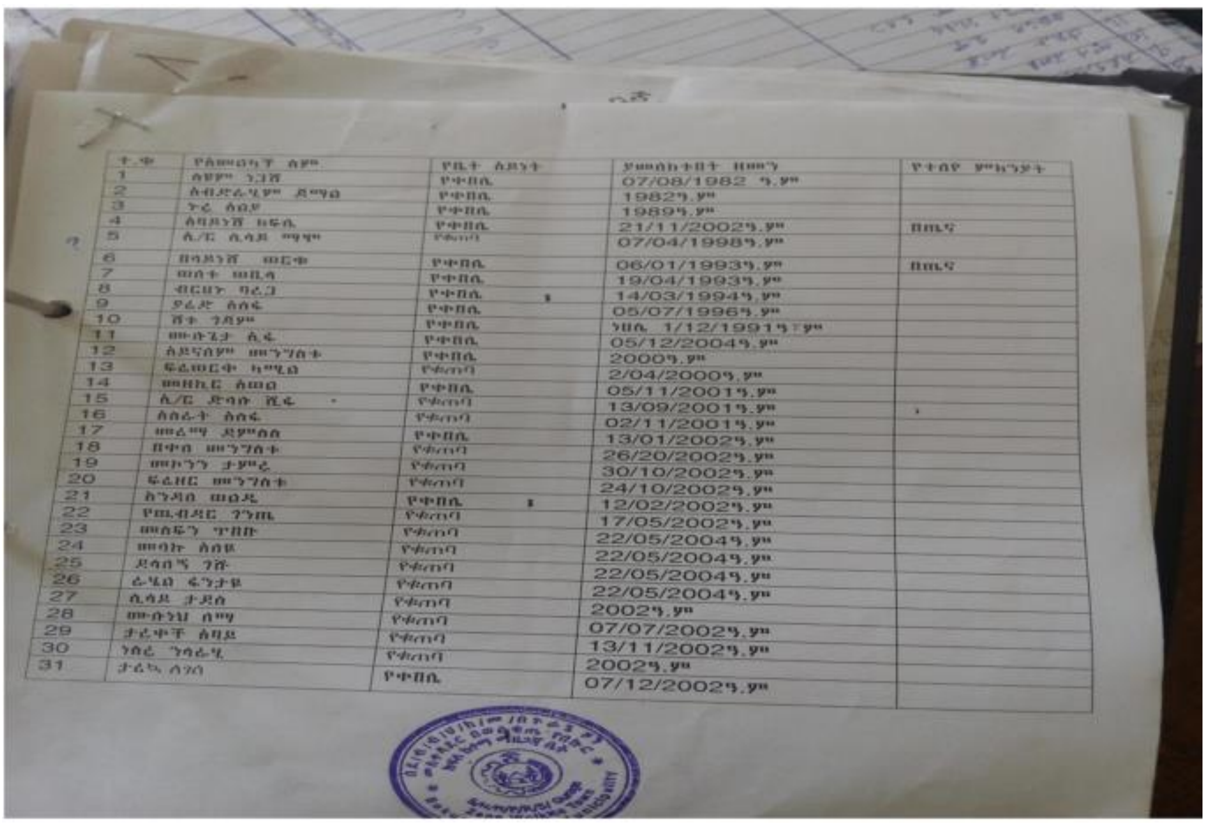
3.6. በዚህ ማመልከቻ ቅጽ የሞላሁትና የሰጠሁት ማረጋገጫ ሀሰተኛ ሆኖ ቢገኝ ቤቱ ከመገንባቱ በፊት ከሆነ የምዝገባው ውል የሚፈርስ መሆኑን፣ ቤቱን ከተረከብኩ በኋላ ከሆነ አግባብ ባለው ህግ ተጠያቂ ለመሆን የምስማማ መሆኑን፤

3.7. ወደፊት የሚጠየቁ መረጃዎችን ለምሳሌ የጣት አሻራና ሌሎች መረጃዎችን ለመስጠት ፈቃደኛ መሆኔን፤

3.8. የመኖሪያ ቤት ህብረት ስራ ማህበር ስም ከሌሎች የማህበሩ አባላት ጋር በፈቃደኝነት ለመደራጀትና የመኖሪያ ቤት ለመስራት የሚያስፈልገውን ቅድመ ሁኔታና መስፈርቶች በአዋጁና በመመሪያው መሰረት ያሟላሁ በመሆኑ ህብረት ስራ ማህበሩ በአባልነት እንዲቀበለኝና ከላይ የሞላሁት መረጃ መሰረት ሆኖ ቢገኝ በህጉ መሠረት ተጠያቂ እንደምሆን በመስማማት ይህን ቅጽ የሞላሁ መሆኑን በፊርማዬ አረጋግጣለሁ፤

የህ/ሥ/ማ/አባሉ ሙሉ ስም---------------------------------ፊርማ -----------------ቀን --------------

Figure 9. 1 Applying for house form



ተ.ቁ	የአመልካች ስም	የቤት አይነት	ያመለከተበት ዘመን	የተሰየ ምክንያት
1	[illegible]	የቀበሌ	07/08/1982 ዓ.ም	
2	[illegible]	የቀበሌ	1982ዓ.ም	
3	[illegible]	የቀበሌ	1989ዓ.ም	
4	[illegible]	የቀበሌ	21/11/2002ዓ.ም	[illegible]
5	[illegible]	የቁጠባ	07/04/1998ዓ.ም	
6	[illegible]	የቀበሌ	06/01/1993ዓ.ም	[illegible]
7	[illegible]	የቀበሌ	19/04/1993ዓ.ም	
8	[illegible]	የቀበሌ	14/03/1994ዓ.ም	
9	[illegible]	የቀበሌ	05/07/1996ዓ.ም	
10	[illegible]	የቀበሌ	ነሀሴ 1/12/1991ዓ.ም	
11	[illegible]	የቀበሌ	05/12/2004ዓ.ም	
12	[illegible]	የቀበሌ	2000ዓ.ም	
13	[illegible]	የቁጠባ	2/04/2000ዓ.ም	
14	[illegible]	የቀበሌ	05/11/2001ዓ.ም	
15	[illegible]	የቁጠባ	13/09/2001ዓ.ም	
16	[illegible]	የቁጠባ	02/11/2001ዓ.ም	
17	[illegible]	የቀበሌ	13/01/2002ዓ.ም	
18	[illegible]	የቁጠባ	26/20/2002ዓ.ም	
19	[illegible]	የቁጠባ	30/10/2002ዓ.ም	
20	[illegible]	የቁጠባ	24/10/2002ዓ.ም	
21	[illegible]	የቀበሌ	12/02/2002ዓ.ም	
22	[illegible]	የቁጠባ	17/05/2002ዓ.ም	
23	[illegible]	የቁጠባ	22/05/2004ዓ.ም	
24	[illegible]	የቁጠባ	22/05/2004ዓ.ም	
25	[illegible]	የቁጠባ	22/05/2004ዓ.ም	
26	[illegible]	የቁጠባ	22/05/2004ዓ.ም	
27	[illegible]	የቁጠባ	2002ዓ.ም	
28	[illegible]	የቁጠባ	07/07/2002ዓ.ም	
29	[illegible]	የቁጠባ	13/11/2002ዓ.ም	
30	[illegible]	የቁጠባ	2002ዓ.ም	
31	[illegible]	የቀበሌ	07/12/2002ዓ.ም	

Figure 9. 2 Placement information

Appendix II: Sample Source Code

Login code

```
<?php
session_start();
include('connect.php');
?>
<? php
        include ("header. php");
        ?>
        </div>

<? php
     if (isset($_POST['submit']))
     {
      $username=$_POST['user'];
      $password=$_POST['pass'];
              $sql ="SELECT * FROM user WHERE username='$username' AND
password='$password'";
      $result = mysqli_query ($conn, $sql);
      // TO check that at least one row was returned
      $rowCheck = mysqli_num_rows ($result);
      if($rowCheck>0){
      $row=mysqli_fetch_array ($result);
       if($row['user_type']=='hm'){
      $_SESSION['user_id']=$row['user_id'];
       echo "<script>window.location='hm.php';</script>";
      }
         else if($row['user_type']=='client'){
      $_SESSION['user_id']=$row['user_id'];
      echo "<script>window.location='client.php';</script>";
         }
         else if($row['user_type']=='admin') {
      $_SESSION['user_id']=$row['user_id'];

      echo "<script>window.location='Sadmin.php';</script>";
      }
      }else {
      ?>
         <script>
```

```
          alert('You enter wrong username or password ,please try again to login by the correct account');
          window.location='login.php';
          </script>
       <?php
       }
       }
   mysqli_close ($conn);
   ?>

   <section class="blog-section spad">
      <div class="container">

   <!--End of Php Code-->
<form action="login.php" method="POST" name="form1">
      <center><table  style='border:50px  solid  skyblue;border-radius:50px;box-shadow:5px 5px  10px #000000;width:100px; height:400px'>
               <tr><td  colspan='2'  bgcolor="blue"  style='text-align:center;'><font color='blueblack'>Login Page</font></td></td>
   </tr>
                <tr><td>Username</td><td><input  type='text'id="user"  name='user' required="required"></td>
   </tr>
             <tr><td>Password</td><td><input  type='password'  id="pass"name='pass' required="required"></td>
   </tr>
          <tr><td></td><td><input  type='submit'  name='submit'  class="btn  btn-primary"value='Login'style="border-radius: 10px;"/>  
   <input type="reset" value="Reset" class="btn btn-danger"style="border-radius: 10px;"/><h6>
                <a  href="search.php"><span style="color: #44a2ee;">If you Forgot password please!!!click Here... </a></h6></span></td>
   </tr>
   <tr></tr>
   </table></center>
   </form>
   </tr>
   </table>
   <br>
</center>
 <div id="footer">
        <?php
        include("footer.php");
        ?>
        </div>
```

```
  </body>
</div>
</html>
```