

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTEMENT OF INFORMATION SYSTEMS

PROJECT TITLE: ONLINE HOUSE RENTAL MANAGEMENT SYSTEM FOR WOLKITE CITY

NAME	ID NO
1. NEGUSSE DUBA	CIR/173/11
2. RUFAEL MULETA	CIR/178/11
3. ASTER GEMEDA	CIR/159/11

ADVISED BY: MRS. HANA HAYESH

WOLKITE UNIVERSITY

WOLKITE, ETHIOPIA

Submission Date: 22/06/2014

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTEMENT OF INFORMATION SYSTEMS

PROJECT TITLE: ONLINE HOUSE RENTAL MANAGEMENT SYSTEM FOR WOLKITE CITY

NAME	ID NO
1. NEGUSSE DUBA	CIR/173/11
2. RUFAEL MULETA	CIR/178/11
3. ASTER GEMEDA	CIR/159/11

ADVISED BY: MRS. HANA HAYRESH

WOLKITE UNIVERSITY

WOLKITE, ETHIOPIA

Submission Date: 08/10/2014

DECLARATION

This is to declare that this project work which is done under the advisor of Mrs. Hana Hayesh and having the online house rental management system is the sole contribution of: Negusse Duba, Rufael Muleta, and Aster Gemeda. 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._____	_____
2._____	_____
3._____	_____

APPROVAL FORM

This is to confirm that the project report entitled online house rental management system submitted to Wolkite University, College of Computing and Informatics, Department of Information systems by Negusse Duba, Rufael Muleta, and Aster Gemeda 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

we would like to express our sincere gratitude to several individuals for supporting us throughout our Industrial Project. First of all, we would like to thank God for helping us to reach this time. Then we would like to thank Wolkite University Specially Department of Information System for providing us the opportunity to do this project & to fix several problems of the organization. Then we would like to express our Special thanks for our advisor Mrs. Hana Hayesh for his enthusiasm, patience, insightful comments, helpful information and practical face to face advice that helped us tremendously at all times throughout our project, and for continuous support in every phase of the project and who continuously provides us his valuable advice to prepare this project document. We also express our gratitude to our instructors who support us in different way by sharing their precious idea and for our university for accepting us into graduate program. Lastly, we would like to acknowledge all our friends for their morals, idea, materials and financial support during the courses of the years.

EXECUTIVE SUMMARY

The project that we are going to develop is web based house rental system for Wolkite city. We inspired to develop this system because renting a house is not an easy activity. This is due to some problems facing the tenants. Such problems are do not knowing the exact location of the house to be rented; this leads to fatigue of finding the house, wastage of renter's time and money. To avoid these and some related problems we designed this web based house rental system. So we hope that the newly automated system provides fast and reliable service for tenants, house owners by minimizing time and resource wastage. The document has clear and concise significance for tenants, house owners. Under this document we specify the functional and nonfunctional requirements of the system and by carefully applying the functional requirements tenants can get quality service from the newly automated system. To achieve this we are using object oriented system develop methodology and using HTML, JavaScript with CSS implementation for the interface and PHP languages to design and for data manipulation and MYSQL as back end database. In General Web based House Rent System for wolkite city is very important task. Its primary goal is to enable tenants to rent houses online. And the second task is for the house owners to advertise their houses and its services to tenants online. To achieve these goals we proposed the new web based system of house rental system. Hence tenants can register online so that no needs waste the time to go for searching the house to rent on foot. The system supports house registration as well as viewing available properties for purposes of rent.

TABLE OF CONTENTS

DECLARATION i
APPROVAL FORM ii
ACKNOWLEDGEMENT iii
EXECUTIVE SUMMARY iv
LIST OF TABLE viii
LIST OF FIGURES ix
CHAPTER ONE 1
1. INTRODUCTION 1
1.1 Background of the project 1
1.2 Statement of the problem 2
1.3 Objective of the project 2
1.3.1 General objective of the system 2
1.3.2 Specific objective of the system 3
1.4 System development approach 3
1.5 Feasibility Study 4
1.5.1 Operational Feasibility 4
1.5.2 Economic feasibility 5
1.5.3 Technical Feasibility 5
1.5.4 Legal feasibility 6
1.6 Project Scope and Limitation 6
1.6.1 Scope of the project 6
1.6.2 Limitation of the project 6
1.7 Significance of the project 7
1.7.1 Tangible benefits 7
1.7.2 Intangible benefits 8
1.8 Beneficiary of the project 8
1.9 System development Methodology 8
1.9.1 Data collection methodology 8
1.9.2 System analysis and design approach 9
1.9.3 System development model 9
1.9.4 System testing methodology 10

1.9.5. Development tools and technology 11
1.10 Document Organization 13
CHAPTER TWO 14
2. DESCRIPTION OF THE EXISTING SYSTEM 14
2.1. Introduction of Existing System 14
2.2. Users of Existing System 14
2.3 Major Functions of the Existing System 14
2.4. Drawbacks of the Existing System 15
2.5. Business Rules of the Existing System 15
CHAPTER THREE 16
3. PROPOSED SYSTEM 16
3.1 Requirement Analysis of the New System 16
3.1.1 Functional Requirements 16
3.1.2 Non-functional Requirements 18
CHAPTER 4 20
4. SYSTEM ANALYSIS 20
4.1. System Model 21
4.1.1 Use Case Model 21
4.2. Object Model 40
4.2.1. Class Diagram 40
4.2.2. Data Dictionary 41
4.3. Dynamic model 42
4.3.1 Sequence diagram 43
4.3.2Activity Diagram 47
4.3.3 State chart 51
5. SYSTEM DESIGN 54
5.1 Design goal 54
5.2 Current System Architecture 57
5.3. Proposed System Architecture 57
5.3.1 Subsystem Decomposition and Description 59
5.3.2. Hardware/Software Mapping 59
5.3.3. Detailed Class Diagram 61
5.3.4. Persistent Data Management 61

5.3.5. Access Control and Security 62
5.4 Packages 63
5.5 Algorithm Design 64
5.6. User Interface Design 65
5.6.1 Home Page 65
5.6.2 Create Account owner Page 66
CHAPTER SIX 67
6. IMPLEMENTATION AND TESTING 67
6.1. Implementation of the Database 67
6.2. Implementation of the Class Diagram 67
6.3. Configuration of the Application Server 67
6.4. Configuration of Application Security 67
6.5. Implementation of User Interface 68
6.6. Testing 68
6.6.1. Test Case 69
6.6.2. Testing Tools and Environment 69
6.6.3. Unit Testing 69
6.6.4. System Testing 70
6.6.5. Integration Testing 70
6.6.6. Acceptance Testing 70
CHAPTER SEVEN 71
CONCLUSION AND RECOMMENDATION 71
7.1 Conclusion 71
7.2 Recommendation 71
REFERANCE 73
APPENDICES 74

LIST OF TABLE

Table 4. 1 Use case model 21
Table4. 2 Create user account use case 24
Table4. 3 Ogin use case description 25
Table4. 4 Send request use case description 26
Table4.5 Change password use case description 27
Table4.6 Use case description of manage user account 28
Table4.7 Register house use case description 29
Table4.8 View house information use case description 30
Table4.9 View response use case description 30
Table4.10 Upload house information use case description 31
Table4.11 View customer request use case description 32
Table4.12 Reserve house for customer request use case description 32
Table4.13 View customer request use case description 33
Table4.14 Update house use case description 34
Table4.15 Generate news use case description 35
Table4.16 Send feedback use case description 36
Table4.17 View feedback use case description 36
Table4.18 Pay payment use case description 37
Table4.19 Logout use case description 38
Table4.20 Use case scenario 39
Table 5. 1 Access Control And Security 63

LIST OF FIGURES

Figure 4. 1 Essential use case diagram 23
Figure 4. 2 System use case diagram 24
Figure 4. 3 Uml class diagram 41
Figure 4. 4 Sequence diagram for login 43
Figure 4. 5 Sequence diagram for change password 44
Figure 4. 6 Sequence diagram for send request 44
Figure 4. 7 Sequence diagram for view request 45
Figure 4. 8 Sequence diagram for actors send response 45
Figure 4. 9 Sequence diagram for send feedback 46
Figure 4. 10 Sequence diagram for logout 47
Figure 4. 11 Activity diagram for log in to the system 48
Figure 4. 12 Activity diagram for customer login to the system 48
Figure 4. 13 Activity diagram for owner login to the system 48
Figure 4. 14 Activity diagram for employee login to the system 49
Figure 4. 15 Activity diagram for create account 49
Figure 4. 16 Activity diagram for change password 50
Figure 4. 17 Activity diagram for view response 51
Figure 4. 18 Activity diagram for upload house 51
Figure 4. 19 State chart diagram for login 52
Figure 4. 20 State chart diagrams for reserve the house 52
Figure 4. 21 State chart diagrams for upload house 53
Figure 5. 2 Web server/application servers 58
Figure 5. 3 Subsystem decomposition 59
Figure 5. 4 Uml deployment diagram 60
Figure 5. 5 Detailed class diagram 61
Figure 5. 6 Persistence diagram 62
Figure 5. 7 Packages 64
Figure 5. 8 Home page interface 65
Figure 5. 9 Create account owner page 66

LIST OF ACRONYM'S

BR……..Business rule

CPU…….Central processing unit

CSS……Cascading Style Sheet

DB…….Database

GPS……..Global positioning system

HTML……....Hyper Text Markup Language

HTTP……Hyper Text Transfer Protocol

ID…….Identification

Mr……Mister

Mrs……..Misters

MYSQL……My Structured Query Language.

OOA…….Object oriented analysis

OOAD……..Object oriented analysis and design

OOD…….Object oriented design

OOP…….Object oriented programming

OOSAD……..…Object oriented system analysis and design

OOSD……...Object oriented system development

PC ……Personal Computer

PHP……...Hyper Text Pre Processer

REQ……...Requirement

SDLC……...Software Development Life Cycle

SQL……..Structured query language

SRS……..Software Requirement Specification

UAT ……....User acceptance testing

UC……Use Case

UI…….User interface

UML……Unified Modeling Language

WAMP……Windows, Apache, MySQL, PHP

CHAPTER ONE

1. INTRODUCTION

Renting a home is popular with peoples from all walks of life. Typically students and employees that are moving temporarily will share house renting. Basically we are initiated to develop this system because all most all of Wolkite city employees get suffer about house renting even they leave with their relatives until they find house that was really difficult to do practically therefore, we are developing this system today.

Due to its numerous benefits nowadays computers are involved in every sectors of life, in wide range of applications. And even in some areas mankind has been totally dependent on computers.

One of the remarkable and much known products of technology advancement is the conversion of manually-operated system into automated system. Automation produces a great impact in the lives of mankind, particularly in the field of industry, business, medicine, technology and education.

The main purpose of this project is to develop and implement a web based application "online house renting system in order to implement web based house property system to help renters to provide reliable services. These systems tend to us to find possible solution about the problem it helps user to access the system easily.

1.1 Background of the project

Wolkite is one of the beautiful cities found in Ethiopia. The city is found in the South-Western parts of Ethiopia. Located in the Gurage Zone of the South Nation Nationalities and Peoples Region and it is 149.6km far from Addis Ababa Ethiopia. Currently the city has no automated house renting system for customers. Since the city is wide it is harder to find houses for renting in such wide city. In wolkite city the population growth is increasing day to day that means different branch offices are opened like Wolkite University, this university and other company's needs employees and teachers to work in their company and these employees need house to serve their life. So we have started to develop new web based application of house

renting relative with current problem of the city we have started by studying the detail of the system.

1.2 Statement of the problem

Wolkite city is a wide city, therefore to rent or finding a house in this city is taken as a problem. The other problem is increasing the number of people come to wolkite from day to day for different reasons and they want to rent a house.

Generally the main and observed problems are:-

- Man power wastage:-renters lose their energy by finding house on foot from place to place even they cannot get house because most of houses are reserved and they cannot know where they can get free house exactly.
- Wastage of money:-renters waste their money for coordinators payment and for taxi in order to find house.
- Difficult for customers to withdraw from house: - Since the system is manual it is difficult to drop the files of customer's on the paper.
- Coordinators or brokers false information:-coordinators give uncertain information for renter when the renters will find the house with coordinators since the renter did not get the house as he/she expect.
- Data Redundancy: - There is redundancy in renting new houses. Coordinators or brokers may take payment from different peoples and they will give the house for the one who gives more money for them this leads to conflict among them.
- Prone to error: - Since manual systems are done by human being's it is prone to error.
- Files are not secure:-There is no authentication mechanism to secure the files. Therefore files are easily accessible by any one.

1.3 Objective of the project

1.3.1 General objective of the system

The general objective of the system is to develop and design online house rental management system for Wolkite city.

1.3.2 Specific objective of the system

To achieve general objective of the system we used the following specific objectives.

- To gather required information for proposed system by using interview, observation, and document analysis.
- To Analyze the gathered information using SRS
- To Design the proposed system using UML diagram.
- To Design a user interface for the proposed system
- To Design a database for the proposed system
- To Design implementation for the proposed system
- To Test the proposed system
- To form an easy platform for owner to post houses.

1.4 System development approach

In this project, we would use object oriented system development methodology (OOSD) for designing. OOSD in modern software engineering is best conducted in an iterative and incremental way. Iteration by iteration, the outputs of OOAD activities, analysis models for OOA and design models for OOD respectively, will be refined and evolve continuously driven by key factors like risks and business value so in this project we use OOSD due to the above task as well as easier to use in our skill and not as much cost.

We select this because of the following advantages:-

- To simplify the design and implementation of complex program.
- To make it easier for teams of designers and programmers to work in a single software project.

- To enable a high degree of reusability of designs and software codes. Increased consistency among analysis, design and programming activities.

1.5 Feasibility Study

A feasibility study is an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. It must account for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. It is the measure of evaluates and analysis of the project's potential for success.

1.5.1 Operational Feasibility

The proposed system has been operationally functional and user friendly. The users of system use the system to perform its regular task without any ambiguity. The user of the system can understand the new system with little trial. Operationally the system has capable of producing quick and qualitative services to users therefore the system developed is operationally more feasible. Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. It determines how the system satisfies the organizations need and it also offers secure, accurate and efficient system to the organization. The system in which we will have developed is also compatible to all operating systems. Operational feasibility also performs the following tasks:-

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.
- User friendly-the system will be easy and user friendly to be acceptable of users.

1.5.2 Economic feasibility

The purpose of economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system provides. It includes quantification and Identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

House rent site is available to anybody and anywhere. So the system is more functional in reducing costs from both users and the owner.

Some benefits the system provides are:-

- Small response time and many services.
- Easy and fast file management.
- Reduce cost for manual data management (Reduced expenses).
- Easy update & retrieval on stored records.

Our system is economically feasible, because it reduces the time needed to perform certain actions such as budget to cover man-power, paper and pen which they are using for manual work.

1.5.3 Technical Feasibility

The technical feasibility is primarily concerned with ascertaining whether the needed technology is existed to accomplish things as have been suggested. Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. Technical feasibility also performs the following tasks. Analyzes the technical skills and capabilities of the software development team members, and determines whether the relevant Technology is stable and established. The proposed system is technically feasible as:

- This project is practically applicable with in the current technology.
- The software used to implement the system is well understood by the group members.

We have technical knowledge about:

- PHP used as an interface with WAMP.
- Notepad/Notepad++ to write and edit the code.
- MYSQL to build the database for storing the data.

- Unified Modeling Language (UML) model to do analyzing and designing in good manner.

1.5.4 Legal feasibility

The proposed system is legally feasible since it does not violet or contradict any law, regulation, custom, value of Ethiopian constitution and Ethiopian statistical agency.

1.6 Project Scope and Limitation

1.6.1 Scope of the project

The scope of the project can be described as the overall features of what the new system is capable of doing. This system has different features which make things easier for the office. Generally our proposed system will support the following features.

- The system registers and post houses.
- The system manages all users account.
- The system updates house information.
- The system view reserved house and free house
- The system sends request.
- The system view request.
- The system sends response to the customers.
- The system view houses information
- The system view customers' requests
- The system send withdrawal request
- The system generates report
- The system send feedback

1.6.2 Limitation of the project

- The limitation of our system is that all users may not be participating directly to the system since it is not done by a local language and due to shortage of devise.

- It does not include kebele houses.

1.7 Significance of the project

After implementing the project our proposed system provides many advantages for customers, employees as well as owners.

The main significance of the project includes:

- Customers will reserve house by selecting the place, type of house and the cost they can pay by viewing all recorded house details.
- Houses will rent with short time.
- Houses can become available to customers within the time of they want to rent.
- The system is more secured than existing one so files cannot be lost easily.
- It increases performance of the system.
- It makes disabled persons participant in the service. .
- Decrease the difficulties of customer.
- Makes the system user friendly.
- Motivate the people to use online system.
- Provide knowledge and internet service for the people.
- Easy and manageable report
- Users can reserve house online even they are not in wolkite city they can reserve before coming and they can directly go to the house after they come.
- Save energy of client so they cannot lose their power to search houses on foot.

1.7.1 Tangible benefits

These are benefits that can be measured in monetary terms. These include:

- Reduce the amount of paper need.
- Reduce the amount of pen need.
- Reduce cost of printer for printing notice.
- Reduce transport cost.

1.7.2 Intangible benefits

Intangible benefits cannot be measured in monetary terms directly but they do have a very significant business impact. These include:

- Reliable data retrieval.
- Increased system performance.
- Data accessed easily.
- Data security is better.
- Improve customer satisfaction
- Decreased response time

1.8 Beneficiary of the project

- Handicap peoples: - by nature or different situations peoples may become disabled so they are not enabled to find things easily so we have to provide easy way to them.
- Customer:-this system prevent renters from extra expense for finding a house to rent and they can get the service everywhere at any time.
- House owners:-The house owners become beneficiary regarding to the houses to rent with in time and also they can save their time, energy and money at the same time
- Admin:-We are also beneficiary from the system because we are going to know detail information about the existence system and we are going to gain knowledge of how to develop new system and also as we are owner of the system you will simply start work after graduation by joining together and start business by owner system.

1.9 System development Methodology

1.9.1 Data collection methodology

Observation

It is fact finding technique where the system analyst either participates in or watches a person performs activities to learn about the system. It gives us better measurement than the authorized personnel and we will be able to see exactly what is being done, it is expensive and highly reliable.

By observing the existing problem we are gathering the data. Also we have referred other senior projects. And we observe the entire broker working procedures and renters.

Interview

It is fact finding technique where by the system analyst collects information from individual face to face interaction. It gives as an opportunity to motivate the interviewee to respond freely and openly and it allows us to prove for more feedback from the interviewee. In addition we believe that we adapt reward questions for each individual. Having an interview making questionnaires and observing in the around problems. Also we collected raw data or documents which are useful for the project should be implemented.

Document analysis

The project team tried to discover written documents about the organizational areas relevant to the project.

1.9.2 System analysis and design approach

There are different system development approaches from those we used:

Object oriented system analysis and design methodology: our team members choose OOSAD rather than others approach through the following reasons:

- Increase re-usability: the object oriented provides opportunities for reuse through the concepts of inheritance, polymorphism, encapsulation and modularity.
- Increased extensibility: when you to need to add new feature to the system you only need to make changes in one part of the applicable class.
- Maintainable: object oriented methods make code more maintainable. Identifying the source of errors becomes easier because objects are self-contained (encapsulation).
- Improved reliability and flexibility: Object-oriented system promise to be far more reliable than traditional systems, primarily because new behaviors can be "built" from existing objects.

1.9.3 System development model

We used Agile SDLC model. Agile SDLC model approach has the following advantages:

- The main objective is satisfaction of customers, so it focuses on single requirement with multiple iterations.
- Its technical excellence and good design.
- It requires the system to be developed in increments, each increment is independent of each other's, and at last all increments are integrated into complete system.

1.9.4 System testing methodology

Unit testing: While using unit testing is done at the source or 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 the testing process, the online house rental management system developers test the different sub procedures and functions.

Sample Tests:

- 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.

Integration testing: Integration testing is performed to test individual components to check how they function together. In other words, it is performed to test the modules which are working fine individually and do not show bugs when integrated. It is the most common functional testing type and performed as automated testing.

Sample Tests:

- Check the interaction between individual functionality which performs the specific tasks.
- Evaluate the functionality of the subsystem after combination all individual functionality.
- Identify the independence of each subsystem with another subsystem.

System testing: It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

Sample Tests: Evaluate the functionality of the subsystem after a combination of individual subsystem whether it works correctly or not.

- Check the coherence and coupling of each subsystem.
- Check the overall functionality of house rental management system that achieves the user's requirement.
- Check the interaction of each subsystem that performs the specified business process.
- Verify the system completeness-based user's requirement.

User acceptance testing (UAT): It is the last phase of the software testing process. In UAT actual software/app users test the software to make sure it can handle required tasks in real-world scenarios. Generally, it is performed at the time of product delivery to the organization as a final checkpoint among all functional testing types. From starting to deployment, the software/app undergoes through various types of testing by testing team and developers. The end goal of all the efforts is to deliver a working software/app that fulfills users' requirements and the client's expectations. The end users such as the owners and customers give feedback. How they are interact with the system that is developed.

1.9.5. Development tools and technology

1.9.5.1. Front end technology

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development.

HTML 5

- Include new attribute, elements, and abilities.
- Makes creating accessible sites easier.
- All browser support HTML.
- Increases the semantic value of the web page.

CSS

- It allows adapting the presentation to different types of devices, such as large screens, small screens, or printers.
- It allows to position and re-positions the components of a web page.
- It is browser compatible.

JavaScript

- To produce the same result on all modern browsers.
- User interface interactivity: JavaScript used to fill web page data dynamically.
- Rapid development: JavaScript syntax are easy and flexible for developers.

1.9.5.2 Back-end Technology

Back-end is the server-side of the website. It stores and arranges data and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. PHP and MySQL are the languages used for back end development.

PHP and MYSQL are software of the database system will be used in developing and managing the back end of the system.

PHP

- It is widely used open source general-purpose scripting languages that are especially suited for web development and can be embedded into HTML.
- PHP scripts are executed on the server.
- It is free to download and use.

MYSQL

- It is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL).
- It runs on virtually all platforms.
- It enables data to be stored and accessed across multiple storage engines.
- It is also capable of replicating data and partitioning tables for better performance and durability.
- MYSQL users aren't required to learn new commands; they can access their data using standard SQL commands.

Apache

- It can be modified to adjust the code and also to fix errors.
- It can be installed easily.
- Apache can run on almost any operating systems.

1.10 Document Organization

Our project documentation contains five chapters including the proposal which contains chapter one. Chapter two we defined and described the system requirements of the project. Then in Chapter three we seen analysis part of the system. In the fourth chapter we seen about system Design are going to be elaborated with regard to online house rental management system service. In the last chapter we seen about system Implementation.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

The existing house rental process in wolkite Town currently uses the traditional way of searching of house. The Admin that wants to rent house will announce the resident by tagging the information on their door. Then the person who wants to rent the house will find out if there is any house owner who announce their available house for renting. Then he will get in the house and asks general information about the utility which is provided by the owner and the price of the house. And if they can reach at an agreement they sign an agreement that deals about price and different conditions. Finally the customer enters to house.

2.2. Users of Existing System

- The owner of house is the person that writes the vacancy about house and tags it on the door then the customer read that vacancy and contact with him for renting purpose.
- Tenant is a person that finds a vacant house by journey and since he get home that fulfills his interest he call the renter of the house by the contact information tagged on the door.
- Admin is the owner of softewere.

2.3 Major Functions of the Existing System

The major function in the current system is renting a house to customers. The system announces the customers that there are houses to be rented. Activities in the existing system are:

Input: when house owners renting the houses they use:

House details as input in order to make the customers aware about the house.

Lease agreement policy between house owner and customers.

Process:

Posting the house details for customers by writing piece of information on computer then print the information; then post it on the door of the house to be rented.

Then waiting for request of the interested customer. If the request reached; the customers personal information, payments status and rent agreements are filled in the house rent agreement form in order to hold legal contract between the customers and house owners for renting the house.

Output: after completing these processes the house will be rented.

2.4. Drawbacks of the Existing System

There is no existing system in the wolkite city currently, but when customer and owner of the home make agreement some problems may be phased.

- From the two, one may not fulfill the agreement they made orally.
- The customer may leave the without pay for the house.
- The owner of the home may increase the price because of there is no legally documented low in wolkite about house rental.

The other problem is that there is no any modern tool to search and find houses for renting as well as showing the location of houses for customers.

2.5. Business Rules of the Existing System

The business rules of the existing system

- BR1: Both owners and customer should made agreement and take full accountabilities.
- BR2: customer and owner meet to submit agreements

CHAPTER THREE

3. PROPOSED SYSTEM

House rental system is designed to provide fast and easy way of controlling all the activities of house renting. It is also used to communicate with customer's using the web, keep the data for the longest time with in the database.

Our proposed system provides an easy way for customer's to check available house's for renting. And also the system allows managing all house related files. This includes posting and updating houses .The system also provides an easy way for owner to see customer's request on the web and sending their response to the customer.

Generally, house rental system is concerned on managing house's, providing easy and understandable way of controlling the activities of house renting, and making easy and fast communication held between tenant, house owner and admin.

The major solutions to address the problems of the existing system are as follows. Better utilization of resources, performance, security, reliability, accuracy and in general better service and the new system is aimed to perform basic and crucial tasks of the organization. It contains a well-organized database server which makes data to retrieve, update easily. Since the computer is capable of performing and processing many and huge tasks too faster, efficient and more correct it is preferred to apply it on the system.

3.1 Requirement Analysis of the New System

The purpose of requirement is used to prepare requirement specification before starting the actual design. This enables the designer to have a clear idea about the structure and content of the information system.

3.1.1 Functional Requirements

Functional requirements consist of a specification of a function that the system must support. In this section we discuss what is our web based system is expect to do. The type of user in this system is customer (tenant), House Owner and admin of the system.

Owner**:**

- REQ-1: the system shall allow user to login before accessing system.
- REQ-2: The system shall allow user to change password.
- REQ-3: The system shall allow user to create user account.
- REQ-4: The system shall allow the user to update house information.
- REQ-5: The system shall allow user to upload houses information.
- REQ-6: The system shall allow the user to view customer request.
- REQ7: The system shall allow the user to send response for customer.
- REQ-8: The system shall allow the user to view customer feedback.
- REQ-9: The system shall allow the user to approve payment.
- REQ-10:-The system shall allow the user to manage user account.
- REQ-11: The system shall allow user to view houses information.
- REQ-12: The system shall allow the user to generate report.
- REQ-13: The system shall allow user to register House.
- REQ-14: The system shall allow user to send request.
- REQ-15: The system shall allow user to send feedback.
- REQ-16: The system shall allow the Customer to view response.
- REQ-17: The system shall allow user to view report
- REQ-18: The system shall allow user to backup and recovery
- REQ-19: The system shall allow user to view notification
- REQ-20: The system shall allow user to update user account
- REQ-21: The system shall allow the user to pay payment.
- REQ-22: The system shall allow user to reserve house for tenant.
- REQ-23: The system shall allow user to cancel reservation request.
- REQ-17: The system shall allow user to manage account.
- REQ-17: The system shall allow user to view payment progress.

3.1.2 Non-functional Requirements

Non-functional requirements are requirement, which are not very basic or essential for the system, but it can support and give more quality for the system.

The non-functional requirement of the system deals with how well the system provides service to the user. They are also called technical requirements.

It concerns to the technical aspects that your system must fulfill, such as performance-related issues, reliability issues, and availability issues.

The new system had the following Non-functional requirements.

3.1.2.1 User Interface and Human Factors

We satisfy all requirement of the user as much as possible by using prioritization and negotiation.

3.1.2.2 Hardware Consideration

Our proposed system is web-based which is implemented by using PHP, HTML, CSS, And JavaScript and it will be accessed on desktop computer and also on PC.

3.1.2.3 Security Issues

When we come to security the proposed system will apply different security mechanisms in which user can only access information and perform any activity through their privilege and credentials.

- User authorization: the system gives different permission to different users of the system to perform their activity.
- Backup data: the system has its own database that helps to store personal data which implies that there is no way of losing data.

Security is the main issue to protect and secure the entire project to unintended loss of data, deletion of data and modification by unauthorized and unauthenticated users. Here in our system, we try to handle this through: -

- Prevent unauthorized access of data: every users of the system must be access in their own privilege that the system grants to them.
- Generate error message: the system generates error message when user forget password and username.

3.1.2.4 Performance Consideration

- Time of response: our system is respond the requested task and operations quickly.
- Processing time: Since the system developed with efficient programming language and database upon request for user's Activities, the system under normal condition will process the request as quickly as possible by using multi-tier architectures.
- Users of the system: our system support multiple users at a time.

3.1.2.5 Error Handling and Validation

- Input error: the system handles many exceptions like inserting empty (null) value to the database, inserting a duplicated client id, inserting invalid data type. To handle the above errors, the proposed system will provide appropriate correction messages for each error.
- Login error: the system handles an attempt to login with an incorrect user account and password by displaying an error message.
- Input Validation: our system uses both JavaScript and php to validate different forms like registration form, on this form if the users are asked to email account and insert it, our system check whether the email is valid or not.

3.1.2.6. Quality Issues

The quality of proposed system will be better due to reduction of error. The system should give correct output for the users when they want to get services. The system will be effective and consistent in performing its required function in short period of time without delaying much time. There are some issues that our system going to solve:

- Availability: the system is available for users to get service at any time anywhere if they got an internet access.
- Reliability: our system is reliable in terms of security, accessibility and also on data storage mechanism.

3.1.2.7. Backup and Recovery

We build the system, we have a database on WampServer as a backup data. the system the admin host the system to make it accessible anywhere for every user and to do this the admin center must buy the storage from the web-hosting company and store personal data and other useful information of the system. Therefore, if our system fails or some things happen to the system, we can get our data from this storage.

3.1.2.8. Physical Environment

The system gives access for wolkite city. And it will be accessible anywhere at any time where the internet access exists. There are no external factors that affect our system.

3.1.2.9. Resource Issues

There are some constraints on resources for the full deployment of system like for hosting.

3.1.2.10. Documentation

We provide documentation for our system to give information or to make peoples deeply know about the system functional and non-functional requirement and also advantages of the proposed system over the existing system. This is helpful for user to be attracted by the system. And also, documentation used for user to help as guidance.

CHAPTER 4

4. SYSTEM ANALYSIS

System analysis is a process of collecting and interpreting facts, identifying the problems and decomposition of a system into its components.it is conducted for the purpose of studying a system or

its parts in order to identify its objectives. In this chapter had design and deal with different system analysis concepts such as use case model, use case description, object model (class diagram and data dictionary), Dynamic Model (sequence and activity diagram) and state chart diagram for our system.

4.1. System Model

System model is the process of developing abstract models of a system, so with each model had shown a different view or perspective of the system using graphical notation, which is now almost based on notations in the Unified modeling language (UML).

4.1.1 Use Case Model

A use-case model is a model of how different types of users that are participating in house rent at wolkite city interact with our system to get access from our system. As such, it describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying the goals of the users in house rent.

- It represents interaction between a user's in and our system.
- It describes the proposed functionality of the new system. It is a single unit of meaning full work. Each Use Case has a description which describes the functionality that had built in the proposed system.

Table 4. 1 Use case model

Actor	Use case
admin	• Login

	• Manage user account. • Update house information. • View customer request. • View customer feedback. • Approve payment. • Generate report. • Update account
House Owner	• Login • Create user account. • View customer feedback. • Reserve house for customer. • Upload houses information. • View customer request. • Collect requirements from customer. • vie w report
customer	• Login • Send request. • Give feedback. • View houses Information. • Pay payment • View customer feedback. • View response for their request. • View reserved house. • Send withdrawal request.

4.1.1.1. Use Case Diagram

A use case diagram contains four components.

- Boundary:-which defines the system of interest in relation to the world around it.
- Actors: -usually individuals involved with the system defined according to their roles.

- Use cases: - the specific roles are played by the actors within and around the system.
- The relationships:-between the actors and the use cases

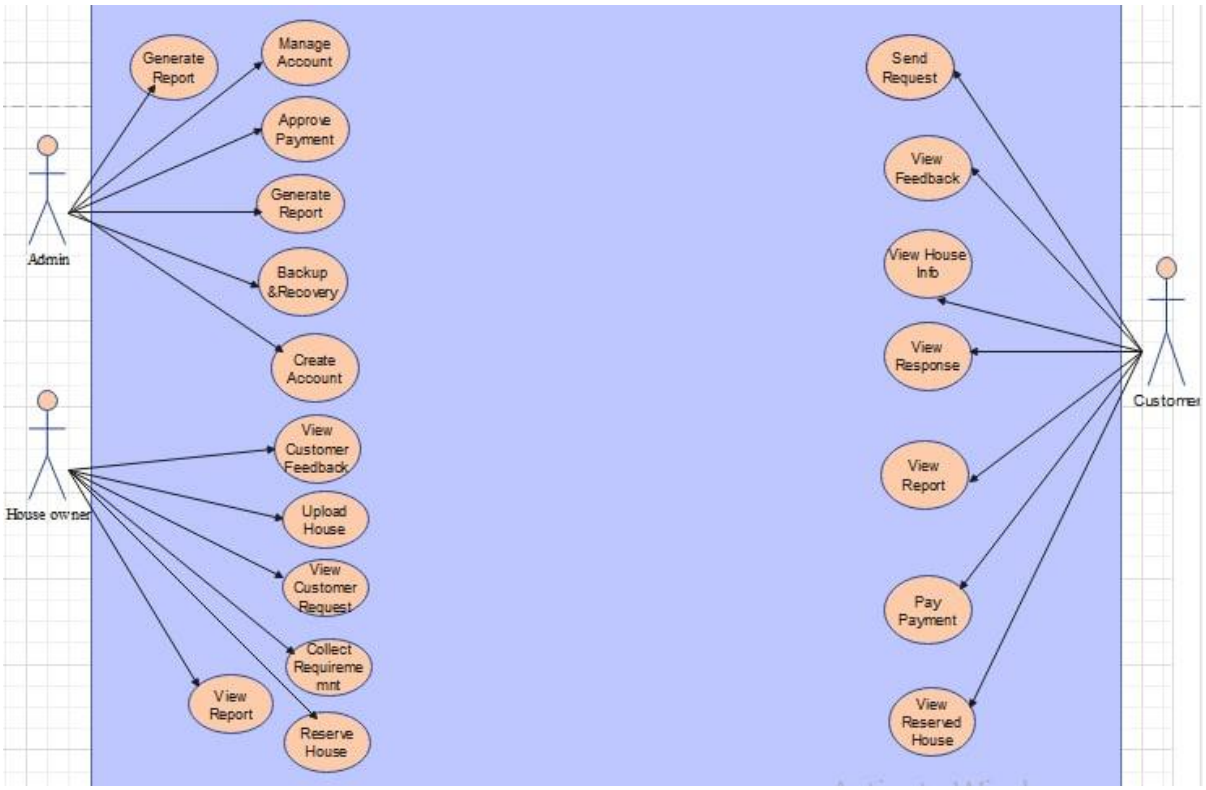


Figure 4. 1 Essential use case diagram

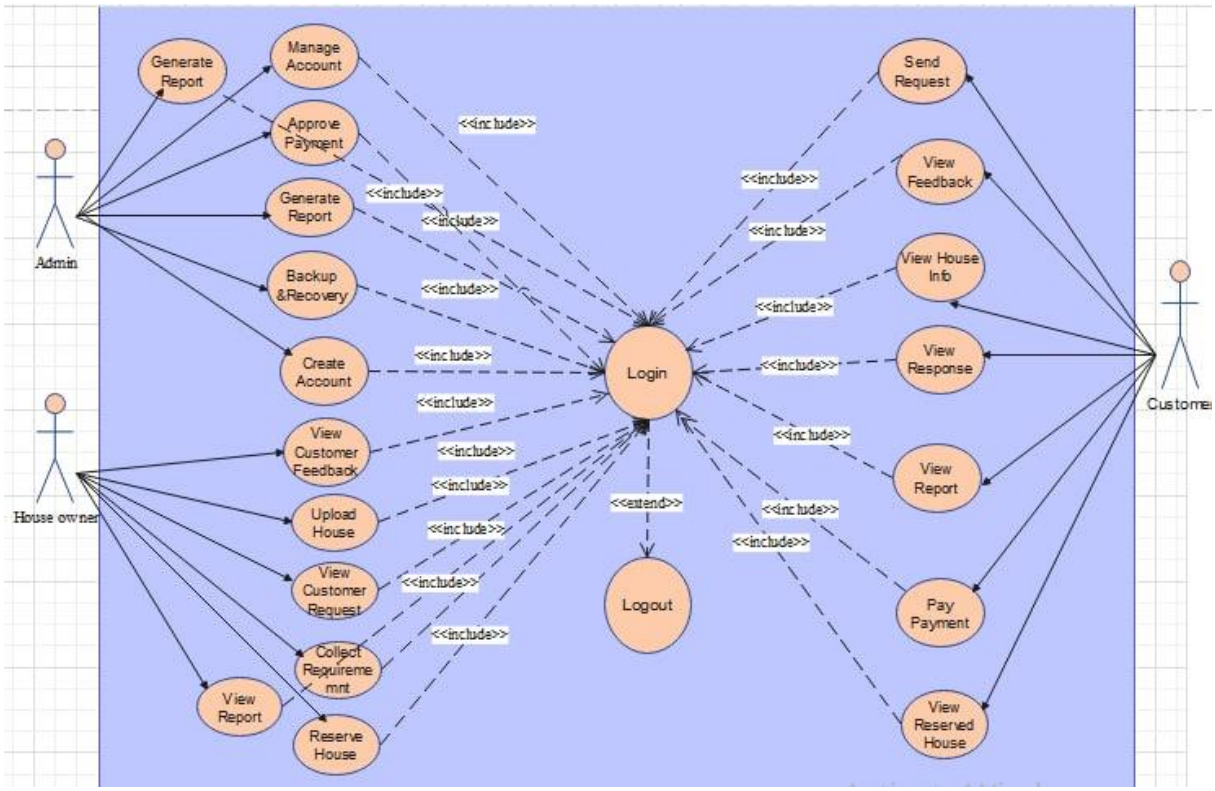


Figure 4. 2 System use case diagram

4.3.1.2. Use Case Description

Table4. 2 Create user account use case

Use case name	Create user account
Actors	House Owner
Description	This state allows these actors, to create User name and password in order to access the system.
Pre-condition	The actors should be new user
Basic flow of event	1. The admin and House Owner enters to home page 2. Then click register button on home page

	3. The user fill valid the required data on the form and click register button 4. The system displays registration form 5. 5. The system can validate all inserted data 6. the system can register on Database 7. The system displays your account is successfully created 8. Use case ends.
Alternate course of action	A1. If the entered required information is not valid it returns step 4 to refill the form.
Post condition	The new user is registered.

Table4. 3 Login use case description

Use case name	Login
Actors	Customer, Admin, House Owner
Description	The Customer, Admin, and House Owner uses this form to log into the system.
Pre-condition	The Customer, Admin, House Owner and should be register first or need to have an account to login.
Basic flow of event	1. The Customer, Admin, House Owner and enter to the home page 2. System displays login form 3. Enter user name and password and click login button 4. The system checks for the validity of the user name and password

	5. The system displays the customer, and Employee page 6. Use case Exit
Alternate course of action	1. If the entered user name and password is not valid it displays “please enter valid user name and password” go to basic flow 2. 2. If the user forgot the user name and password display reset user name and password page.
Post condition	The users (Customer, Admin and House Owner) logged in to the main page.

Table4. 4 Send request use case description

Use case	Send request
Actor	Customer
Description	This use case permits to register rental information of the customers and the house that the customers rent.
Pre-condition	Login
Flow of event	1. This use case is initiated when the actor’s clicks on send request option. 2. System displays the page that contains information to be registered 3. Actors fill all the information 4. Customer clicks or press on the Save or insert button. 5. The system verifies that the fields have been filled out correctly.

	6. System displays send successfully. 7. Use case Exit
Alternative course of action	4.1 If the customers fields are not filled out correctly system goes back or returns to step 4 of basic course of Action to fill valid data.
Post-condition	Customer's information is registered.

Table4.5 Change password Use Case description

Use case name	Admin and House Owner
Actor	Allows the system users to change their password.
Description	Allows the system users to change their password.
Pre-condition	User must have older password
Flow of event	1. User login first into the system. 2. Choose change password option 3. System had display the page. 4. User fills the information required and press change button. 5. The system checks for validity. 6. Display success message. **7.** Use case exits
Alternative course of action	If the entered data is not valid the system displays error message and returns basic course of action 3.
Post-condition	User password changed successfully.

Table4.6 Use Case description of manage user account

Use case	Manage user account
actor	Admin
Description	The system admin can manage the user account. First the actor's login to system and go to account then there is update, insert image and other management activity and they can do everything they want to change.
Pre-condition	Login to the system. Click profile. And manage the user account.
Flow of event	1. After successfully login to the system owner click on account button. The system display registration form. 2. The system display users profile. 3. Admin get profile page and also, they get different management activities such as update, insert, and edit. Admin select the activity they want to perform. 4. System display the page owner select. 5. Admin perform their activity and click validating activity button. 6. System validate the information that admin insert. 7. Perform activity and display change.
Alternative course of action	Steep 6 the system checks the information that user inserts to change or edit and if they insert wrong information the system display error messages and return the process back to step 3 to make actors check their information

Post-condition	Account is managed and you can do everything with managed account. Also, can logout.

Table4.7 Register house use case description

Use case	Register House
Actor	Admin and House Owner
Description	This use case permits to register rental information of the Admin and House Owner and the house that they register
Pre-condition	Log-in
Flow of event	1. This use case is initiated when the actors clicks on register houses option. 2. System displays the page that contains information to have been registered. 3. Admin and House Owner fill all the information 4. Admin and House Owner click or press on the save or register button. 5. The system verifies that fields have been filled out correctly. 6. The system displays inserted successfully message. 7. Use case Exit
Alternative course of action	If landlords fields are not filled out correctly system goes back or returns to step 3 of basic course of action. To fill valid field.

Post-condition	Admin and House Owner information is registered.

Table4.8 View house information use case description

Use case	View house
Actor	Customer, Admin And House Owner
Description	When actors choose view house option, the system displays all available resources.
Pre-condition	User need to have an account if they want to see more house information.
Flow of event	1. This use case is initiated when the actors clicks on view resources option. 2. System displays available name of the Private admin, house owner and some descriptions. Customer click on house option 3. System displays all the available resources recorded in the database. 4. Use case Exit
Post-condition	Customer wants to know if house information is reserved or not.
Alternative flow of event	Actors see only name of the admin and house owner description if they don't want to login.

Table4.9 View response use case description

Use case	View response
actor	Admin and House Owner

description	This option allows the admin and customer to view the response for customer's request.
Pre-condition	There must be sent request earlier.
Flow of event	1. Admin and House Owner login to the page. 2. Select view option response option. 3. System had display the response if any. 4. The use case exit.
Post-condition	Able to view response for customer.

Table4.10 Upload house information use case description

Use case	Upload House info
actor	admin and House Owner
description	This use case permits admin and house owner to post rental information of the house that they will rent.
Pre-condition	Logging into the system Having house information to be posted.
Flow of event	1. This use case is initiated when the actors clicks on register houses option. 2. System displays the page that contains information to will be registered. 3. Admin or House Owner fill all the information 4. Then they click or press on the register button. 5. The system verifies that fields have been filled out correctly. 6. The system displays inserted successfully message. 7. Use case Exit

Alternative course of action	If the fields are not filled out correctly system goes back or returns to step 3 of basic course of Action to fill invalid field.
Post-condition	Customer information is registered.

Table4.11 View customer request use case description

Use case	Send response
Actor	House Owner and Admin
Description	When house owner and admin chooses view customer request option, the system displays information about who is requesting. Then he/she can reply the response to the requester.
Pre-condition	House Owner and Admin must first logged in. There must be request from customers.
Flow of event	1. The house owner click view request link. 2. System displays the response page. 3. House Owner or Admin fills necessary notification and click send button to send it. 4. The system will display success message to user. 5. Use case Exit
Post-condition	Response is sent to the customer.

Table4.12 Reserve house for customer request use case description

Use case name	Reserve the house
Actor	House Owner and Admin
Description	This use case permits house owner and admin to reserve the house for customer and customer make payment to the house owner based on the response from them.

Pre-condition	The customer and the house owner must rich an agreement.
Flow of event	1. The house owner or admin logged in and view all available houses for renting. 2. Then press sends request button. 3. The system displays request page and the user fills the form, press send button and waits for response from the customer. 4. If the customer sends response the house owner or admin press reserve button. 5. The system displays payment form for the customer. 6. Customer fills the form then press ok button. 7. System validates the information. 8. The system displays success message. 9. Use case ends.
Alternative course of action	If the customer filled invalid information, the system goes back to step 6 to enter the invalid field again.
Post-condition	House is rented to the customers.

Table4.13 View customer request use case description

Use case	View customer request
Actor	Admin, House Owner
Description	When admin and house owner choose view customer request option, the system displays information.
Pre-condition	Admin and House Owner need to have an account if they want to see more requests.

Flow of event	1. This use case is initiated when the admin and house owner clicks the view customer request option. 2. The system displays available name of customer and some descriptions. 3. Admin and House Owner click the view request option. 4. System displays all the available requests, recorded in the database. 5. Use case Exit
Post-condition	Admin and House Owner want to know if there is customer's request.
Alternative flow of action	Admin and House Owner only see the name and the description of customers

Table4.14 Update house use case description

Use case	Update house information
Actor	House Owner, Admin
Description	Only House Owner and Admin can update house information.
Pre-condition	Admin and House Owner need to have an account to modify the property
Flow of event	1. This use case is initiated when admin and house owner click on house information option.

	2. System displays the page that contains update buttons. 3. House Owner and Admin update resources. 4. System displays update successfully. 5. Use case Exit
Post-condition	House Owner and Admin want to modify house information

Table4.15 Generate news use case description

Use case	Generate news
Actor	Admin
Pre-condition	To perform this activity the admin must login first. Click view button and select from list the one you want to view.
Description	The system admin views different daily or monthly reports and generate news for house rent.
Flow of event	1. Admin login to system. 2. The system displays the admin home 3. Select the report want to view. 4. View the selected report. 5. Owner generates news according to report viewed.

	6. Actors perform their activity and click validating activity button.
Post-condition	The news is generated.

Table4.16 Send feedback use case description

Use case name	Send Feedback
Actors	Customer
Description:	To enable the users to send the feedback, to comment any Suggestion on text area.
Precondition:	The customer must be having a suggestion about the system as well as houses property.
Flow of event	1. The customer click feedback button. 2. System display text area box. 3. The customer writes some necessary information and click post button. 4. The system can validate if writes on text area or not. 5. The system can make save on database and post on home page. 6. Use case Exit.
Post condition	Suggestion will be posted

Table4.17 View feedback use case description

Use case	View feedback
actor	House Owner and Admin
Pre-condition	To view login to system first and there must be sent feedback by customer.

Description	This helps the service provider to improve services they are providing or to appreciate the services provider.
Flow of event	1. Login to the system first. 2. Select view request link. 3. System displays the requests if available else display there is no available feedback 4. Use case exit.
Post-condition	Able to view feedback

Table4.18 pay payment use case description

Use Case	Pay for rent
Actors	Customers
Description	The customers can pay for house and the house is reserved by owner
Precondition	The customers can view free house and click payment.
Flow of event	1. Customers must login first. 2. System display customer's home page. 3. Customers view house they want to rent 4. The system display free house 5. Click payment to rent house 6. The system checks customer account 7. Customers fill payment form and submit 8. System validate shipment form 9. The system displays the order and payment information in the form of pdf 10. Both customers and admin can view the information

Alternative course of action	• Step 6 the system checks customers account whether it is recharged or not, if it is recharged successfully and the account balance is enough for items customers select to rent if it's not enough the system back customers process to step 4 make customers to check his account • Step 8: the customers correctly fill the payment information if not the its process is aborted if customers fill successfully the system go to next step and if not the system display error message and back the customers to Step 7.
Post-Condition	The customers rent house successfully.

Table4.19Logout use case description

Use case name	Logout
Participating actors:	Customer, House Owner and Admin
Pre- Condition:	Customer, House Owner and Admin should be logged in
Description:	To Exit from services.

Flow of action:	Customer, House Owner and Admin clicks on logout button.

4.1.1.3 Use Case scenario

Table4.20 Use Case scenario

Scenario name: Upload info

Participating actors: Mrs. Saron.

Initial assumption: Mrs. Saron must have a user account and role for owner. Also, she is logged into the system and on the admin page.

The normal flow of events:

- Mrs. Saron clicks on the (upload image) link on the page provided for Admin.
- The System display page for uploading the image.
- Mrs. Saron will upload the student image and then click on the (upload) Button.
- The System preprocesses the input images and stores them in the database.

Alternative flow:

Mrs. Saron enters incorrect image type and the System displays a message that contains “enter the correct image type or you entered wrong value” had displayed.

Scenario name: send response

Participating actors: Mr. Negusse

Initial assumption: Mr. Negusse must have user account and login to system by its page (employee page).

The normal flow of event.

- Mr. Negusse view customer request to return for it response.
- Mr. Negusse clicks on the send response link on the page provided by admin.
- The System display page for sending the response to the customer.

- Mr. Negusse has send response and click on the response button.
- The System preprocesses the sent response and stores them in the database.

Alternative flow:

Mr. Negusse trying to send incorrect response and the System displays a message that contains "enter the correct response or you entered wrong value" had displayed.

4.2. Object Model

Object model is one of the object-oriented programing (OOP) lifecycle which is logical interface or system modeled through the use of object-oriented techniques.

4.2.1. Class Diagram

UML class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association) and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling. A class model is comprised of one or more class diagrams and the supporting specifications that describe model elements including classes, relationships between classes, and interfaces. UML class diagrams show the classes of the system, their inter-relationships, and the operations and attributes of the classes.

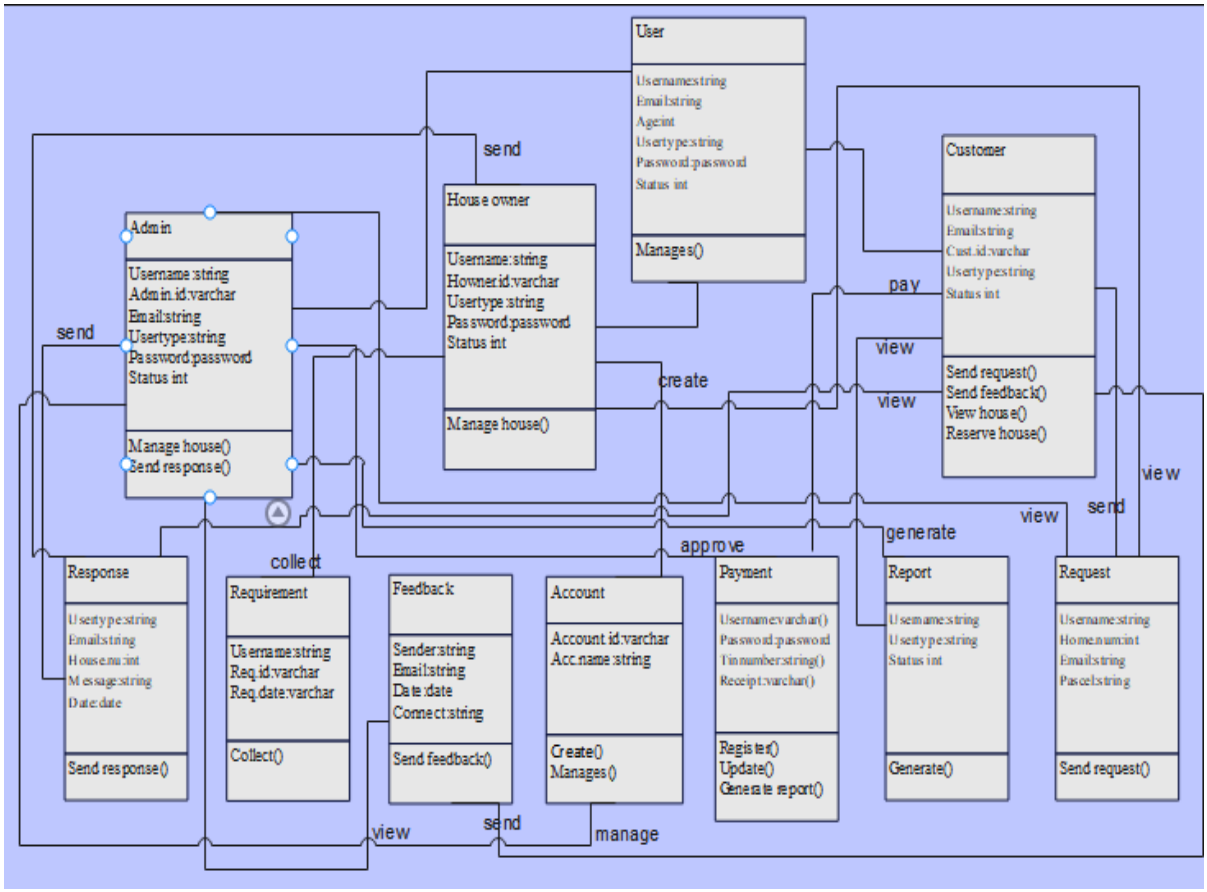


Figure 4. 3 UML class diagram

4.2.2. Data Dictionary

Table 4.22 Data dictionary for customer table

Field Name	Data type	Example	Constraints
Account_id	Varchar (50)	01	Primary key
Customer_id	Varchar (50)	Customer123	Foreign key
Customer_ip	Varchar (50)	127.0.0.1	Not Null
User_email	Varchar (50)	Customer@gmail.com	Not null
Customer_id	Varchar (50)	wolkite	Not null
Customer_contact	Varchar (30)	+251941800702	Not null
Customer_address	Varchar (50)	Wolkite	Not null

Customer_image	Blob	abdi.jpg	Not null

Table 4.23 Data dictionary for admin table

Field Name	Data type	Example	Constraints
Admin_Id	Varchar (50)	Admin123	Primary key
Admin_ name	Varchar (50)	Admin	Not Null
Admin_ email	Varchar (50)	Admin2021@gmail.com	Not Null
Admin_ city	Varchar (10)	wolkite	Not Null
Admin_ contact	Varchar (50)	+251949026480	Not Null
Admin_ address	Varchar (30)	wolkite	Not Null

Table 4.24 Data dictionary for house owner table

Field name	Data type	example	constraint
H.owner-Id	Varchar (50)	H.owner123	Primary key
H.owner-Name	Varchar (50)	House Owner	Not Null
H.owner-email	Varchar (50)	H.owner@gmail.com	Not Null
H.owner-city	Varchar (50)	wolkite	Not Null
H.owner-contact	Varchar (50)	+251941800702	Not Null
H.owmer-address	Varchar (50)	Gubrie	Not Null

4.3. Dynamic model

This model represents the time dependent aspect of the system. It concerned with the temporal changes in the states of the objects in a system. Here had seen sequence diagram to display the interaction between users, screen, Objects and entities with the system, activity diagram to

show how different workflows in the system are constructed this deal with how many decision paths that can be taken from start to finish, State charts are used to detail the transitions or changes of state an object can go through in the system and also process model used to model a business process.

4.3.1 Sequence diagram

Sequence diagrams, commonly used by developers, 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.

Figure 4.4 Sequence diagram for actors 'create account

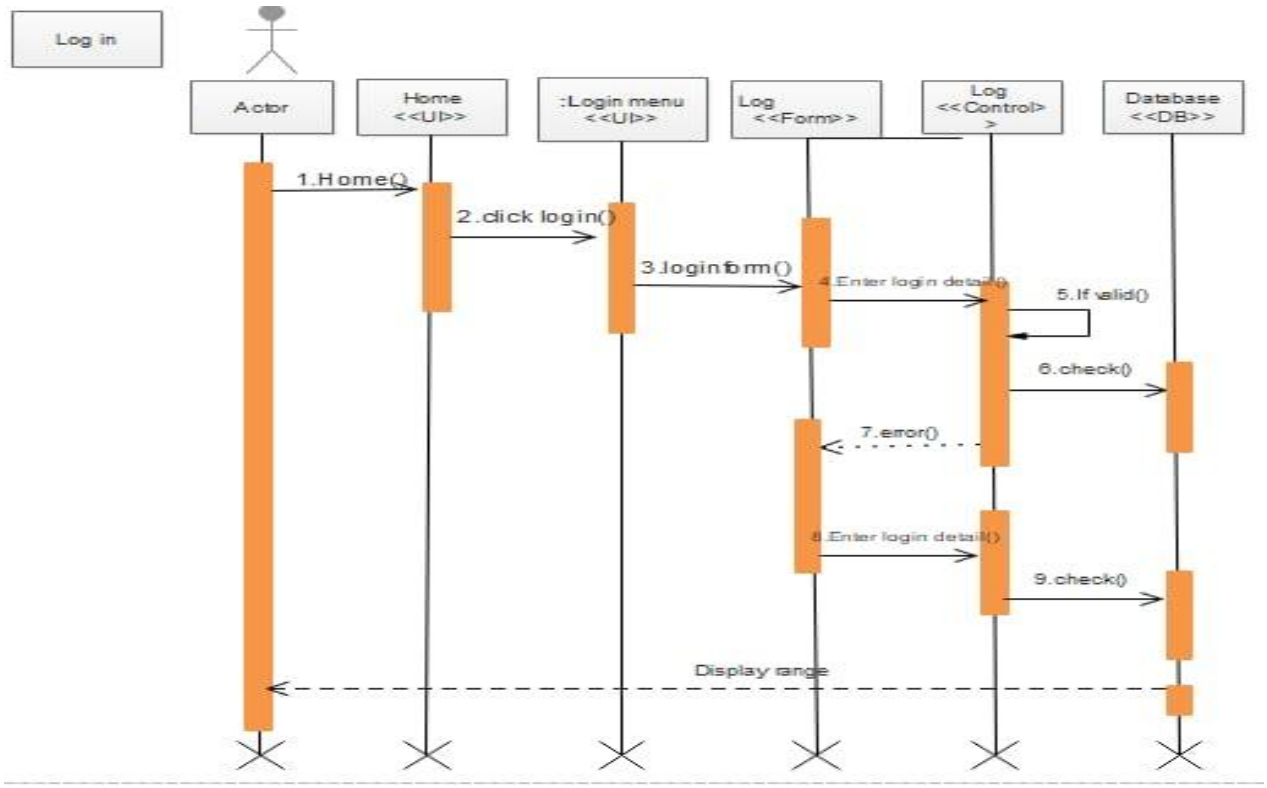


Figure 4. 4 Sequence Diagram for Login

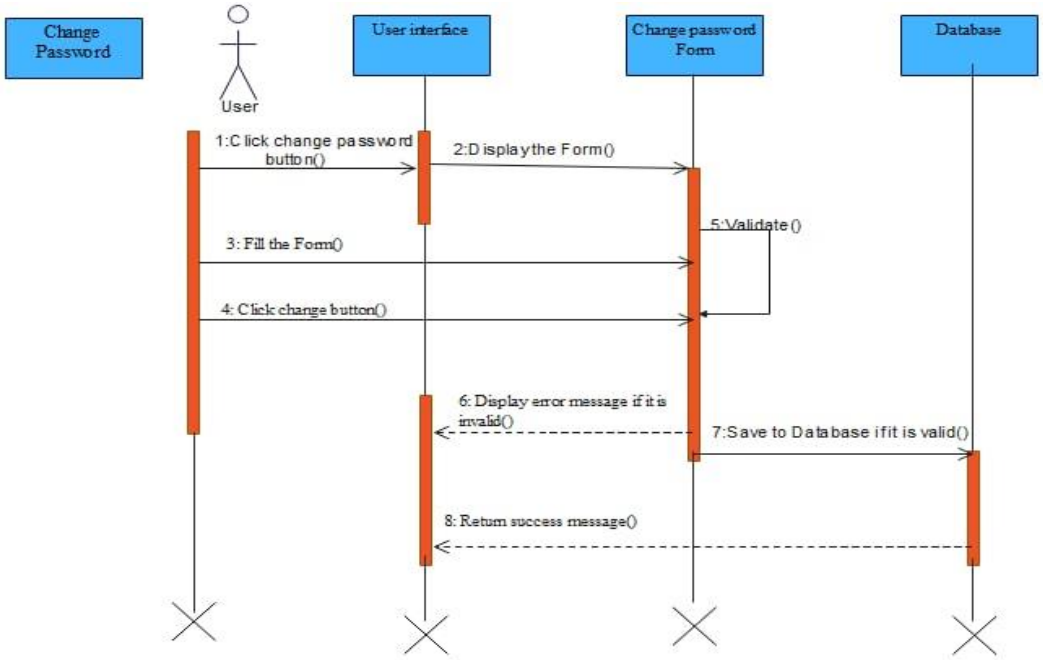


Figure 4. 5 Sequence Diagram for Change Password

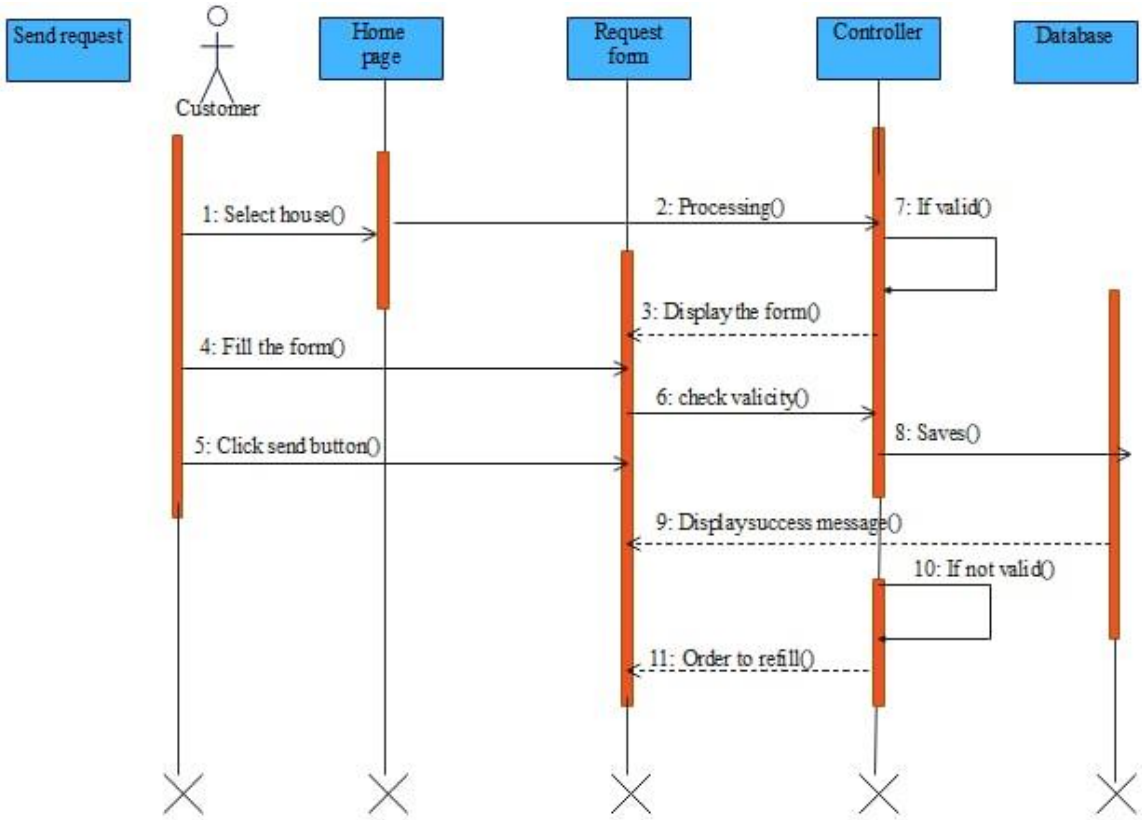


Figure 4. 6 Sequence Diagram for Send Request

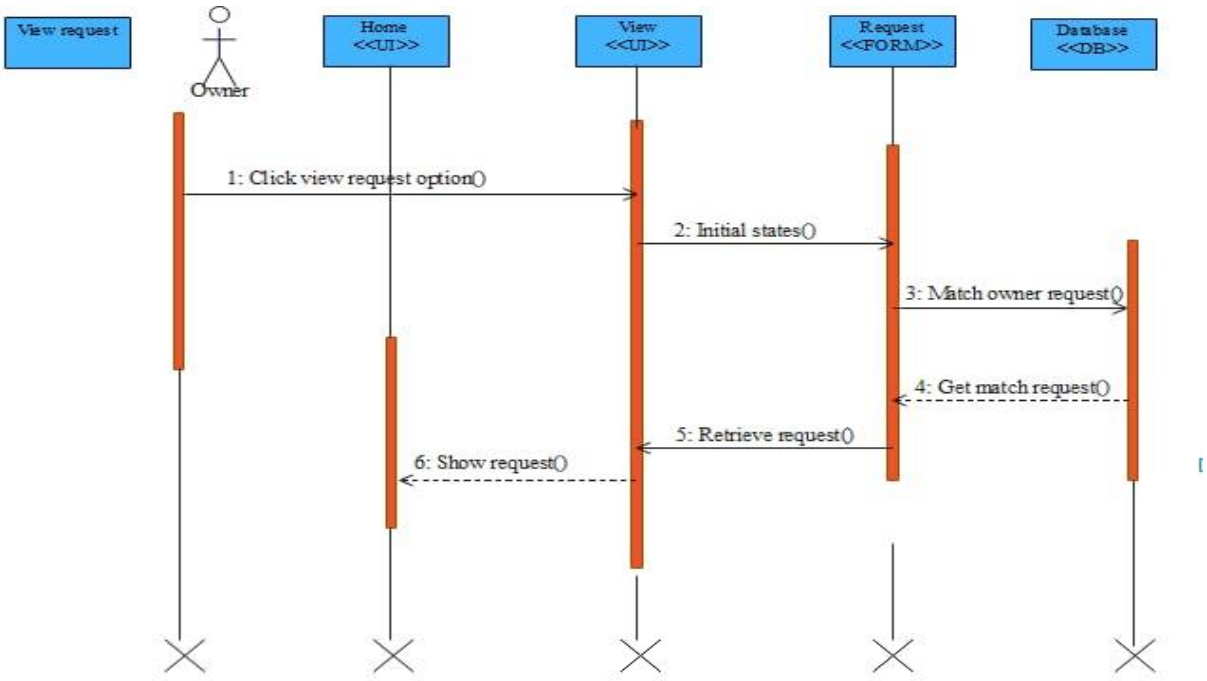


Figure 4. 7 Sequence Diagram for View Request

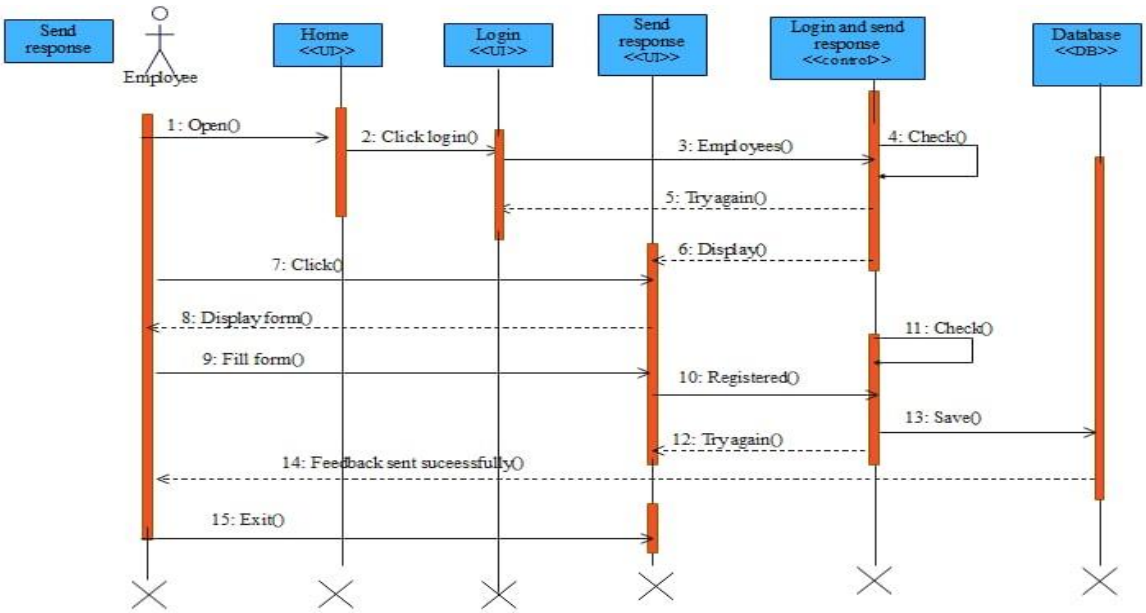


Figure 4. 8 Sequence diagram for actors send response

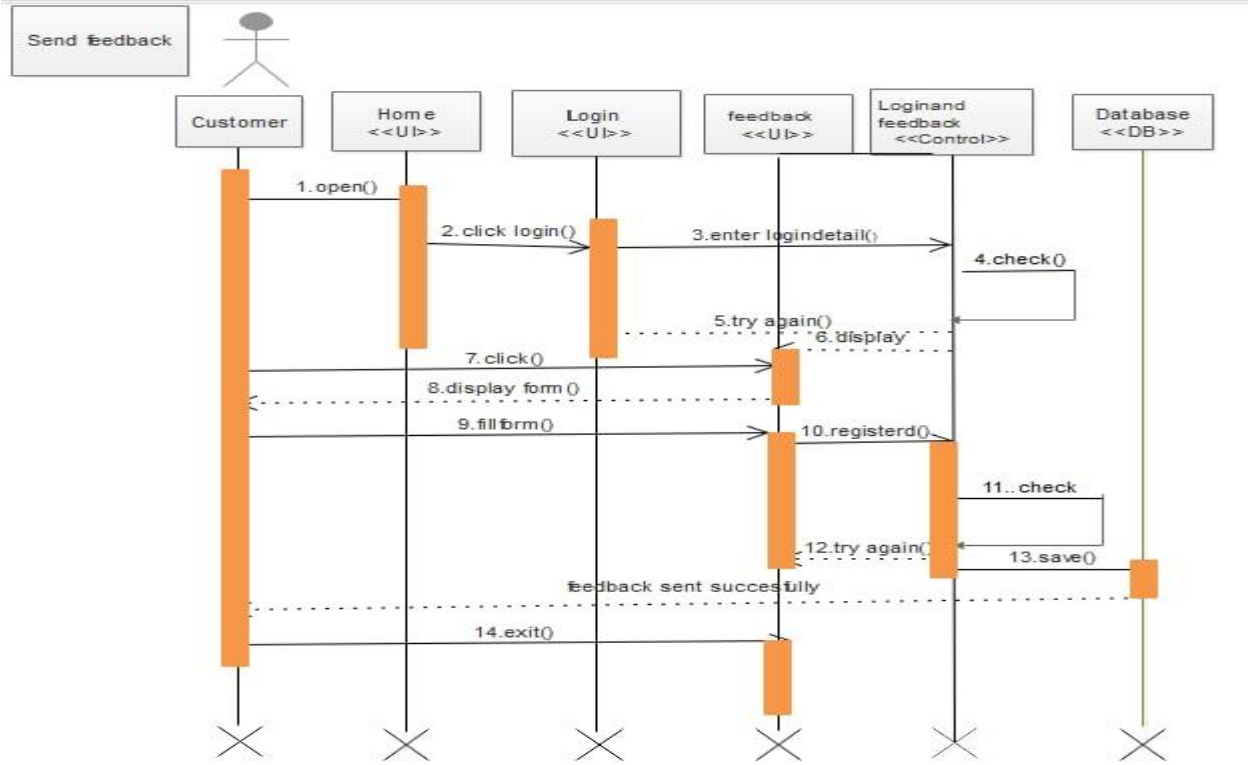


Figure 4. 9 Sequence Diagram for Send Feedback

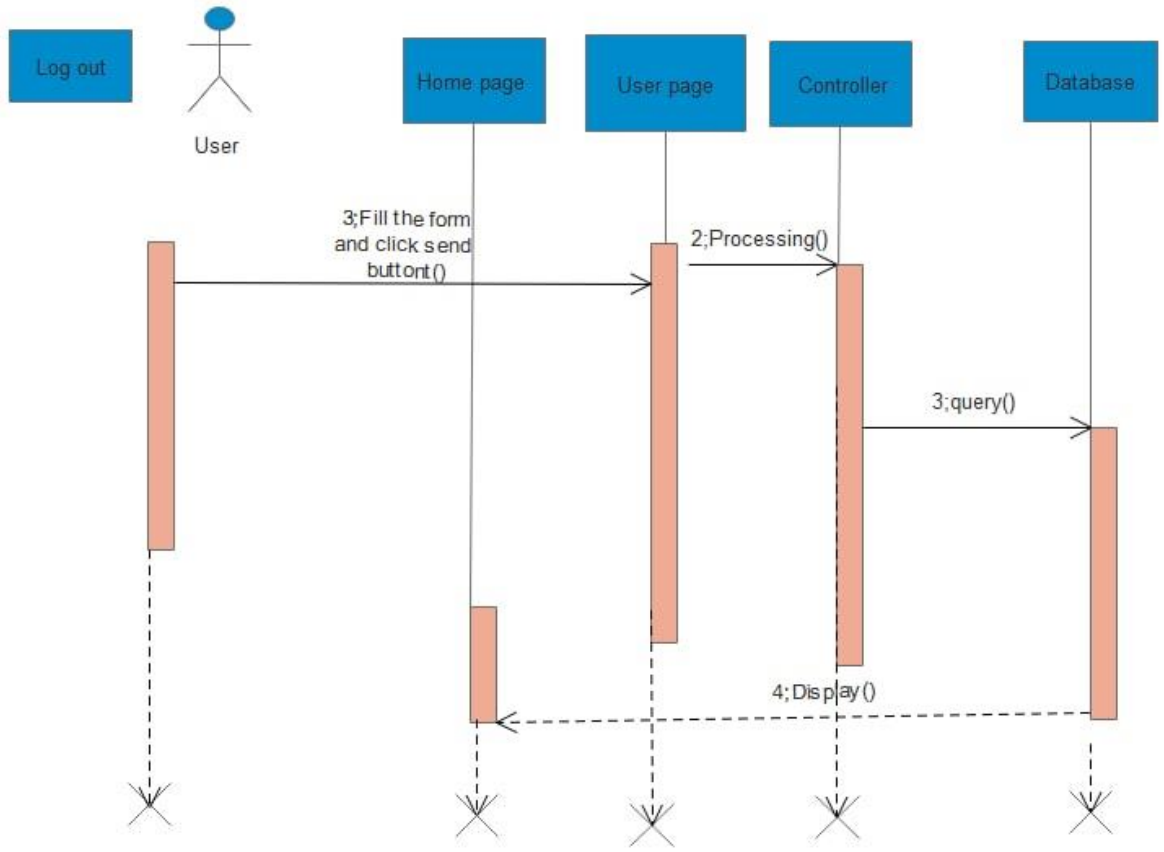


Figure 4. 10 Sequence Diagram for Logout

4.3.2Activity Diagram

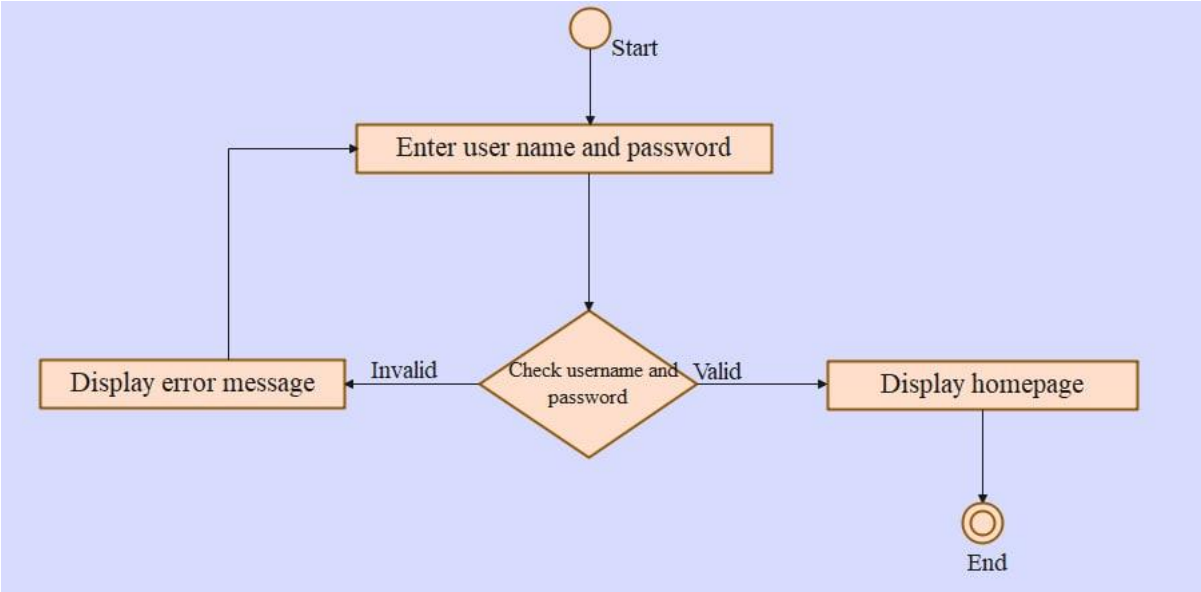


Figure 4. 11 Activity diagram for log in to the system

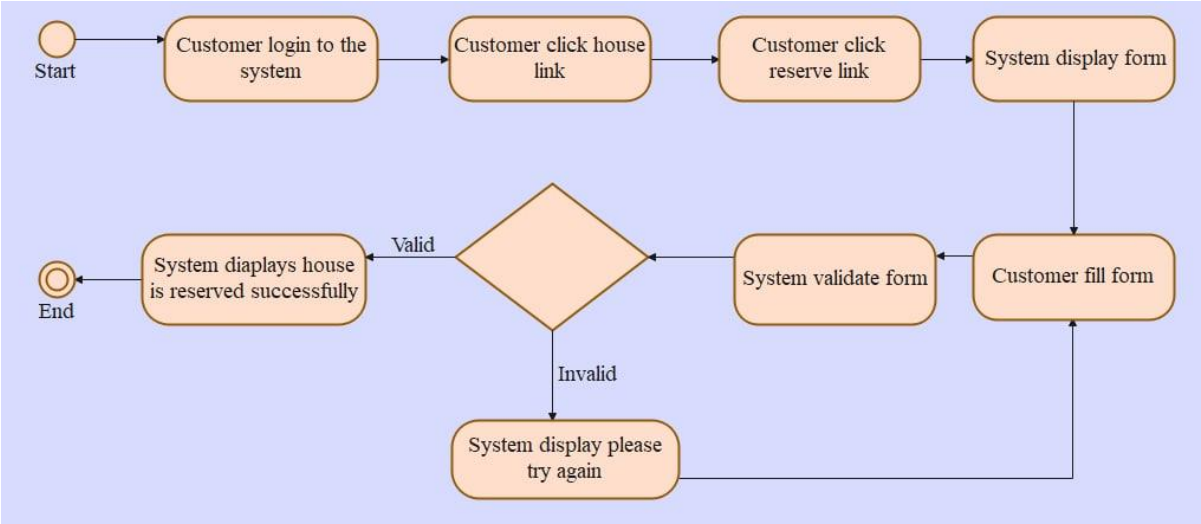


Figure 4. 12 Activity diagram for customer login to the system

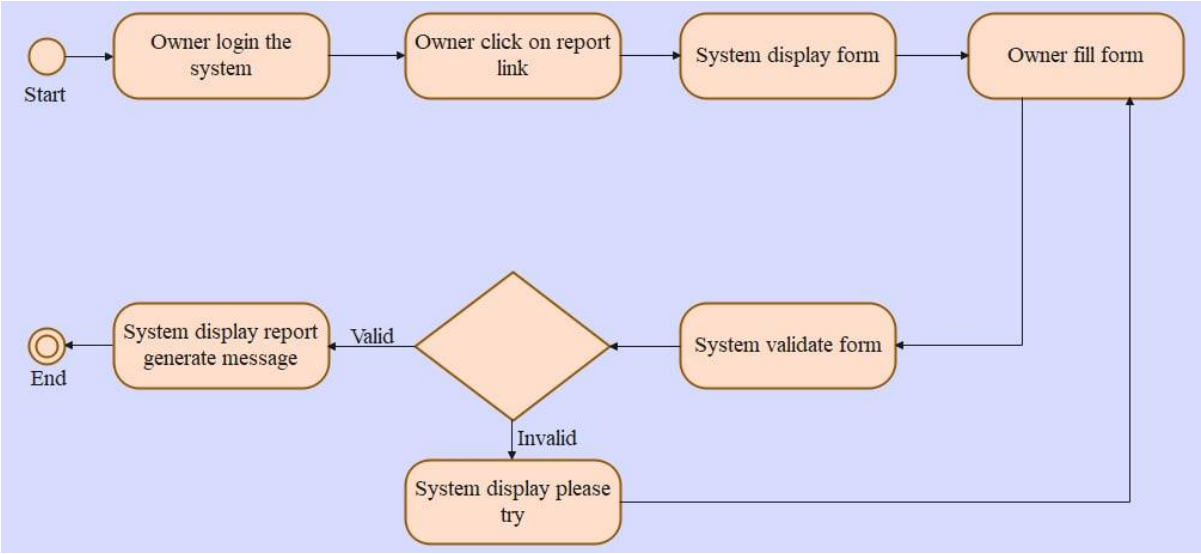


Figure 4. 13 Activity diagram for admin login to the system

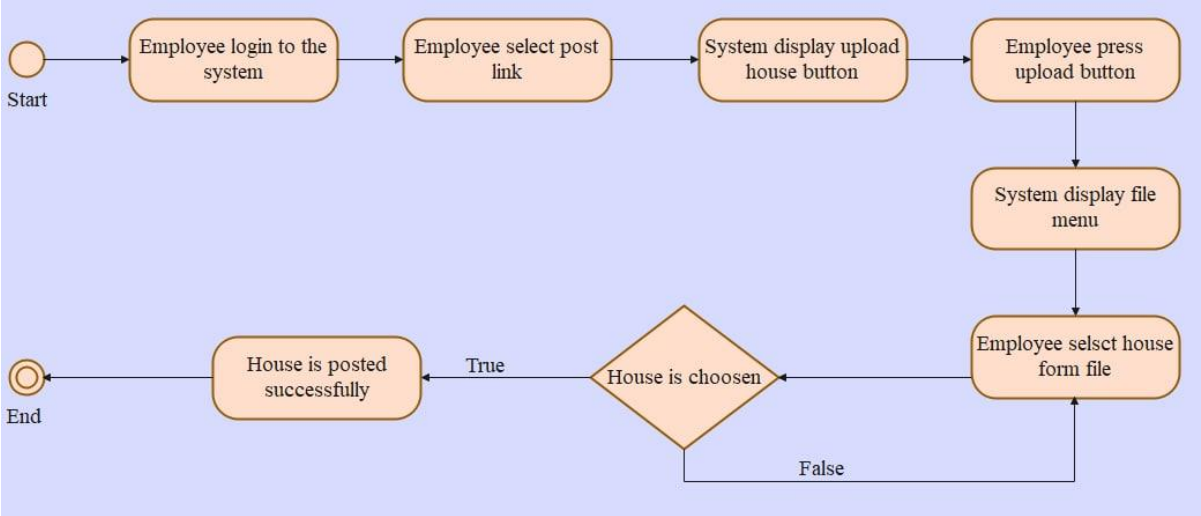


Figure 4. 14 Activity diagram for House Owner login to the system

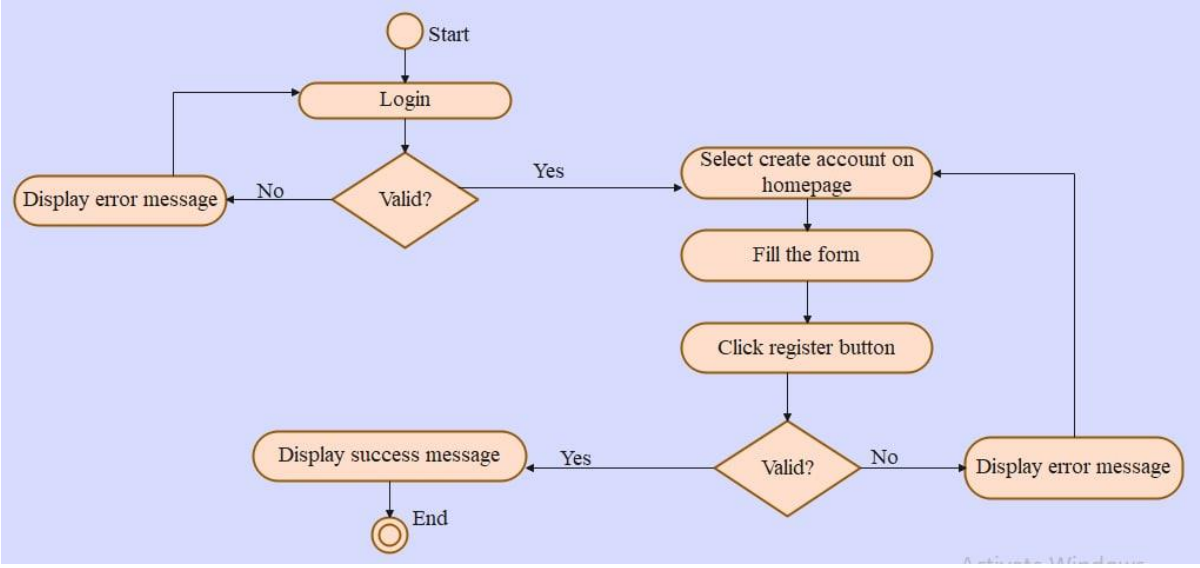


Figure 4. 15 Activity diagram for create account

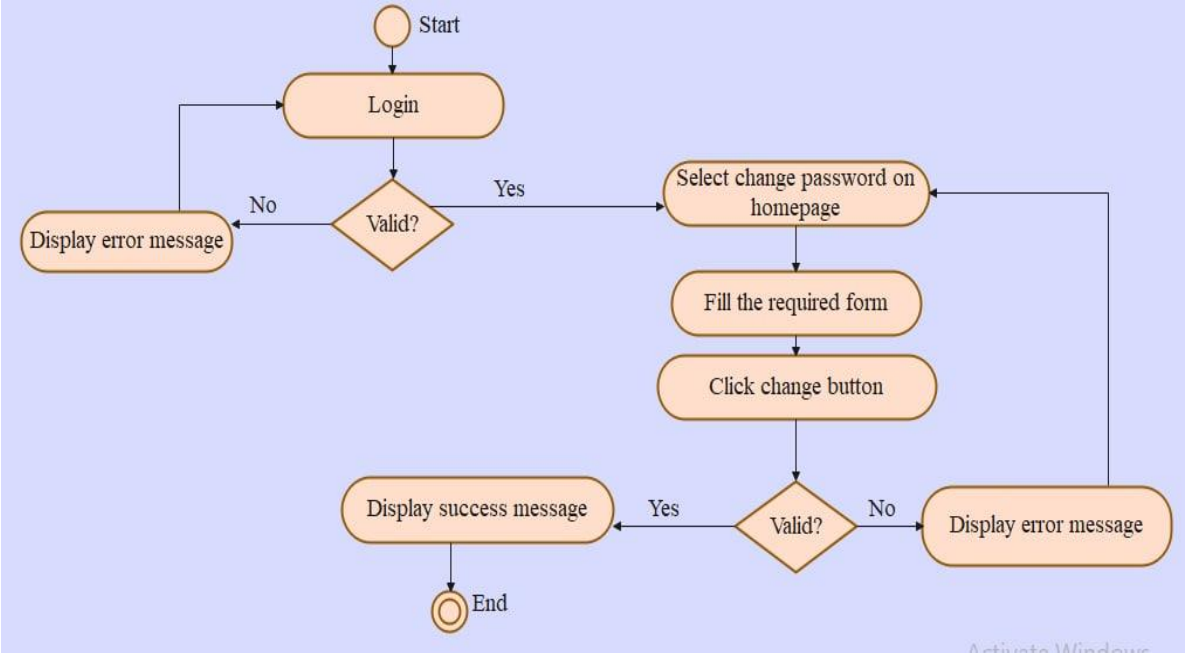


Figure 4. 16 Activity diagram for change password

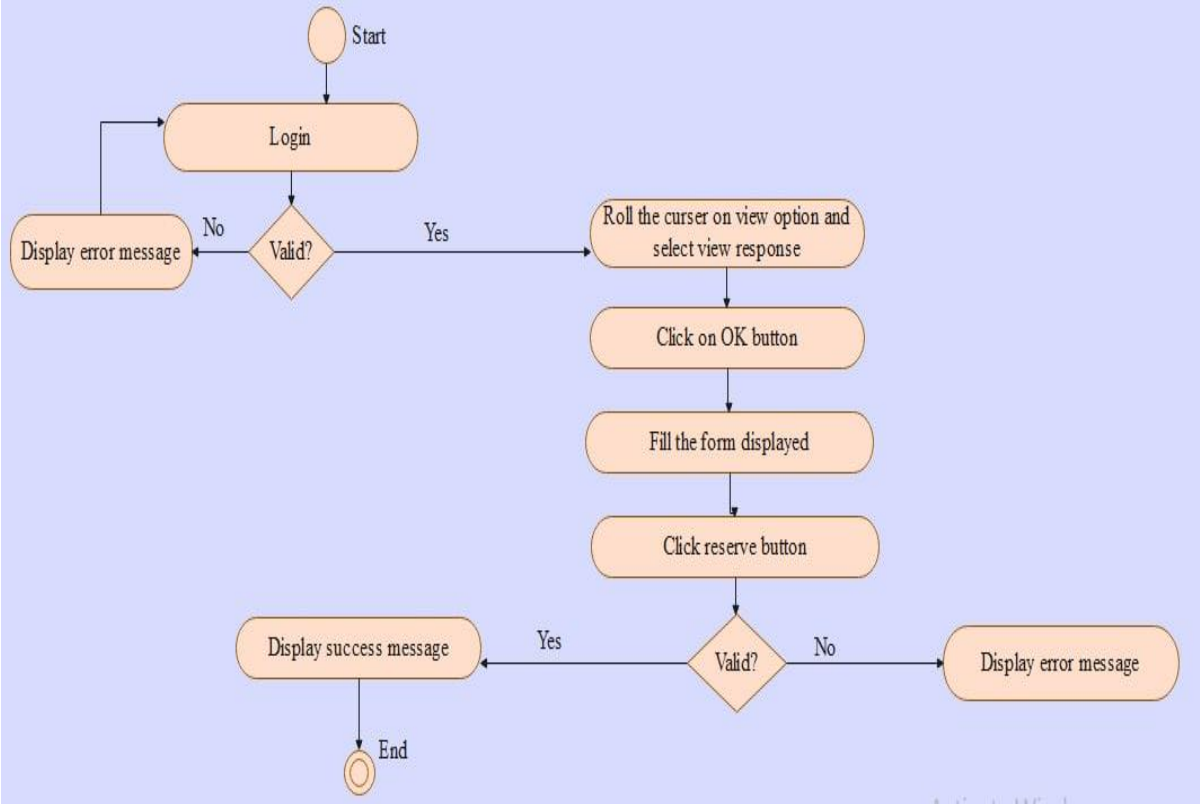


Figure 4. 17 Activity diagram for view response

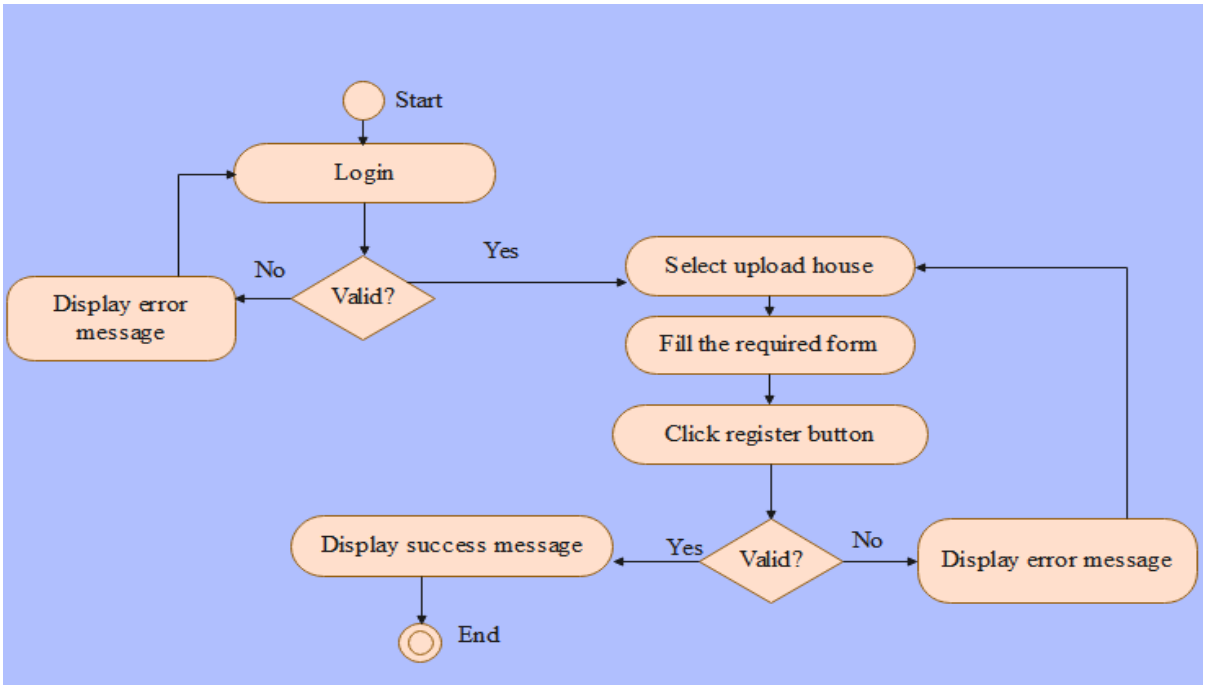


Figure 4. 18 Activity diagram for upload house

4.3.3 State chart

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. A diagram describes the flow of control from one state to another state.

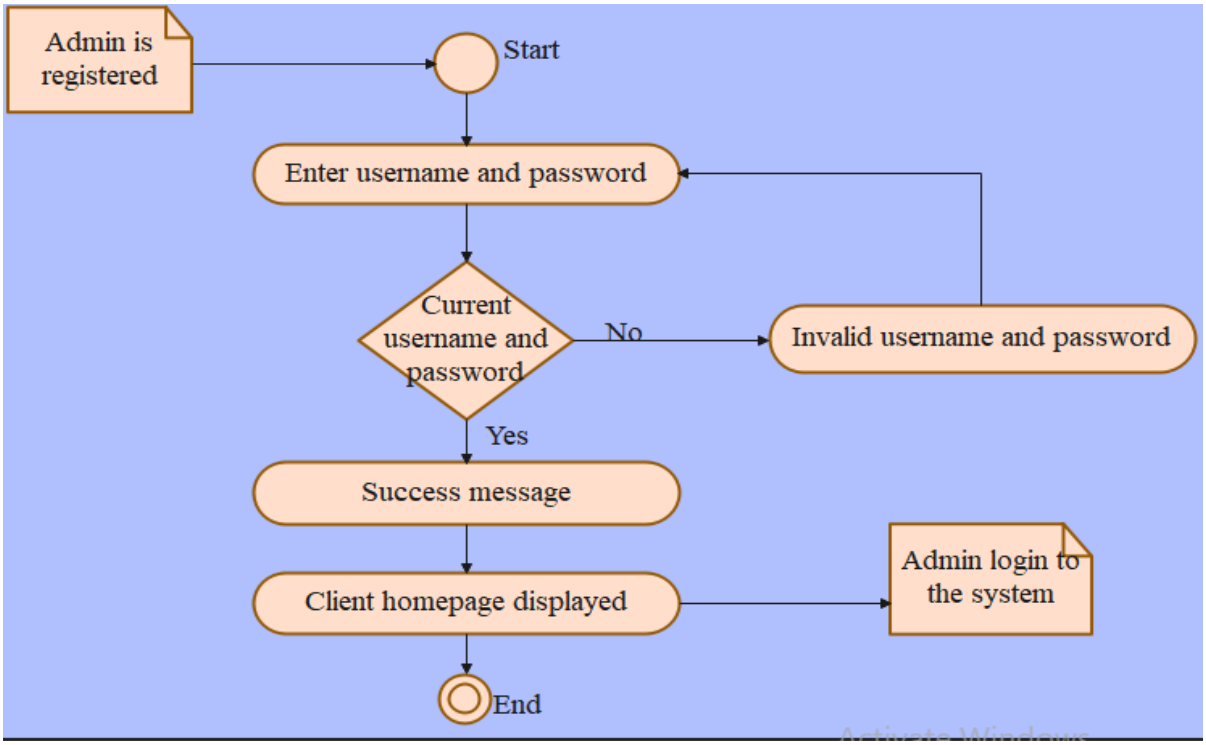


Figure 4. 19 State Chart Diagram for Login

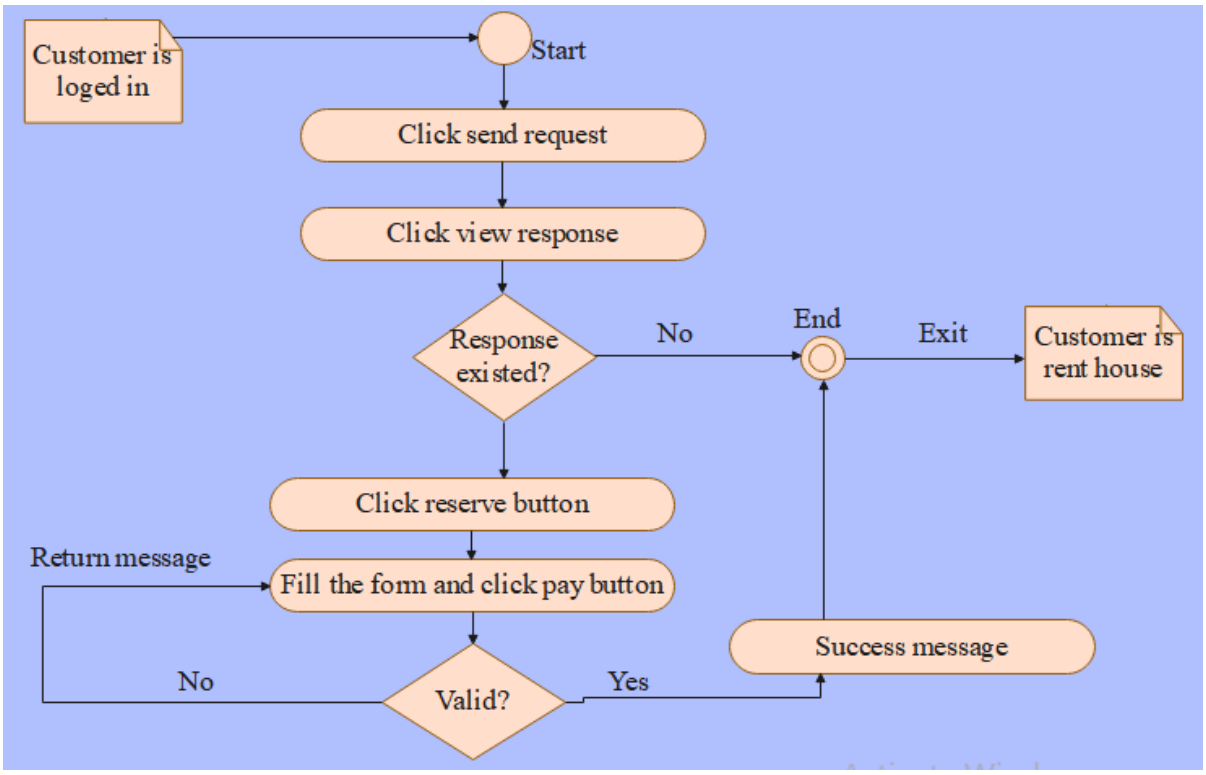


Figure 4. 20 State chart diagrams for reserve the house

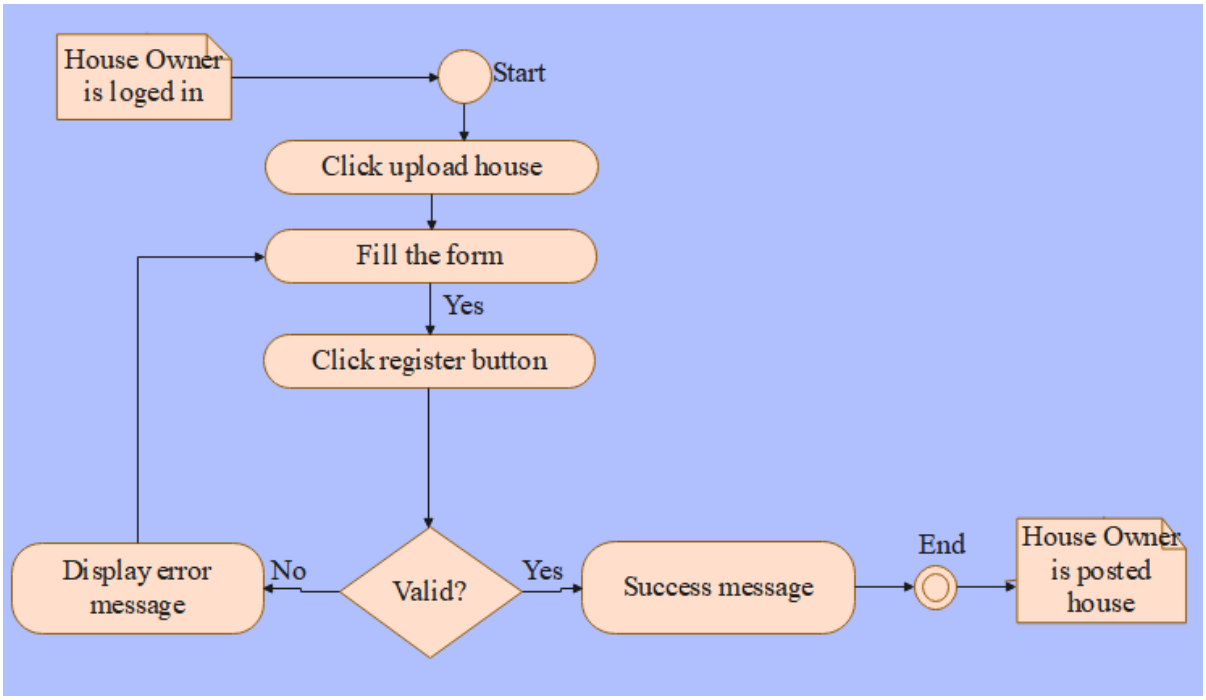


Figure 4. 21 State chart diagrams for Upload house

CHAPTER FIVE

5. SYSTEM DESIGN

In the previous chapter we have discuss the functional and non-functional requirements of the system and produced the analysis model. In this chapter we are going to describe the current and proposed system of architectural design. In addition to this we describe the access control and security of our system. The overall system design objective is to give an efficient, modular design that will reduce the system complexity, facilitate change and leads to an easy implementation.

5.1 Design goal

Design goals describe the important system qualities. Design goals also define the values against which options are evaluated. When designing a new system, a system designer creates a model of the system from requirements made. The quality of the model highly determines the quality of the product and future maintainability. The list down with pair manner as you listed no functionality of your project.

User Interface and Human Factors

The User Interface of the proposed system will be simple to understand, the user interface components are leading and action indicators. Thus, the users of the proposed system can easily perform the tasks that they intended to do. Every component of the user interface is arranged based on understanding our users in terms of their goals, skills, preferences, and tendencies. The proposed system's user interface can be easily operated by a user since most of the users of the proposed system have basic knowledge in related systems like SIMS.

Hardware Consideration

Our proposed system is web-based which is implemented by using PHP, HTML, CSS, And JavaScript and it had be accessed on desktop computer and also on PC.

Security Issues

When we come to security the proposed system has apply different security mechanisms in which user can only access information and perform any activity through their privilege and credentials.

- User authentication: only authenticated user can access the system by inserting their user account and password.
- User authorization: the system gives different permission to different users of the system to perform their activity.
- Backup data: the system has its own database that helps to store personal data which implies that there is no way of losing data.

Security measures are provided to prevent unauthorized access of the system at various levels. Security is the main issue to protect and secure the entire project to unintended loss of data, deletion of data and modification by unauthorized and unauthenticated users. Here in our system, we try to handle this through:-

- Prevent unauthorized access of data: every users of the system must be access in their own privilege that the system grants to them.
- Generate error message: the system generates error message when user forget password and username.
- Generate leased time: our system has lease time for users to try their account and password. One user can only try for three times if it doesn't insert the right password or account, they will wait for 30 second and try it back.
- Secured documents: there are different secured document for the user of the system such as comment from customers to admin, customers to sales person and also other document those are private for some specific user.

Performance Consideration

- Time of response: SESC will respond the requested task and operations quickly.
- Processing time: Since the system had developed with efficient programming language and database upon request for user's Activities, the system under normal

condition will process the request as quickly as possible by using multi-tier architectures.

- Users of the system: the system will support multiple users at a time.

Error Handling and Validation

- Input error: the system handles many exceptions like inserting empty (null) value to the database, inserting a duplicated client id, inserting invalid data type. To handle the above errors, the proposed system will provide appropriate correction messages for each error.
- Login error: the system handles an attempt to login with an incorrect user account and password. By displaying an error message.
- Input Validation: our system uses both JavaScript and php to validate different forms like registration form, on this form if the users are asked to email account and insert it, our system check whether the email is valid or not.

Quality Issues

The quality of proposed system had better due to reduction of error. The system should give correct output for the users when they want to get services. The system had effective and consistent in performing its required function in short period of time without delaying much time. There are some issues that our system going to solve:

- Availability: the system is available for users to get service at any time anywhere if they got an internet access.
- Reliability: our system is reliable in terms of security, accessibility and also on data storage mechanism.

Backup and Recovery

When we build the system, we have a database on Wamp Server as a backup data. When we complete building the system the admin can host the system to make it accessible anywhere for every user and to do this the admin center must buy the storage from the web-hosting

company and store personal data and other useful information of the system. Therefore, if our system fails or some things happen to the system, we can get our data from this storage.

Physical Environment

The system will give access for wolkite city. And it have been accessible anywhere at any time where the internet access exists. There are no external factors that affect our system.

Resource Issues

There are some constraints on resources for the full deployment of system like for hosting.

Documentation

We provide documentation for our system to give information or to make peoples deeply know about the system functional and non-functional requirement and also advantages of the proposed system over the existing system. This is helpful for user to be attracted by the system. And also, documentation used for user to help as guidance.

5.2 Current System Architecture

Since it is manual process there is no current software architecture for house rental system.

5.3. Proposed System Architecture

The proposed system have three tier architecture these are client tier, middle tier and data tier. The three-tier architecture is used when an effective distributed client/server design is needed that provides increased performance, gives greater flexibility to development teams by allowing them to update a specific part of an application independently of other parts, maintainability, reusability and scalability The proposed system will have three tier architecture these are client tier, middle tier and user (data)

The client tier

The user can be browsing the system directly to interact with the system through web browser.

The middle tier (application tier)

The system contains web server. The web server handles all the HTTP requests coming from the client machines.it also manages the responses that are returned to the client machines.

The database tier

Manages textual data that is being stored in the database.

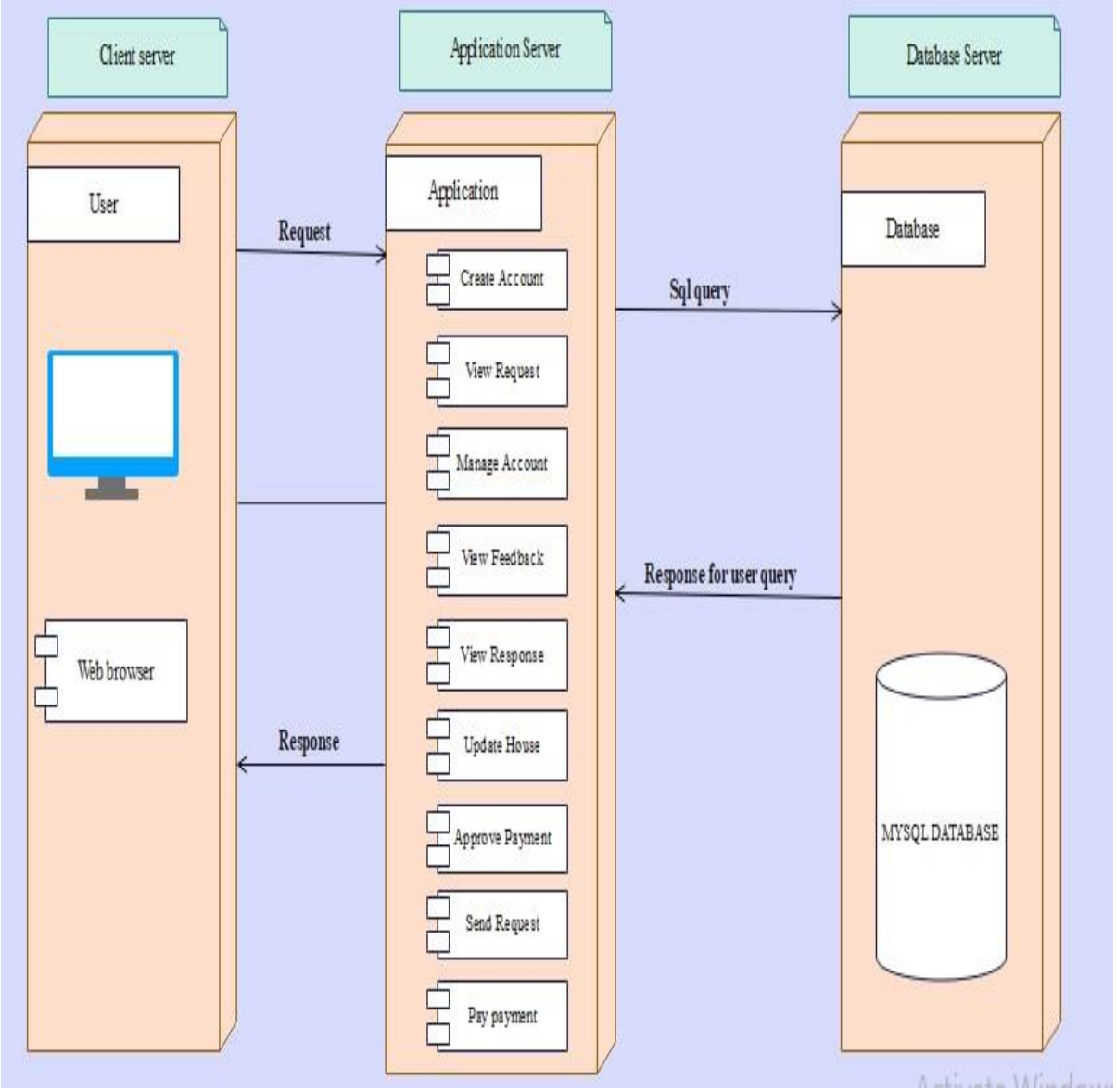


Figure 5. 1 Web server/Application servers

5.3.1 Subsystem Decomposition and Description

System decomposition is undertaken to reduce the complexity of the system and gaining insight into the identity of the constituent components. The system is decomposed in to sub-systems which are a collection of classes, associations, operations, events and constraints that are closely interrelated with each other. The proposed web based house rent system is decomposed in to smaller sub-system as shown in the following figure

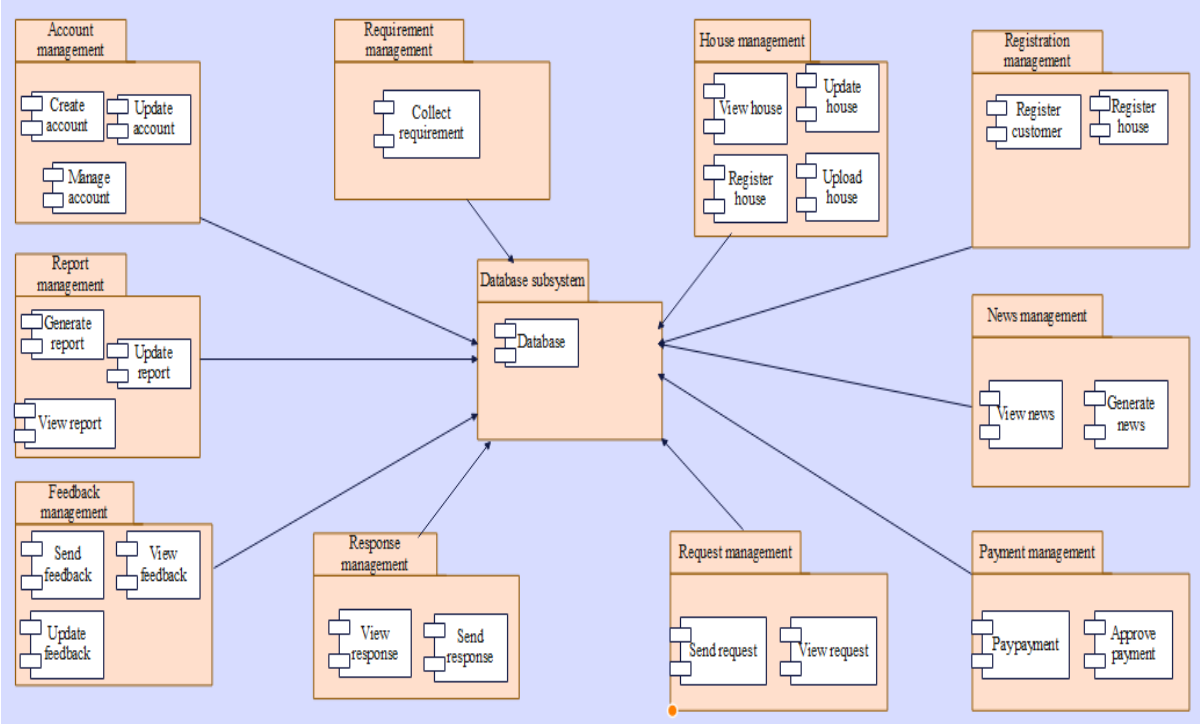


Figure 5. 2 Subsystem decomposition

5.3.2. Hardware/Software Mapping

Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for our system, the software installed on that hardware, and the middleware used to

connect the disparate machines to one another. Accordingly; online house rental system deployment diagram consist the system software installed on electronic devices with browsers acting as middleware to connect client side to the server side.

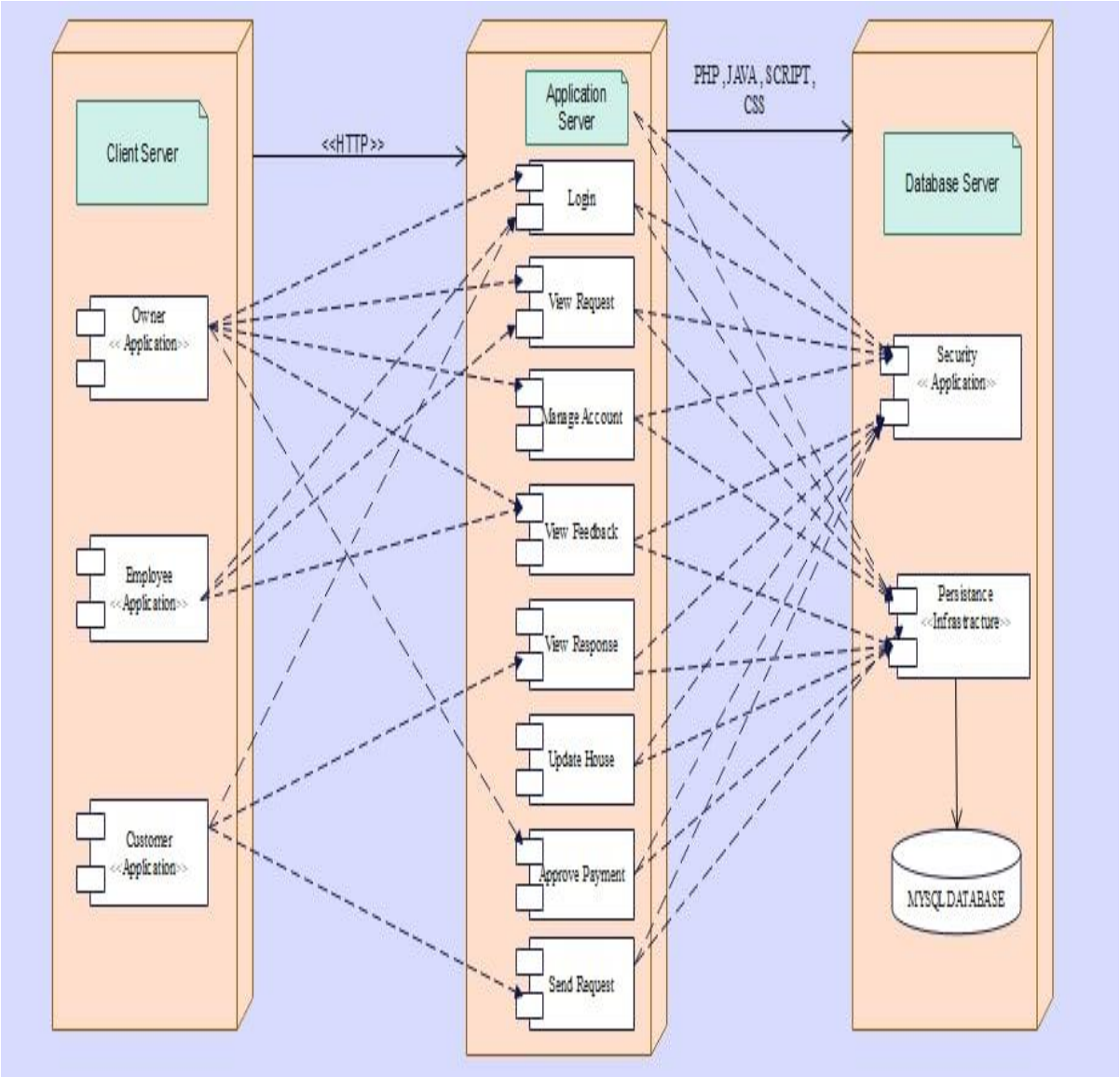


Figure 5. 3 UML Deployment diagram

5.3.3. Detailed Class Diagram

A class diagram is a type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.

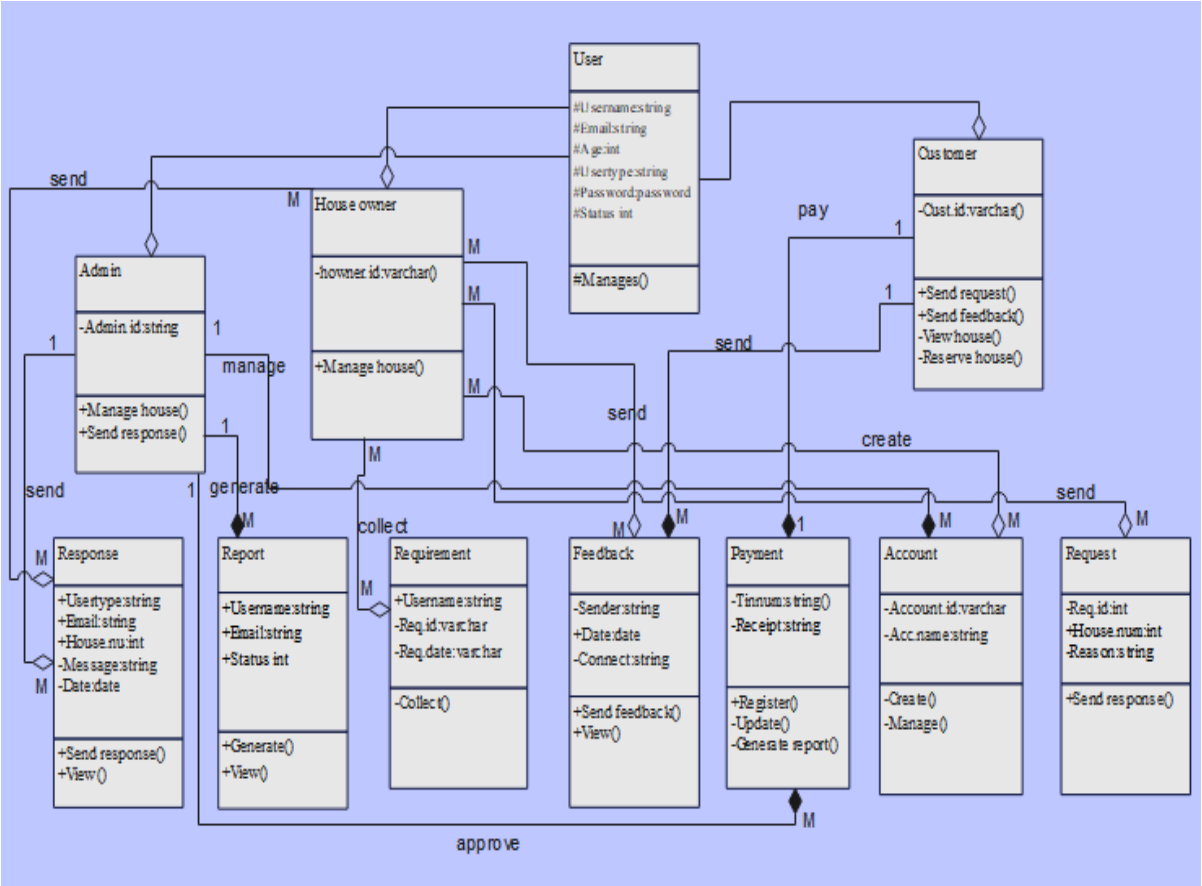


Figure 5. 4 Detailed Class Diagram

5.3.4. Persistent Data Management

Persistent data management is basically used to represent the design of the database. Database design is the process of producing a detailed data model of a database. Online house rental system have database tables to store, access and maintain information that flows in the system.

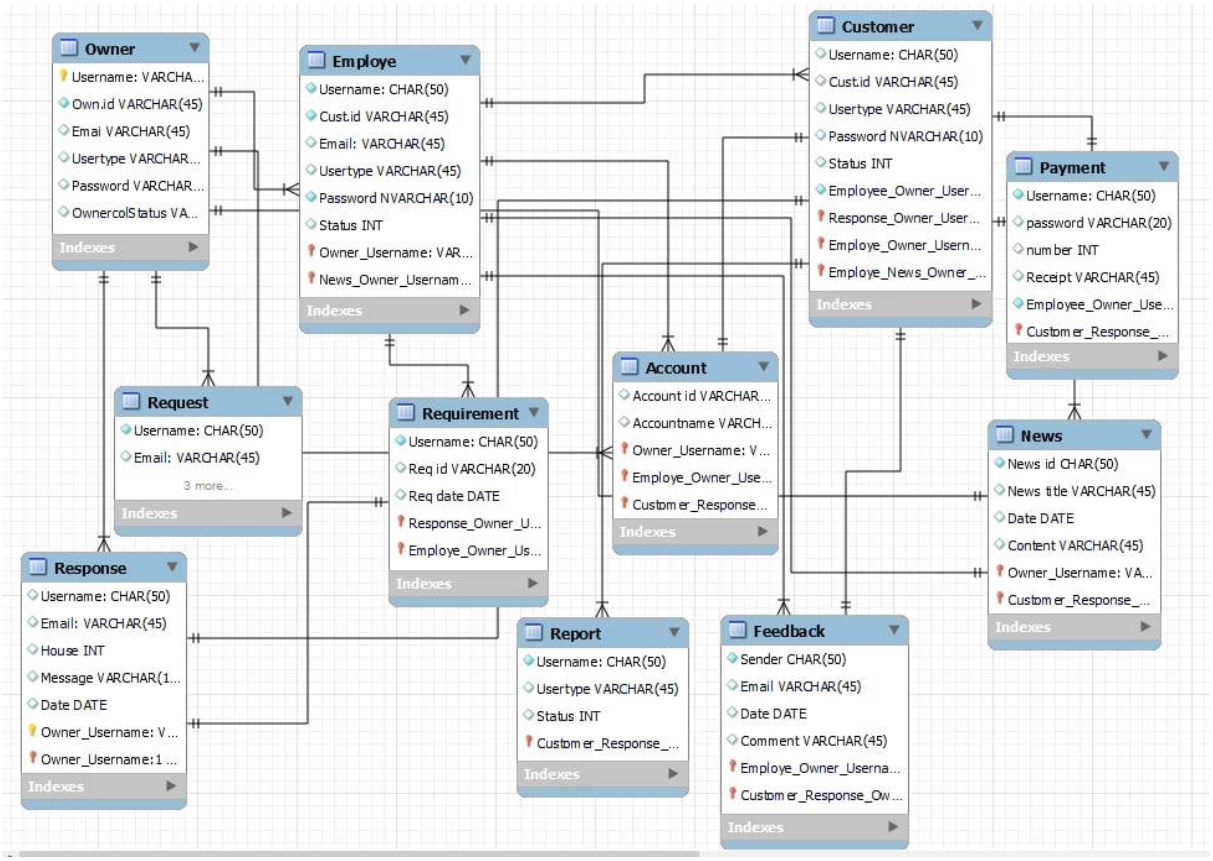


Figure 5. 5 Persistence diagram

5.3.5. Access Control and Security

Different actors have access to different functionality and data. Therefore these privileges prevent unauthorized users from accessing data's which they don't have granted to access. Authentication: This take place by letting users to insert their user name and password in the displayed login form. Authorization: This takes place by preventing users from participating in specific tasks on which he/she doesn't have grant to access.

Table 5. 1 Access Control and Security

Function	Customer	Admin	House Owner
Generate report		✓	
View report	✓	✓	✓
View customer request		✓	✓
Register house information		✓	✓
View house information	✓	✓	✓
Send withdrawal request	✓		
Send response		✓	✓
Update house information		✓	✓
Register customer		✓	
Reserve house		✓	✓
Manage account		✓	
upload house		✓	✓
Send feedback	✓		
View feedback	✓	✓	✓
Pay payment	✓		
Approve payment		✓	

5.4 Packages

Package diagram, a kind of structural diagram, shows the arrangement of organization of model elements in middle to large scale projects. Package diagram can show both structure and dependencies between subsystems or modules, showing different views of a system

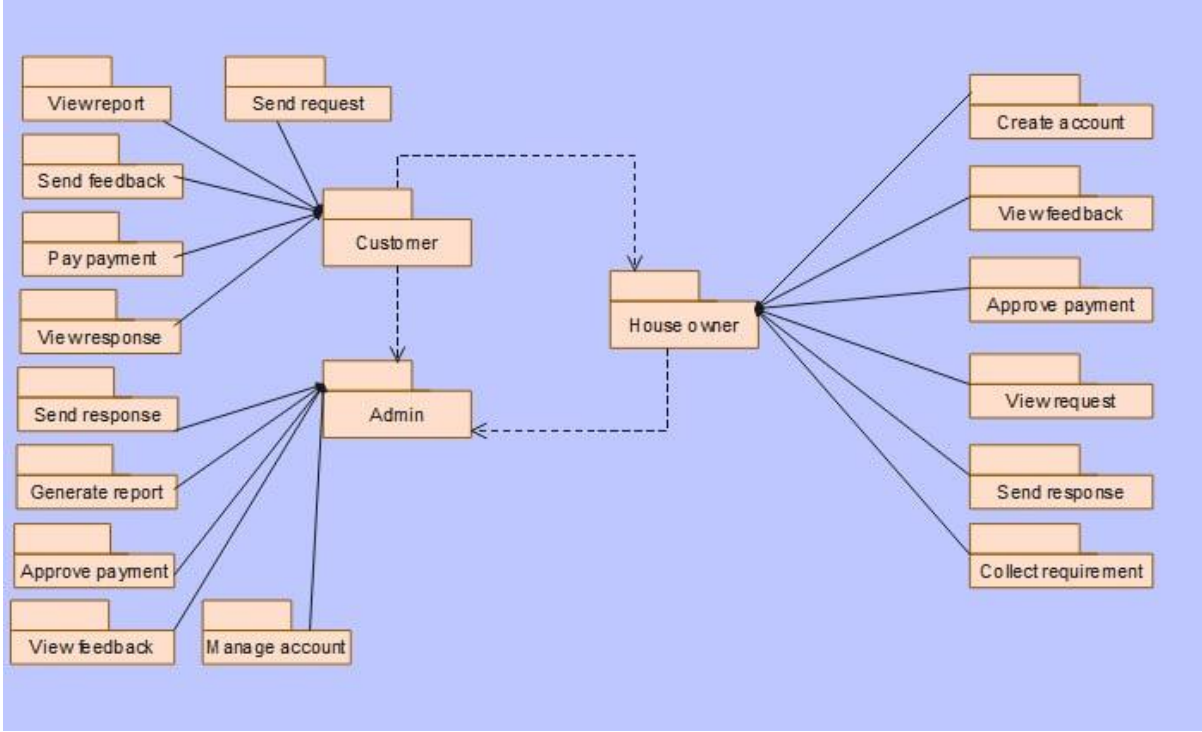


Figure 5. 6 **Packages**

5.5 Algorithm Design

- Algorithm for login:

Step 1: User open the browser.

Step 2: Enter email and password.

Step 3: system validate Email and Password.

Step 4: if it's valid

Step 5: home page is displayed.

Step 6: else.

Step 7: error message is displayed.

- Algorithms for registration:

Step 1: browse the home page.

Step 2: fill the registration form.

Step 3: if input information is valid.

Step 4: the user is added to database and home page is displayed.

Step 5: else invalid input

Step 6: display error message

5.6. User Interface Design

User interface is the front-end application view to which user interacts in order to use the system. The user interface is the main component that is used as a communication medium between the user and the system. We do have a user interface as described below.

5.6.1 Home Page

This page appears as the site in which the system is deployed is opened or accessed and contains some links which lead the user to other pages according to his/her privilege, and if the user is authorized user or has an account, he/she is directly go to the page that he want by entering correct username, password. Otherwise they will sign up to have an account and access the system.



Figure 5. 7 Home Page interface

5.6.2 Create Account owner Page

This page appears as user has navigated to the admin page it has a form that a user fills to create an account. User must fill username, password, Id and type of an account `weather it is an administrator or gust account to create his or her own account.

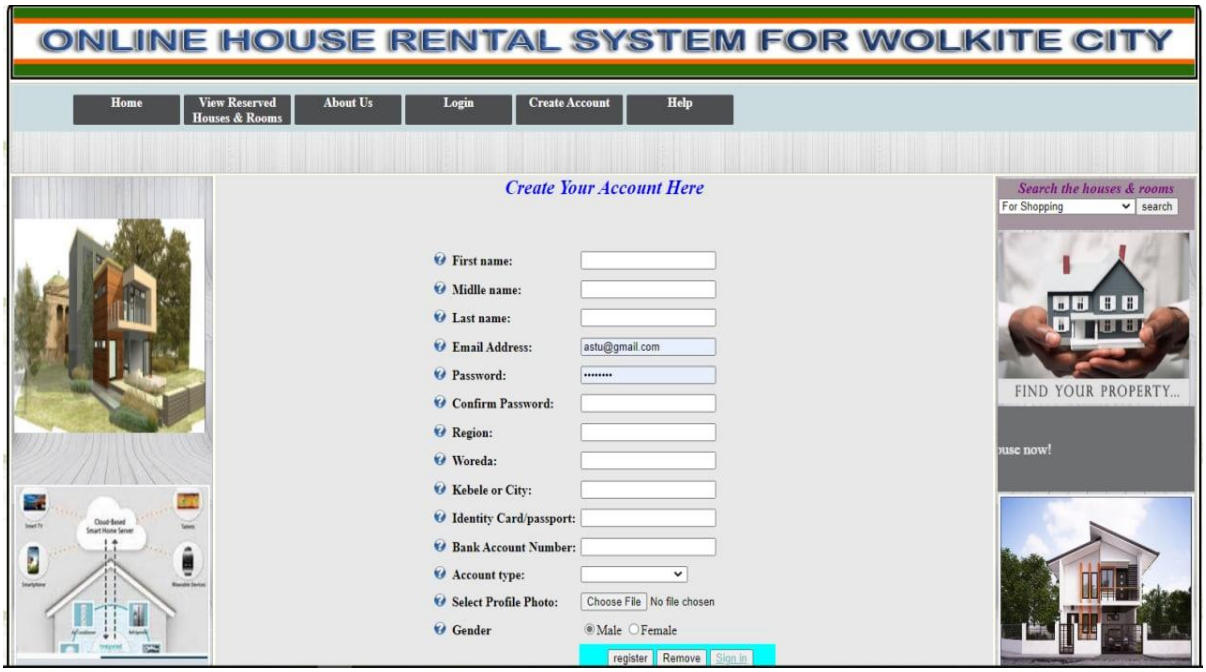


Figure 5. 8 Create Account owner page

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

6.1. Implementation of the Database

In order to implement the main application DBMS 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 online house rental system for wolkite city 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 email & Password
- Test Case 2: Check result on entering Invalid User email& Password
- Test Case 3: Check response when a User email 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.

1. Software testing tools

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

2. 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

In this testing part, all the modules combined to gather and tested for fitness with each other and with the system functionality. If error occurs in gather and tested for the module with problem can be identified and re combined.

6.6.6. Acceptance Testing

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

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATION

7.1 Conclusion

This project has given us vast knowledge on the different computing technologies. We have learned a lot during the documentation of this system project. As the scope of our project describes we developed web based house rental system for Wolkite City.

We were also able to learn a lot of system analysis and design of the project specially object oriented concepts. Security was the main issue in the development of this.

The system that we have developed has two phases; the first phase deals with the analysis phase of the life cycle with the organization, and the next phase addresses the design phase.

As the end of the first phase, we need to review what we have planned at the beginning. We began our work by identifying the problems in existing system and the significance of the new system for WCHRS.This concerned defining the system development methodology, identifying process and resource, and setting the deliverable and scheduled for the project.

The flow of analysis helps the team to understand the major functional areas and processes of the proposed system. Through this flow, we identified the weakness and strength of the existing system by developing the proposed system.

After that, we discovered system requirements. Through this phase, we identify functional and non-functional requirements of the new system. Then we have undertaken a major phase in system development process: object oriented Analysis. Here, we tried to model the new system we proposed using UML diagrams: Use case, sequence, and activity and class diagrams. Also, we designed the new system user interface prototype.

As a result, we proposed the new system to solve the problem of existing system such as time management and security. And also we developed our capacity during working this new system so that the manual system changed to computerized system.

7.2 Recommendation

We recommend that every customer and house or room owner use this system properly they can satisfy their need successfully with in time; in addition enhancing the language by supporting local languages like Amharic, Tigrigna, Oromigna and others to fulfill users' requirements. For the time being we did not do those functionalities due to lack of time.

REFERANCE

([1].activity diagram, Friday January 13, 2022 24:23:15 AM. [Online].Available: https://en.wikipedia.org/wiki/Activity_diagram/).

([2].class-diagram, Monday February 10, 2022 23:23:28 AM. [Online]. Available: https://www.smartdraw.com/class-diagram/).

([3]. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/).

([4]. Kawabata, R. Kasah, K. (2007). "Systems Analysis for Collaborative System by Use Case Diagram", Journal of Integrated Design & Process Science/).

([5]. MORGAN, Tony (2002). Business Rules and Information Systems: Aligning IT with Business Goals. Addison-Wesley. ISBN 0-201-74391-4/).

([6]. OMG (2011). OMG Unified Modeling Language (OMG UML), UML Sequence diagram, V2.4.1, p. 507/).

([7].Sequence diagram, Thursday February 03, 2022 2:47:33 AM. [Online]. Available: https://en.wikipedia.org/wiki/Sequence_diagram/).

APPENDICES

```
<?php
ob_start();
?>

<html>
<head><title>login page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width">
         <script src='java script/search.js'></script>
    <SCRIPT type="text/javascript">
    window.history.forward();
    function noBack() { window.history.forward(); }
</SCRIPT>
<link rel="stylesheet" type="text/css" href="css/myStyle.html" media='all'/>
<script>
function Validator(){

    // Make quick references to our fields
    var email = document.getElementById('email');
    var pas = document.getElementById('password');
    var user = document.getElementById('user');
    // Check each input in the order that it appears in the form!

          if(notEmpty(email, "Please enter your user Email")){
          if(emailValidator(email, "Please enter a valid email address")){
          if(notEmpty(pas, "Please enter your password")){
        if(notEmpty(user, "Please Select your user type")){

                    return true;
          }}}}
```

```
    return false;

}

function notEmpty(elem, helperMsg){
    if(elem.value.length == 0){
        alert(helperMsg);
        elem.focus(); // set the focus to this input
        return false;
    }
    return true;
}

function isNumeric(elem, helperMsg){
    var numericExpression = /^[0-9]+$/;
    if(elem.value.match(numericExpression)){
        return true;
    }else{
        alert(helperMsg);
        elem.focus();
        return false;
    }
}

function isAlphabet(elem, helperMsg){
    var alphaExp = /^[a-zA-Z]+$/;
    if(elem.value.match(alphaExp)){
        return true;
    }else{
        alert(helperMsg);
        elem.focus();
        return false;
    }
}
```

```
function isAlphanumeric(elem, helperMsg){
   var alphaExp = /^[0-9a-zA-Z]+$/;
   if(elem.value.match(alphaExp)){
      return true;
   }else{
      alert(helperMsg);
      elem.focus();
      return false;
   }
}

function lengthRestriction(elem, min, max){
   var uInput = elem.value;
   if(uInput.length >= min && uInput.length <= max){
      return true;
   }else{
      alert("Please enter between " +min+ " and " +max+ " characters ");
      elem.focus();
      return false;
   }
}

function madeSelection(elem, helperMsg){
   if(elem.value == "Select user type"){
      alert(helperMsg);
      elem.focus();
      return false;
   }else{
      return true;
   }
}

function emailValidator(elem, helperMsg){
   var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;
   if(elem.value.match(emailExp)){
```

```
    return true;
  }else{
    alert(helperMsg);
    elem.focus();
    return false;
  }
}</script>

</head>
<body background="images/backg.jpg" onload="noBack();"
onpageshow="if (event.persisted) noBack();" onunload="">
<table  width='1000' height='900'  bgcolor='#fffff' align='center' style='border:3px solid
#000000;width:1350px;height:710px;border-radius:15px;
box-shadow:1px 10px 10px white'>

<tr>
<td  height=20px  colspan=4 style='border:3px solid #000000;width:900px;height:10px;'>

<img src="images/RUFI.jpg" width="1550px" height="80px">
</td>
</td>

</tr>
<tr height=30px>
<td  colspan=3 background="images/back2.jpg"  height="auto"style='border:1px solid
#bbbbbb;width:600px;height:3px;'>
 <div id="navbar">
    <ul id="navmenu">
            <li id="current" ><a href="index.php">Home</a></li>

              <!--li><a href="housedatabase.php"><span> Available Houses and Rooms</span></a-->


                    </li>
```

```
                <li ><a href="reserved.php" target="_parent">View Reserved Houses & Rooms</a></li>
                <li><a href="aboutus.php" target="_parent">About Us</a>
                </li>
                <li><a href="Login.php" target="_parent">Login</a></li>
                <li><a href="cccc1.php" target="_parent">Create Account</a></li>
                <li><a href="help.php" target="_parent">Help</a>
                </li>

    </ul>

    </ul>
  </div>
</td>
</tr>

<td >
<table border=0 width="250px" height="250px" background="images/back.jpg" style='border:1px solid
#bbbbbb;height:100%;'>

<tr>
<td><img src="images/capture5.PNG"height="225px" width="250px"></td></tr>
<tr>
<td width='100px' height='250px'>

<table border=0  bgcolor="#ffffff">
<tr>
<td valign="top" bgcolor="#d8dce4">
<h2><center>
<font color="white"><img src="images/capture1.PNG" height="255px" width="250px"></center></h2>
<font color="black" face="arial" class="just">

</td>
</tr>
```

```
</table>

</td>  </table>




<td  valign="top" bgcolor="#e8e8e8" width="765px">
  <br>
   <form  method="post" action=" " onSubmit="return Validator()">
      <table   class="inputform" align="center" width="1020px" height="400px"cellpadding="2"
cellspacing="2">
 <center><font color="#0000CD" size="5"><b><i>Login Into System
Here</font></b></td></tr></center>
<tr><td><img src="images/tick.png" width="15px" height="15px">  <b>Your Email:
</td><td class="inputform">  <input type="text" name="email" id="email" ></td></tr>
<tr><td><img src="images/tick.png" width="15px" height="15px">  <b>Password:
</td><td class="inputform">  <input type="password" name="password" id="password" ></td></tr>
<tr>
                <td><img src="images/tick.png" width="15px" height="15px">  <b>Login
Us</td>
                <td>
                <select name="user_type" id="user" >
                <option></option>
                   <option value="House Owner">House Owner</option>
                   <option value="Tenant">Tenant</option>
                   <option value="admin">Adminstrator</option>
                </select>
                </td>
             </tr>
<tr>

    <br><br><br>
    <td><td><input type="submit" name="login" value="Login" />  <input type="reset"
value="Clear" /></td>
```

```
 </td></tr>
 <tr/><tr/><tr/><tr/><tr/><tr/><tr/>
      <tr><td colspan=2><center><a href="Forgot3.php" ><img src="images/forgot-password-button-png-
hi.png"width=170" height="30"/><img src="images/Help.png"width=50" height="20"/></a></tr>
          <tr><td colspan=2 rowspan=5><center><font color="#0000CD" size="4">NO Account
  <a href="cccc1.php"><u><font color="red" size="4"><b>Sign Up
?</font></a></td></center>

</center></tr>




 <?php session_start(); ?>
<?php

require("configration.php");
if (isset($_POST['login'])) {

   $un = $_POST['email'];
   $password =$_POST['password'];
   $pass = md5($password);
   $user = $_POST['user_type'];
$con = mysqli_connect('localhost','root', '','house');
 $sql= mysqli_query($con,"SELECT * FROM account WHERE email = '$un' AND password='$pass'
AND usertype='$user'" )or die(mysql_error());
 $result = mysqli_fetch_array($sql);
  if ($result) {
   if($result['account_status']=='1'){
      echo"<p  class='wrong'>You couldn't Sign in b/c your account is inactive</p>";
       echo'<meta content="5;" http-equiv="refresh" />';
   }
   elseif($result['account_status']=='0') {
    $_SESSION['log']['email'] = $result['email'];
      $_SESSION['start'] = time();
```

```
        $_SESSION['expire'] = $_SESSION['start'] + (2 * 60);
            header('Location:login.php');
        if ($user == "Tenant") {
            header("location:customer.php");
        } elseif ($user == "House Owner") {
            header("location:howner.php");
        } elseif ($user == "admin") {
        header("location:admin.php");}

    }
    }
    else{
        ?>
        <script>
            alert('You enter wrong username or password ,please the correct to login again');
            window.location='Login.php';
            </script>
            <?php
    }
}
//mysqli_close($con);
?>
</table></center>

</form>


  </div>

 </td>

<td  width="100%">
<table border=0 width="250px" height="250px" background="images/back.jpg"
align="right"style='border:1px solid #bbbbbb;height:100%;'>
<tr  height="15px" bgcolor="#a4959f">
<td ><form name="fs" action="search.php" method="post" ">
```

```
<center><b><font size="4" color="#800080" face="batang"><i>Search the houses &
rooms</font></b></center>
<select name="houses">
        <option disabled="disabled">Houses & Rooms</option>
        <option>For Shopping</option>
        <option>For Office</option>
        <option>For small business center</option>
        <option>For Banking</option>
        <option>For Residence</option>
        <option>For Hotel Reservation</option>
        <option>For Pharmacy</option>
        <option>For Clinc</option>
        <option>For Hell</option>
        <option>For Depositary</option>
        <option>For Boarding School</option>
        <option>For cafe and Restaurant</option>
        <option>For Other Purposes</option>


</select></th><input type="submit" name="search" value="search" ><br>   </div></tr></form></td>
<tr>
<td><img src="images/ss1.jpg"height="190px" width="250px"></td></tr>

<tr>
<td bgcolor="#6e6f71" height="100px">

<marquee direction="left"><font face="Lucida Calligraphy" color="white" size="4">
<center>Rent the house now!</center></font></marquee></td></tr>
<td width='100px' >

<table border=0  bgcolor="#ffffff">
<tr>
<td valign="top" bgcolor="#d8dce4">
<h2><center>
<font color="white"><img src="images/capture3.PNG" height="190px" width="250px"></center></h2>
```

```
<font color="black" face="arial" class="just">

</td>
</tr>



 </table>

   </table>




</td>

<tr>
<center><td colspan=3 bgcolor="#80584c">
<?php
include('footer.php');
ob_end_flush();
?>
</td></center>
</tr>
</tr>
</table>
</body>
</html>
```