# **ACKNOWLEDGMENT**

First of all we would like to thanks our God helping us in different ways either the way we know or we didn’t know to complete our task. Secondly we wish to express our heart full thanks to our advisor Instructor Shimalis K. (MSc) who is playing a great role by giving a powerful comments and different advices that helps for our project. We appreciate his inspiring guidance and constant encouragement that will help us for completion of our project. Last but not least we would thanks for 4th year Computer science students for sharing useful ideas.

Contents page

[ACKNOWLEDGMET I](#_Toc484593198)

[LIST OF TABLES VI](#_Toc484593199)

[LIST OF FIGURES VII](#_Toc484593200)

[ABSTRACT IX](#_Toc484593201)

[CHAPTER ONE](#_Toc484593202)

[Introduction 1](#_Toc484593203)

[1.1 Introduction 1](#_Toc484593204)

[1.2 Background of the Organization 1](#_Toc484593205)

[*1.2.1* *Vision* 2](#_Toc484593206)

[*1.2.2* *Mission* 2](#_Toc484593207)

[1.3 Background of the Project 2](#_Toc484593208)

[1.4 Statement of the Problem 2](#_Toc484593209)

[1.5 Team composition 3](#_Toc484593210)

[1.6 Objectives of the project 3](#_Toc484593211)

[*1.6.1* *General objective* 3](#_Toc484593212)

[*1.6.2* *Specific objectives* 3](#_Toc484593213)

[1.7 Scope and limitation of the project 4](#_Toc484593214)

[*1.7.1* *Scope of the project* 4](#_Toc484593215)

[*1.7.2* *Limitation of the project* 4](#_Toc484593216)

[1.8 Significance of the project 4](#_Toc484593217)

[1.9 System development Methodology 5](#_Toc484593218)

[*1.9.1 Data gathering techniques* 5](#_Toc484593219)

[1.10 System analysis and design methodology 6](#_Toc484593220)

[1.11 System development Tool 7](#_Toc484593221)

[*1.11.1* *Hardware tools required:* 7](#_Toc484593222)

[1.11.2 *Software tools required*: 7](#_Toc484593223)

[1.12 Beneficiary 7](#_Toc484593224)

[1.13 Feasibility study 8](#_Toc484593225)

[1.13.1 *Technical Feasibility* 8](#_Toc484593226)

[1.13.2 *Operational Feasibility* 8](#_Toc484593227)

[1.13.3 *Economic feasibility* 8](#_Toc484593228)

[1.14 Budget break down 9](#_Toc484593229)

[*1.14.1* *Tangible cost for hardware* 9](#_Toc484593230)

[*1.14.2* *Intangible cost* 9](#_Toc484593231)

[1.15 Work break down structure 10](#_Toc484593232)

[1.16 Project time schedule 10](#_Toc484593233)

[CHAPTER TWO](#_Toc484593234)

[Description of the Existing System 12](#_Toc484593235)

[2.1 Introduction 12](#_Toc484593236)

[2.2 Existing System 12](#_Toc484593237)

[2.3 Players in the Existing System 12](#_Toc484593238)

[2.4 Major Functions/Activities in the Existing System 13](#_Toc484593239)

[2.5 Business Rules 13](#_Toc484593240)

[2.6 Report Generated in the Existing System 14](#_Toc484593241)

[2.7 Forms and other documents of the Existing System 14](#_Toc484593242)

[2.8 Bottlenecks of the Existing System 16](#_Toc484593243)

[*2.8.1 Performance (Response time)* 16](#_Toc484593244)

[*2.8.2 Have Problem in data Storage* 17](#_Toc484593245)

[*2.8.3 Security and control* 17](#_Toc484593246)

[*2.8.4 Efficiency* 17](#_Toc484593247)

[2.8.5 Input and output bottlenecks 17](#_Toc484593248)

[*2.8.6 More costly* 17](#_Toc484593249)

[*2.8.7 Working time wastage*: 17](#_Toc484593250)

[2.9 Practices to be preserved 18](#_Toc484593251)

[2.10 Proposed Solution for the new System 18](#_Toc484593252)

[2.10 Requirements of the Proposed System 19](#_Toc484593253)

[*2.10.2 Functional requirements* 19](#_Toc484593254)

[*2.10.2 Nonfunctional requirement* 20](#_Toc484593255)

[CHAPTER THREE](#_Toc484593256)

[Process Modeling 22](#_Toc484593257)

[3.1 Introduction 22](#_Toc484593258)

[3.2 System Requirement Specifications (SRS) 22](#_Toc484593259)

[*3.2.1 Use case Model* 22](#_Toc484593260)

[*3.2.2* Use case documentation 24](#_Toc484593261)

[*3.2.3 Sequence diagram* 36](#_Toc484593262)

[*3.2.4 Activity Diagrams* 50](#_Toc484593263)

[*3.2.5 Analysis Class Diagram* 53](#_Toc484593264)

[*3.2.6 User Interface Prototype* 55](#_Toc484593265)

[*3.2.7 Supplementary Specifications* 56](#_Toc484593266)

[CHAPTER FOUR SYSTEM DESIGN](#_Toc484593267)

[4.1 Introduction 57](#_Toc484593268)

[4.2 Design goal 57](#_Toc484593269)

[4.3 System Design Architecture 57](#_Toc484593270)

[4.4 Class Type Architecture 59](#_Toc484593271)

[4.5 Class Modeling 60](#_Toc484593272)

[4.6 State Chart Modeling 60](#_Toc484593273)

[4.7 Collaboration Diagrams 62](#_Toc484593274)

[4.8 Component Diagram 63](#_Toc484593275)

[4.9 Deployment Diagram 64](#_Toc484593276)

[4.10 Persistence Modeling 65](#_Toc484593277)

[*4.10.1 Conceptual Database Design* 66](#_Toc484593278)

[*4.10.2 Logical Database Design* 66](#_Toc484593279)

[*4.10.3 Physical Database Design* 68](#_Toc484593280)

[4.11 User Interface Design 69](#_Toc484593281)

[CHAPTER FIVE](#_Toc484593282)

[Implementation and Testing 72](#_Toc484593283)

[5.1 Tools 72](#_Toc484593284)

[5.4.1 System overview 83](#_Toc484593285)

[5.4.2 System Summary 84](#_Toc484593286)

[5.4.2.1 System Configuration 84](#_Toc484593287)

[5.4.2.2 User Access Levels 84](#_Toc484593289)

[5.4.2.2 Getting started 84](#_Toc484593290)

[5.4.3 Using Client Application 84](#_Toc484593291)

[5.4.3.1 Settings 84](#_Toc484593292)

[*5.4.3.2* *Server* 85](#_Toc484593293)

[5.4.4 Users (house or room owners and tenants) Information 85](#_Toc484593294)

[5.4.5 Administrator information (Activities) 85](#_Toc484593295)

[CHAPTER SIX](#_Toc484593296)

[Conclusion and Recommendation 86](#_Toc484593297)

[6.1 Conclusion 86](#_Toc484593298)

[6.2 Recommendation 86](#_Toc484593299)

[References 87](#_Toc484593300)

# 

# LIST OF TABLES

[Table 1: Project team compositions 3](#_Toc472864719)

[Table 2: Cost estimation for the project 9](#_Toc472864720)

[Table 3: Task allocation for group members 10](#_Toc472864721)

[Table 4: Create Account table 25](#_Toc472864722)

[Table 5: Table of login 26](#_Toc472864723)

[Table 6: Use case Description for Change Password 26](#_Toc472864724)

[Table 7: use case description for search information 27](#_Toc472864725)

[Table 8: use case description for Send request 28](#_Toc472864726)

[Table 9: use case description for view response 28](#_Toc472864727)

[Table 10: use case description for Upload house or room information 29](#_Toc472864728)

[Table 11: use case description for Delete house 30](#_Toc472864729)

[Table 12: use case description for View Tenants request 31](#_Toc472864730)

[Table 13: use case description for View house information 32](#_Toc472864731)

[Table 14: use case description for Send Response 33](#_Toc472864732)

[Table 15: use case description for reserve the house 34](#_Toc472864733)

[Table 16: use case description for Update house information 35](#_Toc472864734)

[Table 17: use case description for send feedback 35](#_Toc472864735)

[Table 18: use case description for view feedback 36](#_Toc472864736)

[Table 19: use case description for Logout 36](#_Toc472864737)

# LIST OF FIGURES

[Figure 1: Time Schedule for the project 11](#_Toc484589450)

[Figure 2: agreement form used in the existing system 15](#_Toc484589451)

[Figure 3: agreement form used shop renting in the existing system 16](#_Toc484589452)

[Figure 5: Use case diagrams for online house rent system 24](#_Toc484589453)

[Figure 6: Sequence Diagram for Create account 37](#_Toc484589454)

[Figure 7: Sequence Diagram for Login 38](#_Toc484589455)

[Figure 8: Sequence Diagram for Change Password 39](#_Toc484589456)

[Figure 9: Sequence Diagram for Send Request 40](#_Toc484589457)

[Figure 10: Sequence Diagram for View Request 41](#_Toc484589458)

[Figure 11: Sequence Diagram for Send Response 42](#_Toc484589459)

[Figure 12: Sequence Diagram for View Response 42](#_Toc484589460)

[Figure 13: Sequence Diagram for Reserve House 43](#_Toc484589461)

[Figure 14: Sequence Diagram for Upload House or Room 44](#_Toc484589462)

[Figure 15: Sequence Diagram for View House Information 45](#_Toc484589463)

[Figure 16: Sequence Diagram for Delete House Information 46](#_Toc484589464)

[Figure 17: Sequence Diagram for Update House Information 47](#_Toc484589465)

[Figure 18: Sequence Diagram for Send Feedback 48](#_Toc484589466)

[Figure 19: Sequence Diagram for Search Information 49](#_Toc484589467)

[Figure 20: Sequence Diagram for Logout 50](#_Toc484589468)

[Figure 21: Activity Diagram for Login 51](#_Toc484589469)

[Figure 22: Activity Diagram for Create Account 51](#_Toc484589470)

[Figure 23: Activity Diagram for Change Password 52](#_Toc484589471)

[Figure 24: Activity Diagram for Upload House or room 52](#_Toc484589472)

[Figure 25: Activity Diagram for Reserve House 53](#_Toc484589473)

[Figure 26: Analysis Level Class Diagram for Online House Rental system 54](#_Toc484589474)

[Figure 27: User Interface Prototype for Online House rental system 55](#_Toc484589475)

[Figure 28: System Architecture for Online House Rental System 58](#_Toc484589476)

[Figure 29: Class Type Architecture 59](#_Toc484589477)

[Figure 30: Class modeling for house rental system 60](#_Toc484589478)

[Figure 31: State Chart Diagram for Login 61](#_Toc484589479)

[Figure 32: state chart diagrams for reserve the house 61](#_Toc484589480)

[Figure 33: State chart diagrams for Upload house or room 62](#_Toc484589481)

[Figure 34: Collaboration Diagram for Login 62](#_Toc484589482)

[Figure 35: collaboration Diagram for create account 63](#_Toc484589483)

[Figure 36: Component Diagram for online house rental system 64](#_Toc484589484)

[Figure 37: Deployment Diagram for house rental system 65](#_Toc484589485)

[Figure 38: ER diagram for online house rental system 66](#_Toc484589486)

[Figure 39: Physical Database Design for Online House Rental System 69](#_Toc484589487)

[Figure 40: User interface Design for home Page 70](#_Toc484589488)

[Figure 41: User interface Design for login 70](#_Toc484589489)

# Abbreviations and Acronyms

**CPU**: - Central Processing Unit

**CSS**:-cascading style sheet

**DB**:-database

**HTML**:-Hypertext Markup Language

**HTTP:-**hypertext transfer protocol

**LAN**:-Local Area Network

**PHP**:-Hypertext Preprocessor

**UML**:-Unified Modeling Language

**OOA**: - Object Oriented Analysis

**OOD**: Object Oriented Design

**OHRS**: online house rental system

**PC:** - Personal Computer

**SRS**: - System Requirement Specification

**UI:** - User Interface

# 

# ABSTRACT

*The project that we are going to develop is web based house rental system for Adama 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 Adama 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.*

# **Chapter One**

# **Introduction**

## Introduction

Renting a house is popular activity through walks of life. Typically students and employees that are moving temporarily will share house renting activity.

Basically we are initiated to develop this system because all most all of Adama city employees get suffer about house renting and forced to leave with their relatives until they find house that was really difficult to do practically that’s why we are developing this system today.

The main purpose of this project is to develop and implement a web based application entitled “online house renting system” to help renters and house owners to get and provide reliable services. These systems tend us to find possible solution for the problems existed in manual house renting activity and helps user to access the system easily.

## Background of the Organization

Adama city is one the towns located in which is 67 kilometers far from Addis Ababa. In this city the population growth is increasing from year to year due to different organizations are opened like HarambeUniversity, this university and other company’s needs employees and teachers to work in their company and these employees need shelter to serve their life. Because of this renting house is becoming common activity in the city. There are many city house owners tryingto satisfy the needs of their tenants. They haven’t any advanced system to fully supply the services to the house renters. This is due to tenants not knowing the exact location of the houses to be rented. Following this problem house renters and owners are getting suffered from house renting. They perform all the house renting activities manually with paper based agreements. So we have started to develop new web based application of house renting relative with current problem of the city for house owners and renters (Tenants) to get fast and reliable house renting service.

### *Vision*

To become open up and world class complete house renting service provider and earn the respect of all owners ,customers ,employees and community at large.

### *Mission*

House owners’ mission is to provide the most complete, reliable, and reasonably priced service with the aid of technology and well qualified professionals and high sense of commitment to satisfy their customers.

## Background of the Project

Currently Adama city house ownersprovides house renting services to theirtenants by building houses then advertises the houses to be rentedby city brokers or posting on the door some piece information to inform that the house is available for renting.Then any interested tenant submits his/her request and can get the home depend on the agreement. This activity is done manually so it is labor intensive work, because it needs going to the house owners’ home for tenants to sign the agreement. The house owners try to provide reliable services to their tenants, but since it is manual work it is tedious process. 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, identifying problems appearing in the existing system and propose solutions for the problems identified in the existing system.

## Statement of the Problem

Finding houses to rent in Adama city is challenging users cannot get house to rent easily because in Adama there is no any tool to search the houses; therefore it is difficult to find a good house in short time and knowing the exact location of these houses.

Adama city has different problems in house renting because it is performed by brokers of the city still now. The main serious problems are the following:-

* Man labor wastage:-tenants lose their power by finding house on foot from place to place, even they cannot get house, because most of houses may be reserved and they cannot know where they can get free house exactly.
* Wastage of time:-tenants lose their time by searching houses in different place; this is difficult for new employees or civil servants because they may miss their work time.
* Wastage of money:-tenants waste their money for broker’s payment, taxi and may be for some expenses related .Also house owners and house owners pay for brokers to advertise their house.
* Broker’s false information:-brokers give uncertain information for renter:-when the renters will find the house with brokers since the renter did not get the house as he/she expect.
* Redundancy: - there is redundancy in renting new houses 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 between renters.
* There may be loss of data and redundancy may occur.
* There is no proper data store so this cause danger in data security.

## Team composition

This project team is composed of two males and three females. The following table shows the background information about the project team.

Table 1: Project team compositions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Sex | Email Address | Phone no. |
| 1 | WalelignSisay | M | Walelignsisay19@gmail.com | 0919015091 |
| 2 | Jabir Abdulbari | M |  |  |
| 3 | Basufikad H/miskel | M |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

## Objectives of the project

### *General objectiv**e*

The general objective of our project is to develop web based application house renting system for Adama city.

### *Specific objectives*

* Analyze the details of the existing and identify problems found in the existing system.
* Design a solution for problems identified.
* Identify and design the requirements of the new system.
* Design user friendly interface for the new system.
* Design a database for the new system for efficient data retrieval.
* Generally to avoid the aforementioned problems in house renting process and provide more efficient house renting system for Adama city.

## Scope and limitation of the project

### *Scope of the project*

The scope of this project is to develop web based online house renting system for Adama city. The project willaddress the following activities**:-**

* The system allows house owners to register and post their house.
* The system allows administrator to manage house owners’resource.
* The system allows house owner to update their own resource.
* The system allows tenants to send requests to house owners.
* The system allows house ownersto send response to tenants.
* Message notification for house owners when the tenant reserves the house.
* Making the system more secured by creating login account for all tenant, house ownersand administrator.
* The system providing search tool fortenants, house ownersand administrator in easy and efficient way.
* The system allows administrator and tenants to view house owners’resources.
* The system allows house ownersto view tenants’ requests.
* Some other features may be added according to their necessity.

### *Limitation of the project*

* Our system has also limitations on a place that has not internet accessibility because it is depend on connection since it is web based.
* 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.
* Our system do not support agreement canceling before due date.

## 1.8 Significance of the project

After implementing this project the system will have the following benefits.

* Tenants will reserve house by selecting the place, type of house and the cost they can pay by viewing all recorded house detail online.
* Houses will rent on time with exact payment to the house owner.
* Unrented houses can become reachable to tenants with in time.
* Misinformation provide by brokers will be avoided because there is no need of intermediate person (broker) in our system.
* The system is more secured than existing one so users can simply believe it.
* Saves time ofTenants and house owners, tenants will not lose their time to finding a house and the house ownerswill not lose their time and money to advertise their own house property by broker.
* Save money of tenants as well as the house owners and house owners, they will not give payment to broker during agreement of their needs, and also there is no loss any money for transportation.
* Save energy of tenants, they will not lose their labor to search houses on foot.
* Tenants can reserve house online even they are not in Adama city; they can reserve before coming and they can directly go to the house after they come.

## System development Methodology

### *1.9.1 Data gathering techniques*

* Document review: **-** weread over different written documents, books and looking over internet for additional information.
* 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 are 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,tenants as well as house owners and house owners working procedures.

* Interview:-It is fact finding technique we used to collect information from individual through face to face interaction. It gives us 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 have collected raw data or documents which are useful for the project to be implemented.

## System analysis and design methodology

The team planned to use the object oriented design methodology for the development of the system among the different methodologies. Because it is better way to construct, manage and assemble objects that are implemented in our system. Object oriented design methodology has two phases:-

* Object Oriented Analysis (OOA):This phase enabled us to look at the problem domain and with the aim of producing a conceptual model of the information that exists in the area which we are analyzing. Also it helped us to model the functions of the system (use case modeling), identifying the business objects, organize the objects and also the relationship between them and finally model the behavior of the objects.
* Object Oriented Design (OOD):This phase supported us to model interactions and behaviors that support the use case scenario, and finally update object model to reflect the implementation environment. And also transforms the conceptual model produced in object-oriented analysis to take account of the constraints needed to our system, so that we used this phase to improve the use case model to reflect the implementation environment.

We decide to use object oriented system analysis and design methodology because of the following reason:

* Object-oriented techniques work well in situations where complicated systems are undergoing contentious maintenance, adaptation and design
* **Simplicity:** software objects model real world objects, so the complexity is reduced and the program structure is very clear.
* Reusability: the object oriented provides opportunities for reuse through the concepts of inheritance.
* **Modifiability:-**when we need to add new feature to the system we only need to make changes in one part of the applicable class.

## System development Tool

### *Hardware tools required:*

* Personal computer (PC):almost all tasks of our project are performed on computer.
* Flash disk (8GB): required for data movement to store & transfer data from one PC to another PC.
* Modem and LAN of 512 mbps broad band connection.
* Disks (CD, DVD):necessary for the movement of relevant data and for backup and recovery mechanism.
* Network cable:since our system is web based, it is very necessary requirement. It is also help us to extract relevant information about our project from internet.
* Server: to store the data.
* Stationeries (pen, paper): for writing all necessary documentations associated with the project.
* Note book:to take notes during data collection and for other document.

### *Software tools required*:

* Windows Operating system: will be used for the system since it is readily available in laboratories.
* Browsers: -since our system is web based, it is very necessary requirement.
* PHP and HTML: **-**To design the graphical user interface and the whole application.
* Java scripting- client side scripting (front end).
* MYSQL server :-for designing the database
* Microsoft office Word:-for documenting the corresponding deliverables associated with the project
* Enterprise architecture and Edraw max 6.8: **-**for designing Unified Modeling Language (UML) diagrams.
* Notepad++:- For writing a code or program of the system.
* Wamp Server: **-**to test the system by running.

## Beneficiary

The target beneficiary groups are:-

* 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.

* Tenants**:-**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
* Developers:-We are also beneficiary from the system because we are going to know detail information about the existence system and also we are going to gain knowledge of how to develop new system.

## Feasibility study

It is the measure of, evaluates and analysis of the project’s potential for success. It is based on extensive investigation and research to support the process of decision making.

* + 1. *Technical Feasibility*:-

Technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. For instance we evaluate hard ware and software resources to meet our needs for project in a good manner, thus our system is Technical feasible.

It is the process of evaluating the organization ability to construct a proposed system. Our project is technically feasible. Because it can generate outputs in a given time, response time is minimum, easy to communicate and generally it satisfies the end-user’s requirement.

* + 1. *Operational Feasibility***:-**

Operational feasibility is a measure of how well a proposed system solves the problem based on included the requirement of user, and then it can handle all problems in appropriate manner. Hence our project operates all problem specified by user because the desired operational outcome much during the design and development .since our project is operational feasible.

* + 1. *Economic feasibility*: -

Economic feasibility is used to determine the positive economic benefits to the organization will provide by the proposed system. Some benefits provide the system is:-

* Fastest processing time and small amount of processing error.
* 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

Getting positive result informs the economic feasibility. Therefore, the system is economically feasible.

## Budget break down

Even though it is difficult to provide an accurate cost estimates, the following is a rough estimate of the costs associated with the project.

It is the estimated cost takes to develop the project throughout the year

### *Tangible cost for hardware*

Tangible costs are costs that are used in the development of the project.

This includes:-

* Software development cost
* Hardware development cost
* Other related costs.

Table 2: Cost estimation for the project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Item type | Quantity | Unit price(in birr) | Total price(in birr) |
| Hardware’s | Personal computer | 2 | 14000, | 28000 |
| CD | 4 | 50 | 200 |
| A4 paper | 1dose | 200 | 200 |
| Flash Disk | 3 | 380 | 1140 |
| Printer | 1 | 5000 | 5000 |
| Software | Apache server,  Microsoft office word, MYSQL server | - | For free | 0 |
| Other costs | Any cost we may spend through project time line | - | - | 1000 |
| Total Cost |  |  |  | 35540 |

### *Intangible cost*

Intangible costs are:-

* Knowledge of the member student
* Skill of the member student.
* Time and energy of the member student.

## Work break down structure

Each part of tasks of the project is delivered by collaborating with each of the group members. Each group members has equal responsibility to complete the project successfully.

Table 3: Task allocation for group members

|  |  |
| --- | --- |
| Task | Group Enrollment |
| Title Selection | All group members |
| Requirement Gathering | All group members |
| Project Proposal | All group members |
| Analysis Phase | All group members |
| Design Phase | All group members |
| Implementation | All group members |
| Testing | All group members |
| Conclusion and Recommendation | All group members |

## Project time schedule

Our project contains list of activities, number of days used to accomplish a task with its start and end date, deliverables of each phase. The figure below describes time schedule of or project is the following.

|  |
| --- |
|  |

Figure 1: Time Schedule for the project

# **Chapter Two**

# **Description of the Existing System**

## 2.1Introduction

Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult for the proper understanding. This chapter formally defines the detailed functional user requirements using high-level requirements identified in the initiation, System Concept, and Planning phases. It also defines the requirements in terms of data, system performance, security, scalability and maintainability requirements for this project. The requirements are defined in this part to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the initiation Phase. This chapter in general focuses on overview of the existing system, overview of the proposed system, functional and non-functional requirements of the proposedsystem and proposed solution for the new system to address problems of the existing system.

## 2.2 Existing System

The main purpose of studying the existing system is to develop a new system which efficiently performs activities better than current one and for understanding existing problems.

Adama city house owners give house renting service to their customer’s (tenants). But, they haven’t an organized system to perform this activity, registering the new tenants and rental process is done manually, that leads the this system inefficient and time consuming, detailed information about the house is invisible to tenantss. The new tenantshares house renting by physicalypresenting to the house owner home. Then register and document their agreement on the paper.Tenantneed to go to the place where the house to be rented exists. So the new system consumes tenant time and money.

During renting a house the tenant personal information, payments status and rent agreements are filled in the house rent agreement form in order to hold legal contract between the tenant and house owner for renting a house.

## 2.3 Players in the Existing System

* House Owner:building houses then advertises that the houses will be rented either by city brokers or posting on the door some piece of information to inform that the house will be rented. They manage over house properties, payments and may also have full responsibility to carry out the renting activities.
* Renters (tenants): finda house to rent on foot or by paying for broker, then after informed that there is a house to be rented he/she submits his/her request to house owner and can get the home depend on the agreement between them.
* House Brokers:playing an intermediate role between the house owner and renters, by sharing information from tenants to house owner and vice versa to make agreement between them. They receive money from both of them.

## 2.4 Major Functions/Activities in the Existing System

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

Activities in the existing system are:

Input**:** when house ownersrenting the houses they use:

* House details as input in order to make the tenant aware about the house.
* Brokers’ information; telling full information about the house with its payment per month or year to help brokers to do their work easily.
* Lease agreement policy between house owner and tenants.

Process:

* Posting the house details for tenants by writing piece of information on computer then print the information; then post it on the door of the house to be rented or by advertising through city brokers.
* Then waiting for request of the interested tenant.

If the request reached;the tenant’s personal information, payments status and rent agreements are filled in the house rent agreement form in order to hold legal contract between the tenant and house ownersfor renting the house.

Output:after completing these processesthe house will be rented.

## 2.5 Business Rules

* System users should have secured private username and password in order to access the system.
* Tenants and house ownershould reach an agreement to rent the house.
* The tenant should have bank account and should have enough deposit in order to rent the house.
* Our system do not support agreement canceling before due date.
* If one tenant wants to cancel the agreement and leave the house or room he/she must inform the house or room owner before three days of leaving.
* Let payments will punish the tenant 2% of the total payment.

## 2.6 Report Generated in the Existing System

The house rental process reports are concerned on the following.

The report is generated based on the number of rented houses, the number of unrented houses and payment status of the houses. This report is generated by the house ownerto him/herself to announce available houses for renters.

## 2.7 Forms and other documents of the Existing System

Here are the following agreement forms that are filled by house ownerand the tenant. The consists date at which the agreement filled, renter full information, house ownerdetail information, the amount of money to be paid for the house per year and the amount of birr to be punished if someone violates the agreement. It also encompasses witness full names and signature.

## 2.8 Bottlenecks of the Existing System

### *2.8.1 Performance (Response time)*

The other problem which is resided in the current system is performance or response time for user requests. We are measuring the performance of the current system in terms of time. The time took by the current system to proving information about the houses to the tenants (renters) is so slow.This is because tenants need to search for the houses to rent on foot, these delay tenants’ requirements.

### *2.8.2 Have Problem in data Storage*

As the number of houses rented increases, storage of data also increases. So, it is very difficult to store and maintain all the data manuallyin the current system. Therefore, proper and systematic database should be used tostore these data growth and to allow user to update or maintain data efficiently.

### *2.8.3 Security and control*

Due to un organized storage of data there is data loss and redundancy of data occurs, so that the data is unsecured. There is no control mechanism since it is manual process, thus there may be vulnerability of data to unauthorized person.

### *2.8.4 Efficiency*

The other crucial problem which is found in the existing system is luck efficiency. We are measuring the efficiency of the current system in terms of resource: human and time resources. The existing system labor intensive and needs more time and expenses more money to get the service.

The current system uses more stationary materials like papers and pen frequently to store data. Because of this the current system spent more cost for data storage mechanism.

### 2.8.5 Input and output bottlenecks

As we try to list out the main inputs to our system the existing system inputs are detailed information about the house, use of information from the brokers and agreement signing form. This all are recorded manually and it is tedious activity.

### *2.8.6 More costly*

when tenant want to rent house he or she need to make physicalcontact to the house owners’ house; this exposestenants unwanted expense of time ,money and labor this makes tenant’s unsatisfide.

### *2.8.7 Working time wastage*:

Tenants do not gain access all over the week because of the house owners are limited to give access through specific time. This causes repetitive coming of tenants to the house owners’house until he get the house owner to make agreement.

Generally the following problems are exists in the existing system:-

* Most of the time uses manual ways to carryout its tasks.
* It takes more time and effort.
* Difficult for getting feedbacks from the tenants.
* It danger tenants’ economically.
* Problems of data loss and data redundancy
* Difficulties in checking house status whether the house is available for renting.
* Tenant does not know the detailed information about the house that they are renting.
* Difficult for tenants to get the house for the purpose they want to reserve for. For example if someone wants to rent house for apartment he must search throughout the city until he find the house, also if some tenant wants to rent the house for other purposes like business, for schooling and for other different services he may didn’t know the exact location of the house/s, and also do not knows what the house looks like.

## 2.9 Practices to be preserved

After we investigate the existing system we have got some strong sides in the existing system. These are listed below:

* Signing lease agreement must be manually signed in front of the witness.
* The signed lease agreement should be in the hand of house owner, one copy on the hand of tenant, and one copy in the hand of the witness (observer).
* After the house is rented; monthly or yearly payments should be managed manually.
* Penalizing a one violates the lease agreement according to constitution is totally preserved.

## 2.10 Proposed Solution for the new System

After deep identification of the real problems of the existing system, the team suggests an alternative solutions to overcome the problems.Since the new system we are interested to develop is advanced than the existing system, it is believed to solve the short comings in the existing system. We set the following alternative solutions to address these problems.

These alternative solutions are:-

* Changing the manual system into a online system that works on web based environment.
* Uses technological way to providereliable services.
* There will be fair service that will tend for satisfaction.
* Develop web based system to satisfy tenants.
* Develop online web based system in order to secure tenants from unwanted cost and also make the house ownersprofitable.

We are trying to analyze all of the alternative solutions based on the existing system performance, information flow, efficiency and service provided to the users. This analysis has enforced us to select the online web based system.

design user friendly and interactive graphical user interface; requirement analysis is the key phase in project development. Requirement is divided into two: function requirement and non-functional requirements

### *2.10.2 Functional requirements*

Functional requirements are intended behaviors of the system. These behaviors include statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. It specifies the software functionality that the developers must build into the product to enable users to accomplish their tasks.

The main functional requirements of this system are:-

* View:-The system allows tenants, house owners and administrator to view posted resources.
* The system allows tenants to send their requests to house ownerto reserve the house.
* Send response:- when the house owneragree with tenants’ requests then, they can send acceptance message to the tenant.
* The system allows users to make payments to the house owner and follow the durational payment progress.
* Delete:-delete unnecessarily information of tenants request and house owner resources
* Updating information: -the system allows updating the existing resource information by the house owners.
* Create account:-Making the system more secured by creating login account for all tenants, house owners.
* Upload:-The system allows to house owners to register and upload their house or room and its properties.
* Search:-The system is providing search tool for a tenants, house owners, and administrator to search for houses.
* Feedback:-The tenants can send any suggestion regarding the system.
* Send request:-the system allows tenants to make request for the house he selected to rent.

Performance Requirements:-

Some performance requirements are listed here

* The database should accommodate multiple records.
* The system should support multiple users simultaneously at a time.
* Users need to have basic computer knowledge.

Process Requirement:-

The system will provide the following data processing.

* Data validation; only valid data is required to be stored in the system MYSQL server.
* The system will support data deletions, insertions by authorized users.

Input/output Requirements:-

*Data entry*

The system has different fields to perform different tasks and used to manage data entry mechanism.

* Login: to identify the authorized person to use the system.
* Data update: needs to update data, upload new houses or rooms, and delete houses from the system when it is necessary.
* Search for information: needed when the user wants to search for houses to view posted houses.
* Request for renting house.

*Data processing*

The system on input data will provide the following data processing.

* New house registration.
* House updating (deletion, add).

Storage Related Requirement:-

The system will store all in formations in MySQL server. The system needs 350-4500MB storage to run its tasks.

### *2.10.2 Nonfunctional requirement*

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to evolving system properties such as reliability, response time and performance. Alternatively, they may define constraints on the system implementation such as the capabilities of input output devices or the data representations used in interfaces with other systems. Non-functional requirements, such as performance, security, or availability, usually specify characteristics of the system as a whole.

It describes aspects of the system that are concerned with how the system provides the functional requirements. They are:

* Performance (Response time)

Since the system is designed to be accessed by different users with different needs, it is capable of handling and processing their queries quickly. Usually it is hard to tell exactly how many users will be using the system at a time. However, if the system is being accessed by many, all the users must feel that they are the only one using the system.

* User interface:-since the system is designed with attractive anduser friendly interface;users feel comfort and are interested to use it.
* Error handling: - when the user makes some mistakes the system responds that error is occurred using easily understandable messages and allows the user to recover from the error.
* Availability:-the system is available at any time with internet connection.
* Security and controls:-Access to data must be controlled. No one can log into the system without a registered user name and corresponding password. No other program except the system client can be used to log onto the system server.
* Efficiency: - The system responds to user requests as fast as possible and fully operational every day for 24 hours depending on availability of internet access.
* Time minimization:the time taken to find a house to rent will be reduced.
* Cost minimization:since searching is online expenses are reduced.

# **Chapter Three**

# **Process Modeling**

## 3.1 Introduction

Under this topic we use different kinds of UML diagrams to model the functionality, structure and Sequence of activities of the system. In this chapter the major activities performed or identified are: modeling the function of the use case, identifying actors, identifying use case, constructing use case model and use case scenarios and finally designing of user interface.

System modeling involves the evaluation of system components in relationship with one another to determine their requirements and how to satisfy them. Some system modeling tools will be employed during the course of this projectthat will support development tasks, from analysis to design, then to implementation. This will be represented with theuse of the sequence diagram, activity diagram, state chart diagram, collaboration diagram and class diagram for the online house rental system**.**

Here are the scopes of the system modeling phase.

* Functionality description of the system by using use case diagrams.
* Structural description of the system in class diagram.
* Communication of the objects of the system by using sequence diagrams.
* Flow of actions using activity diagrams.

## 3.2 System Requirement Specifications (SRS)

### *3.2.1 Use case Model*

The use case model describes the proposed functionality of the new system. A use case represents a discrete (distinct) unit of interaction between users and the system. Use cases are tasks of the proposed system.

* Actor Specification

Actors are external factors that interact with the system. This may include house owners, tenants, and administrator. An actor initiates a use case and receives something of value from the use case. Actors are always external to the system being modeled i.e. they are not parts of the system.

One can use the following criteria to find actors: Who is using the system, who is affected by & affects the system, what other system interact with this system, where does the system get information, who install the system, who obtain information from this system and the like. Based on these criteria the following actors are identified.

* Name: -Administrator

Description: -aperson who owns the system and can manage the whole system including house owner’s houses information and manages tenants.

* Name: - Tenant

Description: - Tenants are persons who search a house to rent in this system.

* Name:-House owner

Description: - a persons who own houses, can post and delete Tenants’ request as well as who can send response if the tenants request and house owner’s resource match.

Use Case Identification

* Create Account
* Login
* Change password
* View Resource
* Send request
* View request
* Send response
* View response
* Pay money to rent house
* view payment progress
* Upload house or Room
* Update Resource
* Delete Resource
* Search Information
* Send feedback
* Flexibility
* Includes high quality shapes, examples, templates
* Works with MS office well

Notepad++:-used to editor

Snipping tool:-used to cut and transfer graphs or images from one document to another.

* + 1. Hardware Tool
* Computer Specification: Hard disk ------- 80 GB
* View feedback
* Logout

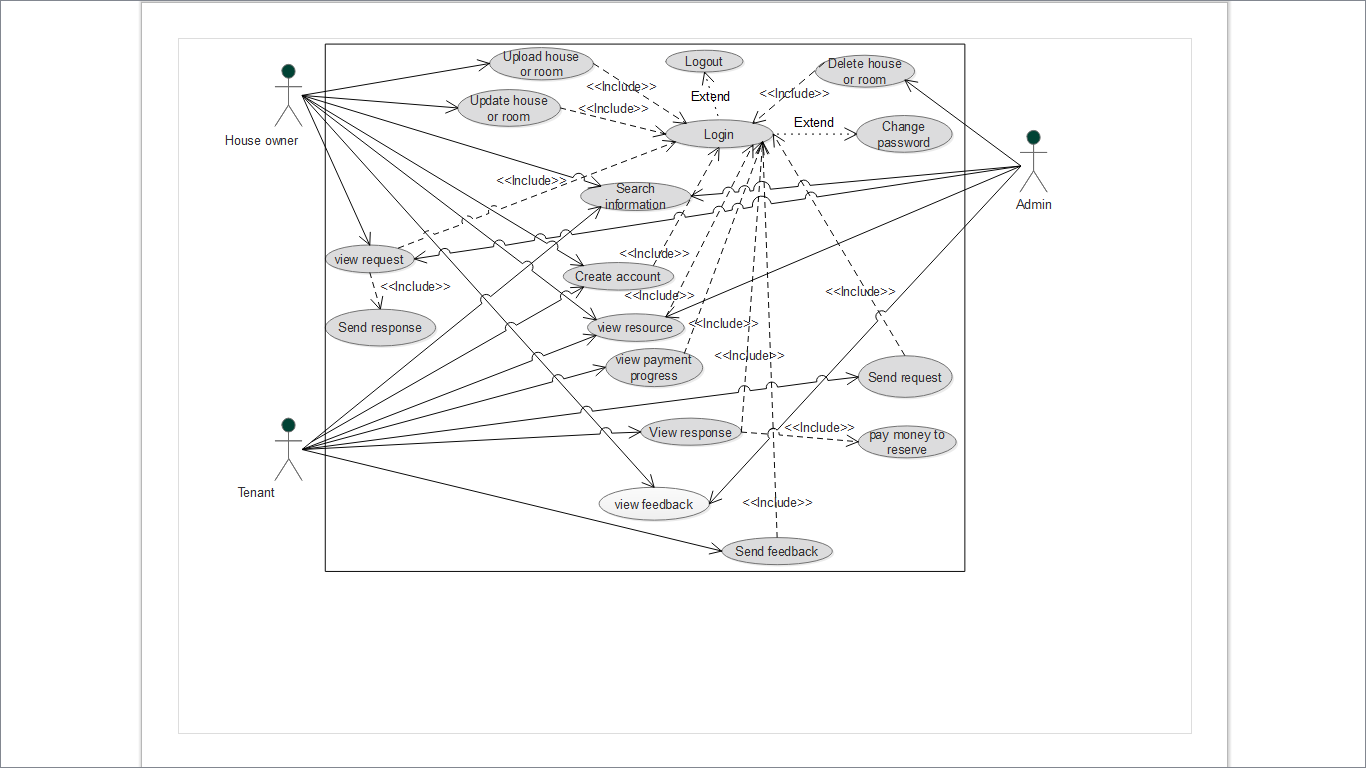


Figure 4: Use case diagrams for online house rent system

### *3.2.2*Use case documentation

Use case description includes descriptions of the use case, preconditions, post conditions, flow of event and whatever which is important in modeling the user goal.

Create Account use case Description

Table 4: Create Account table

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | | | Create Account |
| Actors | | | Tenant and 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 course of action | | | 1. The tenants, house owner and house mangers 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.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 flow 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. |

Login Use Case Description

Table 5: Table of login

|  |  |
| --- | --- |
| Use Case name | Login |
| Actors | Tenants, house owners, and administrator |
| Description | Users use this use case to log into the system. |
| Pre-condition | The user should be registered first or need to have account to login |
| Basic course of action | 1. The user enters to the home page. 2. Select login link, enter name and password and click login button. 3. The system checks for validity of user name and password entered. 4. The System displays user page if user name and password is correct. 5. Use case exit. |
| Alternative course of action | A1 If the entered user name and password is not valid it displays“please enter valid user name and password” go to basic flow 2.  A2 If the user forgot the user name and password display reset user name and password page. |
| Post Condition | The users logged in to the main page. |

Change password Use Case Description

Table 6: Use case Description for Change Password

|  |  |
| --- | --- |
| Use case name | Change password |
| Actors | Tenants, house owner, and admin |
| Description | Allows the system users to change their password. |
| Pre-condition | User must have older password |
| Basic course of action | 1. User login first into the system. 2. Choose change password option 3. System will 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. |
| Alternate 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. |

Search House Information Use Case Description

Table 7: use case description for search information

|  |  |
| --- | --- |
| Use case name | Search Information |
| Actor | Tenant. House owner, and admin |
| Description | When tenant enters their information needed and click search option, the information needed for the tenant is displayed if available, otherwise the item is not found prompt is displayed. |
| Pre-condition | Tenant does not need to have an account to search. |
| Flow of action | 1. This use case is initiated when the actors clicks on search option. 2. The system displays the page that contains a text box and search button. 3. Tenant enters their need and press search button. 4. System displays the requested information if available otherwise item is not found prompt will be displayed. 5. Use case Exit |
| Post condition | Users able to find the information they want. |

|  |  |
| --- | --- |
|  | |
|  |

Send Request use case Description

Table 8: use case description for Send request

|  |  |  |
| --- | --- | --- |
| Use case name | Send request | |
| Actor | Tenant | |
| Description | This use case permits to register rental information of the tenant and the house that the tenant rents. | |
| Pre-condition | Tenant first needs to login to the system and choose the house to rent on the home page. | |
| 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: Tenant fill all the information  4: Tenant clicks or presses on the save or insert button.  5: The system verifies that the fields have been filled out correctly.  6: System displays inserted successfully.  7: Use case Exit | |
| Alternative course of action | A.1 If the tenant’s fields are not filled out correctly system goes back or returns to step 4 of basic course of action to fill invalid field. | |
| Post condition | Tenant’s request is registered. | |
| View Response Use case Description  Table 9: use case description for view response | |
| Use case name | View responses |
| Actor | Tenants, house owners, and Admin |
| Description | This option allows the Tenants, house owners and admin to view the response for tenants request from the house owner to tenant. |
| Pre-condition | There must be sent request earlier. |
| Basic flow of action | 1. Tenants, house owner, and admin login to the page. 2. Select view option response option. 3. System will display the response if any. 4. The use case exit. |
| Post condition | Able to view response from house owner to tenants. |

Upload House or Room Information Use Case Description

Table 10: use case description for Upload house or room information

|  |  |
| --- | --- |
| Use case | Upload House or Room Information |
| Actor | House owner |
| Description | This use case permits house owner to register rental information of the house or room that they will rent. |
| Pre-condition | * Logging into the system * Having house information to be posted. |
| Flow of action | 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: 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 | A.1 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 | House owner information is registered. |

Delete House Information use case Description

Table 11: use case description for Delete house

|  |  |
| --- | --- |
| Use case | Delete house information |
| Actor | Admin |
| Description | It allows admin to delete house owner’shouse information from the system. |
| Pre-condition | The admin must have an account and must have list of available houses deletion requests from the house owners. |
| Flow of event | 1. This use case is initiated when admin click on manage houses resources option. 2. System displays the page that contains delete buttons. 3. Admin delete resources. 4. System displays deleted successfully if admin delete house information. 5. Use case Exit |
| Post condition | House owners’house information is deleted from the system. |

View tenants request use case Description

Table 12: use case description for View Tenants request

|  |  |
| --- | --- |
| Use case | View tenants request |
| Actor | House owner |
| Description | When house owner choose view tenant request option, the system displays information about tenants. |
| Pre-condition | 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 house owner clicks the view tenant request option.  2: The system displays available name of tenant and some descriptions.  3: house owner clicks the view request option.  4: System displays all the available requests, recorded in the database.  7: Use case Exit |
| Alternative flow of action | House owner only see the name and the description of tenant. |
| Post condition | House owner knows if there is tenant’s request. |

View house information use case Description

Table 13: use case description for View house information

|  |  |
| --- | --- |
| Use case | View house information |
| Actor | Tenant, house owner, and administrator |
| Description | When actors choose view house information option, the system displays information. |
| Pre-condition | Users need to have an account if they want to see more information. |
| Flow of event | 1. This use case is initiated when the actor clicks on view house owners’ resources option. 2. System displays available name of the house owner and some descriptions. 3. Admin and tenant click on renting option. 4. System displays all the available resources recorded in the database. 5. Use case Exit |
| Post-condition | Admin and tenant know if house information is reserved or not. |
| Alternative flow of action | Actors see only name of the house owner and the description if they don’t want to login. |

Send Request use case Description

Table 14: use case description for Send Response

|  |  |
| --- | --- |
| Use case | Send response |
| Actor | House owner |
| Description | When house owner chooses view tenant request option, the system displays information about who is requesting. Then he/she canreplies the response to the requester. |
| Pre-condition | * House owner must first logged in * There must be request from tenant(s). |
| Flow of events | 1. The house owner click view request link. 2. System displaysthe response page. 3. User 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 send to the tenant |

Reserve the House use case Description

Table 15: use case description for reserve the house

|  |  |
| --- | --- |
| Use case name | Reserve the house |
| Actor | Tenants |
| Description | This use case permits tenants to reserve the house and make payment to the house owner based on the response from them. |
| Pre-condition | The tenants and the house owner must rich an agreement. |
| Basic course of action | 1. The tenants 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 house owner. 4. If the house owner sends response the tenants press reserve button. 5. The system displays payment form for the tenants. 6. Tenants fill 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 | A.1 If the tenants filled invalid information, the system goes back to step 6 to enter the invalid field again. |
| Post condition | House is rented to the tenants. |

Update House information use case Description

Table 16: use case description for Update house information

|  |  |
| --- | --- |
| Use case name | Update house information |
| Actor | House owner |
| Description | Only house owners can update their own house information |
| Pre-condition | * House owner needs to have an account to modify his property. * House owner should have resource posted first to update it. |
| Basic flow of action | 1. This use case is initiated when they click on update house link. 2. System displays the page with house information with update button. 3. House owner updates the fields. 4. Then press update button. 5. System displays success message. 6. Use case exit. |
| Post condition | House information is modified. |

Send feedback use case Description

Table 17: use case description for send feedback

|  |  |
| --- | --- |
| Use case name | Send feedback |
| Actor | Tenant |
| Description | To enable the users to send the feedback, to comment any Suggestion on text area. |
| Pre-condition | The tenants must be having a suggestion about the system as well as houses property. |
| Basic course of action | 1. The tenants click feedback button. 2. System display text area box. 3. The tenants write some necessary information and click post button. 4. The system validates the entered data. 5. The system can make save on database and post on home page 6. Use case Exit |
| Post condition | Suggestion will be posted |

View feedback Use Case Description

Table 18: use case description for view feedback

|  |  |
| --- | --- |
| Use case name | Actor |
| View feedback | Admin |
| Pre-condition | To view login to system first and there must be sent feedback. |
| Description | This helps the service provider to improve services they are providing or to appreciate the services provider. |
| Basic course of action | 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 response. 4. Use case exit. |
| Post condition | Able to view feedback. |

Logout use case Description

Table 19: use case description for Logout

|  |  |
| --- | --- |
| Use case name | Logout |
| Actors | Tenant, house owners, and admin |
| Pre-condition | Administrator, tenants, and house owner should be logged in first. |
| Description | To out from services. |
| Basic flow of action | The user clicks on logout button. |
| Post condition | users logged out from the system |

### *3.2.3 Sequence diagram*

UML sequence diagrams model the flow of logic within our system in a visual manner thatenableus to document and validate our logic and are commonly used for both analysis and design purposes. This sequence diagrams are dynamic modeling, which focuses on identifying the behavior within our system.

Sequence Diagram for Create Account

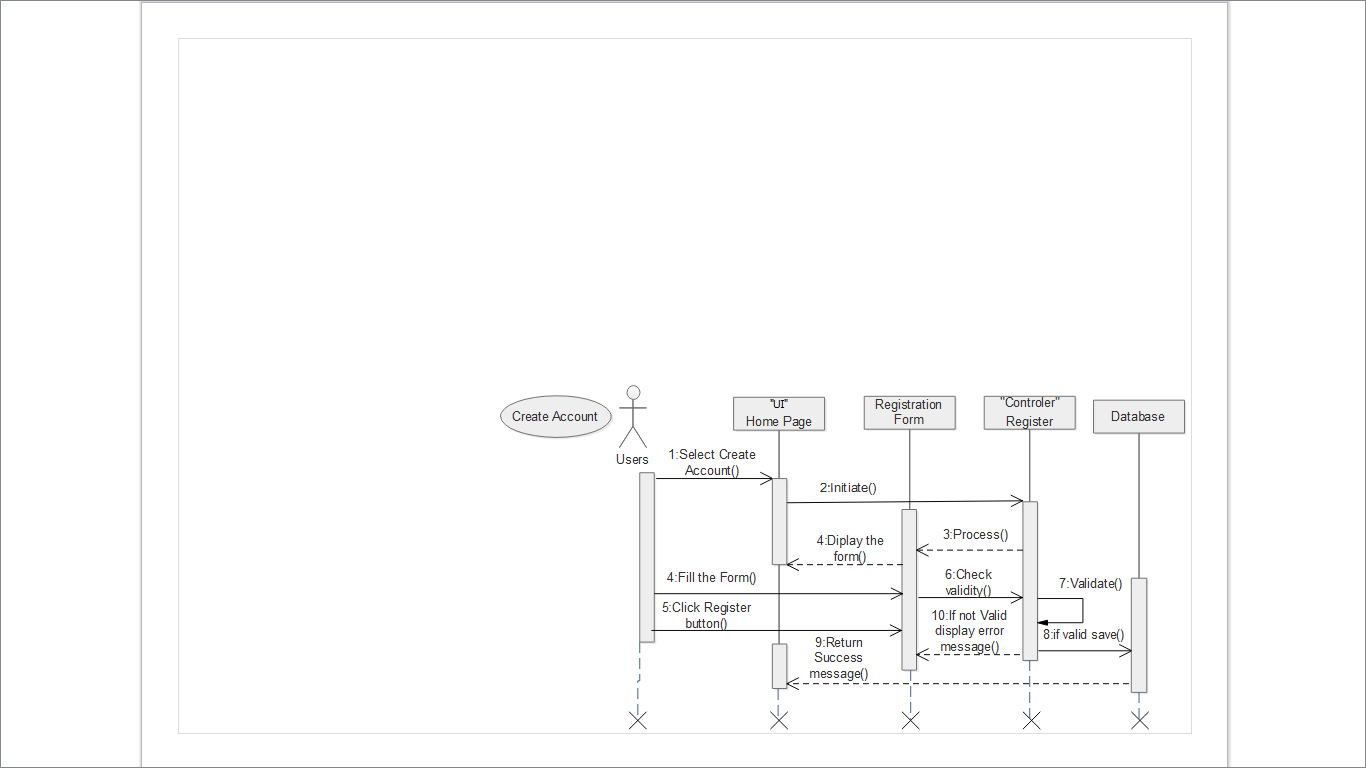


Figure 5: Sequence Diagram for Create account

Sequence Diagram for Login

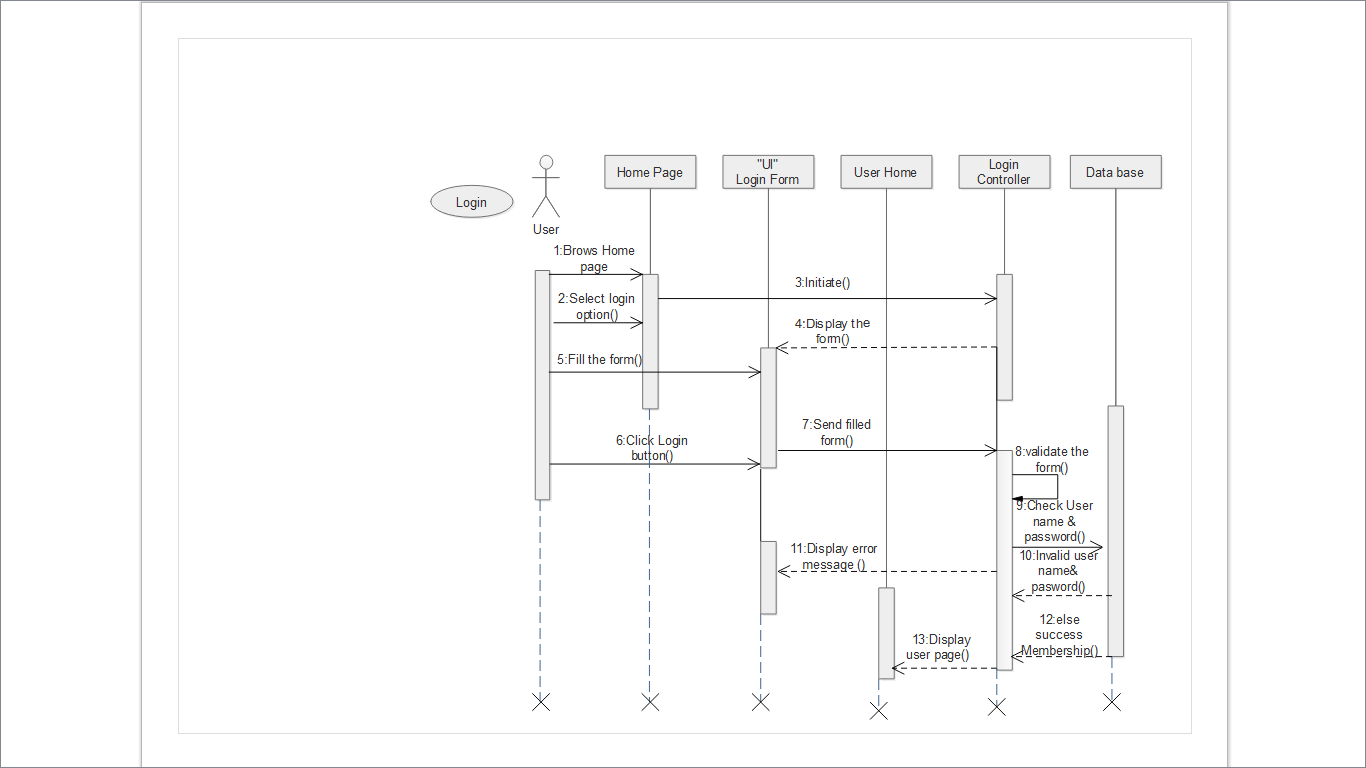


Figure 6: Sequence Diagram for Login

Sequence Diagram for Change Password

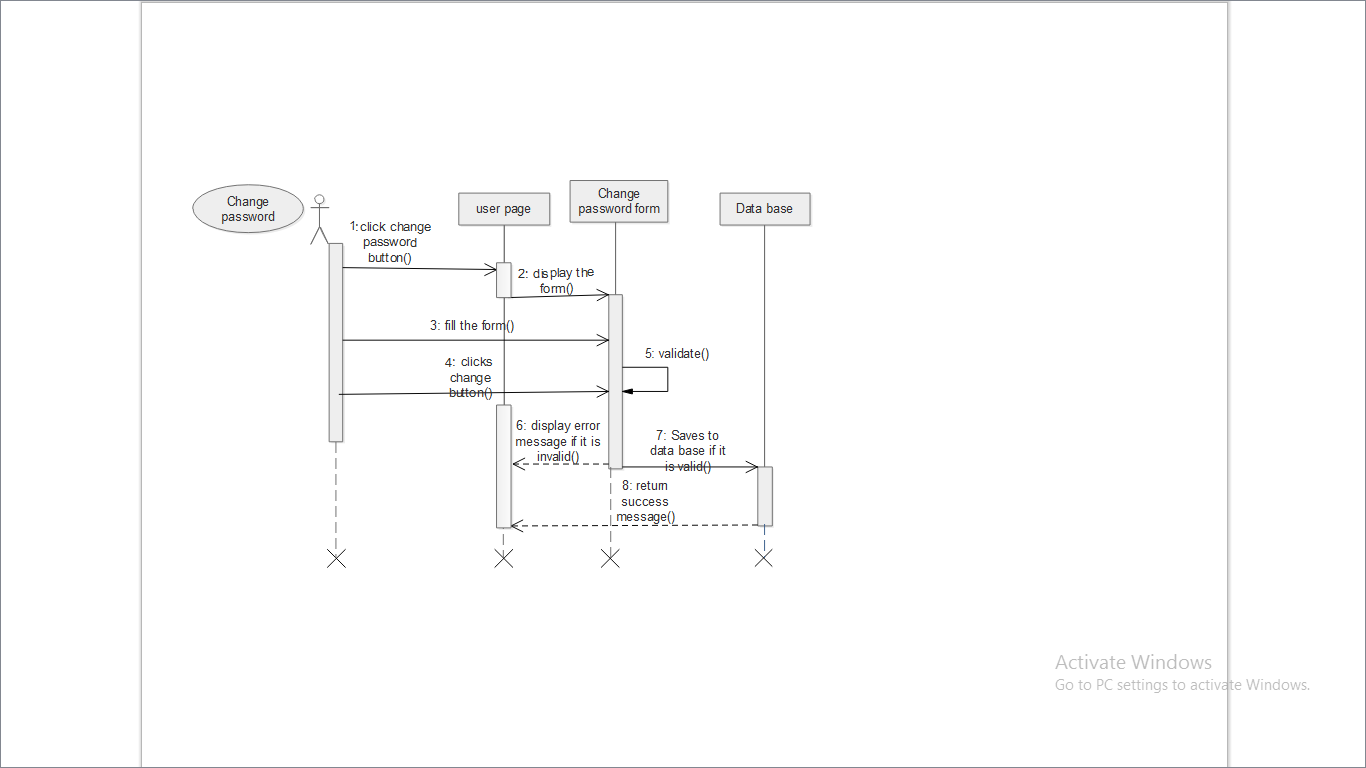


Figure 7: Sequence Diagram for Change Password

Sequence Diagram for Send Request

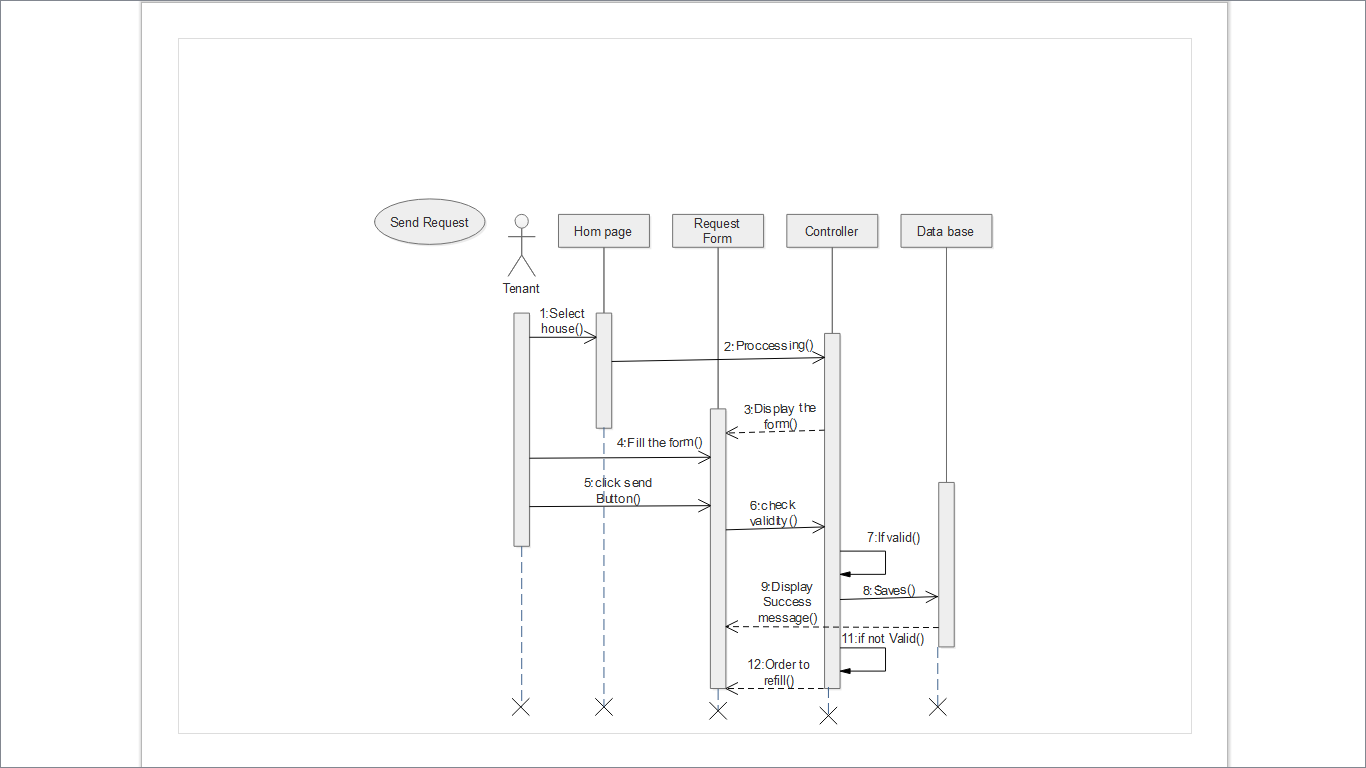


Figure 8: Sequence Diagram for Send Request

Sequence Diagram view request

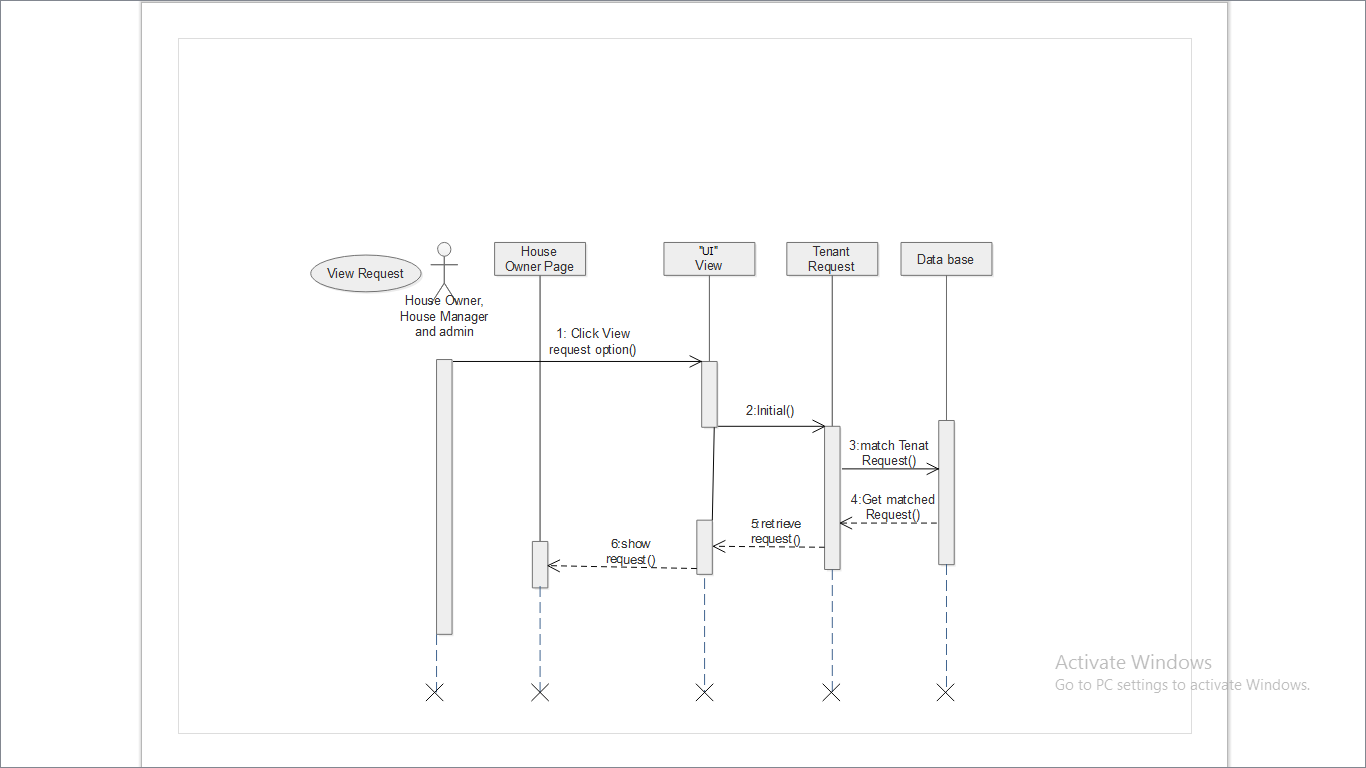


Figure 9: Sequence Diagram for View Request

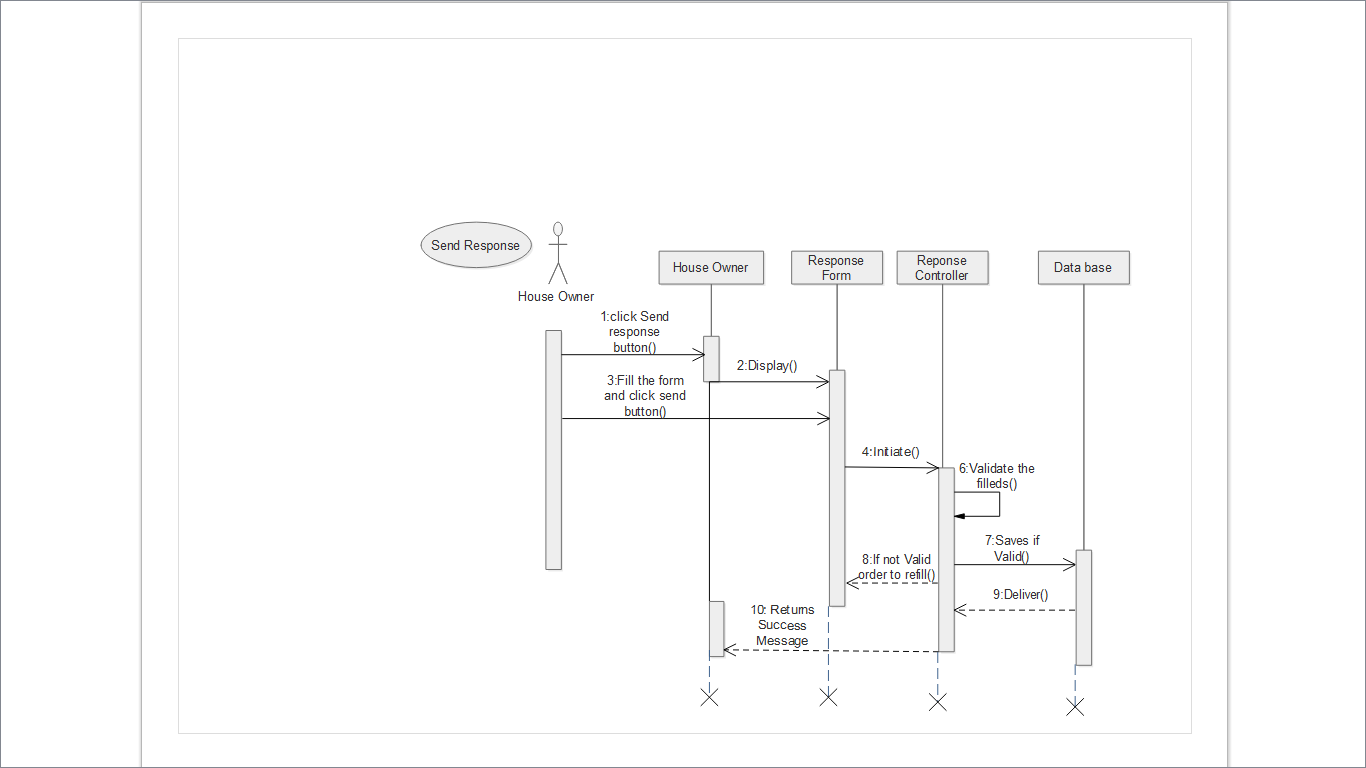
Sequence Diagram Send Response 

Figure 10: Sequence Diagram for Send Response

Sequence Diagram for view Response

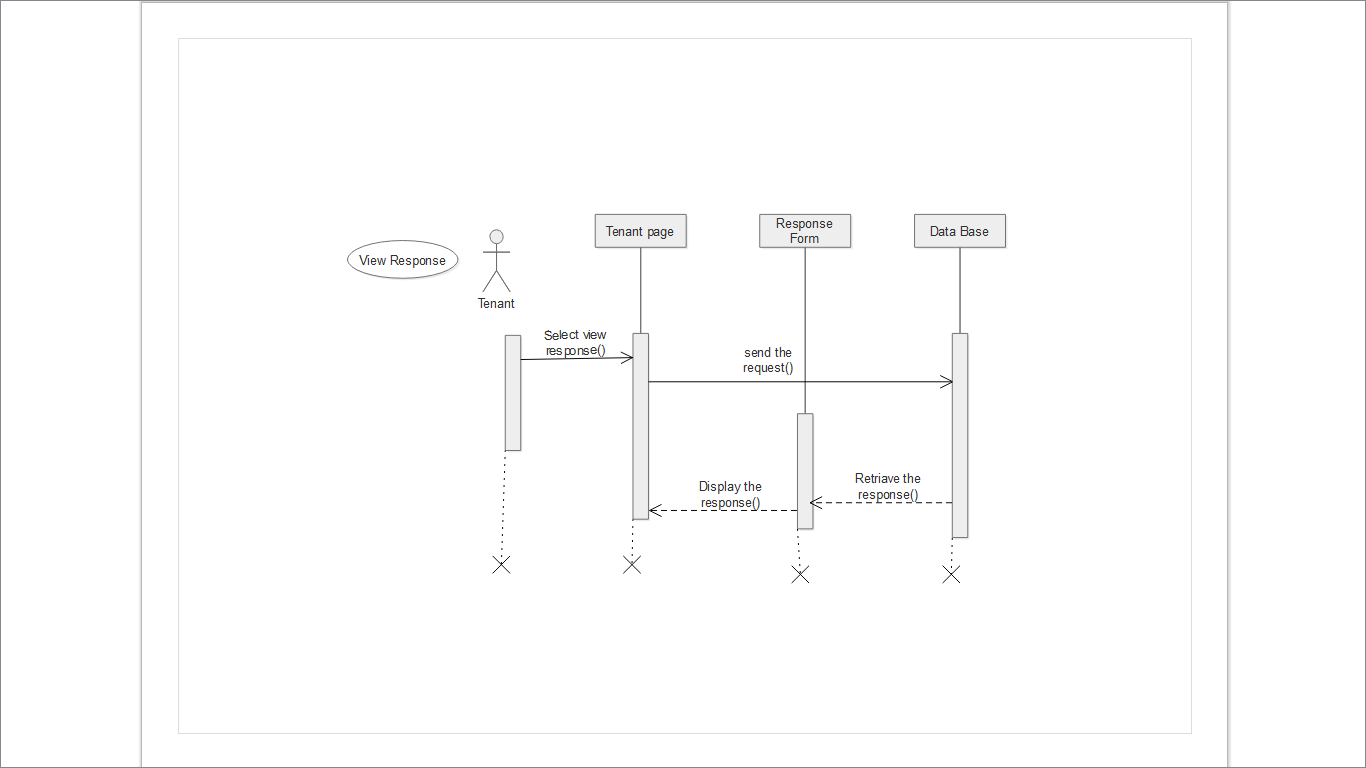


Figure 11: Sequence Diagram for View Response

Sequence Diagram for Reserve House

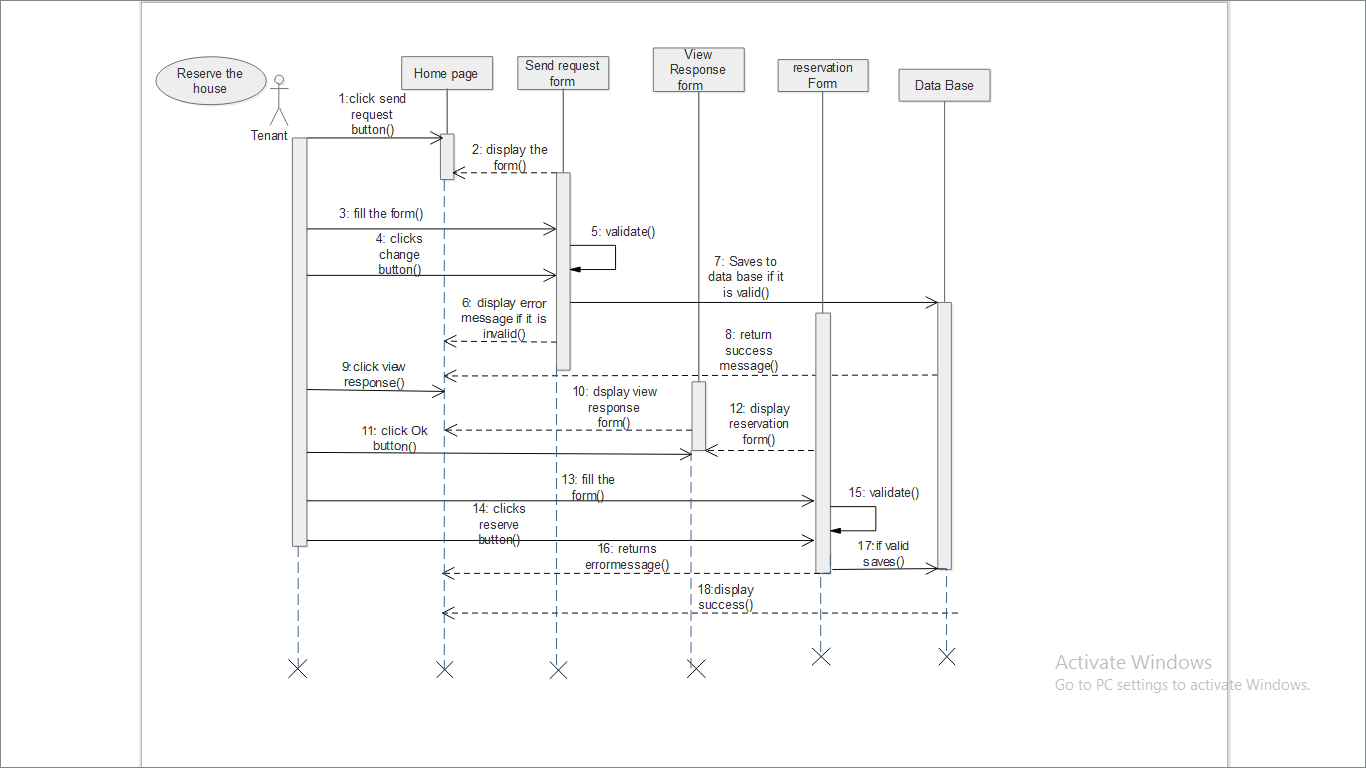


Figure 12: Sequence Diagram for Reserve House

Sequence Diagram for Upload House or Room

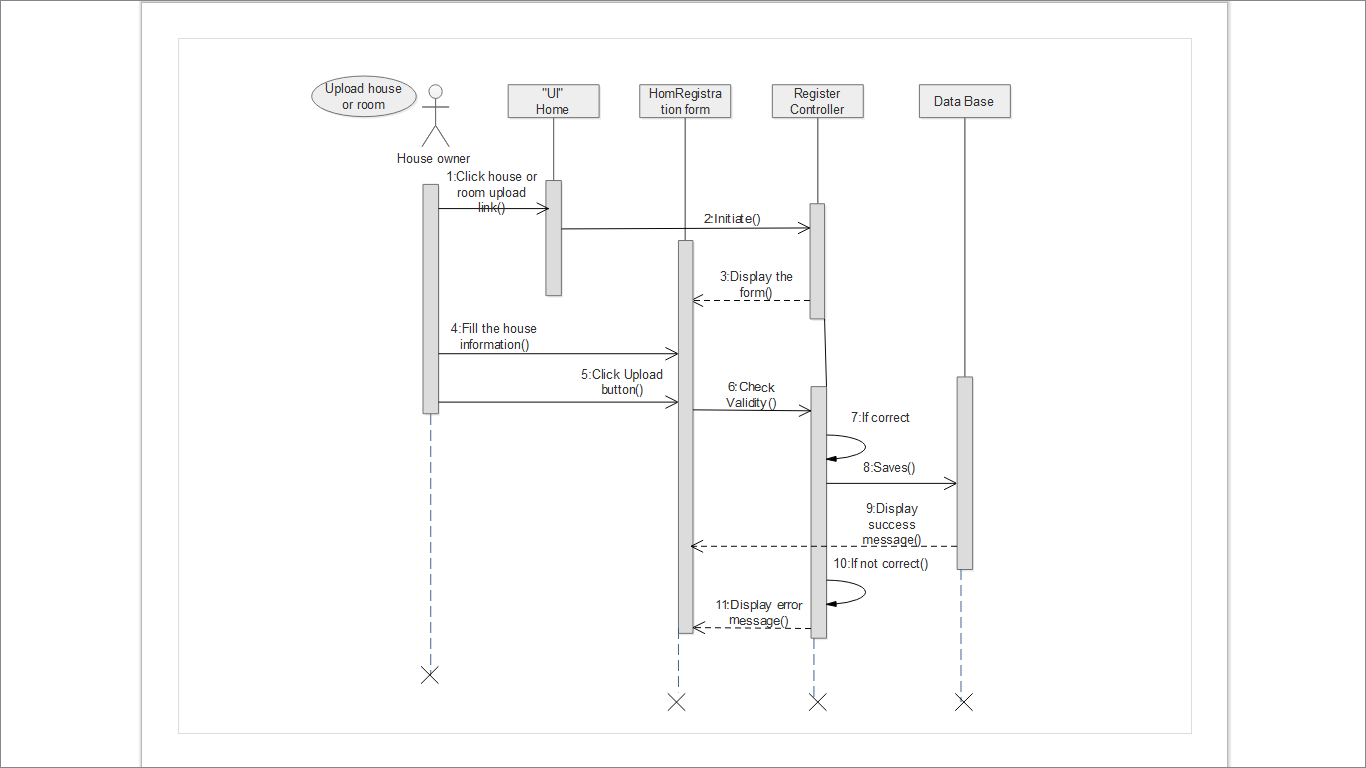


Figure 13: Sequence Diagram for Upload House or Room

Sequence Diagram for View House Information

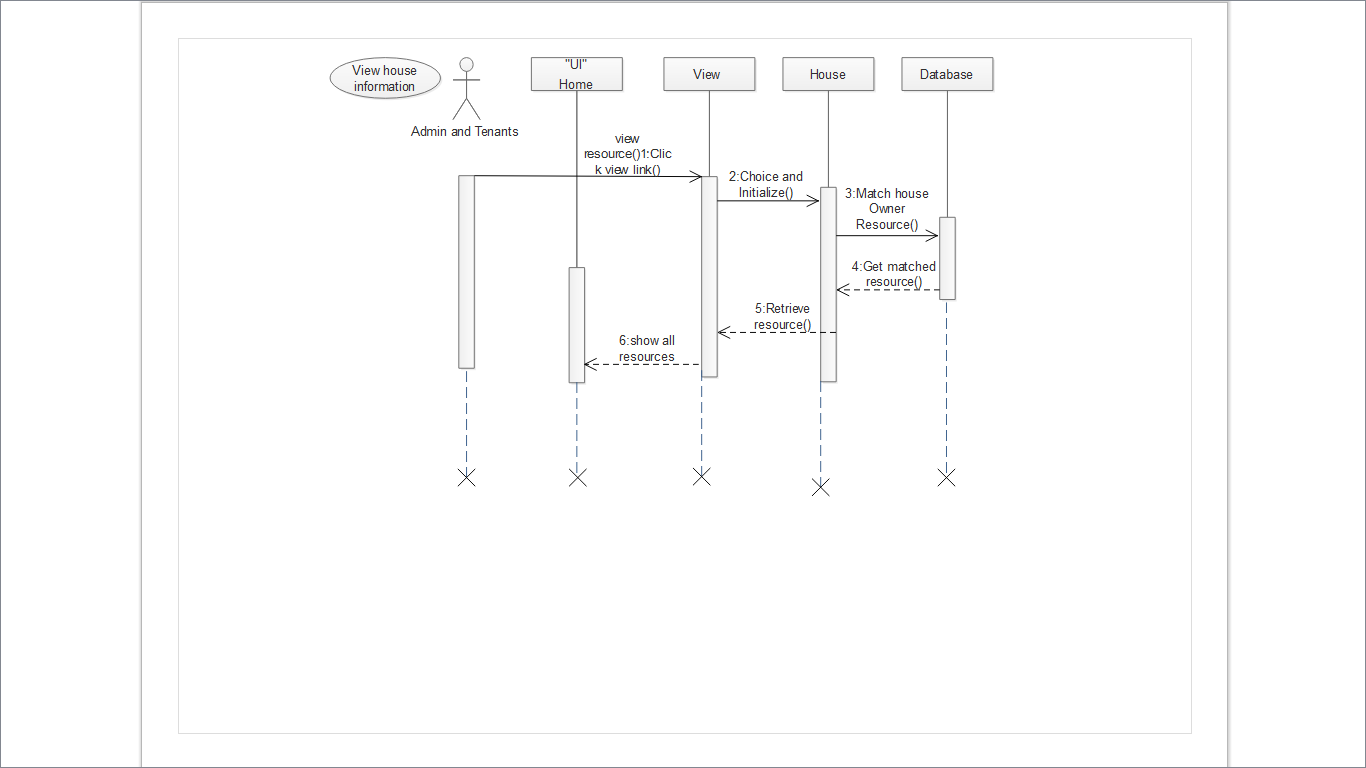


Figure 14: Sequence Diagram for View House Information

Sequence Diagram for Delete House Information

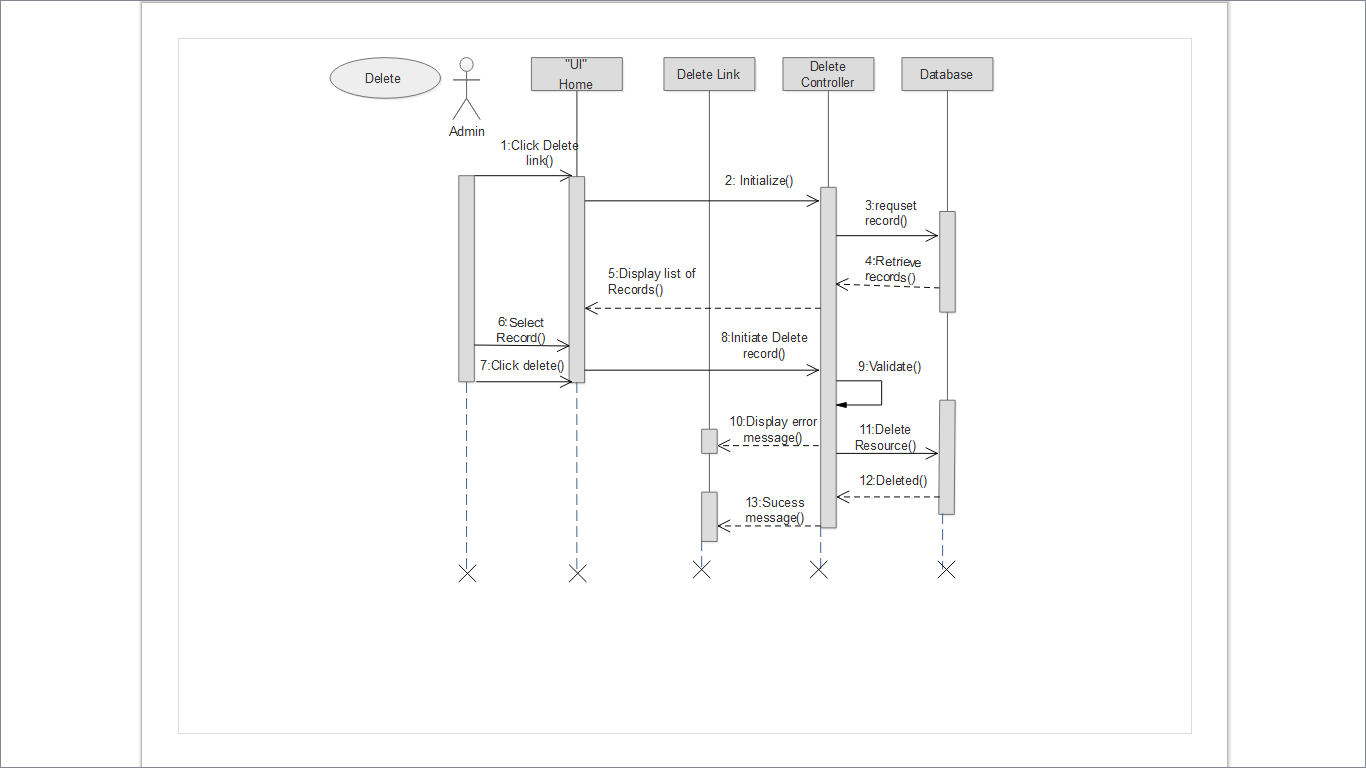


Figure 15: Sequence Diagram for Delete House Information

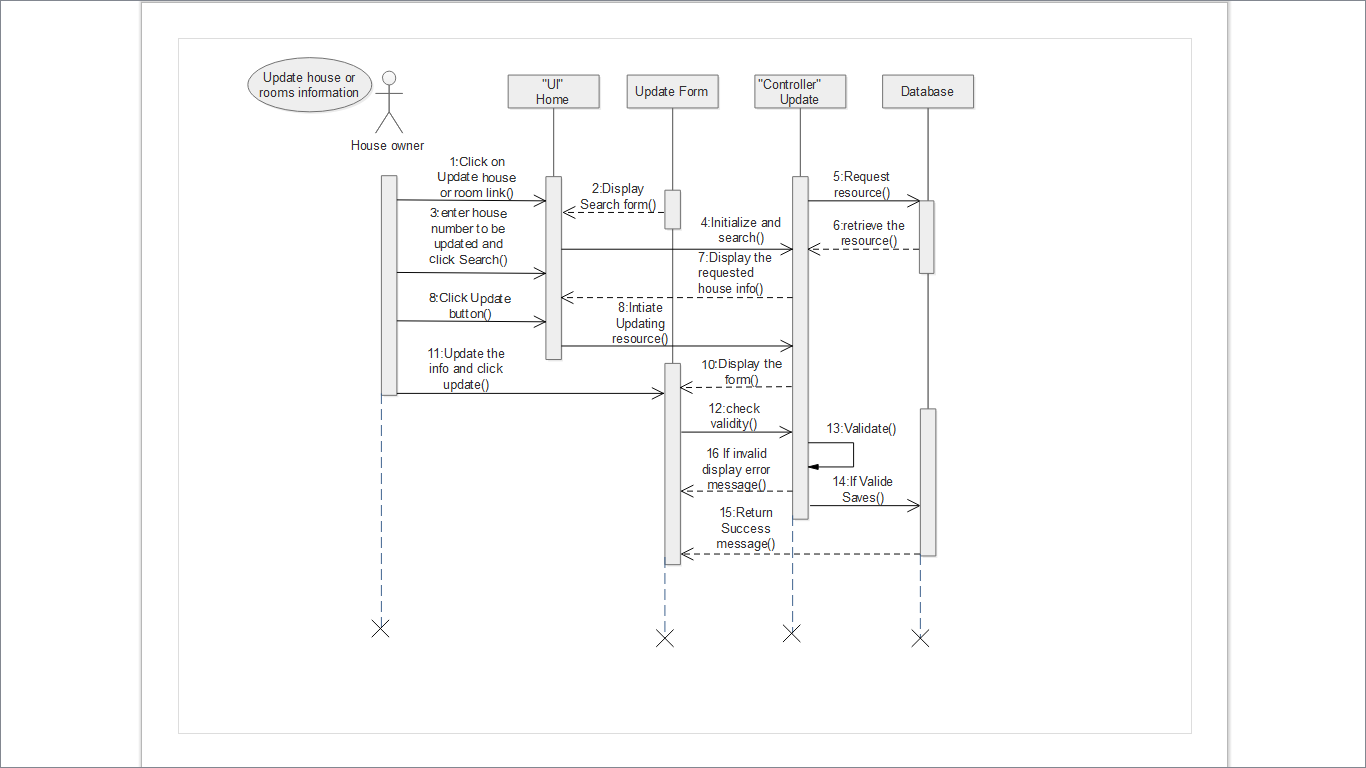
Sequence Diagram for Update House Information

Figure 16: Sequence Diagram for Update House Information

Sequence Diagram for Send Feedback

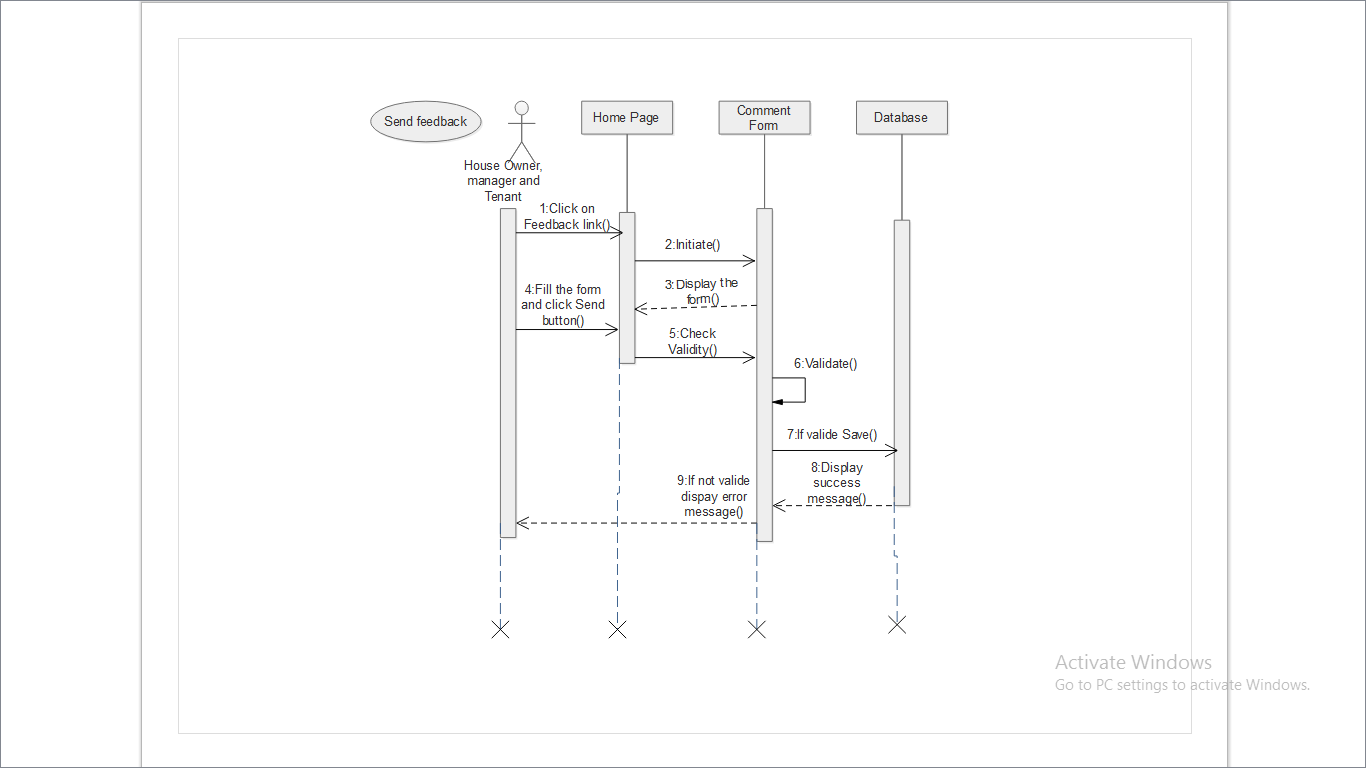


Figure 17: Sequence Diagram for Send Feedback

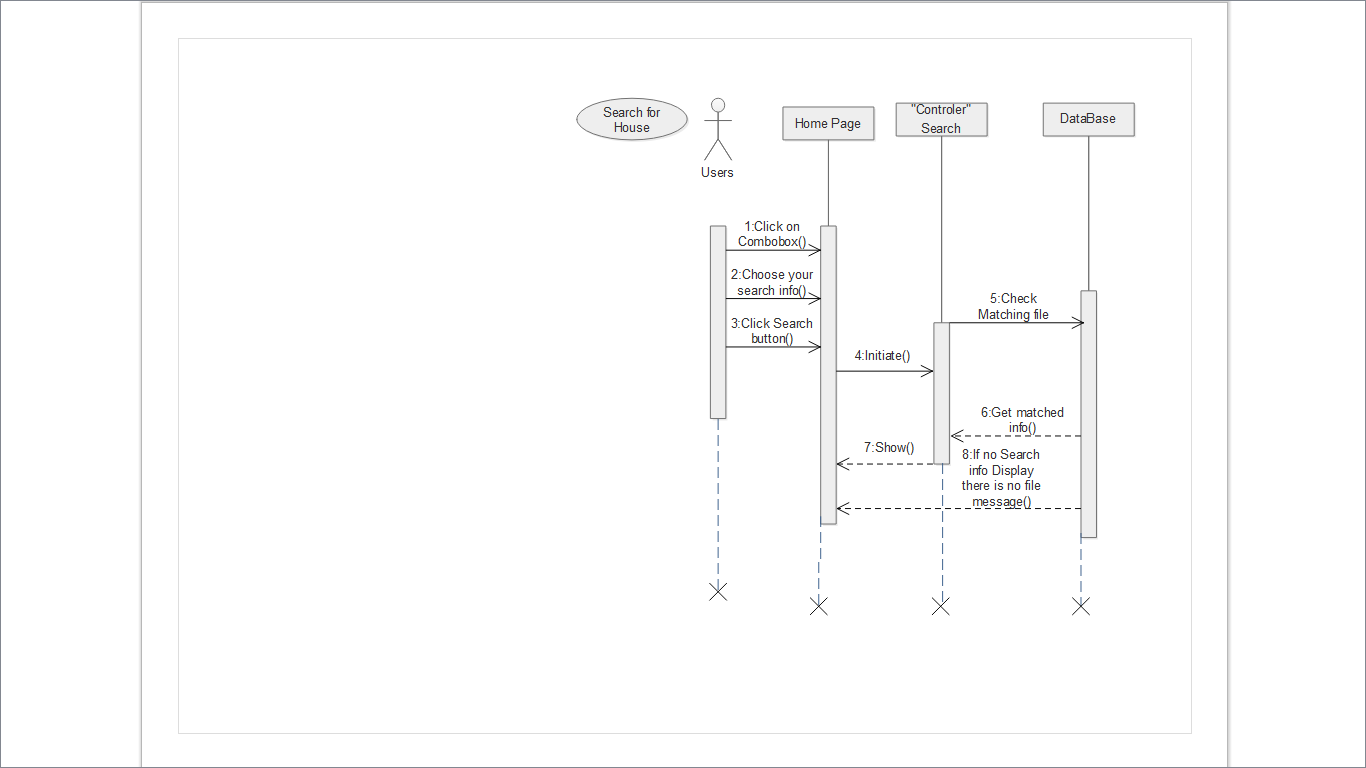


Figure 18: Sequence Diagram for Search Information

Sequence Diagram for Logout

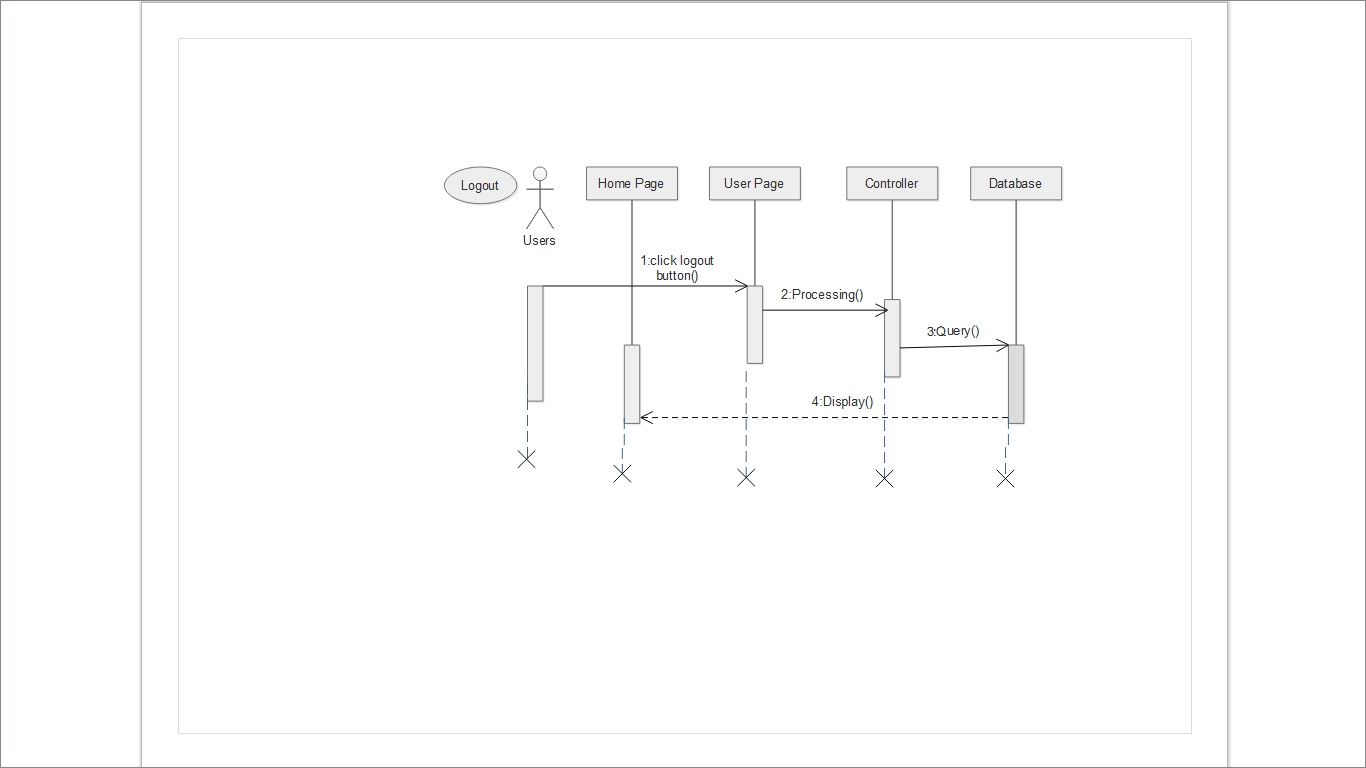


Figure 19: Sequence Diagram for Logout

Destroy

### *3.2.4 Activity Diagrams*

Online house rental system activity diagram represents the flow from one activity to another activity. They focus on the flow of activities involved in a single process. It also shows how these single-process activities depend on one another.

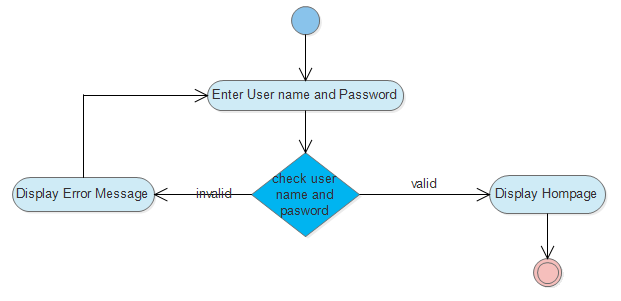


Figure 20: Activity Diagram for Login

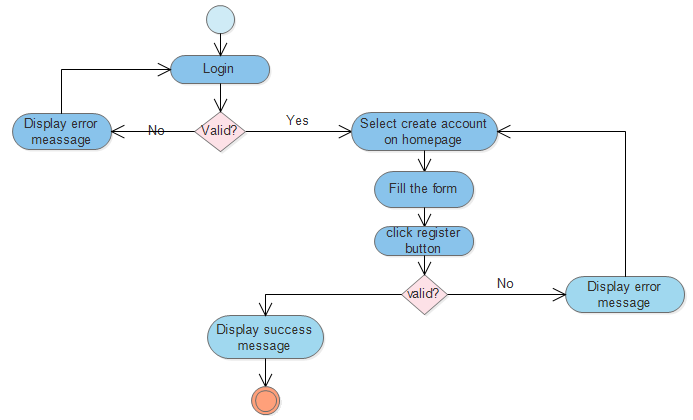


Figure 21: Activity Diagram for Create Account

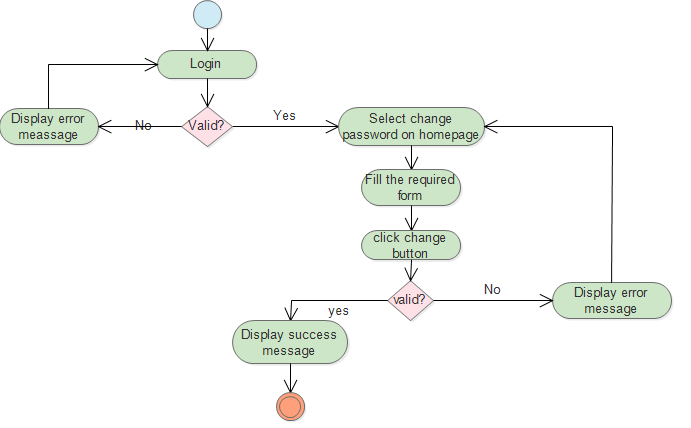


Figure 22: Activity Diagram for Change Password

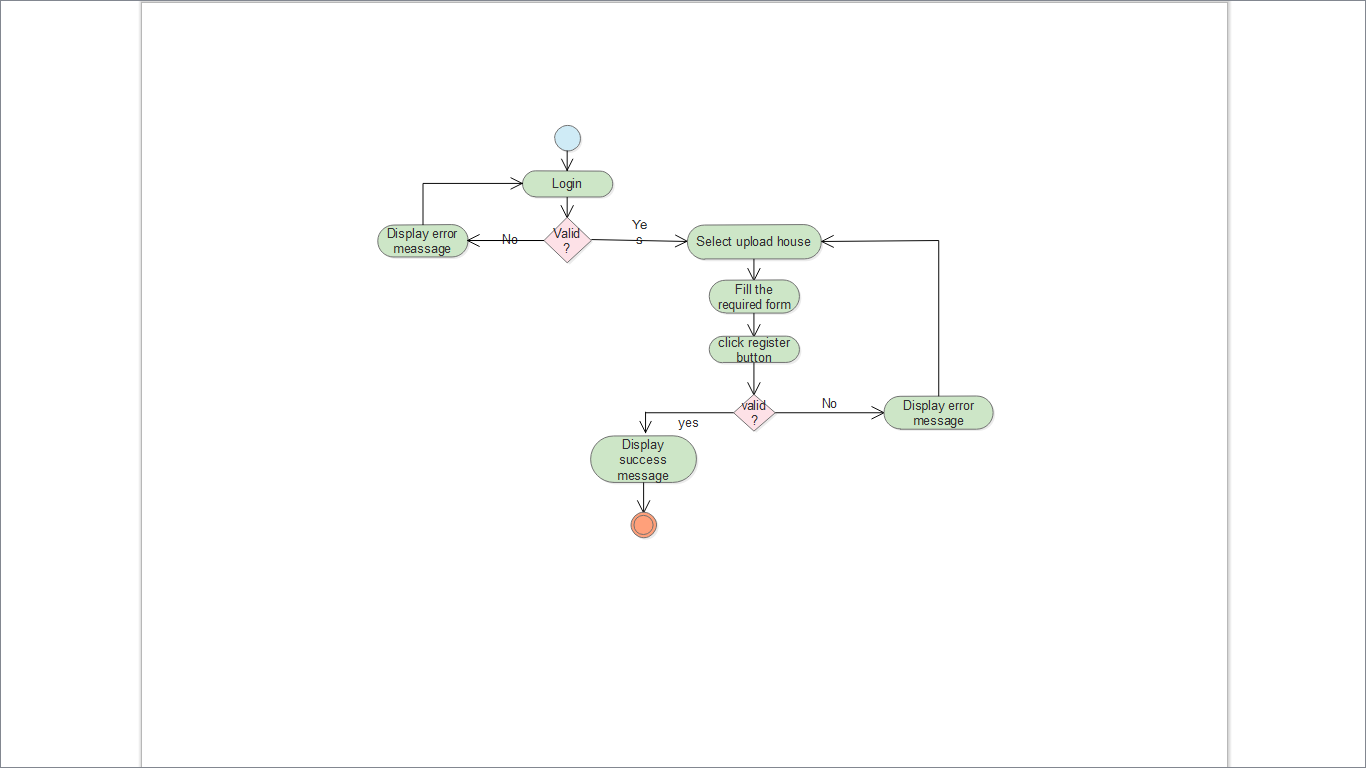


Figure 23: Activity Diagram for Upload House or room

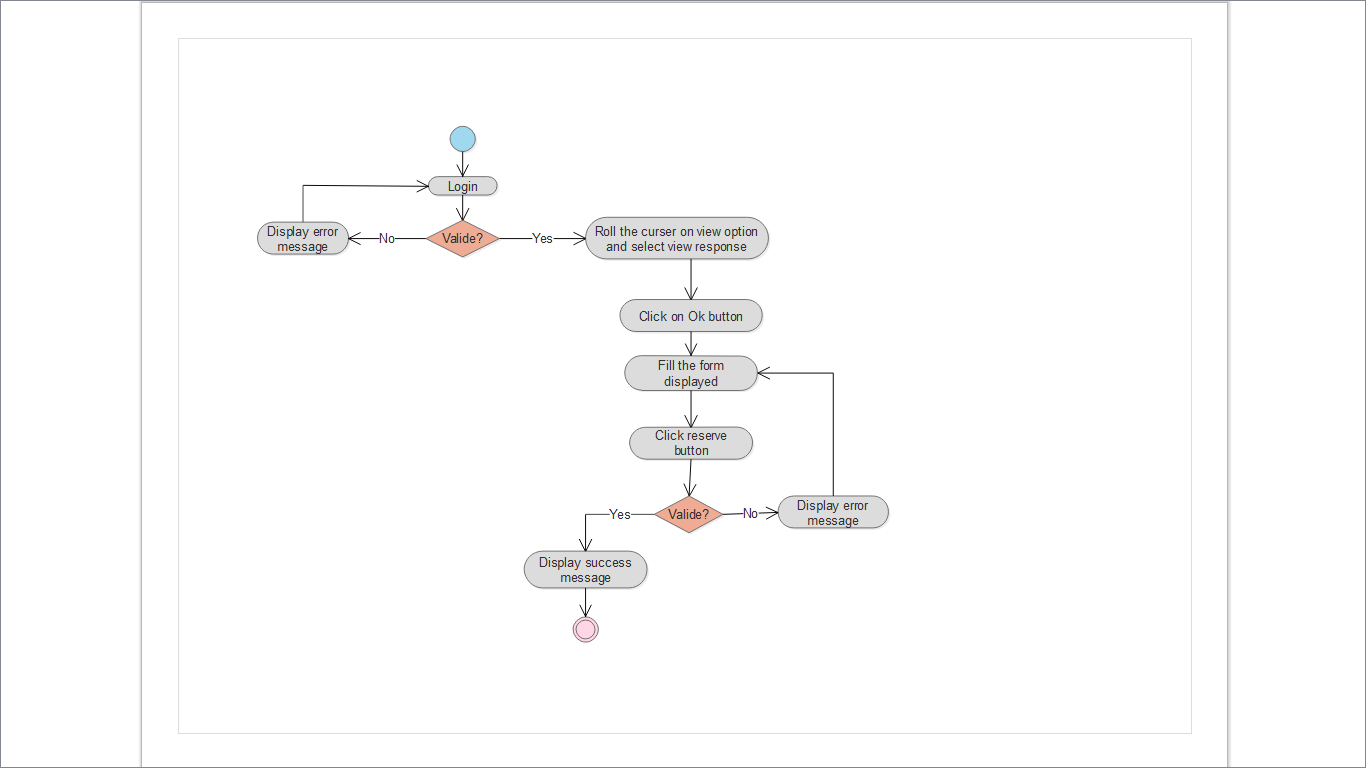


Figure 24: Activity Diagram for Reserve House

### *3.2.5 Analysis Class Diagram*

Our UML analysis level class diagramis representation of meaningful concepts in a problem domain. Itfocuses on domain concepts(i.e., ideas, things, or objects), associations between concepts and attributes of concepts not software entities.

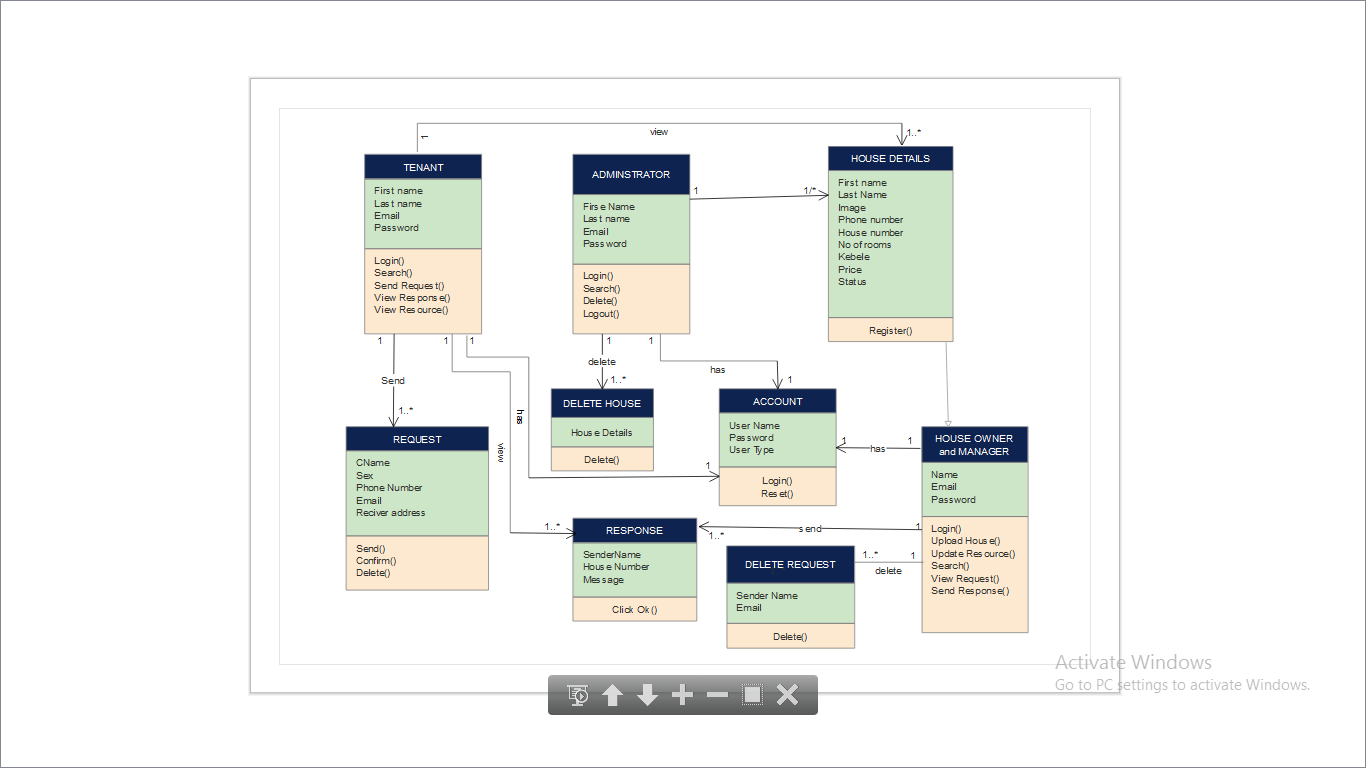


Figure 25: Analysis Level Class Diagram for Online House Rental system

### *3.2.6 User Interface Prototype*

User interface prototyping is an iterative development technique in which users are actively involved in the user interface of a system. It is place where user directly interacts with systems. It represents the general ideas behind the UI, but not the exact details. According to these criteria here is user interface design for online house rental system.

**HOME**

**Help**

**View Reserved House**

**Search Houses**

**View House Owner Resources**

**Create account**

**About Us**

**Login**

**Tenant**

**House owner**

**Administrator**

**Change password**

**View response**

**Send Request**

**Feedback**

**Reserve**

**Logout**

**Change Password**

**Logout**

**View**

**Delete**

**House Owner**

**Tenant**

**House**

**Request**

**Feedback**

**View responses**

**Change Password**

**House Upload or Room**

**Logout**

**View**

**Update House**

**Feedback**

**View Request**

Figure 26: User Interface Prototype for Online House rental system

### *3.2.7 Supplementary Specifications*

Adama city house rental system has the following supplementary specification for the proposed system:

* The design of the pages should be self-descriptive, which means it should not mislead the user from using it correctly.
* Color used as background should be user friendly, do not harm the users’eyes.
* The system doesn’t work for those haven’t valid account.

Chapter Four System Design

## 4.1 Introduction

System design defines the architectures, components, modules, and data base for a system to satisfy specified requirements. It is concerned with how the data is represented and stored within the system. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. It is related to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified /authenticated, how it is processed and how it is displayed.

## 4.2 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.

Design goals:-

* Response time: taking less time for system to respond to the request.
* Simplicity: we develop the new system can make user friendly graphical user interface.
* Reliability: the system must perform its intended functions and operations in a system's environment without experiencing failure or system crash.
* Security: the system must be protected from an authorized access, threats, attacks and vulnerabilities.
* Fault tolerance: the system must have the ability to satisfy requirements despite failures such as hardware, software or network failures.

## 4.3 System Design Architecture

Online house rental system is developed based on two components: the server side and the client side, which runs on the browser. The server is a set of applications that serve the requests made by clients. Data servers are responsible for managing tabular data at the server side and sending the information to the client. This system is viewed in any standard web browser. The performance of the system is measured by the ability to respond to requests faster, and the reliability of the system. The diagram below illustrates a general view of the system design architecture.

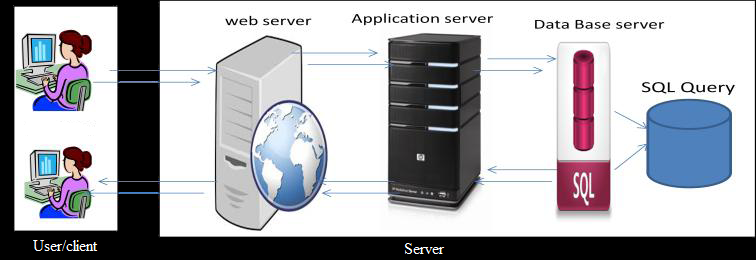


Figure 27: System Architecture for Online House Rental System

As shown above our system is 3-tier architecture. Here the client application no longer directly queries the database; it queries the server application, which in turn queries the data server. The advantage of this system is, when users retrieve data the following happens: The system users on browser asks the web server, the web server asks the server application and the server application queries the data server, the data server serves up a record set with all the information’s available. The server application does all the processing to determine the data and serves up the final data to the web server. The web server displays the final data information to the system users. The three tiers in three-tier architecture are:

* Web server known as Presentation Tier**:** it interacts closely with the user. Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the client and other tiers in the network.
* Application server known as Application Tier: Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.
* Data server also known as Data Tier: This is the most critical aspect of the application; it is where the user data, operational data and metadata are stored for easy access and retrieval. Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic. Thus, a database is an organized collection of structured data, to serve many applications with minimum redundancy.

## 4.4 Class Type Architecture

A class-type architecture provides a strategy for layering the classes of the system to distribute the functionality of the system among classes. Furthermore, class-type architectures provide guidance as to what other types of classes a given type of class will interact with, and how that inter­action will occur. This increases the extensibility, maintainability, and portability of the systems you create.

It includes:-

* User Interface classes: - implements a major UI element of the online house rental system.
* Control/Process classes:- implement business logic that involves collaborating with several busi­ness/domain classes or even other controller/process classes.
* Business/Domain classes: - implement the concepts related to the business domain focusing on the data aspects of the business objects, plus behaviors spe­cific to individual objects.
* Persistent classes: - encapsulate the capability to store, retrieve, and delete objects permanently without revealing details of the underlying storage technol­ogy.
* System classes: -provide operating-system-specific functional­ity for our applications, isolating your software from the operating system (OS) by wrapping OS-specific features, increasing the portability of our application.

The following figure depicts five-layer class-type architecture for the design of object-oriented software which is described in the above in our system.

**System** (Infrastructure Platform)

**User Interface Class**

**Application Controller**

**Domain(**Business Layer**)**

**Persistence Class (**Data**)**

**Data Source**

Figure 28: Class Type Architecture

## 4.5 Class Modeling

Class modeling is static modeling technique that looks at representation of each entity in a system, showing the attribute and activity of each describes one or more objects in uniform set of attribute and services including a description of how to create new objects in the class. Online house rental system class model show the classes of the system with interrelationship and attributes of the classes. It is used to describe the structure of the system. The classes presented here are abstractions that specify the common structure and behavior of a set of objects in our system. Class diagram helps the system during analysis to represent our conceptual model and used to depict the detail the problem of the system.

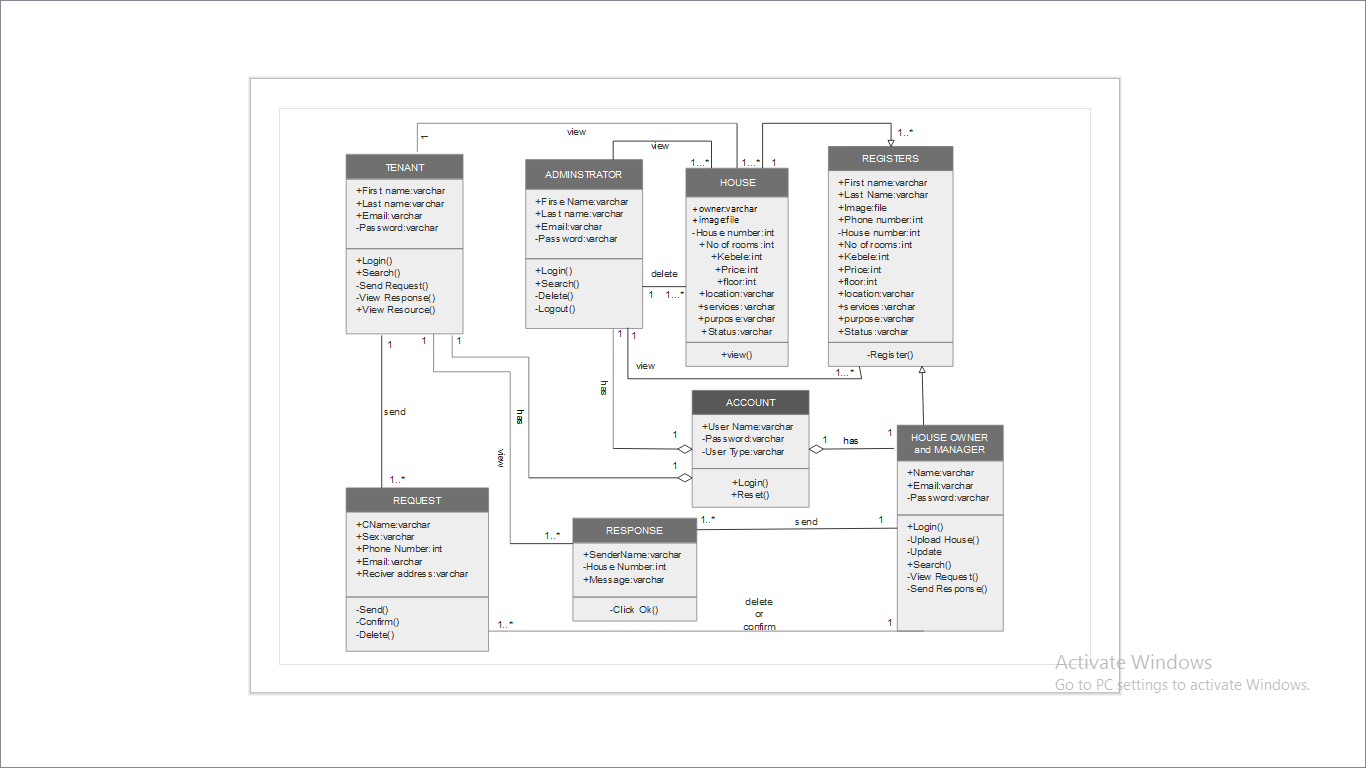


Figure 29: Class modeling for house rental system

## 4.6 State Chart Modeling

The state chart diagram of online house renting system shows the sequence of states that an object goes through the events that cause the transition from one state to the other and the actions that result from a state change.

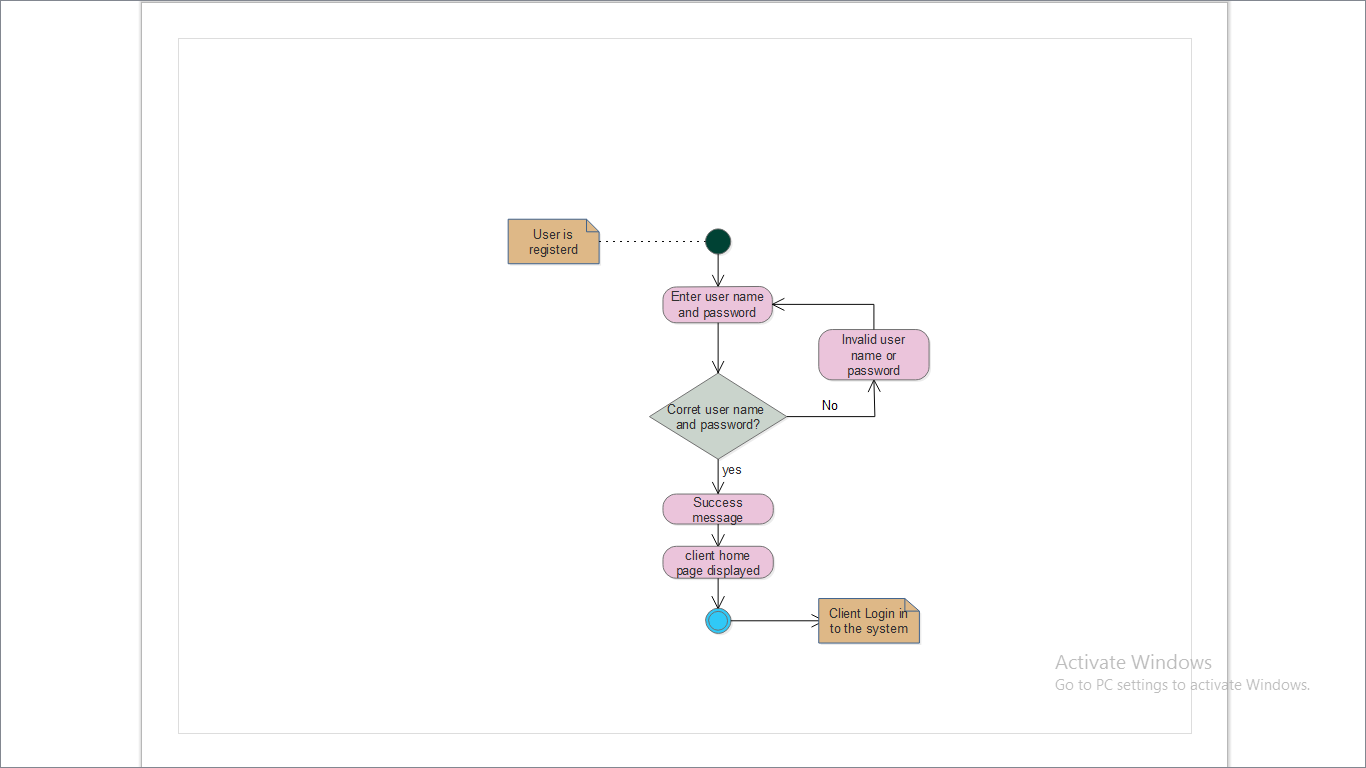


Figure 30: State Chart Diagram for Login

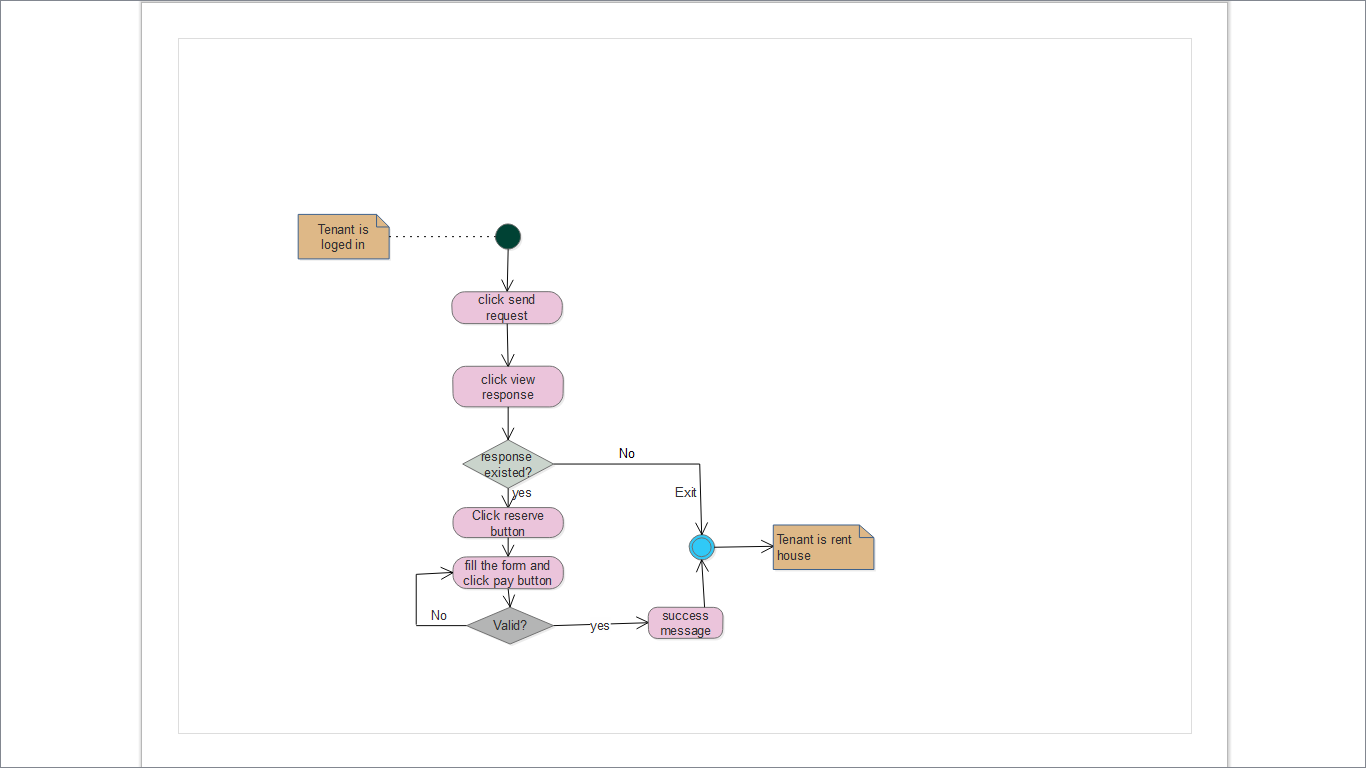


Figure 31: state chart diagrams for reserve the house

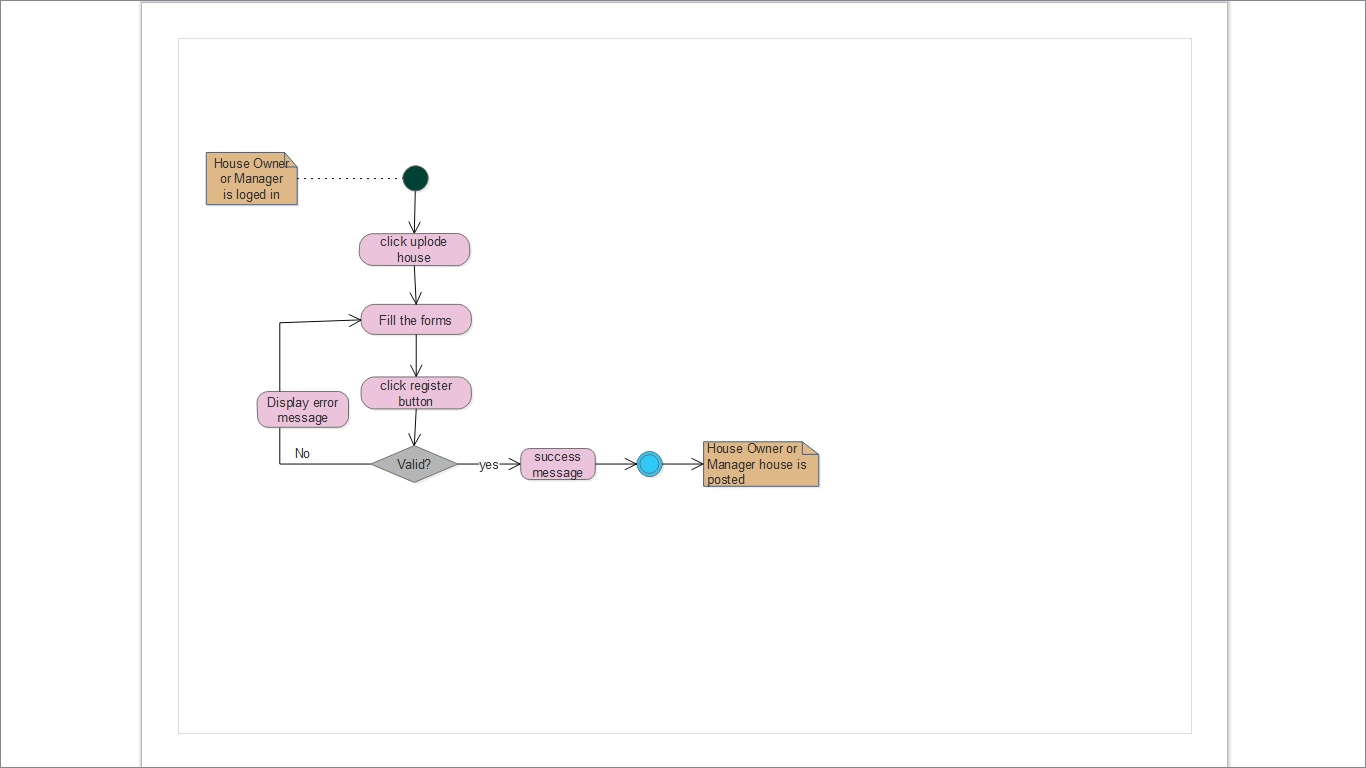


Figure 32: State chart diagrams for Upload house or room

## 4.7 Collaboration Diagrams

A collaboration diagram for online house renting system illustrates the relationships and interactions among its objects.Unlike a sequence diagram, it shows the relationships among the a system objects. On the other hand, a collaboration diagram does not show time as a separate dimension, so sequence numbers determine the sequence of messages and the concurrent threads.

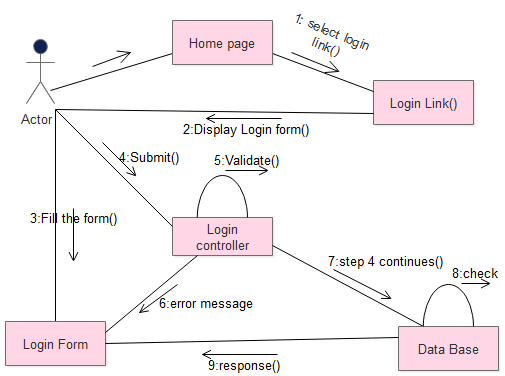


Figure 33: Collaboration Diagram for Login

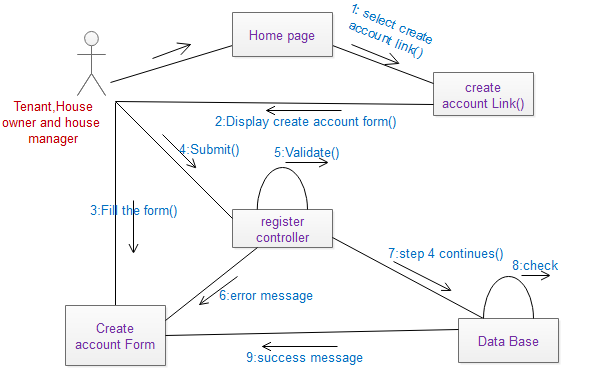


Figure 34: collaboration Diagram for create account

## 4.8 Component Diagram

Online house rental system component diagram show how its Components are wired together to form larger components and or software system they are used to illustrate the structure of arbitrarily complex house renting system. This diagram helps us to model the physical aspect of our system. It illustrates the architectures of the software components and the dependencies between them.

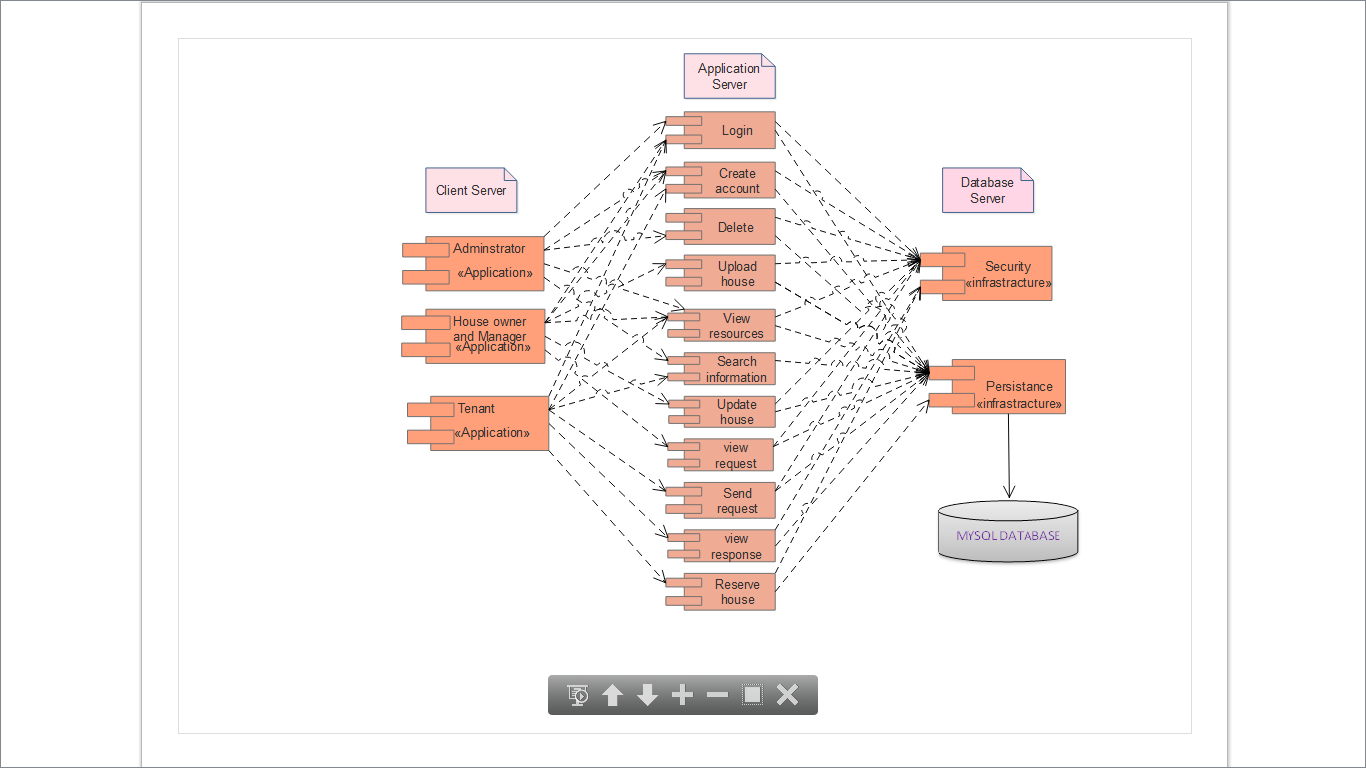


Figure 35: Component Diagram for online house rental system

## 4.9 Deployment Diagram

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.

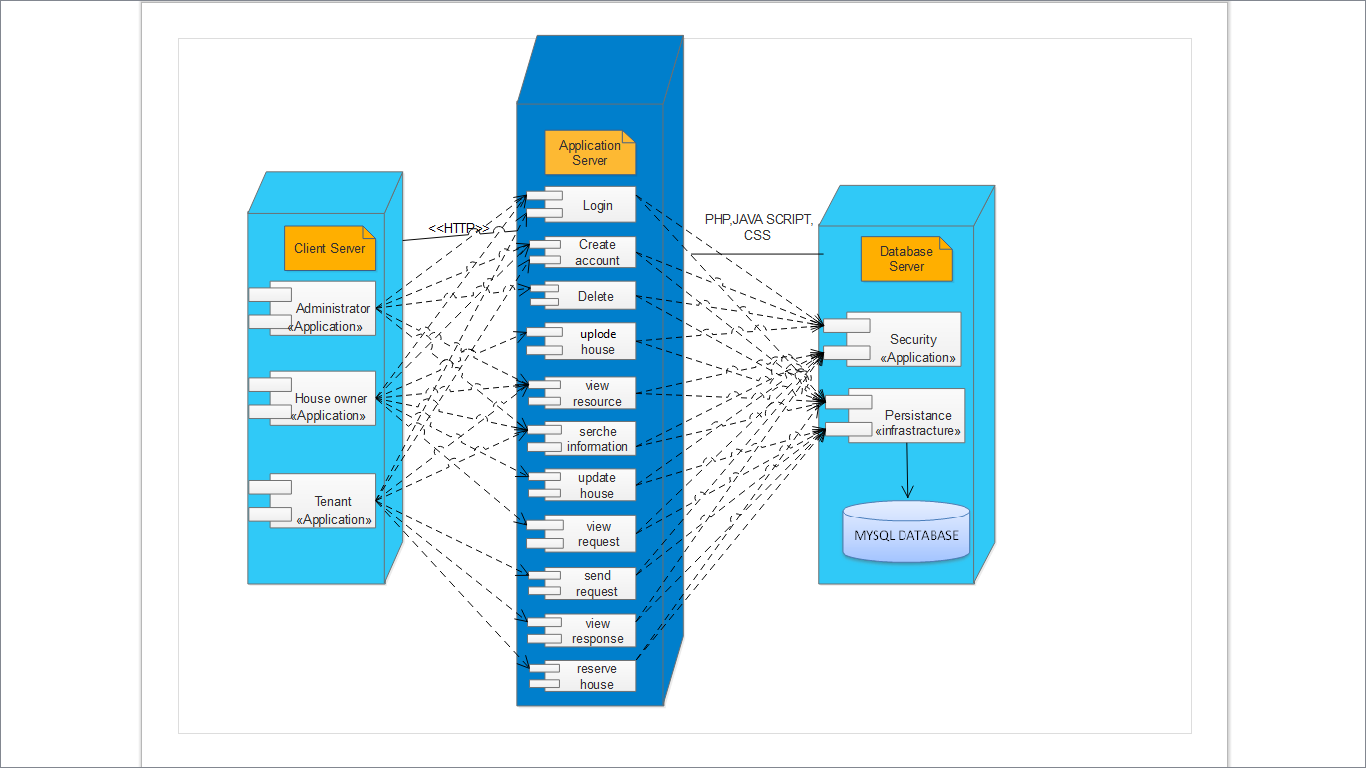


Figure 36: Deployment Diagram for house rental system

## 4.10 Persistence Modeling

Persistence models in our system are used to communicate the design of database. This is basically the entity relation in database application. The design of overall persistent modelingin our system is described at class modeling part. To design the database we will use the table to indicate what our database looks like and we define appropriate field name, data types and length of the input, primary key, foreign key and comment.

The physical data model of a database describes database tables, attributes and relationship between tables. To develop this data model we used ER diagram which designed for to create a common understanding between the developer and end users. Therefore, the ER diagram was mapped to database schema by using different sequential mapping algorithms. After the database schema is created the next step is normalizing the database schema by insert some values to its attribute.

### *4.10.1 Conceptual Database Design*

Once all the requirements have been collected and analyzed, the next step was to create a conceptual schema for the database, using a high level conceptual data model. The result of this phase is an Entity-Relationship (ER) diagram or UML class diagram.

Online house rental Entity Relationship (ER) Diagram is type of flowchart that illustrates how “entities” such as actors, objects or concepts relate to each other within our system. This ER diagram is used to design database in our system. It uses a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

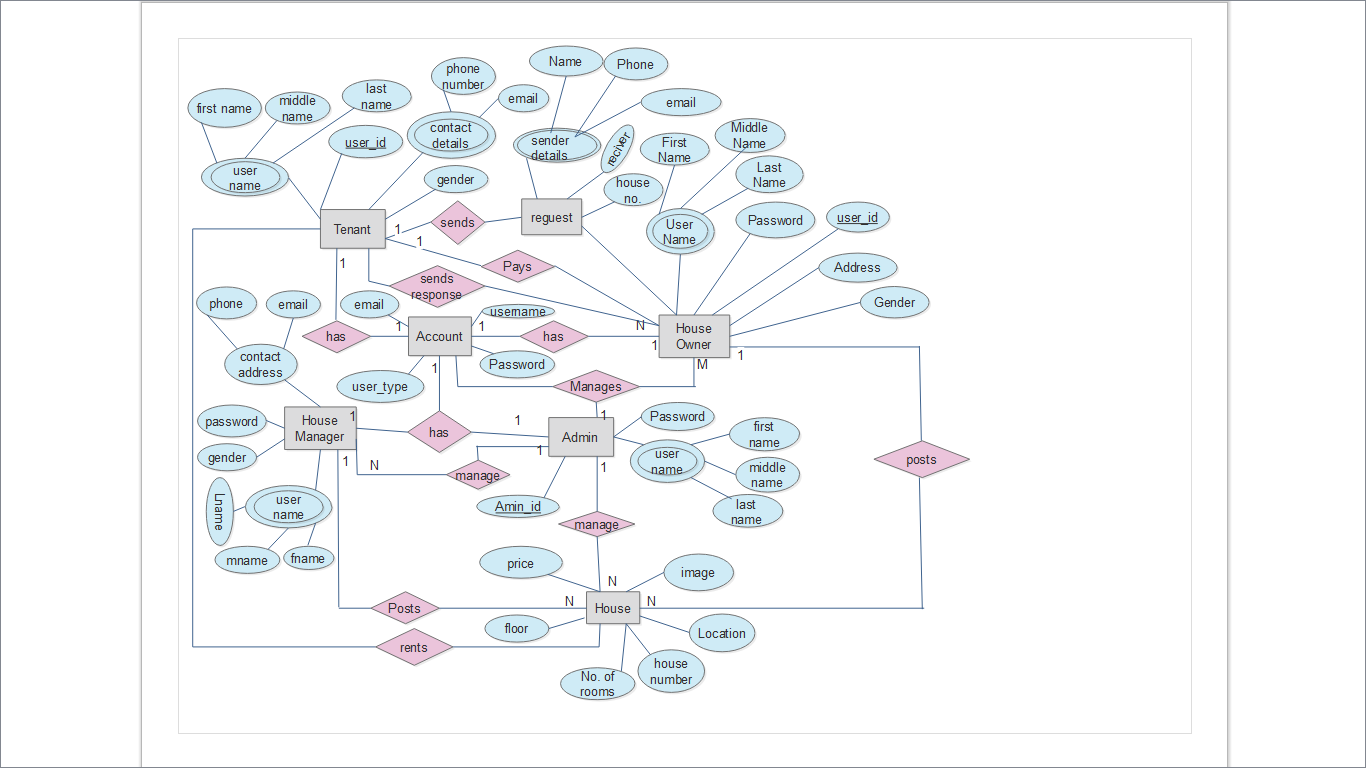


Figure 37: ER diagram for online house rental system

### *4.10.2 Logical Database Design*

The result of this design phase is a set of relation schemas. The ER diagram is the basis for these relation schemas. To create the relation schemas is quite a mechanical operation. There are rules how the ER model is transferred to relation schemas. The relation schemas are the basis for table definitions. In this the primary keys and foreign keys were defined which were not defined in previous phase. By applying mapping algorithm to the above ER diagram the following database schema was created for online house rental system using normalization.

Normalization

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the defined relationships.

1st Normalized Form

1NF is the most basic of normal forms - each cell in a table must contain only one piece of information, and there can be no duplicate rows.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Account | | | | | |
| fname | mname | lname | Email | User\_type | password |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Registers | | | | | | | | | | | | | |
| image | Fname | Lname | Email | Phone\_no | kebele | house\_no | room\_no | price | floor | services | location | purpose | status |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requests | | | | | | |
| first\_name | last\_name | gender | phone\_no | Temail | receiver | house\_number |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Responses | | | | |
| owner\_name | house\_no | message | Temail | date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Payments | | | | |
| Pay\_id | Payer\_name | payed\_to | amount\_payed | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feedback | | | |  |  |
| Name | Email | subject | message | date |

2nd Normalized Form

A database is in second normal form if it satisfies the following conditions:

* It is in first normal form
* All non-key attributes are fully functional dependent on the primary key

For example:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Registers | | | | | | | | | | | | | |
| image | Fname | Lname | email | phone | Kebele | house\_no | no\_rooms | price | floor | services | location | purpose | status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Response | | | | |
| Ownername | house\_no | message | Temail | date |

|  |  |  |
| --- | --- | --- |
| The new introduced view is | | |
| house\_no | Temail | house\_no |

This table is in 2NF because it is

* It is in first normal form
* All non-key attributes are fully functional dependent on the primary key

There must not be any partial dependency of any column on primary key.

3rd Normalized Form

A database is in third normal form if it satisfies the following conditions:

* It is in second normal form
* There is no transitive functional dependency

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Registers | | | | | | | | | | | | | |
| image | Fname | Lname | email | phone | Kebele | house\_no | no\_rooms | price | floor | services | location | purpose | status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Response | | | | |
| Ownername | house\_no | message | Temail | date |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Request | | | | | | | |
| fname | Mname | lname | Gender | phone | Temail | reciver | house\_no |

Look here there is no transition dependency from this tables.

### *4.10.3 Physical Database Design*

The goal of the last phase of database design, physical design, was to implement the database. The physical design of a database describes database tables, attributes and relationship between tables. To develop this database design we used ER diagram in conceptual design. After that, ER diagram was mapped to database schema by using different sequential mapping algorithms. Finally, the database schema normalized by using first, second and third normal forms to create the physical design of the online house rental system database. Look the following figure.

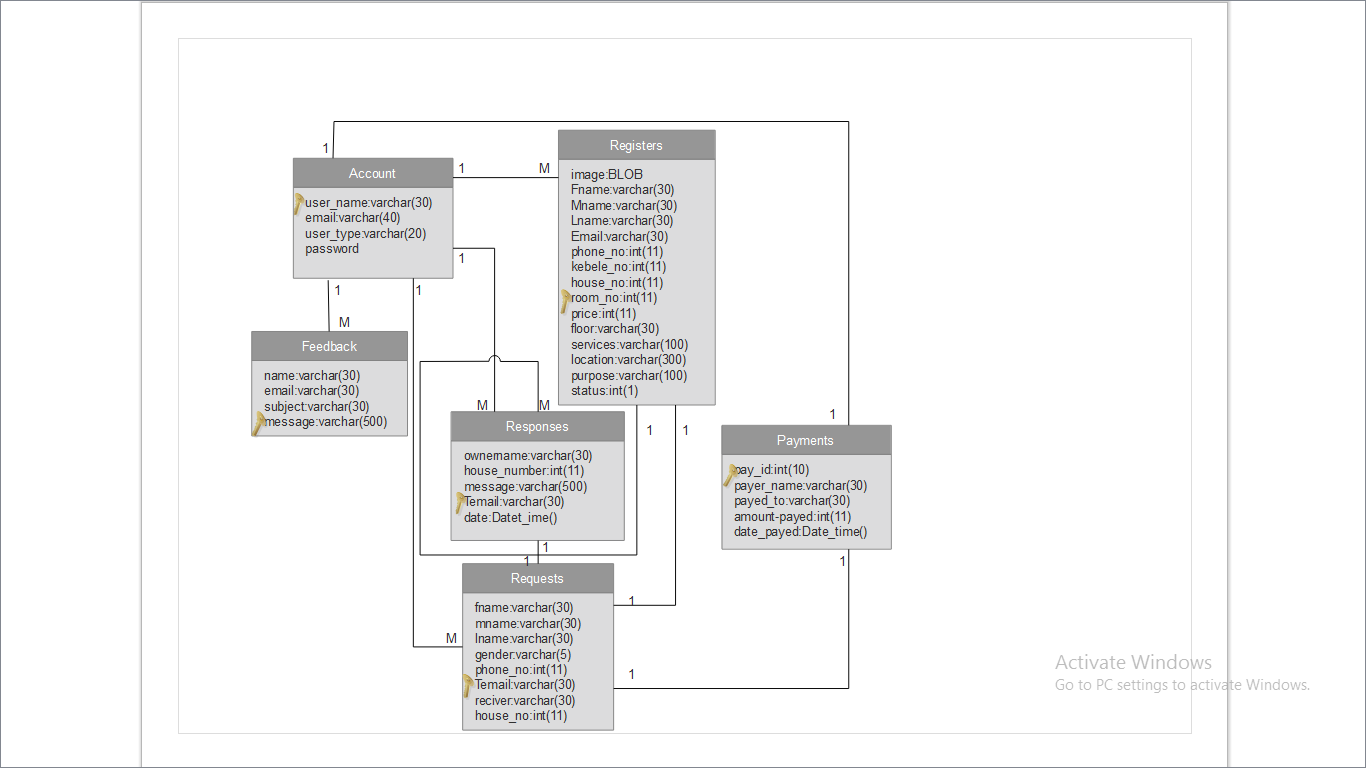


Figure 38: Physical Database Design for Online House Rental System

## 4.11 User Interface Design

User Interface Design of online house rental system is the design of websites and software applications with the focus on the user’s experience and interaction to the system. The goal of this user interface design is to make the user’s interaction as simple and efficient as possible, in terms of accomplishing house renting activities easier. It is user centered design for the system. The following are some user interface designs of our system.

# Chapter Five

# Implementation and Testing

## 5.1 Tools

5.1.1 Software Tools for implementation

There are some tools that we have used for developing the system. These are:-

PHP:we used it as a server side script (back end) program because of it run on different platform like Linux, windows, and on other servers and using integrate database with the site. The main reasons why we selected PHP programming to develop our system described as follows:

* Simple and easy to learn
* User Support
* Easier to fix problems
* Scalability
* Speed
* Portable
* Dynamic computer program

HTML5, JavaScript and CSS: for describing web pages and interface application for users satisfy user interface requirement perform decoration and interactivity with CSS and in order to validate the input information with JavaScript.

HTML5, JavaScript and CSS: Reasons why we selected them are:

* better user experience
* Offline browsing

MY SQL 2005 server: for developing database of the system.

It has main features why we select it as follows:

* Performance
* Reliability

Microsoft Office Word 2010: We used this software for documentation purpose of our project. The main reasons why we choose this software are:

* Found improved search and navigation experience
* Add visual effects to the text
* Add visual effects to the document
* E-draw max UML diagram 7.9:-We used this to draw UML diagrams for our project. The main reason why we choose this software as follows:Simple and clear to create professional looking for different chart and diagram
* Ram ------------- 1 GB
* CPU -------------- 2.8
* Modem and LAN of 512 mbps broad band connection.
* Server computer and client computer.
* Flash disc 8 GB

E-draw max UML diagram 7.9:-We used this to draw UML diagrams for our project. The main reason why we choose this software as follows:

* Simple and clear to create professional looking for different chart and diagram  Flexibility
* Includes high quality shapes, examples, templates  Works with MS office well
  1. Unit testing

The following are main modules of our system

* Register
* Login
* Manage
* Search
* View
* Reserve
* Pay money
  1. Functionality

Register module: is tested to check if it registers correctly.

Login module: tested if actors can login by their account only.

Manage module: is tested if resources and requests are managed correctly.

Search module: is tested to check if it displays the requested information.

View module: is tested to check if it views all the available resources as well as requests.

For example if a house or room owner inputs all the available resources the system displays insertion successful if there is no error otherwise failed to register.

* 1. User Manual

### 5.4.1 System overview

The Online House Rental System (OHRS) is a web application designed utilizes an intuitive user interface that enables house or room owners to create their account, register their house and room into the site to be available for renting to customers, update their own information they posted, and if there is request from customers (tenants) respond to requests; and follow the payment progress online. The system also provides a fast and reliable way for tenants (customers) to look over the site or search for the house with their purpose they are renting for and select the house or room they interested on; it allow them to make requests to the house or room owners, wait for response from the owners; if the owner accepted the request make payments to the owners to reserve the house or room; follow the payment progress online and can also send cancel reservation requests to the corresponding house owner in order to cancel reservation tribes. The admin has control all over the system. This application provides a graphical user interface that allows a user to input all information and store it electronically to eliminate the need for using paper records.

The application uses a client/server based model. The client is what the house or room owners and tenants use to add their records as well as manage them. The client application will communicate with a data base server that saves all the information for each house or room owners and tenants (customers).

## 5.4.2 System Summary

### 5.4.2.1 System Configuration

Each customer (property owners and tenants) and administrator uses their own computer running Windows OS or Linux OS. On those computers is the client application that will connect to a central database running Windows. Computers running OHRS applications are connected on a WAN with no outside connection to the internet. This is because sensitive information such as social security numbers are being sent across the network to control security issues.

OHRS central database server

## WANconnection

## 5.4.2.2User Access Levels

There are three different types of users as depicted in the above diagram; Admin, house or room owners and tenants. Admin has full privileges to alter the data at the backend and control over all system and system users within the database. Tenants as well as house or room owners have permission to view and read partial information about system and use other applications, but cannot change any of them.

### Getting started

Setting Online House Rental System software:

* You first start by running “OHRS -setup.exe” using cd drive or usb drive. This will begin the installation of OHRS Software.
* Select “Typical” installation then click the “Next >” button
* Click the “Install” button to begin installation
* Once the installation completes click “finish >” twice
* Now insure “Configure the OHRS software now”.

## Using Client Application

### 5.4.3.1**Settings**

Changing the settings to fit the current user is the first step that should be taken by a new user to the computer before using the application. The settings can be access from the “Settings” menu in the main application window. There are seven (7) settings that can be changed to fit the current user’s needs. When the settings are changed, the changes will be saved even when the program is restarted. Users may need to contact the network administrator to find out all of the details for these settings.

### *Server*

It is IP address of the computer on the LAN that is running the database server. You may either type the IP address of the database server or the network name of the computer

## 5.4.4 Users (house or room owners and tenants) Information

First the customer information is stored in to your system data base only through registration form. The new user to the system should be clicking the “No account? Sign up” link and insert all needed information through the given fields in to the system data base. After he or she was successfully registered, he/she begins to carry out all the tasks mentioned in over view of the system above.

## Administrator information (Activities)

* **Manage Users**

The manager can control over system users by activating and deactivating system users’ account.

* **House or room Delete**

If there is the delete house or room request from the owners he can delete the house or the room.

* **Manage account**

Admin also can create users account.

* **Generate report**

The administrator can generate report consisting of number of posted houses, rooms, system users, currently reserved houses or rooms, available houses or rooms for renting tenants and the number of house or room owners .

**View information**

Administrator can view all responses and reservations made between system users.

# Chapter Six

# Conclusion and Recommendation

# 6.1 Conclusion

The online house rental system is centralized system it can be used to reduce resource lose and to increase the efficiency. The main focus of this project is to satisfy users need for easy finding property, to maximize property choice, to save time and money .The maintenance of the records is made efficient, as all the records are stored in the database, and also data can be retrieved easily. The editing is also made simpler. The user has to just type in the required field and press the update button to update the information.

# 6.2 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 and others to fulfill users’ requirements. For the time being we did not do those functionalities due to lack of time.

# References

1. Renting a house, [www.rentals.com](http://www.rentals.com), Oct15, 2014
2. Renting a car, [www.car](http://www.car) rent.com, Oct15, 2014
3. Tenantbackground,<http://ohmyapt.apartmentratings.com/what-does-a-tenant-background-check-cover.html#sthash.0bNJWyRM.dpuf>,Oct17,2014
4. Introduction to tenants, [www.homes.com](http://www.homes.com), Oct29, 2014
5. Landlord’s resource, [www.wikihow.com/rent-your-house](http://www.wikihow.com/rent-your-house), Nov5, 2014
6. Rent system, <https://www.rentalsystem.com/site/register>, Nov5, 2014
7. Programming language, [www.w3school.com](http://www.w3school.com), Dec25, 2014