Wollo University

Kombolcha Institution of Technology

College Of Informatics

Department Of Information Technology

Final Year Project Documentation

Title: - Online Hotel Management System for Lucy Hotel

Prepared By: 1. Temesgen Meles

2. Yadelew Kebede

3. Selamawit Tilaye

Adviser's Name: - Molla Tesfaye

Kombolcha, Wollo, Ethiopia

July,2014 E.C

Declaration

This is to declare that this project titled online hotel management system for Lucy hotel is done by 4th year Information Technology students for partial fulfillment of BSc in Information Technology. It is submitted to Department of Information Technology.

Approval sheet

Examining committee members Signature Date

1. Examiner 1 ______________________________ __________ __________

2. Examiner 2 ______________________________ __________ __________

3. Examiner 3______________________________ __________ __________

4. Examiner 4______________________________ __________ __________

Advisor's name Signature Date

Molla Tesfaye ____________ ______________

Group members Signature Date

1. Temesgen Meles _____________ ______________
2. Yadelew Kebede _____________ ______________
3. Selamawit Tilaye _____________ ______________

Acknowledgement

First of all, we want to say thanks for our God who made all things good. Secondly thanks for our adviser, Mr. Molla for continuous support in every phase of the project and who continuously provides us his valuable advice to prepare this project document. Also, thanks to Lucy hotel workers who give for us a correct requirement about the existing system. Lastly, we would like to acknowledge all our friends for their morals, idea, materials and financial support during the courses of the years.

Abstract

This proposed system explains about Online Hotel Management System which we would develop for Lucy hotel. Currently the services that are given in Lucy hotel is mostly handled manually. So, to solve this manual system this Online Hotel Management System is proposed. This system aims at creating an Online Hotel Management system which can be used by customers to reserve hotel rooms, order meals and hire out meeting hall. Users can check the availability of rooms, meals, meeting hall and other facilities. The users can register and log into the system. The receptionist will know the details of reservations and customer order and doing other activities. This system allows a simple communication between the customer and the hotel manager. Therefore, this proposed system is more efficient and effective than the existing system, because the proposed system will done in online. To develop this proposed system, the team member will use HTML, PHP, CSS and JAVA SCRIPT to write the code, and also WAMMP server to compile (run) server-side code.

List of Acronyms

AD	Activity Diagram
CRC	Class Responsibility Collaboration
CSS	Cast Style Sheet
DB	Database
GB	Gigabyte
HTML	Hypertext Markup Language
ID	Identification
JS	JavaScript
PC	Personal Computer
PHP	Hypertext Preprocessor
RAM	Random Access Memory
SD	Sequence Diagram
SQL	Structure Query Language
UC	Use Case
UI	User Interface
UML	Unifying Modeling Language

Table of Contents

Declaration..i

Acknowledgement..ii

Abstract..iii

List of Acronyms..iv

List of Table..viii

List of Figure..ix

Chapter One: Introduction..1

1.1 Background of The Study..1

1.2 Statement of The Problem..2

1.3 Objective..2

1.3.1 General Objective..2

1.3.2 Specific Objectives..2

1.4 Methodology..3

1.4.1 Data Gathering Technique..3

1.4.2 System Methodology..3

1.4.3 Tools Used for Implementation..4

1.4.4 Testing Methodology..4

1.5 Feasibility Analysis..5

1.5.1 Technical Feasibility..5

1.5.2 Operational Feasibility..5

1.5.3 Economic Feasibility..6

1.6 Risks and Assumptions..6

1.7 Scope and Limitation of the Project..6

1.7.1 Scope of The Project..6

1.7.2 Limitation of the Project..7

1.8 Significance and Beneficiaries of The Project..7

1.8.1 Significance of The Project..7

1.8.2 Beneficiary of The Project..7

1.9 Time Line of The Project..8

1.10 Budget of The Project..9

1.11 Organization of The Project..9

Chapter Two: Analysis..10

2.1 Introduction 10

2.2 The Existing System 10

2.2.1 Description of Existing System 10

2.2.2 Requirement gathering 11

2.2.3 Supplementary Requirements 11

2.3 The Proposed System 12

2.3.1 Overview of the Proposed System 12

2.3.2 Software requirement specification (SRS) 12

2.3.3 Existing System Modelling 14

2.3.4 Domain Modelling with Class Responsibility Collaboration (CRC) 15

2.3.5 System Modeling 17

2.3.6 Key Abstraction with CRC Analysis 28

2.3.7 Sequence Diagram 31

2.3.8 Activity Diagram 42

2.3.9 Conceptual Modelling: Class Diagram 51

2.3.10 User Interface Prototyping 53

2.3.11 Identifying change cases 55

CHAPTER THREE: SYSTEM DESIGN 56

3.1 Introduction 56

3.2 Purpose and goals of design 56

3.3 Class Modelling Diagram 57

3.4 Proposed software architecture 59

3.4.1 System architecture 59

3.4.2 Subsystem Decomposition 59

3.4.3 Component diagram 62

3.4.4 Deployment diagram 64

3.4.5 Persistence Modelling 66

3.4.6 Access Control and security 68

3.5 **User-Interface Design** 70

CHAPTER FOUR: IMPLEMENTATION AND TESTING 74

4.1 Implementation 74

4.1.1 Sample code for login: 74

4.2 Choose Your Test Approach 82

4.2.1 Functional Test Specifications 83
CHAPTER FIVE: CONCLUSION AND RECOMMANDATION 85
5.1 Conclusion..85
5.2 Recommendations 86
Reference..87

List of Table

Table 1. 1 Risk and assumption........6
Table 1. 2 Budget of the Project........9
Table 1. 3 CRC analysis of the existing system for Manager........15
Table 1. 4 CRC analysis of the existing system for Receptionist........16
Table 1. 5 CRC analysis of the existing system for Customer........16
Table 1. 6 Use case description for login........19
Table 1. 7 Use case description for Create Account........20
Table 1. 8 Use case view reports........21
Table 1. 9 Use case description for reserve room........22
Table 1. 10 Use case description for order meal........23
Table 1. 11 Use case description for add meal item........24
Table 1. 12 Use case description for delete user Account........25
Table 1. 13 Use case description for add room........26
Table 1. 14 Use case description for Logout........27
Table 1. 15 CRC standard index cards........28
Table 1. 16System Administrator CRC........28
Table 1. 17 Manager CRC........29
Table 1. 18 Receptionist CRC........29
Table 1. 19 Chef CRC........30
Table 1. 20 Customer CRC........30
Table 1. 21 System access control........69
Table 1. 22 Test case specification........84

List of Figure

Figure 1. 1 Gantt chart for Time Schedule 8
Figure 1. 2 Essential use case diagram for the existing system 15
Figure 1. 3 Use Case Diagram 18
Figure 1. 4 Sequences diagram of login into the system 31
Figure 1. 5 Sequences diagram for create account. 32
Figure 1. 6Sequences diagram for generate report. 33
Figure 1. 7Sequences diagram for reserve room. 34
Figure 1. 8 Sequences diagram for meal order. 35
Figure 1. 9 Sequences diagram for add meal 36
Figure 1. 10 Sequence diagram for leave comment 37
Figure 1. 11 Sequence diagram for Add room 38
Figure 1. 12 Sequence diagram for search customer information 39
Figure 1. 13 Sequence diagram for delete user account 40
Figure 1. 14 Sequence diagram for logout 41
Figure 1. 15 Activity diagram of login to the system 42
Figure 1. 16 Activity diagram for create account 43
Figure 1. 17 Activity diagram for generate report 44
Figure 1. 18 Activity diagram for room reservation 45
Figure 1. 19 Activity diagram for Meal order 46
Figure 1. 20Activity diagram for give comment 47
Figure 1. 21 Activity diagram for Add room 48
Figure 1. 22 Activity diagram for View customer information 49
Figure 1. 23 Activity diagram for delete user account 50
Figure 1. 24 Activity diagram for logout 51
Figure 1. 25 Conceptual Class Modelling diagram 52
Figure 1. 26 User Interface Prototype 53
Figure 1. 27 Login form prototype 54
Figure 1. 28 Customer Registration form prototype 54
Figure 1. 29 Class modeling diagram 58
Figure 1. 30 System architecture 59
Figure 1. 31 System Decomposition Diagram 61
Figure 1. 32 Component diagram 63
Figure 1. 33 Deployment Diagram 65
Figure 1. 34 Persistence modelling 67
Figure 1. 35 user interface home page 70
Figure 1. 36 *customer login page* 71
Figure 1. 37 *staff login page* 71
Figure 1. 38 *view reservation for reception* 72
Figure 1. 39 *view order information for chefs* 72
Figure 1. 40 *manage room information in reception page* 73

Chapter One: Introduction

1.1 Background of The Study

A ***Hotel*** is an establishment that provides paid lodging on a short-term basis. Facilities provided inside a hotel room may range from a modest-quality mattress in a small room to large suites with bigger, higher-quality beds, a dresser, a refrigerator and other kitchen facilities, upholstered chairs, a flat screen television, and bathrooms. Small, lower-priced hotels may offer only the most basic guest services and facilities. Larger, higher-priced hotels may provide additional guest facilities such as a swimming pool, business centre (with computers, printers, and other office equipment), childcare, conference and event facilities, tennis or basketball courts, gymnasium, restaurants, spa, and social function services [1].

Hotel Management System is a type of properly management system that facilitates the management of hotel operations and functions.

Lucy Hotel is one of the biggest 4-star hotel in the heart of Dessie, which is a city and a zone in north-central Ethiopia. Located in the Amhara Region, it sits at a latitude and longitude of, with an elevation between 2,470 and 2,550 metres above sea. Dessie is 400 km to the north of the capital Addis Ababa. It has a population of more than 200,000 people in over 30 districts [2].

Lucy Hotel is established in 2013 E.C by Mr. Eshetu Hassen. It consists of 130 employees and this hotel offers many services for customer such as bed rooms (5 luxurious Residential suite rooms, 15 Deluxe suites and 45 standard comfortable rooms), 24/7 open 3 restaurants, top floor VIP lounge, reception bar, basement nightclub, spa and fitness centre, shared swimming pool, green outdoor entertainment multi-functional meeting & event halls, ATM services, Free internet access, Free airport shuttle, and safe parking services. As the hotel vice manager Mr. Kidus says the hotel uses a software called CNET for managing some of the hotel activity, and there is a simple hotel information system with booking feature that gives information about the hotel. Sometimes the customer reserves the bedroom by using phone calling. Even though, this is not effective for the hotel and also for the customer, because the user can’t reserve meeting(event) halls online and also can’t order meals. Because of those problem, we want to develop an online system to fix the problem that is existing in Lucy hotel.

1.2 Statement of The Problem

Lucy Hotel uses the CNET software to manage and control the hotel activity. This existing system has a number of problems in the working procedure for the hotel workers (manager, receptionist, and casher, customers etc...**).** Records are managed manually in the software so it is very hard to maintain the records systematically. There is no data security in the present system (since, any user can view and change the data easily). Difficult to generate report (because, reports are generated manually using this software). The system is too difficult for users to do some operations (Update, Search, Delete, Edit) because to do those operation the user need to use phone calling or face to face communication. May be the user can get hotel information easily, user can reserve bedrooms online or using phone call, but users can't order meals online and also users can't hire out meeting hall in simple way. The system is not giving more satisfaction for the hotel and also for the customer. The other problem in this existing system is, there is no simple and clear communication between the hotel manager and the customer. I.e., When the customer wants to leave or give comment, the customer needed to go to the manager and communicate each other (face to face communication) or by using phone calling.

1.3 Objective

1.3.1 General Objective

The general objective of this project is to design and develop **online hotel management system for Lucy hotel** located in Dessie Ethiopia.

1.3.2 Specific Objectives

In addition to our general objective the project will also contains the following specific Objective: -

- Gathering information about the hotel and the system it uses
- Identify and analysis the problems of the existing(current) system.
- Identify functional and non-functional requirements for the new system.
- Identify system requirement (hardware and software requirements) for the new system.
- Build system design and object design for implementation.
- Build the front-end of the website based on the design by using HTML, CSS, and JavaScript language.

- Build the back-end of the website using PHP language.
- Design a database to manage the information in all over the hotel.
- Test the system validity using some testing methodology

1.4 Methodology

1.4.1 Data Gathering Technique

To design and develop the system, we used the following methods to gather information about the current system.

By Observation:

The reason why selecting the observation is to gather information as it gives direct (primary) information. This is used to analyses the situation in which the current system is acting.

By Interview:

By interviewing hotel manager and customers, we have a well-informed understanding about the current system. This helps to assure the information gathered from observation.

1.4.2 System Methodology

To design the system the project team has choose Object Oriented System Development methodology.

We choose object-oriented development design because of: -

It Improved our software-development productivity: object-oriented programming provides improved software-development productivity over traditional procedure-based programming techniques because of these three factors modularity, extensibility, and reusability.

- Modularity (Inheritance)- object-oriented programming is modular, as it provides separation of duties in object-based program development.
- Extensibility- as objects can be extended to include new attributes and behaviors.
- Reusability- objects can also be reused within an across applications.

1.4.3 Tools Used for Implementation

To implement the system, we used hardware and software tools.

Hardware Tools:

To implement our document, the team members used the following hardware devices(tools).

- Computer with minimum of 4GB RAM
- Flash disk-To hold data at least 4GB
- Printer- To have a hard copy of the data

Software Tools

Software is a program that enables a computer to perform a specific task. To implement our document, we used the following software.

- Apache Web server (WAMP server): is software that will use to run Server-side application.
- Visual Studio code: Used to write the specific code.
- Microsoft office word 2019 - for writing documentation
- Microsoft PowerPoint 2019 - for presentation
- Windows 10 Operating system
- Microsoft Visio 2016: is software which we used to draw modelling diagram. Such as sequence diagram, activity diagram, class diagram.
- Bootstrap and Bootswatch- for cascade style sheet (CSS) framework.
- Web browser: the web browser such as Mozilla, Internet explorer and Google chrome use to see our system.
- CSS (to style the user interface), HTML (to make user interface) and JavaScript (used to validate user entry in the form).
- PHP: hypertext pre-processor language. It will use to implement server-side sub system of the system.

1.4.4 Testing Methodology

Testing is the process of validating and verifying programs or applications. Testing is used test the validity of system by giving real data and wrong data. When the valid data is added to the system it functions as planned. However, when invalid or wrong data is added to the system it returns error message.

Among those testing methodology the following are explained as: -

Unit Testing

It tests that verify the functionality of a specific section of code. In this testing each data in the database is tested through form with a specific data. It uses to catch errors in the form. Each module is tested alone in attempt to discover any error in its code. In this phase all the drivers are willing to test they are rightly connected or not.

Integration Testing

All we want to test modules are combined into sub systems, which are then testing. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is we will test.

Acceptance Testing (Alpha and Beta Testing)

- **Alpha Testing**: When our system has been completed, we have examined it by actual group members and an independent test team and test the system by himself whether it meets their need or not.
- **Beta Testing**: The system would have been tested by the users at their own working place whether it meets their needs or not. (Need permission from the hotel)

1.5 Feasibility Analysis

Project feasibility deals how beneficial the project could be to the organization that it is being developed from the point of perspectives such as technical, economical and operational feasibility.

1.5.1 Technical Feasibility

In this system we will use languages such as HTML, PHP, Java script and CSS to develop the new system. All these are the technology side and once the module is developed it can be easily held by non-technical person, it doesn’t require any technical expertise.

1.5.2 Operational Feasibility

To measure how much the proposed system solves the existing systems problems. This project is surely operationally feasible because of the following points: -

- The proposed system is a good solution maker of the problem or specific solution will work in the existing system.
- The proposed system creates a good environment towards the user of the system.

1.5.3 Economic Feasibility

The proposed system has tangible and intangible benefits.

Tangible Benefits:

Those benefits that are easily quantified from our system. These are:

- Reduce the cost
- Decrease response time

Intangible Benefits:

Those benefits that are believed to be difficult or impossible to quantify from our system are:

- Improving resource utilization and control.
- To increasing security of data and information accuracy.
- To give better services to customers.
- User satisfaction.

1.6 Risks and Assumptions

NO	Risk	Impact	Probability	Mitigation
1	Attacking a computer by a virus	computer by damaging programs, deleting files, and replicate themselves.	likely	Update the pc/computer and install antivirus software on the pc.
2	Shortage of time	increased stress within my group team.	Very likely	Does not consume any time.

Table 1. 1 Risk and assumption

1.7 Scope and Limitation of the Project

The implementation of this project is on online hotel management system for Lucy hotel. In this section the team identify the scope and limitation of the project.

1.7.1 Scope of The Project

- Providing online room reservation.
- Provide online meal order
- Provide online soft drink, and alcoholic drink order.
- The proposed system can allow the customer give comment to the hotel manager in different aspects.

- The system allows the manager replays the comments to the customer.
- The system administrator manages employee information.
- The system allows the system receptionist to add and delete room.
- The system can generate report and the manager can view the report.
- The system allows customer to hire out meeting hall.
- The system allows customer to visit Hotel information on the web site.
- The system allows the receptionist to cancel reserved room after the customer is leaving from the hotel.

1.7.2 Limitation of the Project

- The system uses only one language (only English)
- The system doesn't work without an internet connection (doesn't work offline).
- The system isn't prepared for blind person.

1.8 Significance and Beneficiaries of The Project

1.8.1 Significance of The Project

The significance of the project means the important role of the project to all the hotel employee and the users. Some of significance of the projects are listed below: -

- To reduce time taking for the customer.
- Minimize manpower.
- It reduces working load of the employee (staff members).
- Reduce the complexity of data arrangement.
- Increases accuracy.
- Secure the hotel resources.
- Provide efficient and quick access service to the customers.
- It is able to get fair order of the customer.
- Reduce the expenditure of the owner by reducing manpower.
- It is very simple to use and order the users need.
- It is very important to store user information unambiguously

1.8.2 Beneficiary of The Project

Customers: The customer can gain fast service, because of this new online system. The customer can reserve bedroom, hire out meeting hall, order meals and other service online.

Employee: It is used for the employee. Since, it reduces workload and the employee can't be confusing when they serve the customer order.

The Hotel: It is more profitable by increasing the number of customer because, the hotel will have simple and efficient service, and it reduce expenditure by decreasing the number of employees. Because the system by itself will solve some problem that is covered by some staff members in the existing system.

1.9 Time Line of The Project

Schedule describes the time frame given for every activity and making measurement whether activity Completion date can be met or not. To develop the whole system, we need at minimum six months' time duration.

40 days for requirement analysis

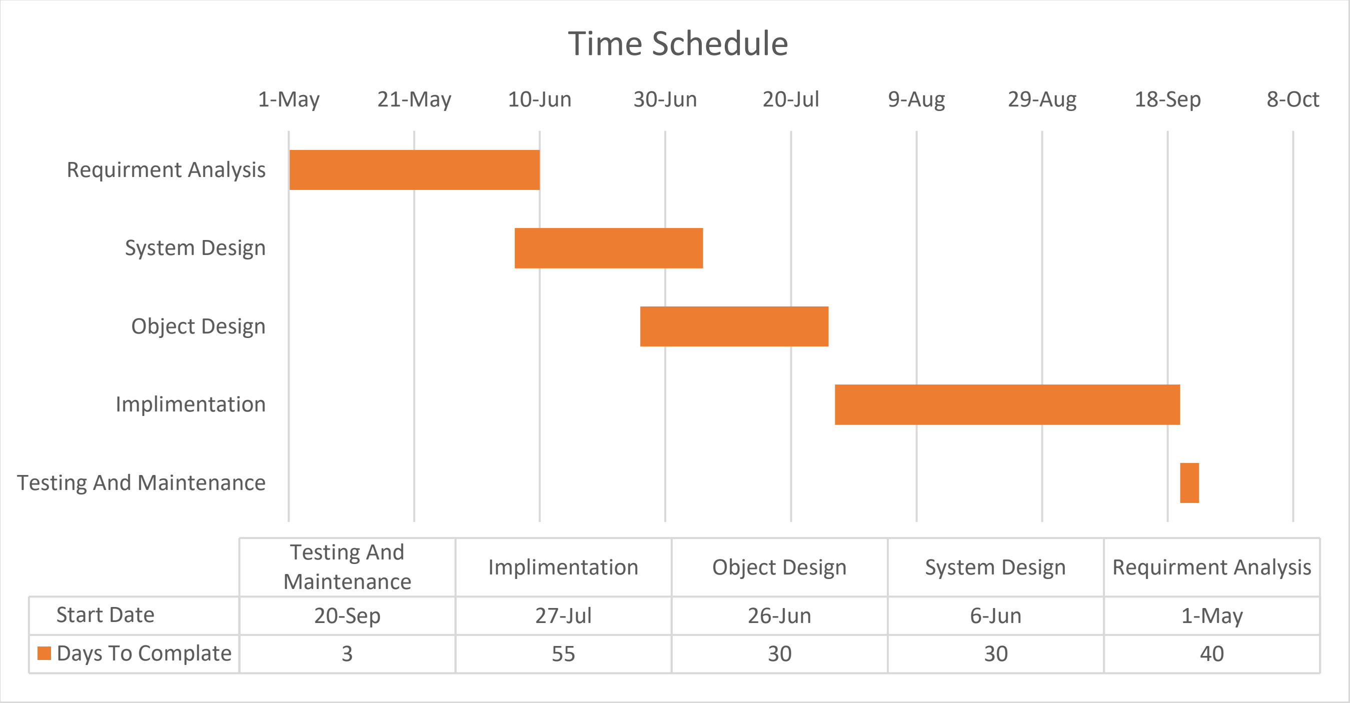
30 days (one month) for system design

30 days (one month) for object design

55 days (two months) for implementation

3 days for testing and maintenance

Generally, it is presented in the graph below:



	Testing And Maintenance	Implimentation	Object Design	System Design	Requirment Analysis
Start Date	20-Sep	27-Jul	26-Jun	6-Jun	1-May
■ Days To Complate	3	55	30	30	40

Figure 1. 1 Gantt chart for Time Schedule

1.10 Budget of The Project

This is deciding the cost of the entire product after analyzing the requirements cost estimated. While developing the project, the cost is not an exact amount.

Item	Quantity	Price per Item	Total Price
Paper	100	1	100
Pen	5	15	75
Mobile card	5	50	250
Color Print	80	15	1200
USB flash	1	350	350
Transport	3	180	540
Total			2,515

Table 1. 2 Budget of the Project

1.11 Organization of The Project

This project document is organized into Two chapters. Chapter one contains an introduction, background of organization, statement of the problem, objective of the project, methodology, feasibility of the project, significant of the project. The second chapter of the project document contains the system requirement analysis part of the system, which shows the activities that will be done in the system functional requirement and non-function requirement, an overview of the proposed system, and modeling the existing system (essential use case modeling, essential user interface prototyping, essential user interface prototyping flow diagram).and also system modeling which contains System use case diagrams. Use case documentation for each use case identified sequence diagram, activity diagram, analysis level class diagram (conceptual modeling), and user interface prototyping.

Chapter Two: Analysis

2.1 Introduction

Analyzing the existing system ensures that a new system which would deliver efficiency, effectiveness, accuracy, reliability and accuracy in the management of a hotel's affair will be designed and implemented. However, the aim of a comprehensive and thorough analysis of the operation of an existing system is to get the required data that will assist tremendously in the design of a new system.

2.2 The Existing System

2.2.1 Description of Existing System

The organization in which the project is applied on, is Lucy Hotel. The hotel has different departments which are Manager, Reception (front office), Guard, Purchase, Main Stores, Security, Kitchen, Housekeeping, Laundry, Waiter and Cashier. The departments interact with each other in order to carry out the hotel's activities. The hotel runs many of its activities manually. The reception uses CNET software in a computer to store user information and the casher uses cash register to carry out the monetary transaction. The user can reserve room online but they can't update or delete rooms reservation. It uses direct conversation to manage the hotel like accepting comment from the user and reporting day to day activity.

Generally, the existing system is not more developed, that means they are not using automated system of operation and managing process. It is not better for workers and it can use more effort from workers and time killing in their work, therefore our system should solve all these problems by automated online system.

Advantage of Existing System

- Employees don't need special computer skills to run the manual system
- Relatively low running costs as the system requires no electricity, internet services as the computerized system would.

Problems of Existing System

- Customer files can easily get lost or mix up with other customer file documents.
- Unnecessary duplication of data.
- Files are prone to theft unauthorized modification due to low data security levels and standards.
- Retrieval of customer records is extremely difficult.

- Data entry procedure is prone to errors.
- Customer records are extremely difficult to modify.

2.2.2 Requirement gathering

Activities of each department

As mentioned earlier the hotel consists of different departments which consist of actors within the hotel. Each actor has its own responsibilities.

Manager: -Responsible for allotting specific job for the employees, conduct meeting for employee and prepare budget for hotel.

Reception: -Hospitable reception of guests, customer registration, prepare guest profile and transfer guest profile to casher for bill generation.

Chef: -Manage the resource consumption of the hotel and buy products for the hotel to use and resale.

Kitchens: -Takes care of all the cooking done for the hotel.

Guard: -Keeps the building safe and **e**mergency communication information

Housekeeping: -Keeps all the rooms clean when it is vacant or unoccupied

Waiter: -Takes food order**,** serves drinks and food items**,** forwards orders to the accountant for bill generation**,** receive payment and finally submit payment.

2.2.3 Supplementary Requirements

1. **Business Rules**

This hotel has its own business rule to give service for the customer and other users. Among these business rules: -

- The customer must register to get the hotel service.
- User need Email address to reserve room online.
- Loss of key is subject to an extra charge birr for buy key.
- When leaving the room, customers are expected to check whether the door is properly locked.

2. Constraints

The problems that we might face while doing this project: -

- The graphical user interface (GUI) supports only one language (English)
- Except the customers those about to reserve room online, other users should have username and password (for identification) in order to login and use the system.
- Any customer who has no email account cannot reserve a room online.

2.3 The Proposed System

2.3.1 Overview of the Proposed System

The proposed system which will provide online reservation and ordering will make it easier for both the hotel and customers for reserving room, hire out meeting hall and ordering meal. And also, communication between the manager and the customer freely. The new system will have the following purposes: To overcome the problems of manual system, Online hotel management system is proposed, the central objective of *Online Hotel Management System* is to provide online facility for booking rooms, order meals and hire out meeting halls, this project can manage and keep records of hotel, introduce what the hotel gives the service, prepare reports about record system, display the availability of bed rooms and meals, and to make simple communication between the customer and the hotel manager.

2.3.2 Software requirement specification (SRS)

Functional Requirement

For Customers

- The system allows customer to register when they use for first time.
- The system allows customer to login in to the system.
- The system allows customer to reserve room.
- The system allows customer to hire out meeting hall.
- The system allows customer to order meal.
- The system allows customer to give comment to hotel manager.
- The system allows customer to visit Hotel information on the web site.
- The system allows customer to logout from the system.

For Manager

- The system allows the manager to login in the system.
- The system allows the manager to view the report that is generated by the system.
- The system allows the manager to view customer information.
- The system allows the manager to view comment.
- The system allows the manager to replay the comment to the customer.
- The system allows the manager to logout from the system.

For System Administrator

- The system allows the system administrator to login in the system.
- The system allows the system administrator to create user account.
- The system allows the system administrator to update user account.
- The system allows the system administrator to activate user account.
- The system allows the system administrator to deactivate user account.
- The system allows the system administrator to delete user account.
- The system allows the system administrator to logout from the system.

For Receptionist

- The system allows the receptionist to login into the system.
- The system allows the receptionist to add room.
- The system allows the receptionist to delete room.
- The system allows the receptionist to cancel reserved room after the customer is leaving from the hotel.
- The system allows the receptionist to view available room.
- The system allows the receptionist to view customer information
- The system allows the receptionist to register new customer.
- The system allows the receptionist to logout from the system.

For Chef

- The system allows the Chef to login in the system.
- The system allows the Chef to add meal item
- The system allows the Chef to delete meal item
- The system allows the Chef to view meal item.
- The system allows the Chef to view ordered meal.
- The system allows the Chef to logout from the system.

Non-Functional Requirement

Non-Functional requirements describe the quality of the system. So, there are a number of non-functional requirements in the proposed system. Among those non-functional requirements the following are listed below: -

Scalability: -The system will upgrade when some updates will be available and the system will operate correctly when the number of recorded information is being increased.

Speed: -The system will have quick response time.

Security: -The authenticate users should have a privilege to access the database. On the other hand, the system does not allow unauthorized users to login.

Usability: -The proposed system gives direct input on how real users use the system.

Portability: -The system supports every operating system.

Efficiency: -The system gives appropriate output based the list of inputs and also the system operates in the shortest time with the least number of resources.

2.3.3 Existing System Modelling

Essential Use Case

Use case diagrams are used to show graphically for the interaction between the system and users. In other words, they graphically describe who will use the system and in what ways the user expects to textually describe the sequence of steps of each interaction.

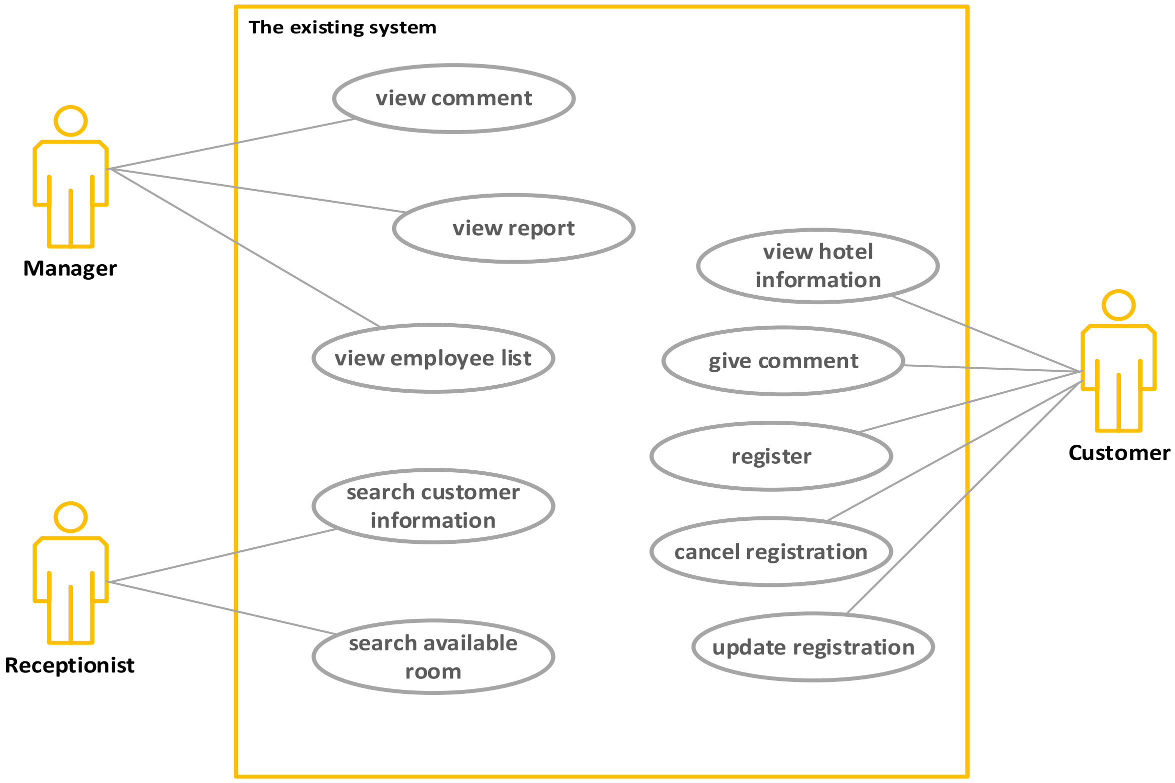


Figure 1. 2 Essential use case diagram for the existing system

2.3.4 Domain Modelling with Class Responsibility Collaboration (CRC)

Class responsibility collaboration is a collection of standard index cards that have been divided into three sections. A class represents a collection of similar objects, a responsibility is something that a class knows or does, and a collaborator is another class that a class interacts with to fulfill its responsibilities. In existing system there are classes. These are Receptionist, Customer, and Hotel Manager.

Manager	
▪ View comment () ▪ View employee () ▪ View report ()	Receptionist and Customer

Table 1. 3 CRC analysis of the existing system for Manager

Receptionist	
▪ Search available room () ▪ Search customer information ()	Manager and Customer

Table 1. 4 CRC analysis of the existing system for Receptionist

Customer	
▪ View hotel information () ▪ Give comment () ▪ Register () ▪ Update registration () ▪ Cancel registration ()	Receptionist and Manger

Table 1. 5 CRC analysis of the existing system for Customer

2.3.5 System Modeling

Use Case Modeling

The use case model provides detailed information about the behaviors of the system. A use case represents a discrete unit of interaction between a user and the system. A use case is a single unit of meaningful work. Each use case has a description which describes the functionality that will be built in the proposed system. Use case may include another use case functionality or extend another use case with its own behavior use cases are typically relate to actors.

Actor Specification

Actors are a person who are external agent to the system. Those persons have directly interact to the system. In this system there are a number of actors in the system. Each actors are their own responsibility in the system.

Those actors are: -

1. **Administrator: -**Create, update, and delete user account like manager, Chef and receptionist, activate user account, and also deactivate user account.
2. **Manager: -**View report, view customer comment, replay customer comment, view customer information, and view employee list.
3. **Chef: -**Add meal item, delete meal item, view meals item, and view ordered meal.
4. **Receptionist: -**Cancel reserved room after the customer is leaving from the room, add room, delete room, view available room, view customer information, register new customer, and accept reservation.
5. **Customer: -**Reserve room, order meal, hire out meeting hall, give comment and view hotel information, and create his/her account.

Use case Diagram

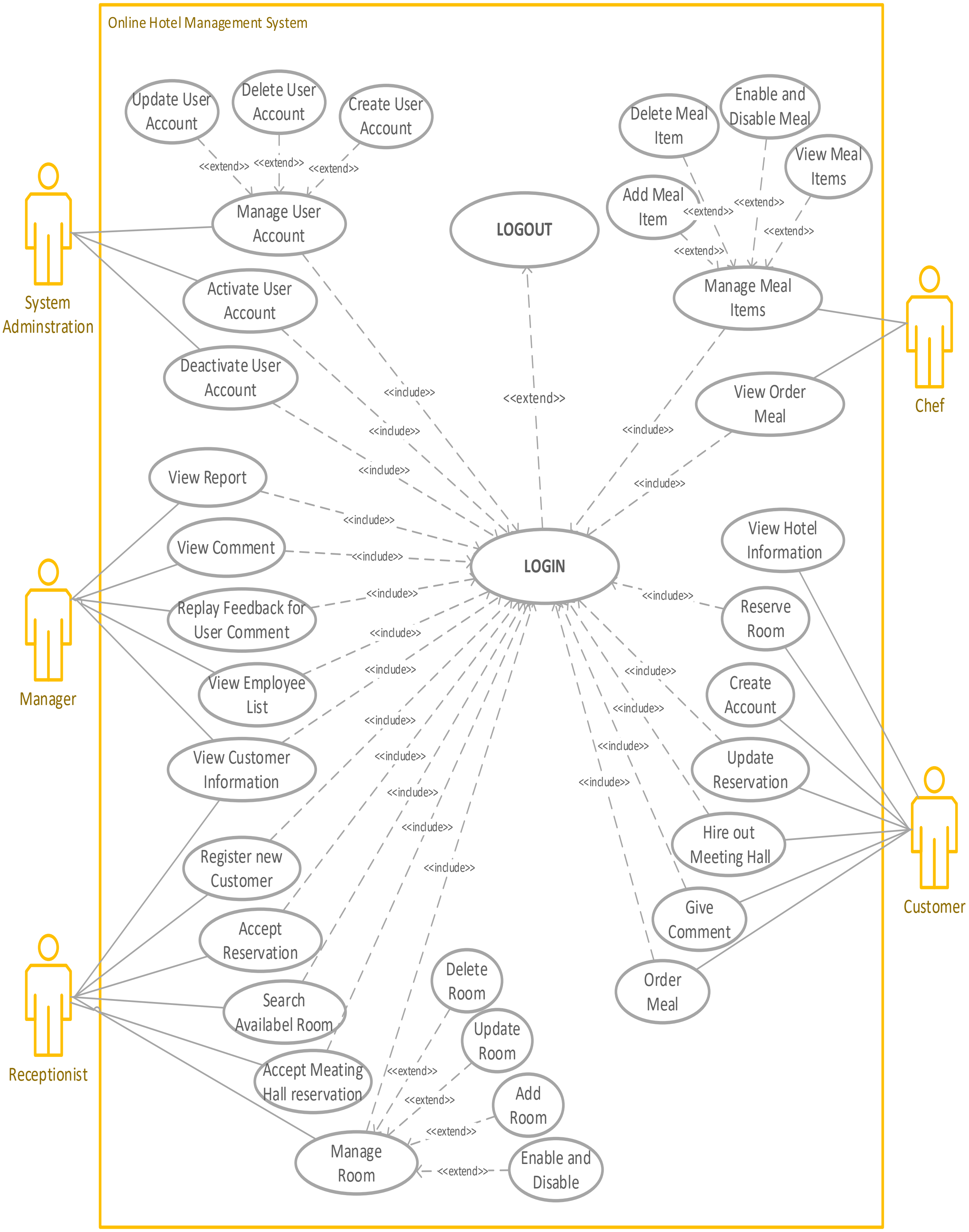


Figure 1. 3 Use Case Diagram

System Use case Documentation

Use case ID	UC-01	
Name	**Login**	
Actor	Customer, Manager, Chef, Administrator and Receptionist	
Description	This describes how the users log into the system	
Precondition	Users of the system should have User name and password	
Post condition	The Customer, Manager, Chef, Receptionist and Administrator successfully login.	
Basic course of action	**User action**	**System response**
	Step1: The user open the website.	**Step2:** The system display homepage.
	Step3: The user click login link.	**Step4:** The system display login form.
	Step5: The user fill the form.	**Step7:** The system validate the input.
	Step6: The user click login button.	**Step8:** The system authenticate user information
	Step10: Use case end.	**Step9:** The system display “you are successfully logged into the system.
Alternative course action	If the login name or password is not valid, the login screen is redisplayed with an error message. ✓ The login page is redisplayed. ✓ Go to step 4 and fill again. Otherwise, the system asks in order to forget the password.	

Table 1. 6 Use case description for login

Use case ID	UC-02	
Use case name	**Create Account**	
Description	Used to create account for customers and other users.	
Actor	Customer and administrator.	
Pre- condition:	The customer must enter a valid input in the registration form.	
Post- condition	The user account successfully created.	
Basic course of action (Flow of event):	**User action**	**System response**
	Step1. The customer or administrator open the system and click create account link. **Step3.**The customer or administrator fills the required information and submits it. **Step7.**The use case ends.	**Step2.**The system displays create account page. **Step4.**The system validates the information. **Step5.**The system registers the customers into the system. **Step6.**The system displays a message “your account has been successfully created”.
Alternative course of action (Flow of event):	If invalid information entry. **1.** The system displays error message. **2.** Go to **step 2** to fill again.	

Table 1. 7 **Use case description for Create Account**

Use case ID	UC-03	
Name	**View reports**	
Actor	Manager	
Description	The Manager finally see the reports from the system	
Precondition:	Report should be prepared by the system.	
Post condition	Generate final Report Information	
Basic course of action	**User action**	**System response**
	Step1:-The manager open home page. **Step3:-**The manager login to the system. **Step5:-**The manager click in report link. **Step7: -**The manager selects type of report. **Step9:-**Use case end.	**Step2:-**The system display home page. **Step4:-**The system display manager home page. **Step6:-** The system display report page. **Step8:-**The system display report page successfully.
Alternative course of action	If the report is not found try again in the system.	

Table 1. 8 **Use case view reports**

Use case ID	UC-04	
Name	**Reserve Room**	
Actor	Customer	
Description	This explains room reservation process by using the system.	
Precondition:	Customer must have email address.	
Post condition	Reserve room successfully.	
Basic course of action	**User action**	**System response**
	Step1:-The customer opens the system. **Step3:-**The customer click on room reservation link. **Step5:-**The customer selects the room. **Step8:-**The customer fill the form. **Step9:-**The customer click reserve button. **Step11:-**Use case end.	**Step2:-**The system display home page. **Step4:-**The system displays rooms. **Step6:-**The system check availability. **Step7:-**The system display reservation form. **Step10:-**The system displays the message “you are successfully reserved the room”.
Alternative course of action	If invalid entry :- The system displays error message and go to step 7 in user action.	

Table 1. 9 **Use case description for reserve room**

Use case ID	UC-05	
Name	**Order meal**	
Actor	Customer	
Description	This explains meal ordering process by using the system.	
Precondition:	Customer should have user account in the system.	
Post condition	Order meal successfully.	
Basic course of action	**User action**	**System response**
	Step1: -The customer opens the system. **Step3: -**The customer login to the system. **Step5: -**The customer clicks on meal order link. **Step7: -**The customer selects the meal and fill the form. **Step8: -**The customer clicks order button. **Step12: -**Use case end.	**Step2:-**The system display home page. **Step4:-**The system display customer home page. **Step6:-**The system displays meal order page. **Step9:-**The system validate input **Step10:-**The system check availability **Step11:-**The system displays the message "you are successfully order the meal".
Alternative course of action	If invalid entry: The system displays error message and go to step 6 in user action.	

Table 1. 10 **Use case description for order meal**

Use case ID	UC-06	
Name	**Add meal item**	
Actor	Chef.	
Description	This explains adding meal item process by using the system.	
Precondition:	Chef must have user account in the system.	
Post condition	Adding meal item successfully.	
Basic course of action	**User action**	**System response**
	Step1:-Open website and login into the system. **Step3:-**The Chef clicks on Add meal item link. **Step5:-**The Chef fill the form **Step6:-**The Chef click Add button. **Step9:-**Use case ends.	**Step2:-**The system display Chef page. **Step4:-**The system display the form of add meal item page. **Step7:-**The system validate the user entry. **Step8:-**You add successfully message is displayed in the system.
Alternative course of action	If invalid entry: The system displays error message and go to step 4 in user action.	

Table 1. 11 **Use case description for add meal item**

Use case ID	UC-7	
Use case name	**Deactivate User Account**	
Description	Used to deactivate user account for employee.	
Actor	Administrator	
Pre- condition:	1. The administrator must login into the system. 2. There should be previous account of user or employee to deactivate.	
Post- condition	The user account successfully updated.	
Basic course of action (Flow of event):	**User action**	**System response**
	Step1. The administrator wants to deactivate user account and login into the system. **Step3.**The administrator press the button deactivate. **Step5.**The use case ends.	**Step2.**The system displays user account details and display deactivate account button in front of user details. **Step4.**The system deactivates the user account from the system.
Alternative course of action (Flow of event):	If the operation is not success the system displays error message and go to step 3.	

Table 1. 12 **Use case description for delete user Account**

Use case ID	UC-8	
Name	**Add Room**	
Actor	Receptionist	
Description	This explains adding room process by using the system.	
Precondition:	Receptionist must have user account and login into the system.	
Post condition	Add room successfully.	
Basic course of action	**User action**	**System response**
	Step1:-Open website and login into the system. **Step3:-**The receptionist clicks on Add Room link. **Step5:-**The receptionist fill the form. **Step6:-**The receptionist click Add button. **Step9:-**Use case ends.	**Step2:-**The system display receptionist page. **Step4:-**The system display the form of add room page. **Step7:-**The system validate the user entry. **Step8:-**You add successfully message is displayed in the system.
Alternative course of action	If invalid entry: The system displays error message and go to step 4 in user action.	

Table 1. 13 **Use case description for add room**

Use case ID	UC-9	
Name	Logout	
Actor	Manager, Receptionist, Customer, Chef and System administrator	
Description:	When the user logouts when he/she wants to exit from the system.	
Precondition	The user should login first.	
Post condition	home page opened	
Basic course of action	**User action**	**System response**
	1. The user clicks the logout button. 3. End use case.	2. The system will display the home page.

Table 1. 14 **Use case description for Logout**

2.3.6 Key Abstraction with CRC Analysis

Class responsibility collaboration is a collection of standard index cards that have been divided into three sections. A **class** represents a collection of similar objects, a **responsibility** is something that a class knows or does, and a **collaborator** is another class that a class interacts with to fulfill its responsibilities. If the class has attribute it will included.

Class	
• Attribute	• Collaboration
• Responsibility	

Table 1. 15 **CRC standard index cards**

System Administrator	
✓ First Name ✓ Last Name ✓ Profile ✓ Address ✓ Sex ✓ Email ✓ User Name ✓ Password	▪ Manager, Receptionist, Chef, Customer
• Create user Account () • Update user Account () • Delete user Account () • Activate and Deactivate ()	

Table 1. 16**System Administrator CRC**

Manager	
✓ First Name ✓ Last Name ✓ ID ✓ Address ✓ Sex ✓ Email ✓ User Name ✓ Password	▪ Customer
• View report () • View customer () • View comment () • Replay comment ()	

Table 1. 17 **Manager CRC**

Receptionist	
✓ First Name ✓ Last Name ✓ ID ✓ Address ✓ Sex ✓ Email ✓ User Name ✓ Password	▪ Customer
• Add room () • Delete room () • View customer () • Registration () • Accept reservation () • Search Available room ()	

Table 1. 18 **Receptionist CRC**

Chef	
✓ First Name ✓ Last Name ✓ Profile ✓ Address ✓ ID ✓ Sex ✓ Email ✓ User Name ✓ Password	▪ Customer
• Add Meal () • Delete Meal () • View Order () • View Meal Item ()	

Table 1. 19 **Chef CRC**

Customer	
✓ First Name ✓ Last Name ✓ Email ✓ Country ✓ City ✓ User Name ✓ Password	▪ Manager, receptionist, Chef
• Create Account () • Reservation () • Meal order () • Give Comment () • Hire out meeting hall () • Search available room ()	

Table 1. 20 **Customer CRC**

2.3.7 Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows.

Login sequence diagram

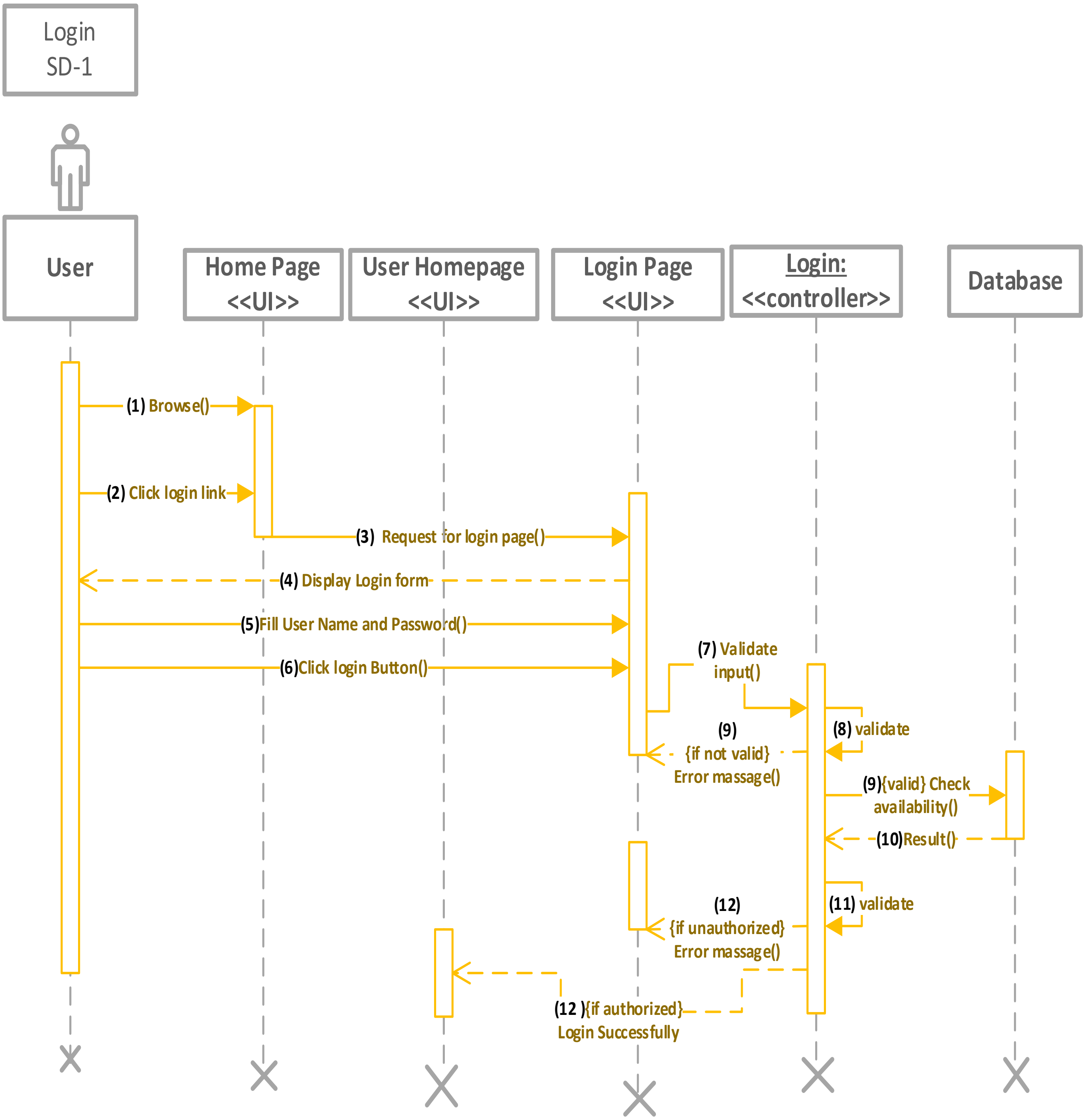


Figure 1. 4 Sequences diagram of login into the system

Create Account sequence diagram

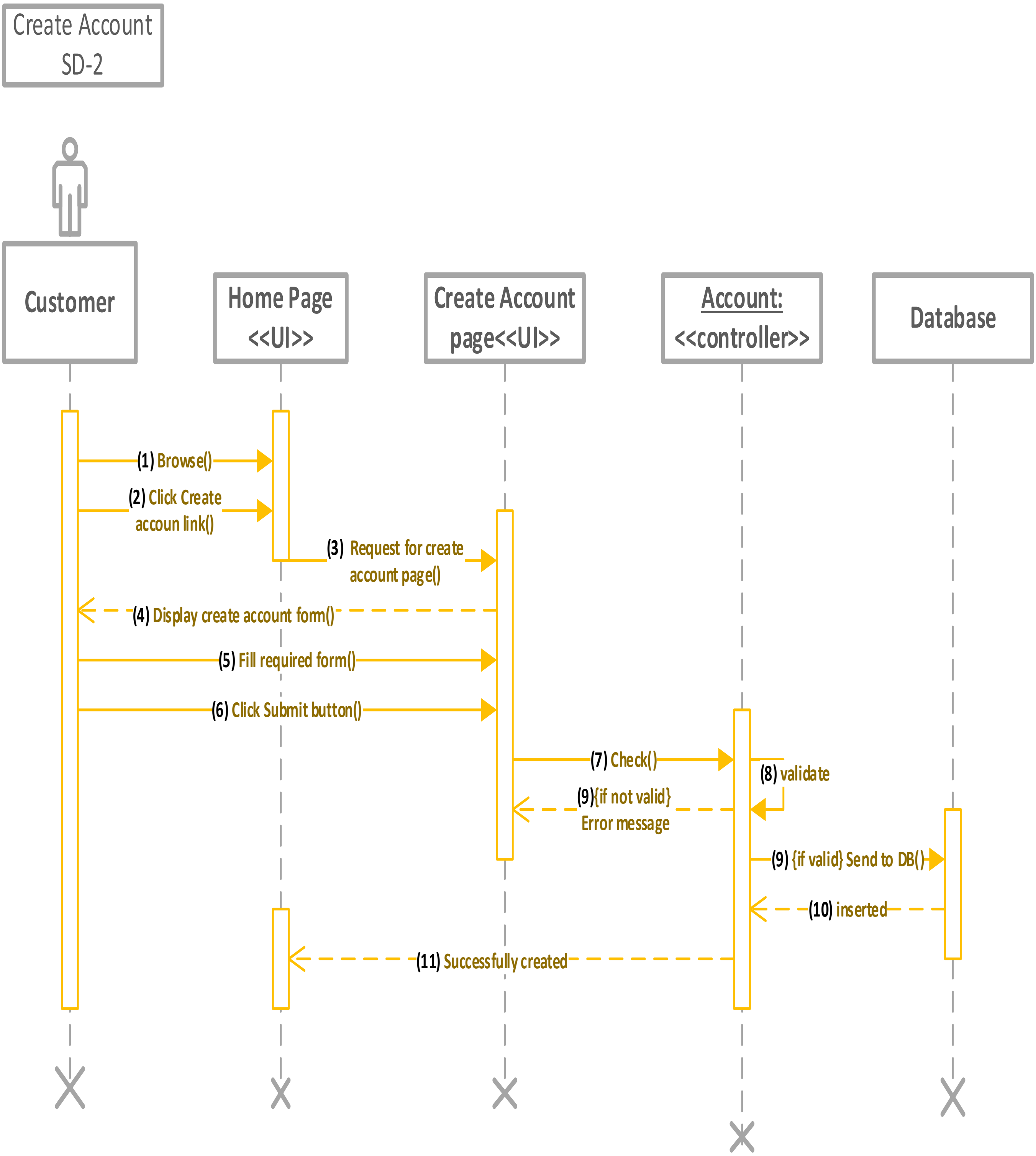

Create Account
SD-2
Customer
Home Page <<UI>>
Create Account page<<UI>>
Account: <<controller>>
Database
(1) Browse()
(2) Click Create accoun link()
(3) Request for create account page()
(4) Display create account form()
(5) Fill required form()
(6) Click Submit button()
(7) Check()
(8) validate
(9){if not valid} Error message
(9) {if valid} Send to DB()
(10) inserted
(11) Successfully created

Figure 1. 5 Sequences diagram for create account.

Generate Report sequence diagram

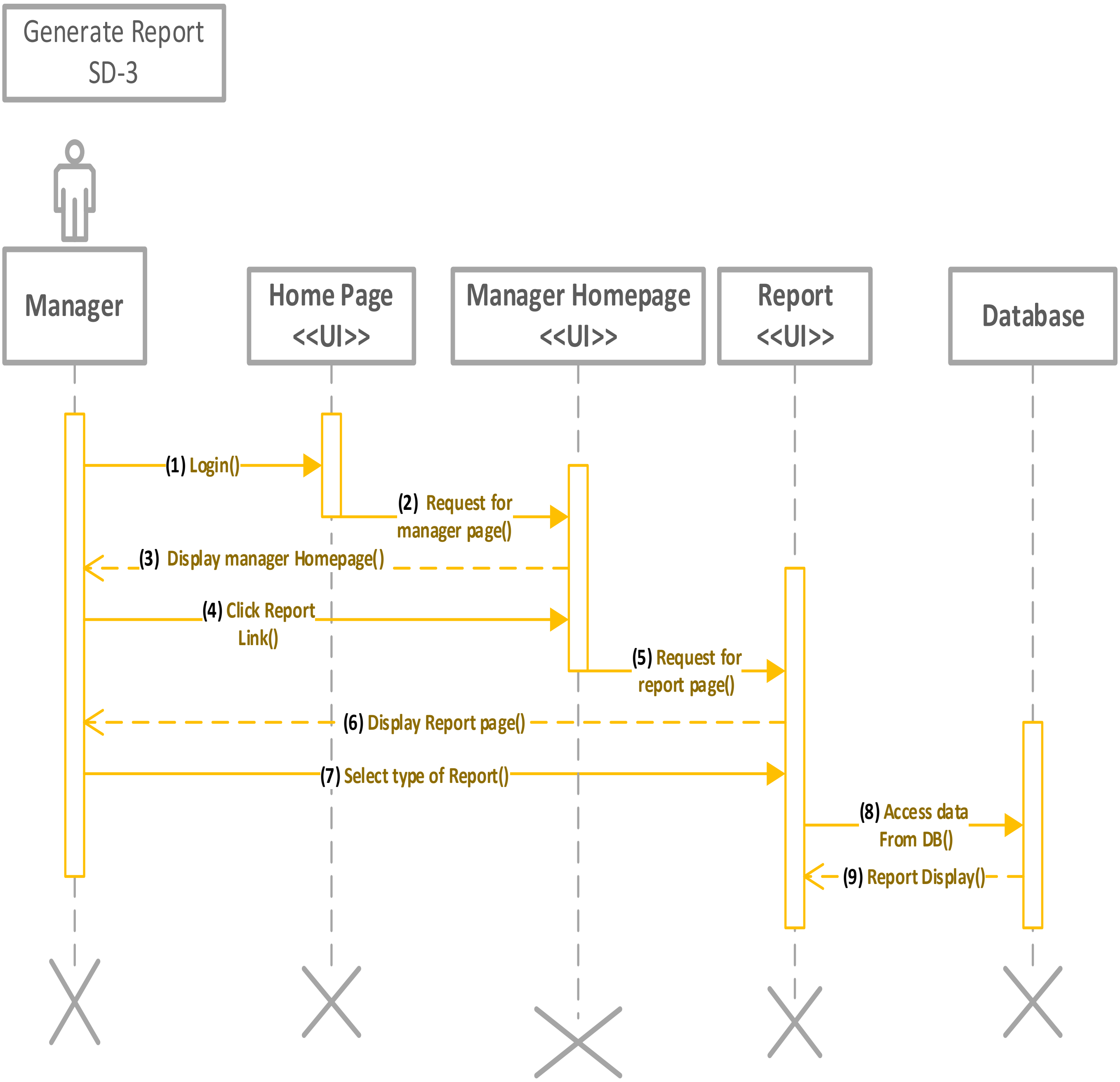

Generate Report
SD-3
Manager
Home Page
<<UI>>
Manager Homepage
<<UI>>
Report
<<UI>>
Database
(1) Login()
(2) Request for manager page()
(3) Display manager Homepage()
(4) Click Report Link()
(5) Request for report page()
(6) Display Report page()
(7) Select type of Report()
(8) Access data From DB()
(9) Report Display()

Figure 1. 6 Sequences diagram for generate report.

Reserve Room sequence diagram

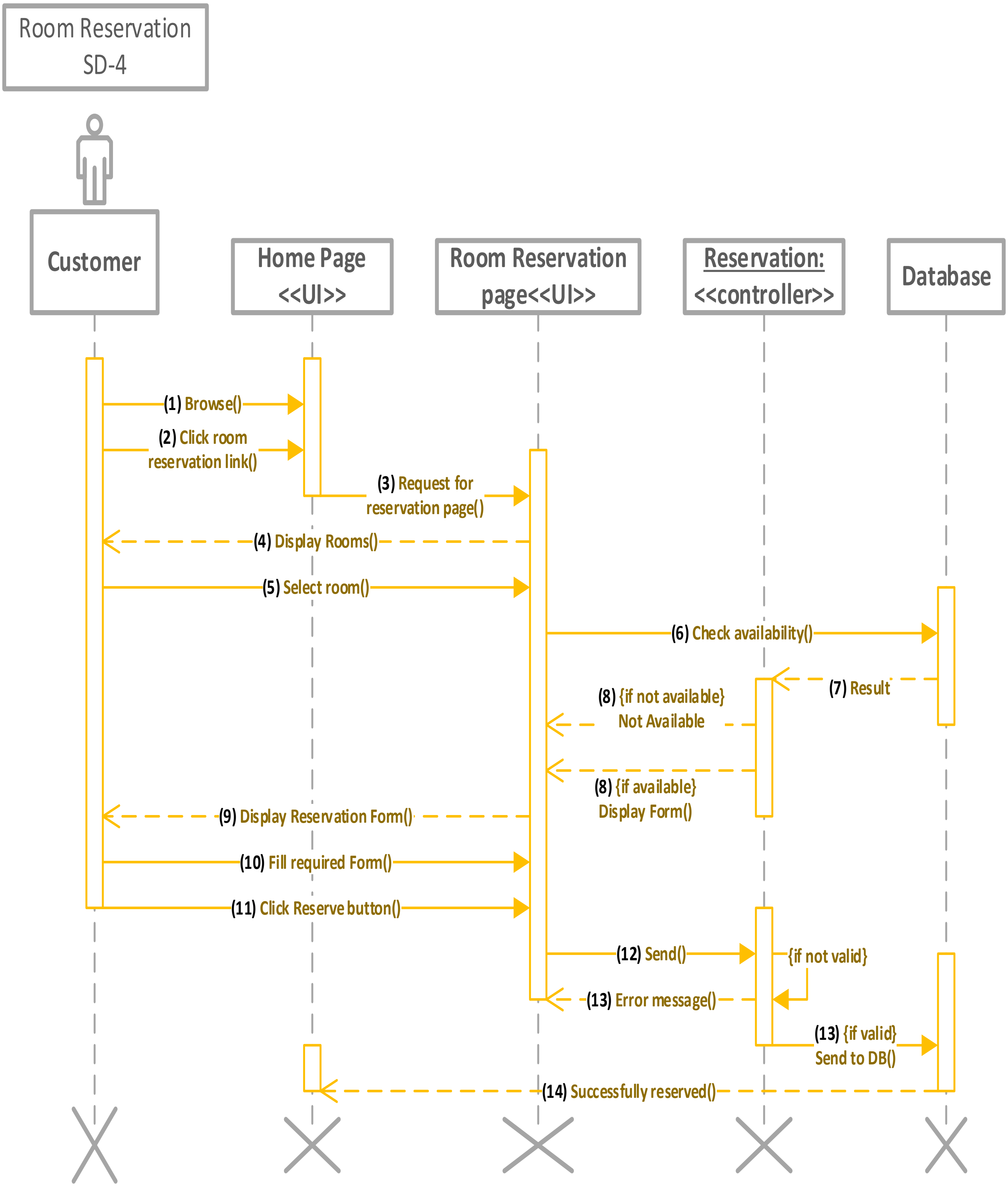

Room Reservation
SD-4
Customer
Home Page <<UI>>
Room Reservation page<<UI>>
Reservation: <<controller>>
Database
(1) Browse()
(2) Click room reservation link()
(3) Request for reservation page()
(4) Display Rooms()
(5) Select room()
(6) Check availability()
(7) Result
(8) {if not available} Not Available
(8) {if available} Display Form()
(9) Display Reservation Form()
(10) Fill required Form()
(11) Click Reserve button()
(12) Send()
{if not valid}
(13) Error message()
(13) {if valid} Send to DB()
(14) Successfully reserved()

Figure 1. 7 Sequences diagram for reserve room.

Meal Order sequence diagram

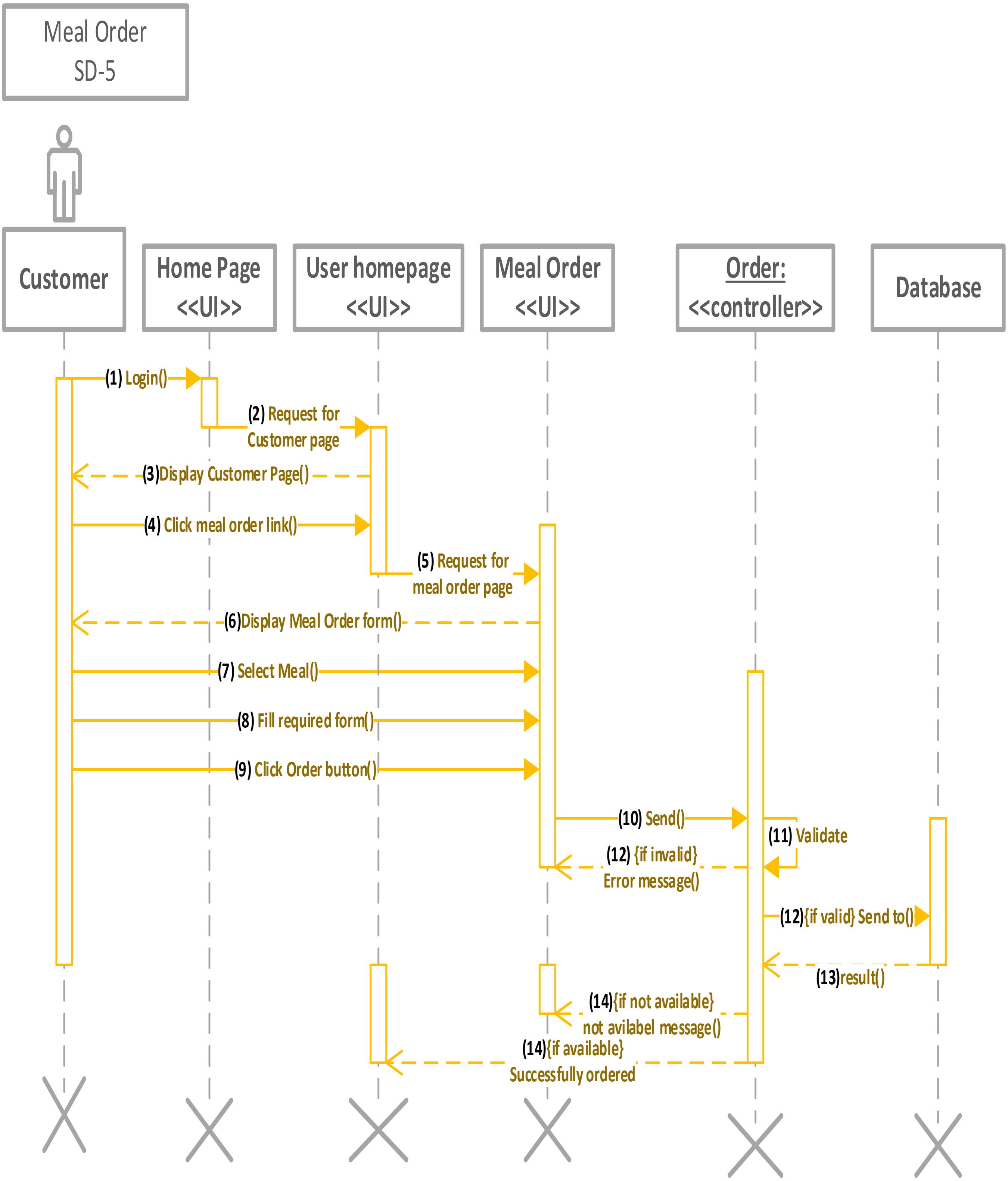

Meal Order
SD-5
Customer
Home Page
<<UI>>
User homepage
<<UI>>
Meal Order
<<UI>>
Order:
<<controller>>
Database
(1) Login()
(2) Request for Customer page
(3)Display Customer Page()
(4) Click meal order link()
(5) Request for meal order page
(6)Display Meal Order form()
(7) Select Meal()
(8) Fill required form()
(9) Click Order button()
(10) Send()
(11) Validate
(12) {if invalid} Error message()
(12){if valid} Send to()
(13)result()
(14){if not available} not avilabel message()
(14){if available} Successfully ordered

Figure 1. 8 Sequences diagram for meal order.

Add Meal sequence diagram

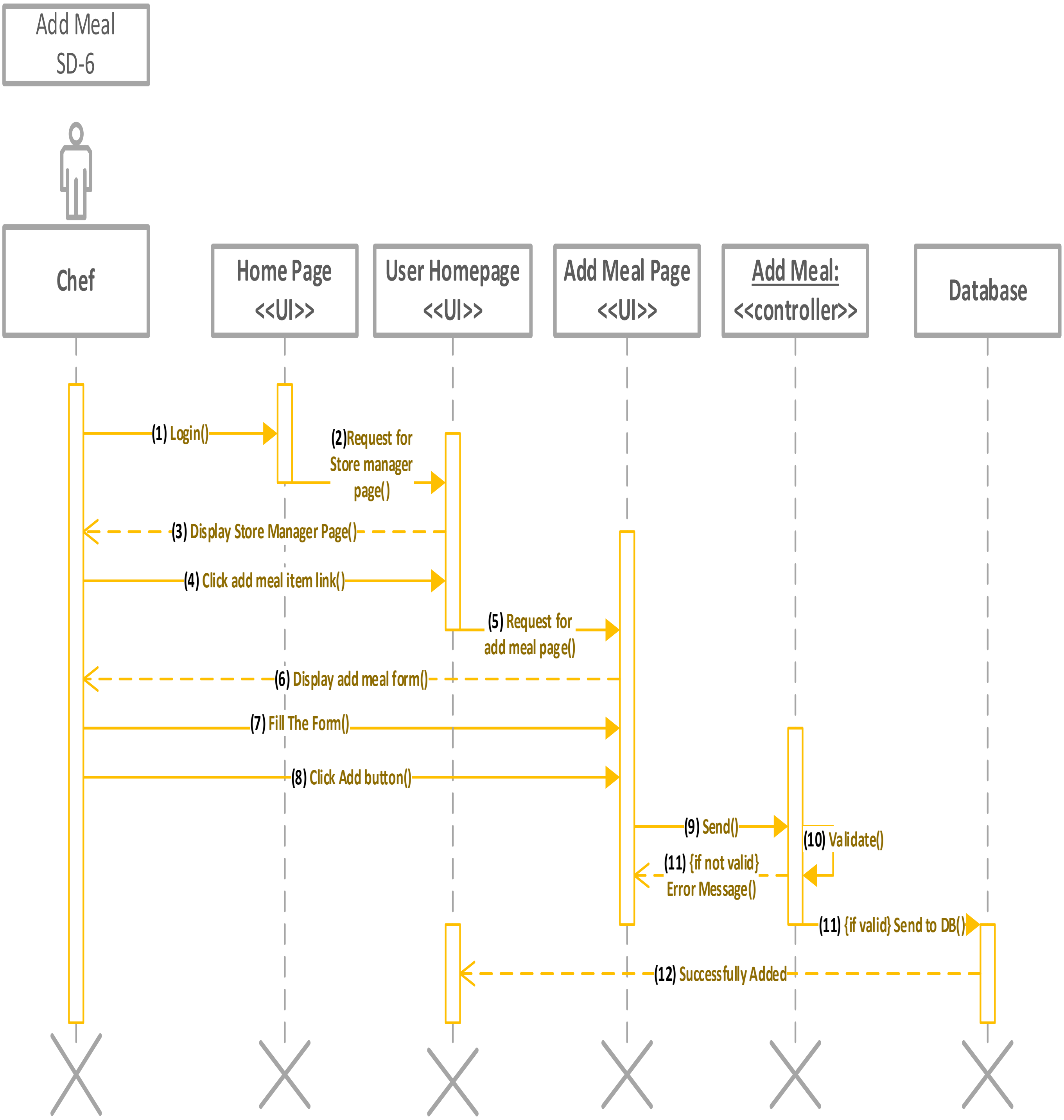

Add Meal
SD-6
Chef
Home Page
<<UI>>
User Homepage
<<UI>>
Add Meal Page
<<UI>>
Add Meal:
<<controller>>
Database
(1) Login()
(2)Request for Store manager page()
(3) Display Store Manager Page()
(4) Click add meal item link()
(5) Request for add meal page()
(6) Display add meal form()
(7) Fill The Form()
(8) Click Add button()
(9) Send()
(10) Validate()
(11) {if not valid} Error Message()
(11) {if valid} Send to DB()
(12) Successfully Added

Figure 1. 9 Sequences diagram for add meal

Sequence Diagram for Leave Comment

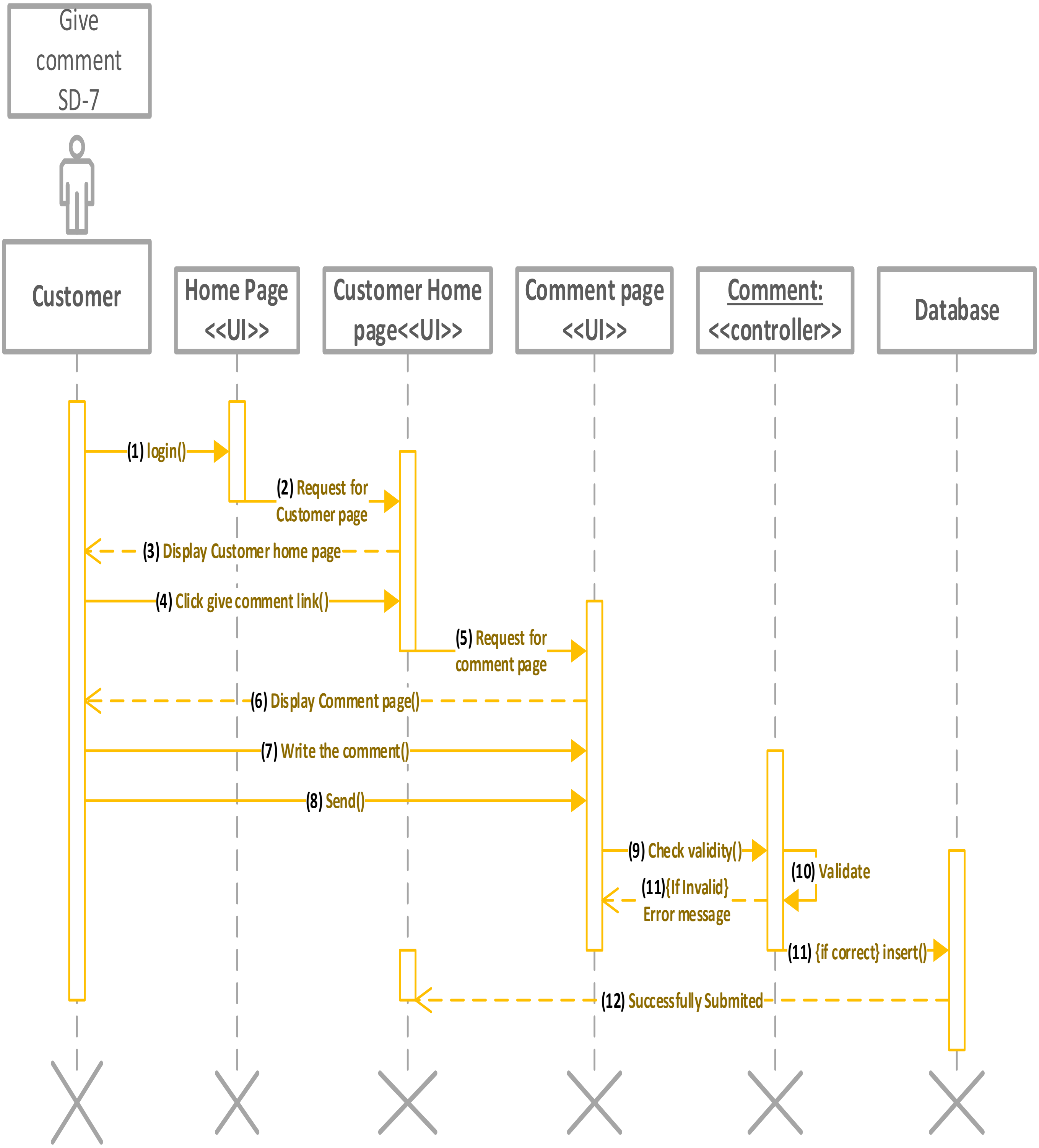

Give comment SD-7
Customer
Home Page <<UI>>
Customer Home page<<UI>>
Comment page <<UI>>
Comment: <<controller>>
Database
(1) login()
(2) Request for Customer page
(3) Display Customer home page
(4) Click give comment link()
(5) Request for comment page
(6) Display Comment page()
(7) Write the comment()
(8) Send()
(9) Check validity()
(10) Validate
(11){If Invalid} Error message
(11) {if correct} insert()
(12) Successfully Submited

Figure 1. 10 Sequence diagram for leave comment

Sequence diagram for Add Room

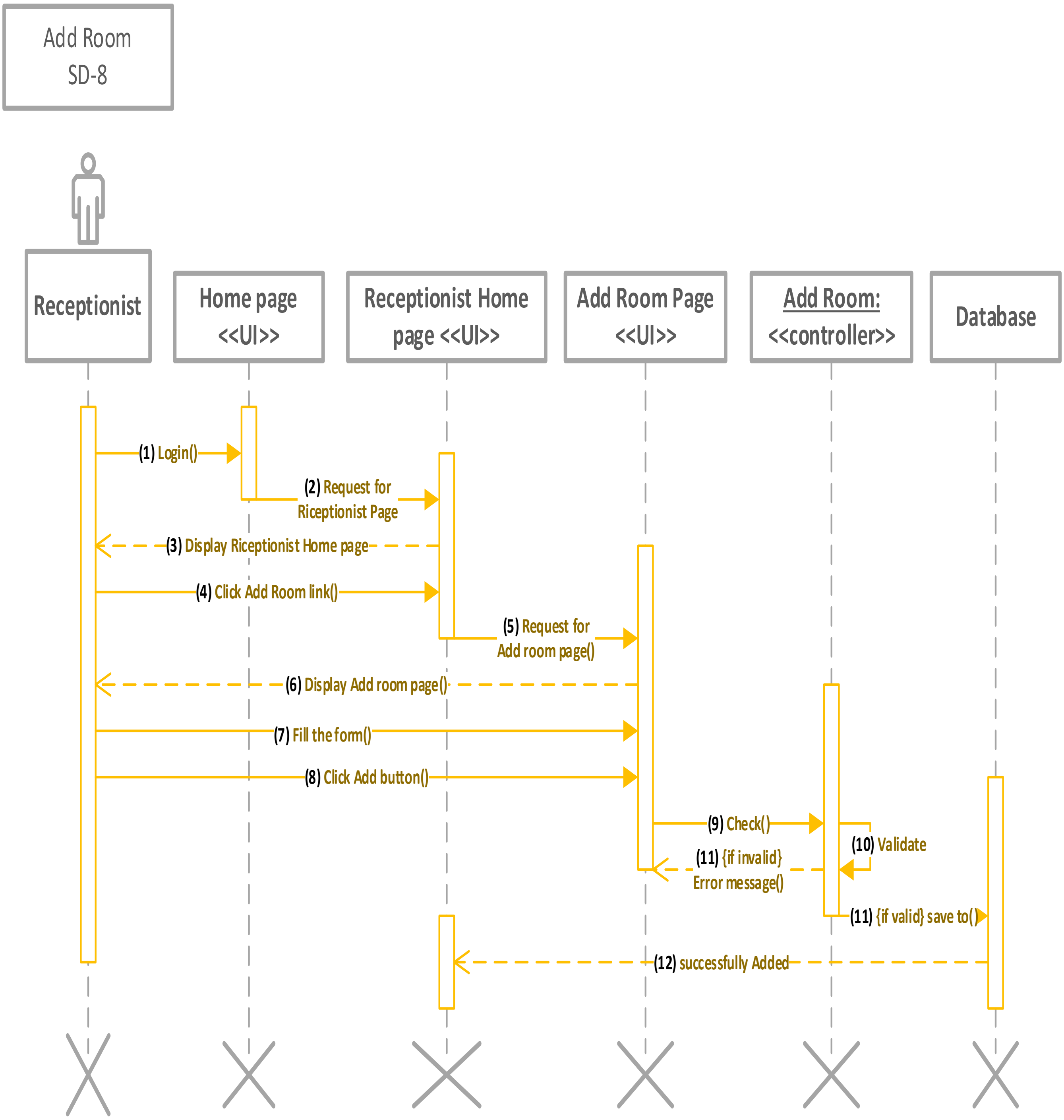

Add Room
SD-8
Receptionist
Home page <<UI>>
Receptionist Home page <<UI>>
Add Room Page <<UI>>
Add Room: <<controller>>
Database
(1) Login()
(2) Request for Riceptionist Page
(3) Display Riceptionist Home page
(4) Click Add Room link()
(5) Request for Add room page()
(6) Display Add room page()
(7) Fill the form()
(8) Click Add button()
(9) Check()
(10) Validate
(11) {if invalid} Error message()
(11) {if valid} save to()
(12) successfully Added

Figure 1. 11 Sequence diagram for Add room

Sequence Diagram for Search Customer Information

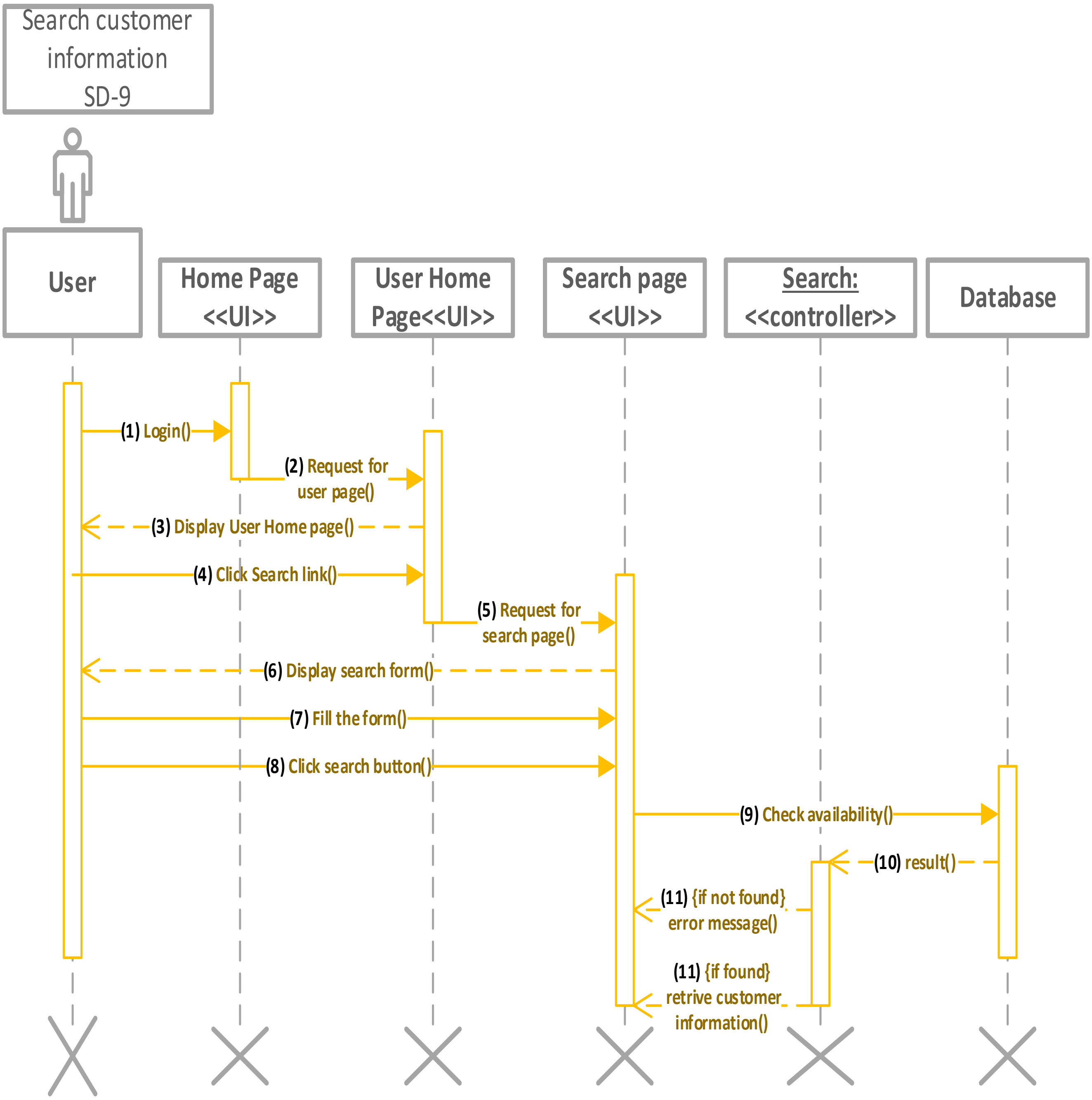

Search customer information SD-9
User
Home Page <<UI>>
User Home Page<<UI>>
Search page <<UI>>
Search: <<controller>>
Database
(1) Login()
(2) Request for user page()
(3) Display User Home page()
(4) Click Search link()
(5) Request for search page()
(6) Display search form()
(7) Fill the form()
(8) Click search button()
(9) Check availability()
(10) result()
(11) {if not found} error message()
(11) {if found} retrive customer information()

Figure 1. 12 Sequence diagram for search customer information

Sequence Diagram for Delete User Account

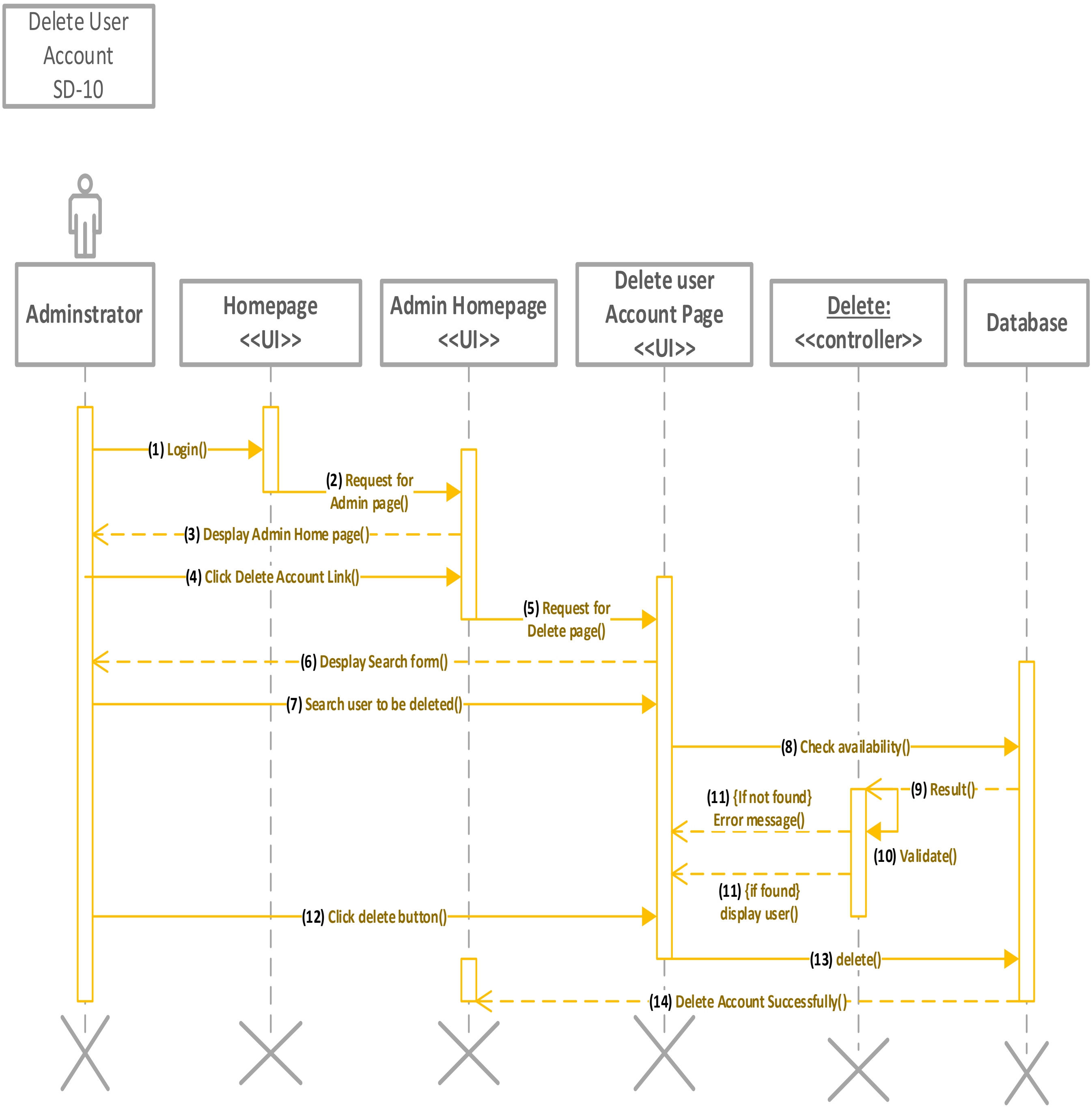


Figure 1. 13 Sequence diagram for delete user account

Sequence Diagram for Logout

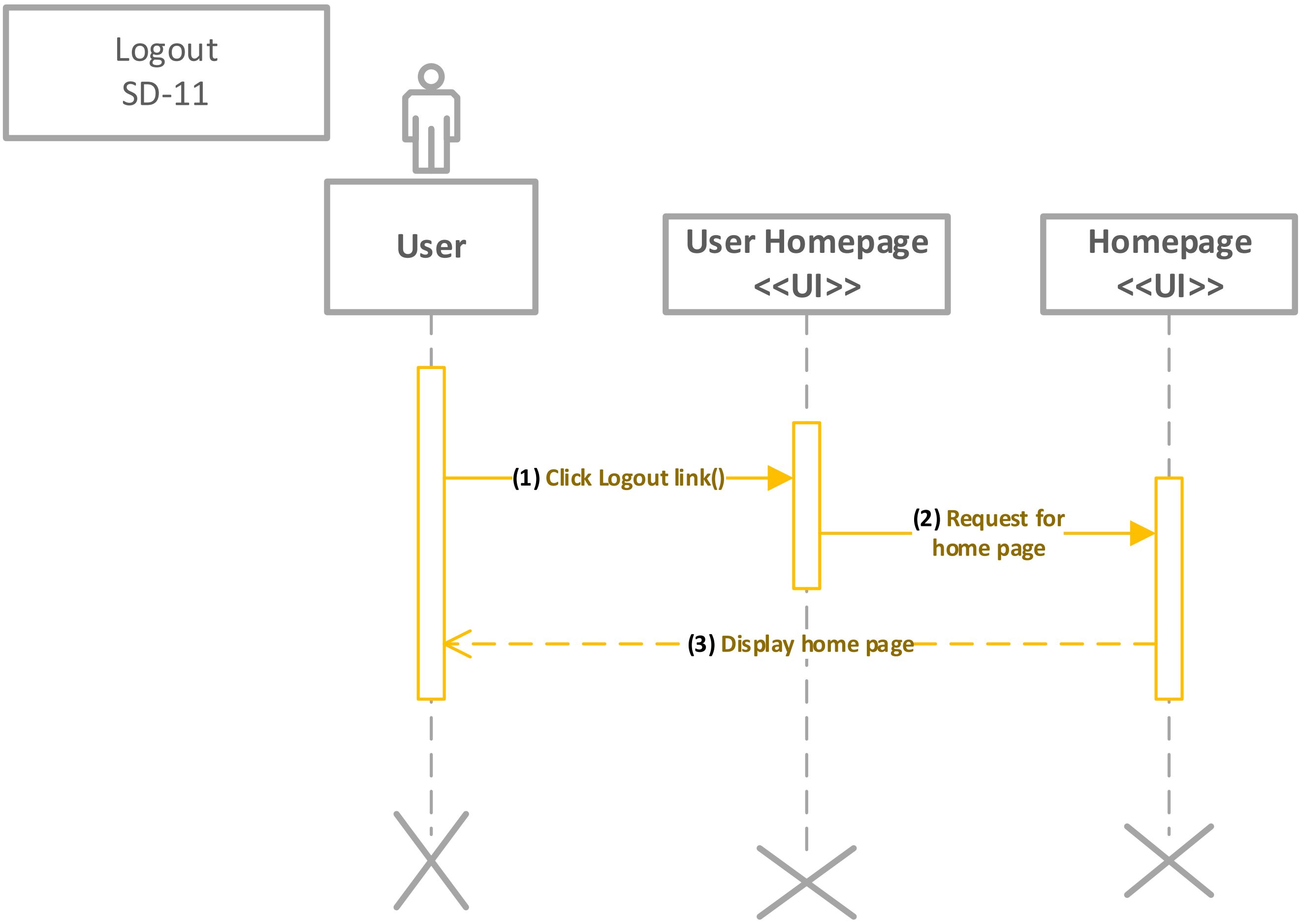


Figure 1. 14 Sequence diagram for logout

2.3.8 Activity Diagram

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. In the unified modeling language (UML), activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagram of login to the system

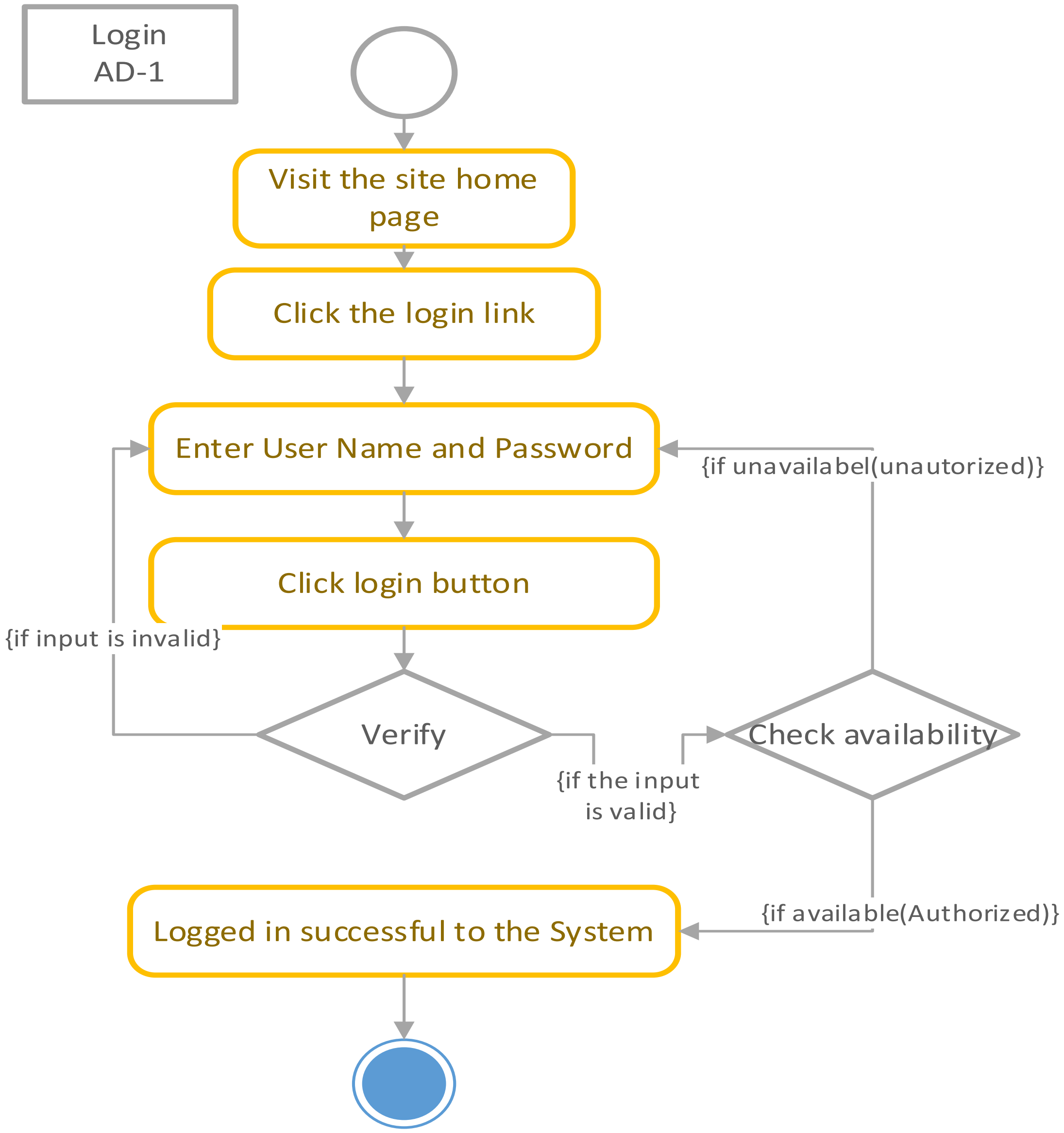


Figure 1. 15 Activity diagram of login to the system

Activity diagram for create account

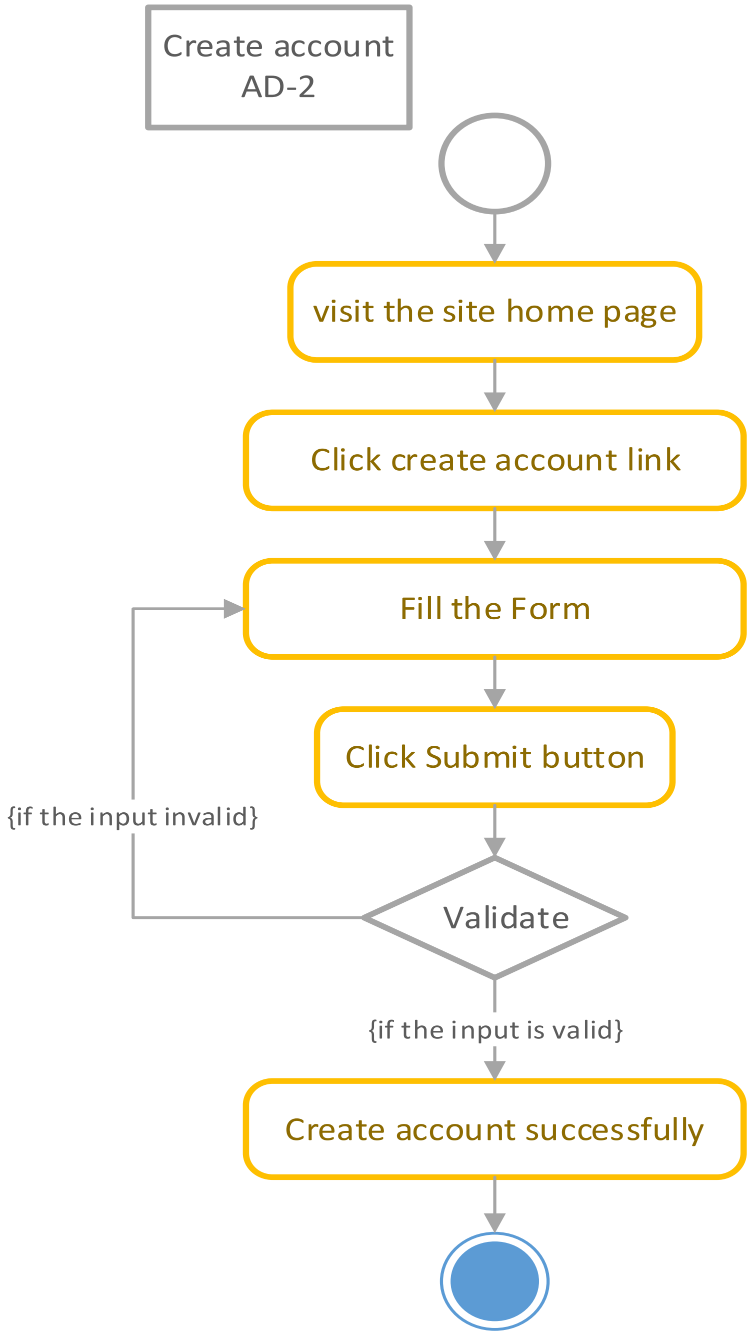


Figure 1. 16 Activity diagram for create account

Activity diagram for generate report

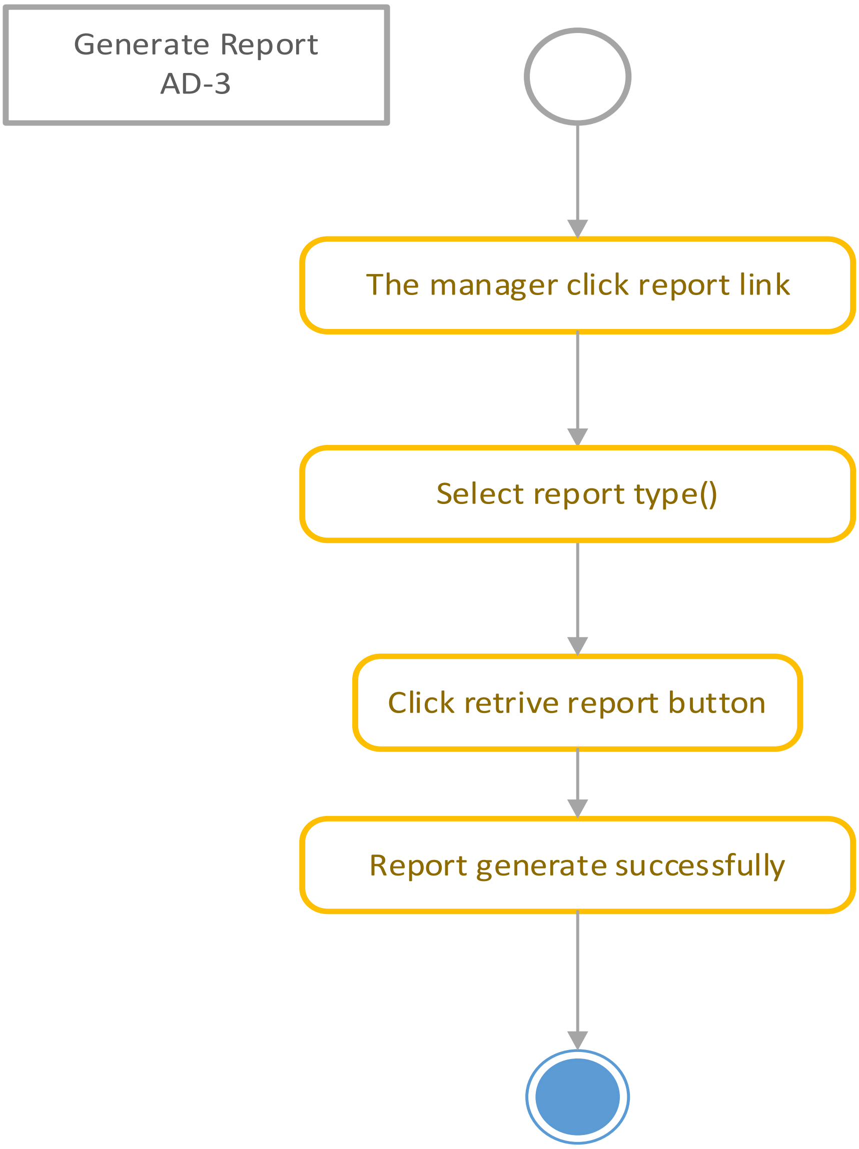

Generate Report
AD-3
The manager click report link
Select report type()
Click retrive report button
Report generate successfully

Figure 1. 17 Activity diagram for generate report

Activity diagram for reserve room

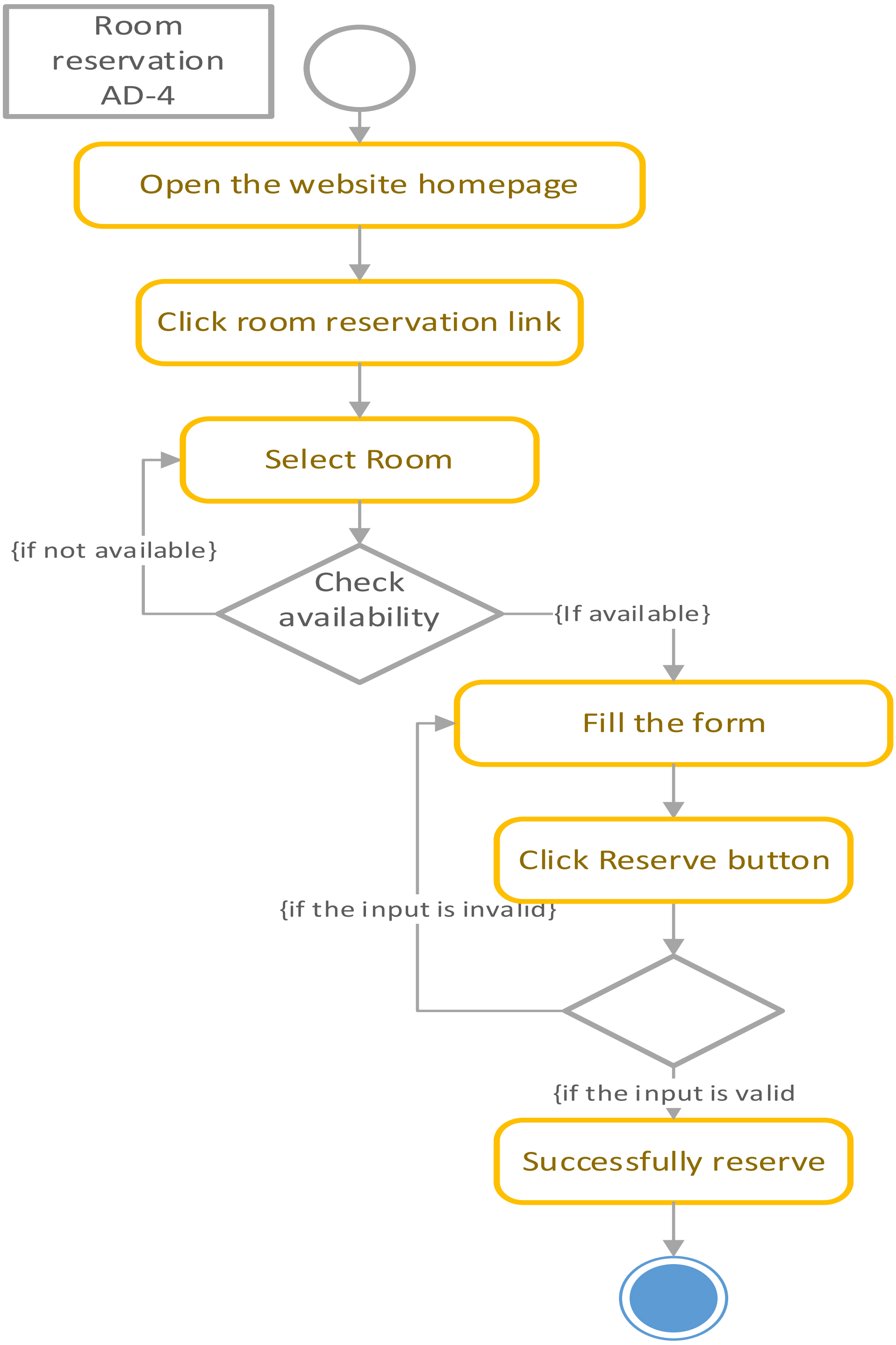


Figure 1. 18 Activity diagram for room reservation

Activity diagram for Meal order

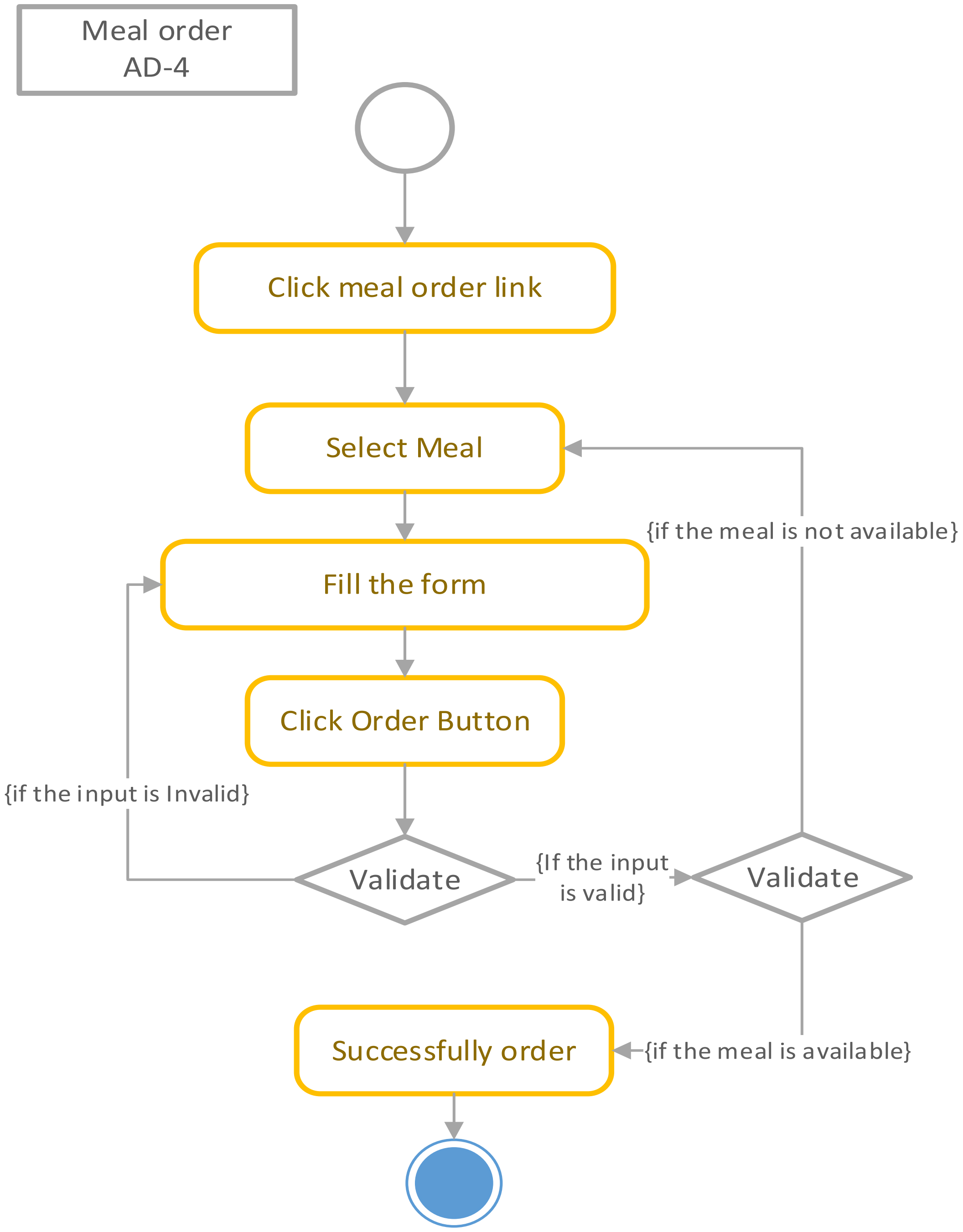


Figure 1. 19 Activity diagram for Meal order

Activity diagram for give comment

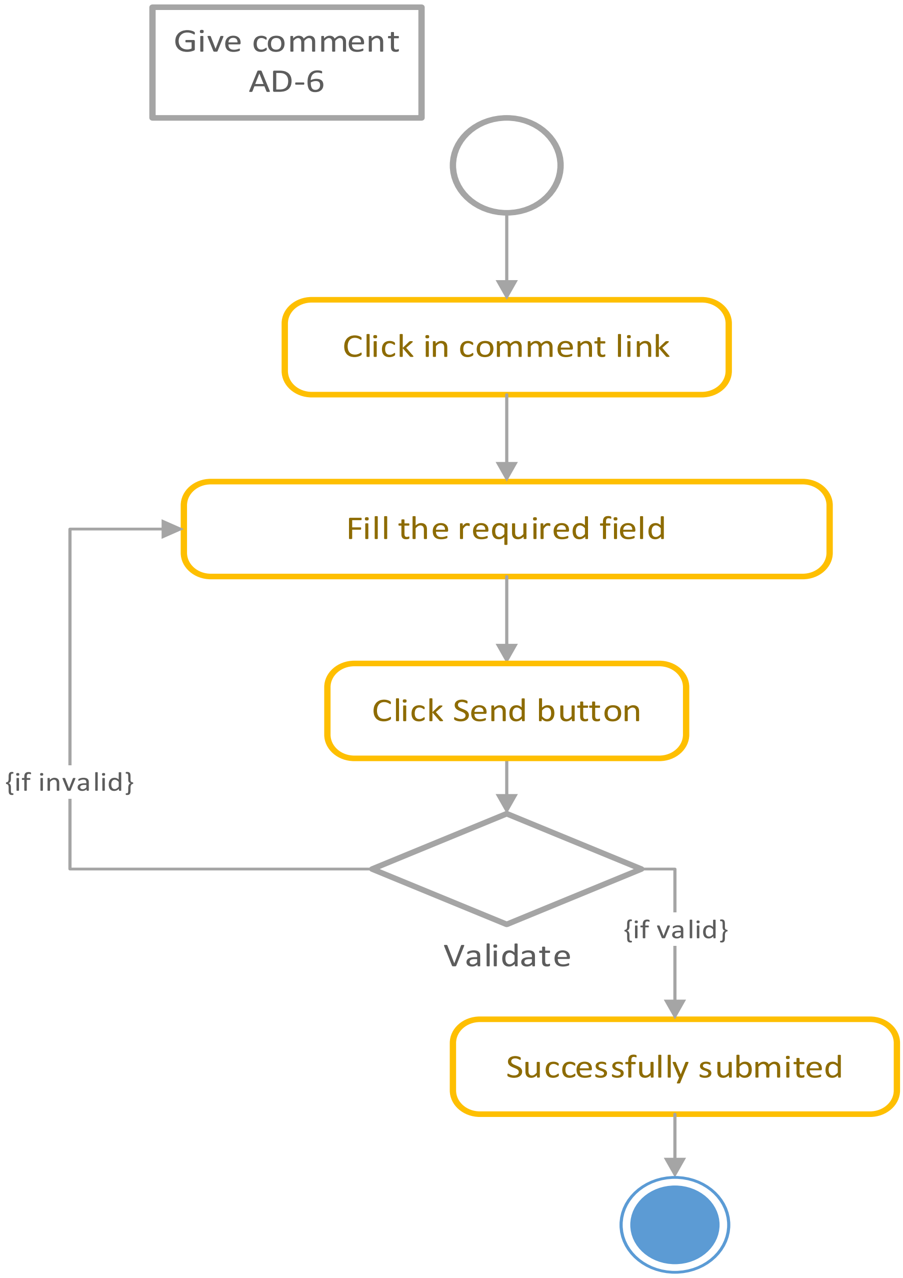


Figure 1. 20 Activity diagram for give comment

Activity diagram for Add room

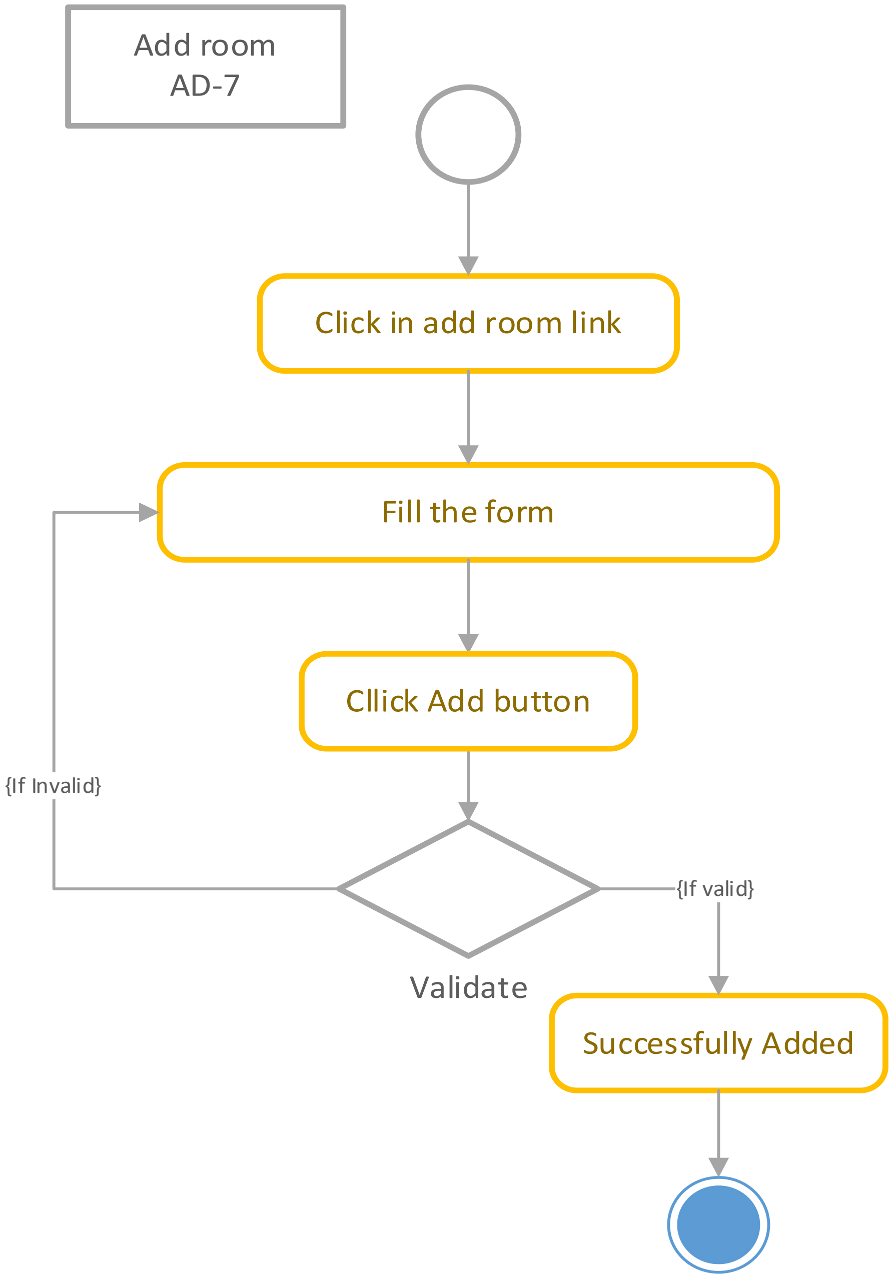


Figure 1. 21 Activity diagram for Add room

Activity diagram for View customer information

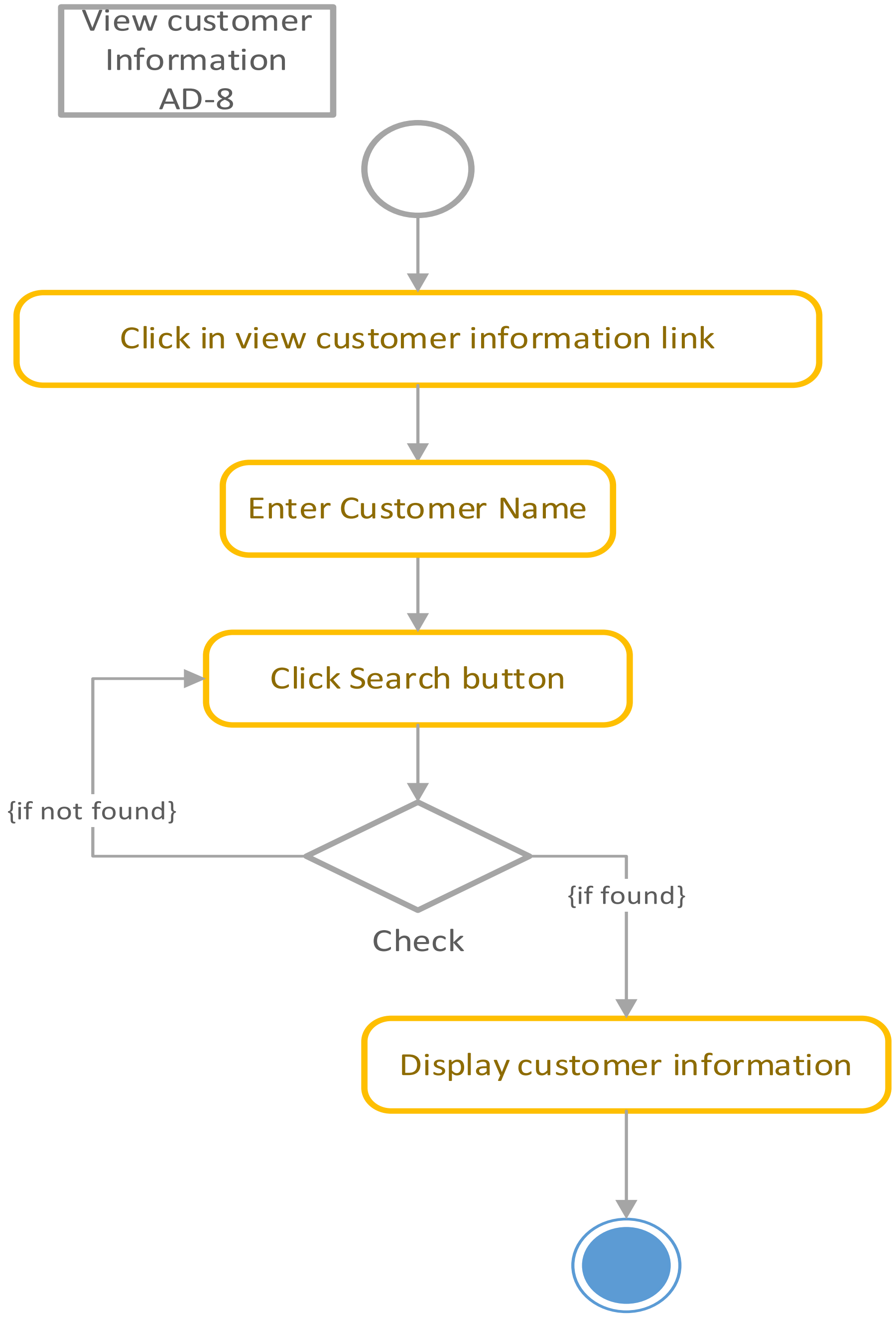


Figure 1. 22 Activity diagram for View customer information

Activity diagram for Delete User Account

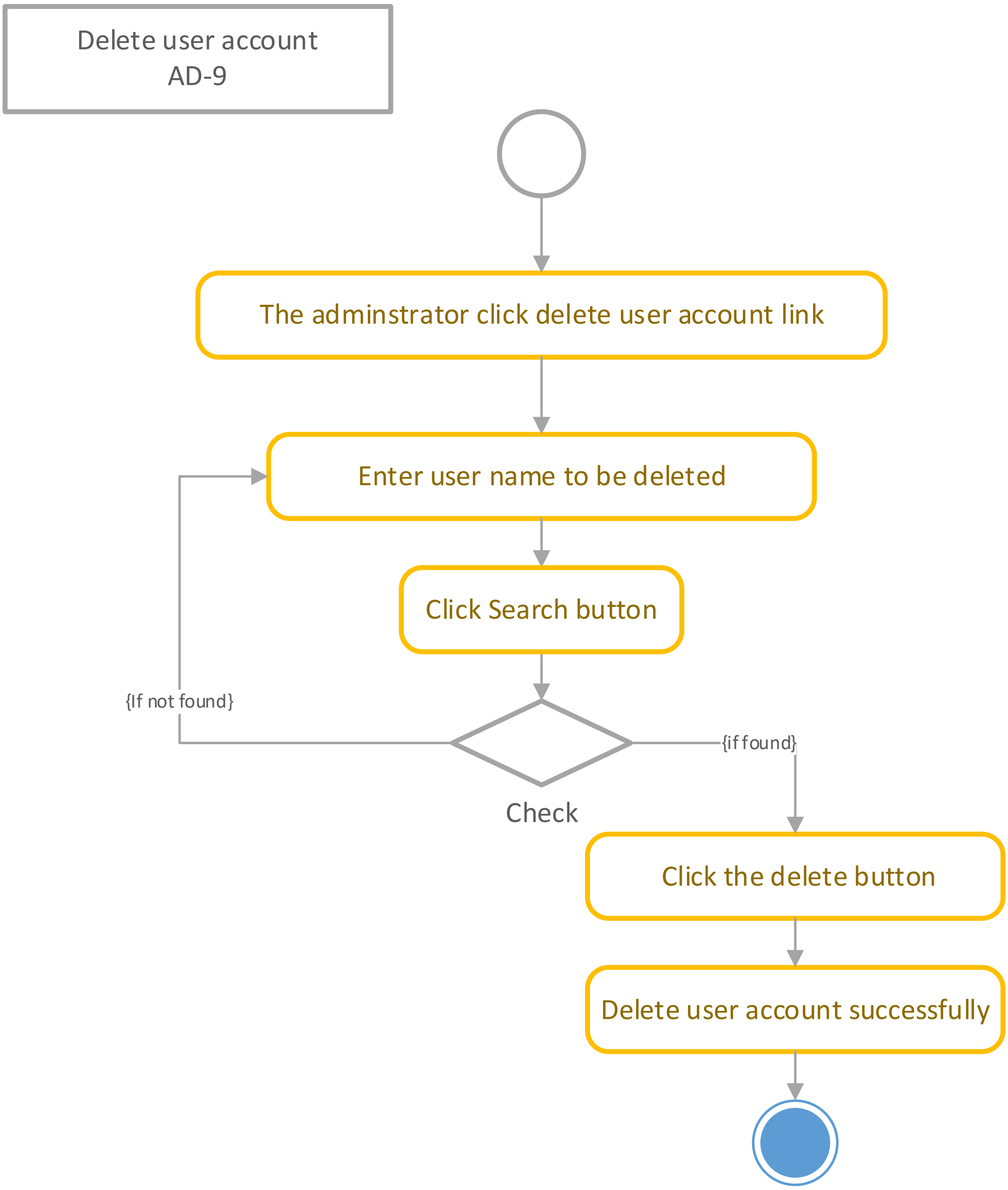

Delete user account
AD-9
The adminstrator click delete user account link
Enter user name to be deleted
Click Search button
{If not found}
{if found}
Check
Click the delete button
Delete user account successfully

Figure 1. 23 Activity diagram for delete user account

Activity Diagram for Logout

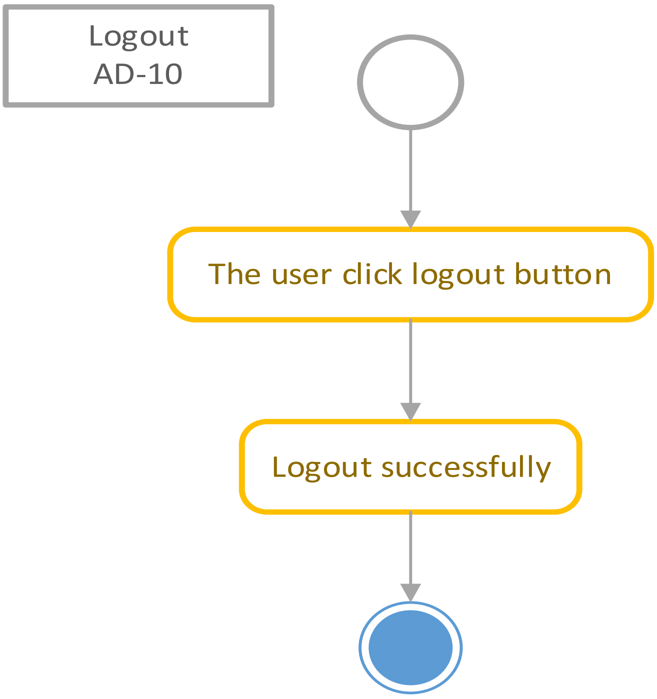


Figure 1. 24 Activity diagram for logout

2.3.9 Conceptual Modelling: Class Diagram

Class diagram is a type of structure diagram that describes the structure of a system by showing the system classes, their attributes, operations (or methods), and the relationships among objects. For this project the project team develops the following conceptual class modeling diagrams.

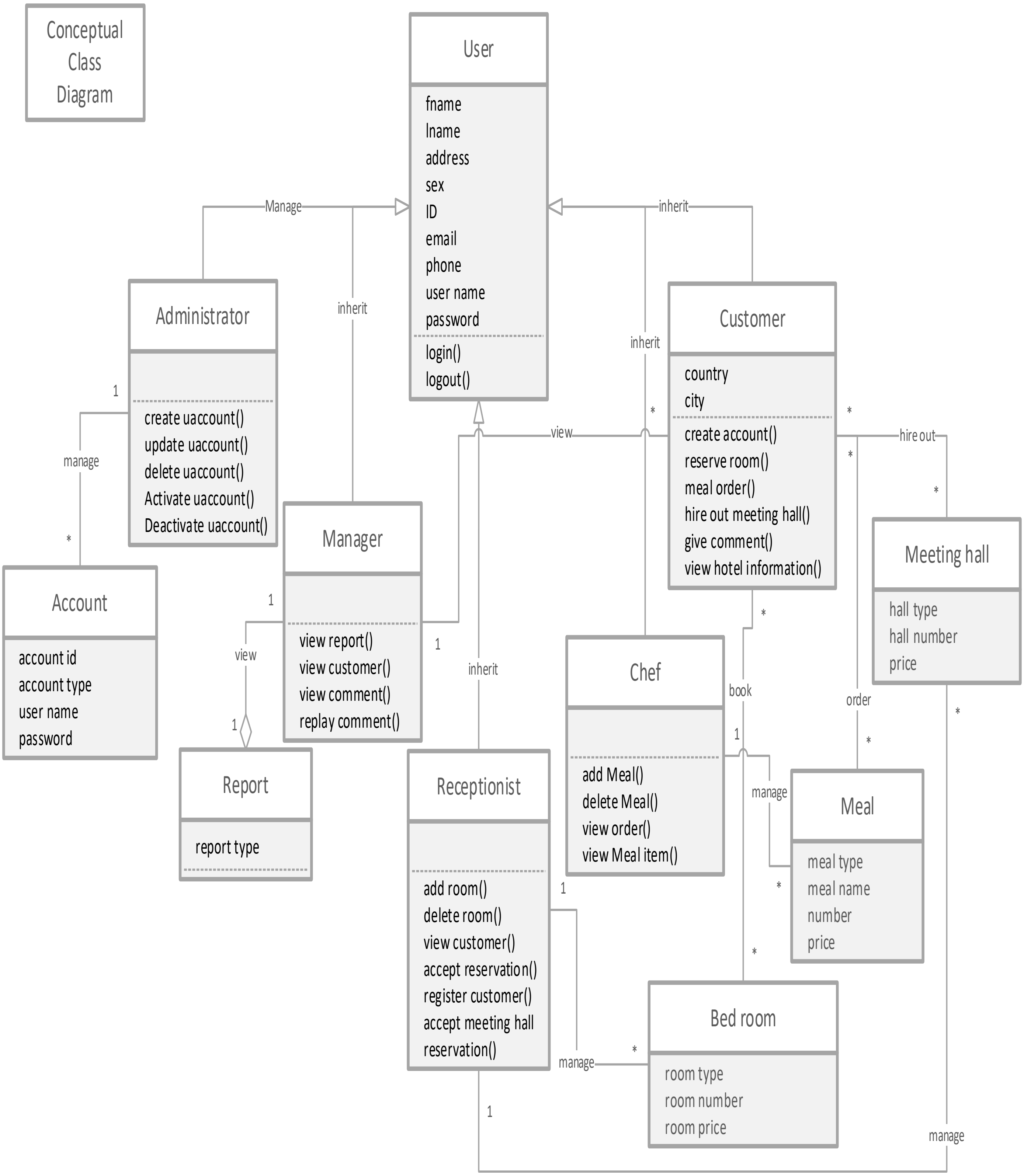


Figure 1. 25 Conceptual Class Modelling diagram

2.3.10 User Interface Prototyping

A user interface (UI) prototype is an iterative analysis technique in which users are actively involved in the make-up of the UI for a system as a design artifact that enables you to explore the solution space of your system.

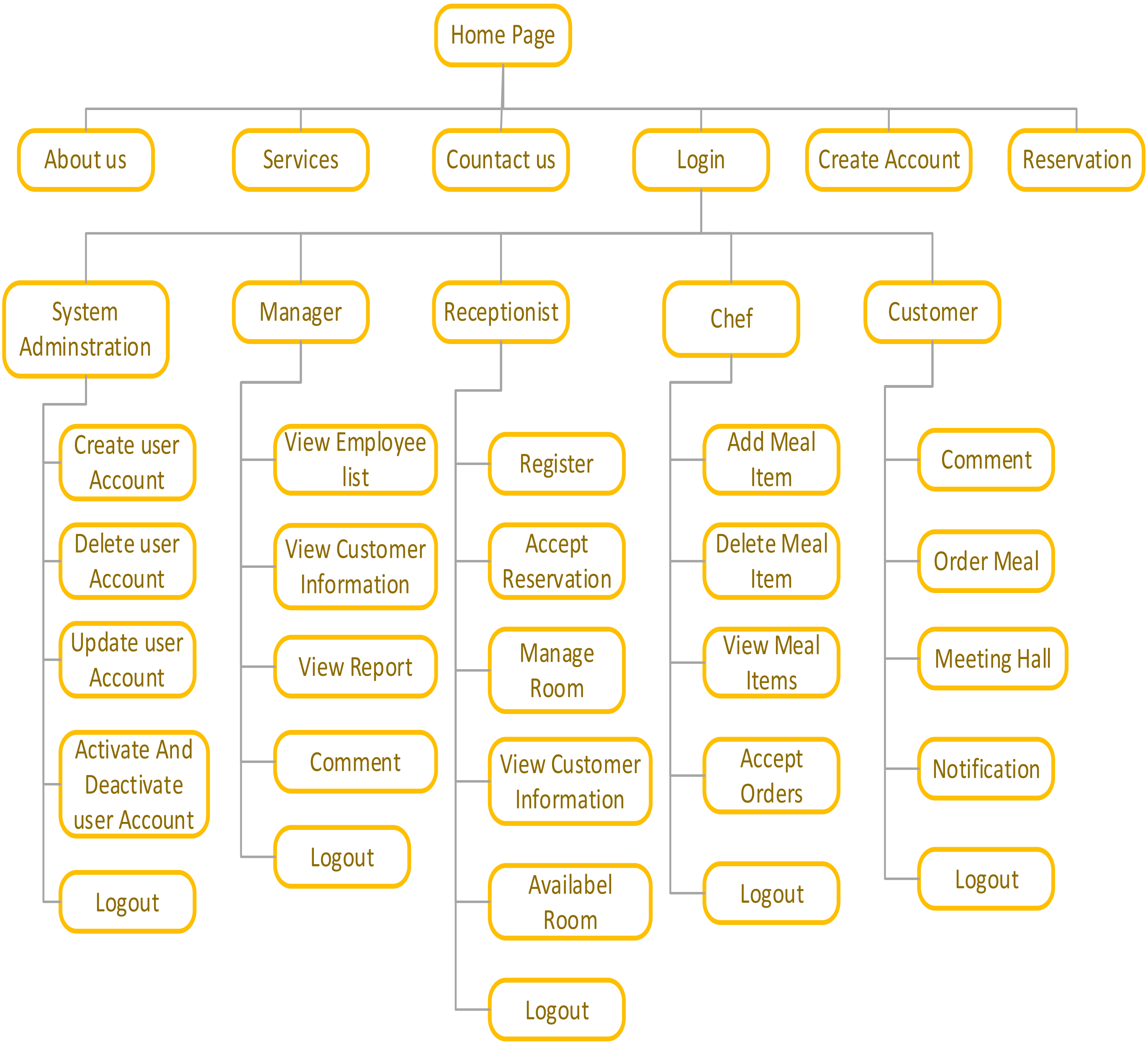


Figure 1. 26 User Interface Prototype

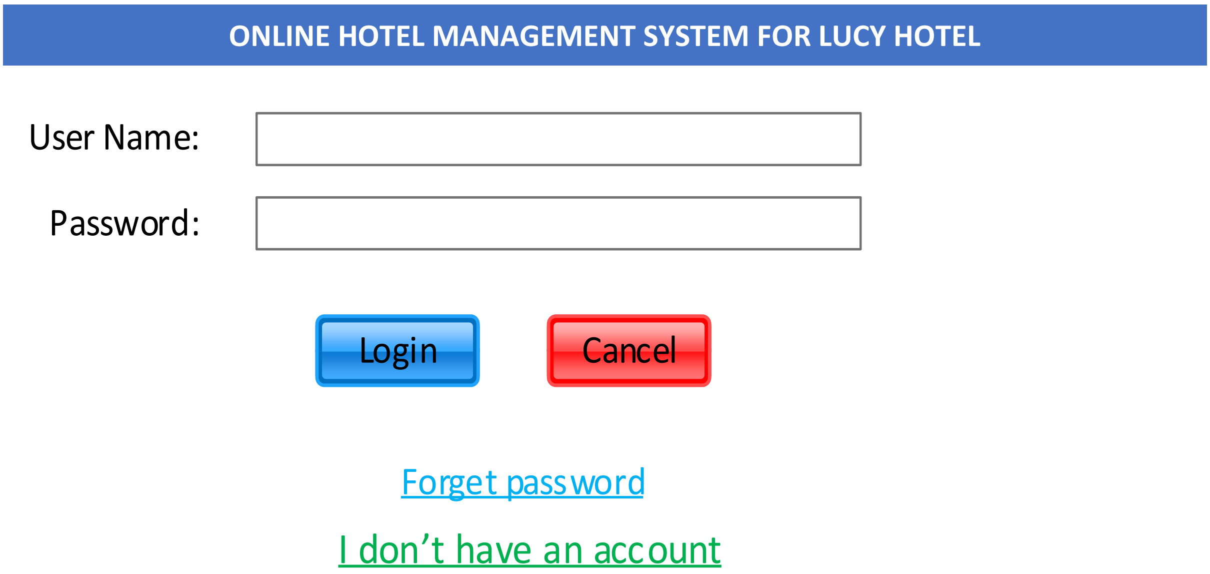


Figure 1. 27 Login form prototype

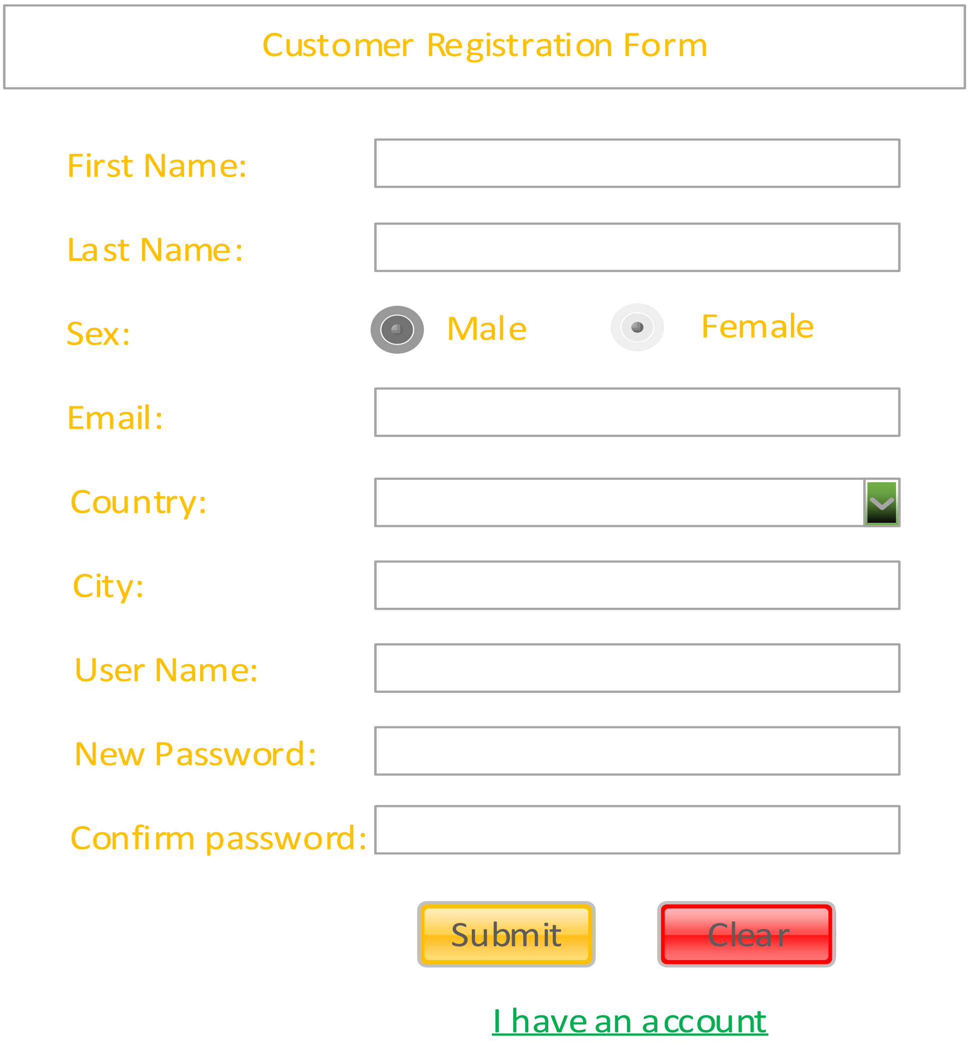


Figure 1. 28 Customer Registration form prototype

2.3.11 Identifying change cases

Change case will be used to describe potential modification requirements to the system.

Many cases change the content of the project. From these: -

- Missing activities: -Where there were important activities missed in each project development phase and the developing team latterly understood them, they should include them.

- Important comments: - When crucial comments are raised from the advisor, teachers, and examiners that should be included and excluded the developing team assesses the project again.

- Lack of resources: - If resources are scarce occurred to develop many system activities, the developing team also restructured the contents of the project.

CHAPTER THREE: SYSTEM DESIGN

3.1 Introduction

System design is the transformation of the analysis model into a system design model. Up to now we were in the problem domain. System design is the first part to get into the solution domain in a software development. This chapter focuses on transforming the analysis model into the design model that takes into account the non-functional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier.

The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on understanding of the model the software built on. The objectives of design are to model the system with high quality. Implementing of high-quality system depend on the nature of design created by the designer. If one wants to change to the system after it has been put in to operation depends on the quality of the system design. So, if the system is design effetely, it will be easy to make changes to it.

3.2 Purpose and goals of design

The objectives of design are to model the system with high quality. The design goals are derived from non-functional requirements that means non-functional requirement is the description of the feature characteristics and attribute of the system as well as any constraints that may limit the boundary of the proposed solution.

Design goals describe the qualities of the system that the developers should consider: -

Easy to use: All the pages of the system are easy and user friendly.

Response time: The hotel management system is developed by taking the last optimized way in each operation performed in the system. This will decrease the response time of the system.

Security: In using such system the security issue is essential because data that is going to be stored in database secure information is stored and this type of data should be kept safely. In this system we use appropriate security measurements to secure the system from hackers, cracker, and the system should be secured that unauthorized user cannot access the system.

Fault tolerance: The system should be able to give response (error message) when the user enters incorrect input. If the user enters incorrect data like username, password and other invalid inputs, then the system displays error message that inform the user to enter the correct data, to do so, we apply validation to our system using java script.

Response time: The hotel management system is developed by taking the last optimized way in each operation performed in the system. This will decrease the response time of the system.

Easy maintenance: The proposed system is using object-oriented programming; the overall system is divided into classes so it will be easy to identify where the failure (error) is found and maintain it.

Portability: In this case the proposed system is platform independent. Because we use PHP as server-side scripting language which is supported by many platforms.

User friendly interface: -Users can easily input or access in simple way. This means that the system must accommodate a clearly understandable user interface.

3.3 Class Modelling Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The classes diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

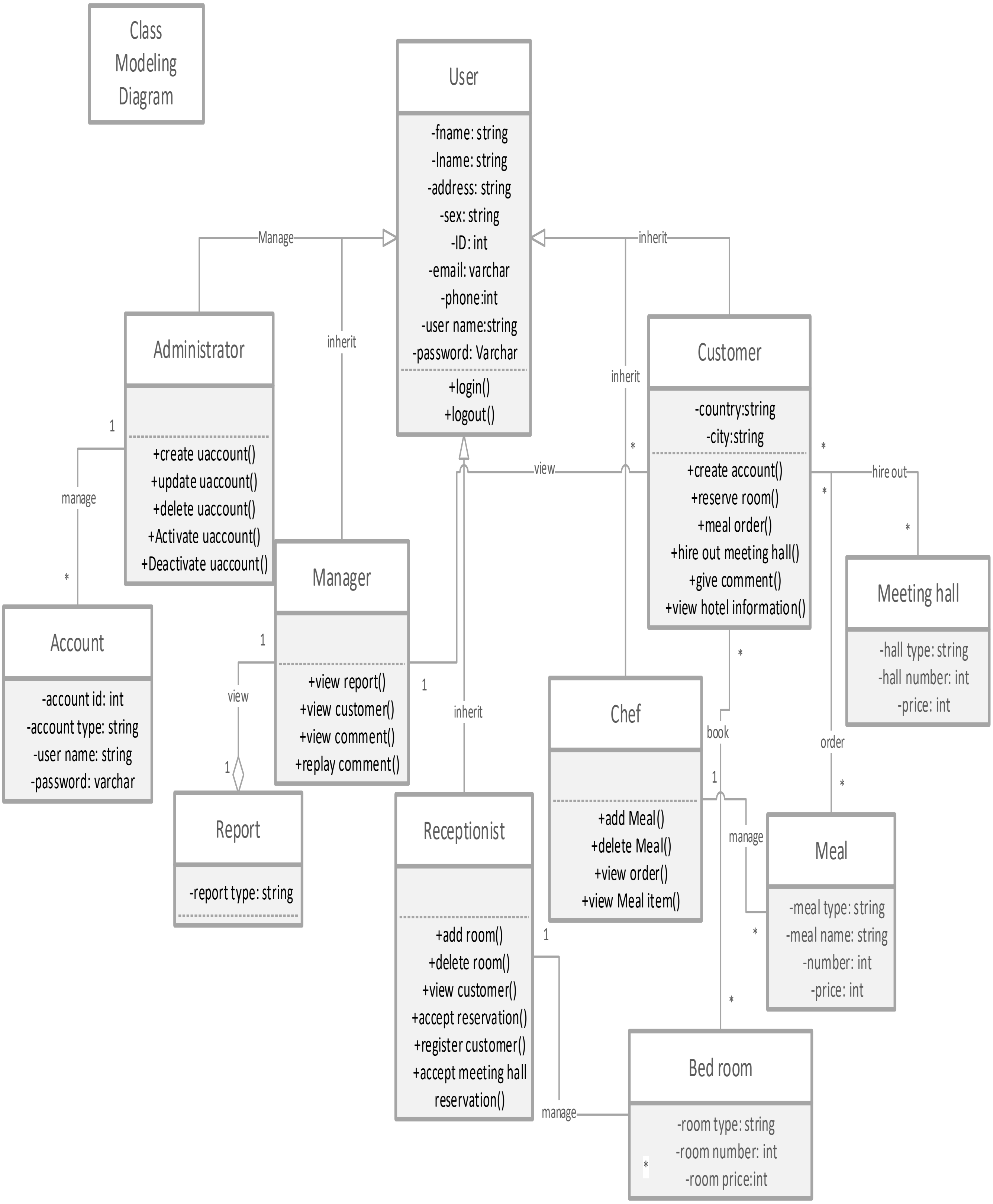


Figure 1. 29 Class modeling diagram

3.4 Proposed software architecture

3.4.1 System architecture

A system architecture is the conceptual model that defines the structure, behaviour, and more views of a system. The system architecture diagram shows the connection between different machines and the path they are connected to each other. The newly developed system uses client/server architecture. In this type of architecture the server is responsible to receive request from the client through internet and respond to the request, whereas the client is responsible to interact with the users of the system.

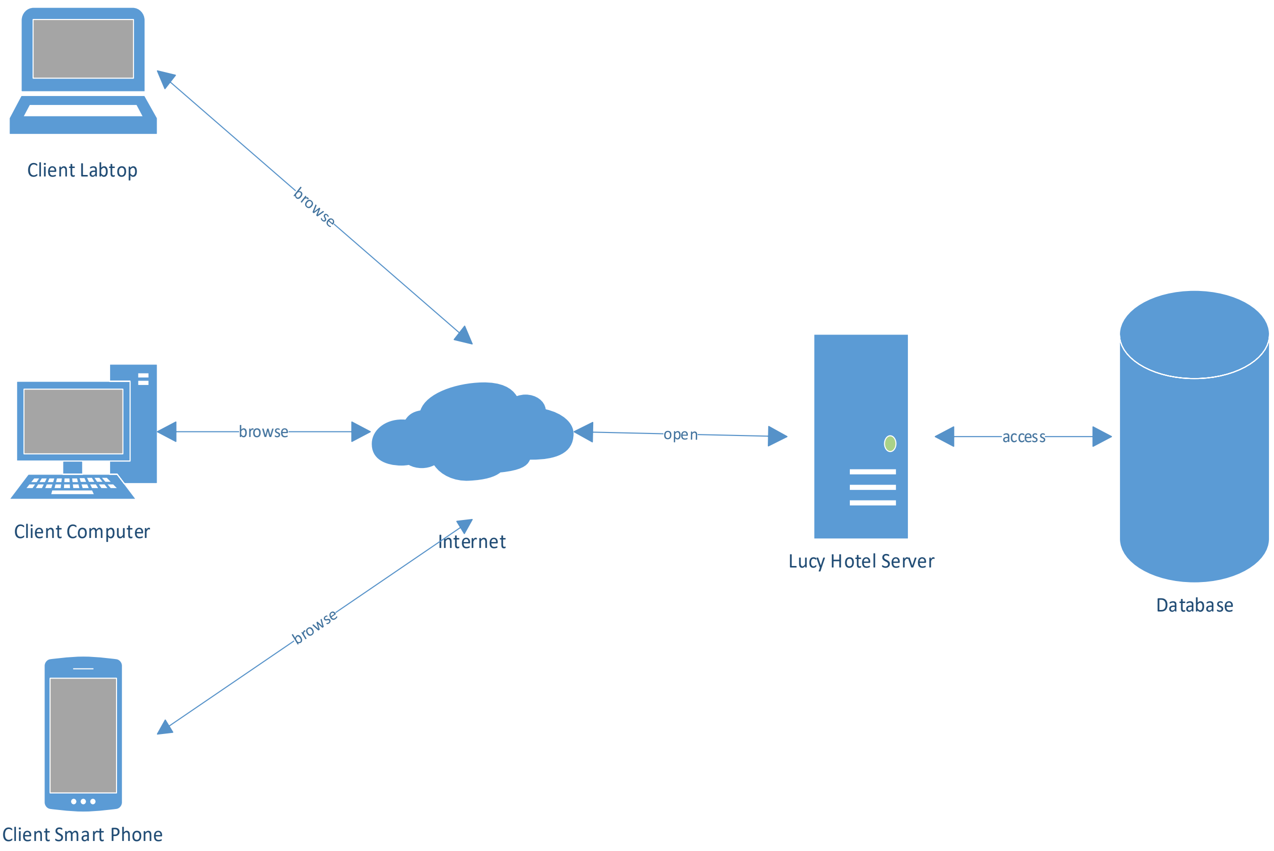


Figure 1. 30 System architecture

3.4.2 Subsystem Decomposition

System decomposition refers to breaking down the system into sub system or easily manageable parts. Subsystem decompositions help reduce the complexity of the system. The subsystems can be considered as packages holding related classes/objects. These subsystems are further decomposed into other subsystems.

To manage or to make simple, this proposed system is divided into many subsystems. Such subsystems are: -

- **Manage room: -** This subsystem also further divided into subsystems such as add room, delete room, and reserve room.
- **Manage meal: -** This subsystem also further divided into subsystems such as add meal, delete meal, and order meal.
- **Manage account: -** This subsystem also further divided into subsystems such as create account, update account, activate account, and deactivate account.
- **Manage report: -** This subsystem also further divided into subsystems such as generate report, and view report.
- **Login: -** This subsystem also further divided into subsystems such as customer login, manager login, receptionist login, Chef login, and administrator login.

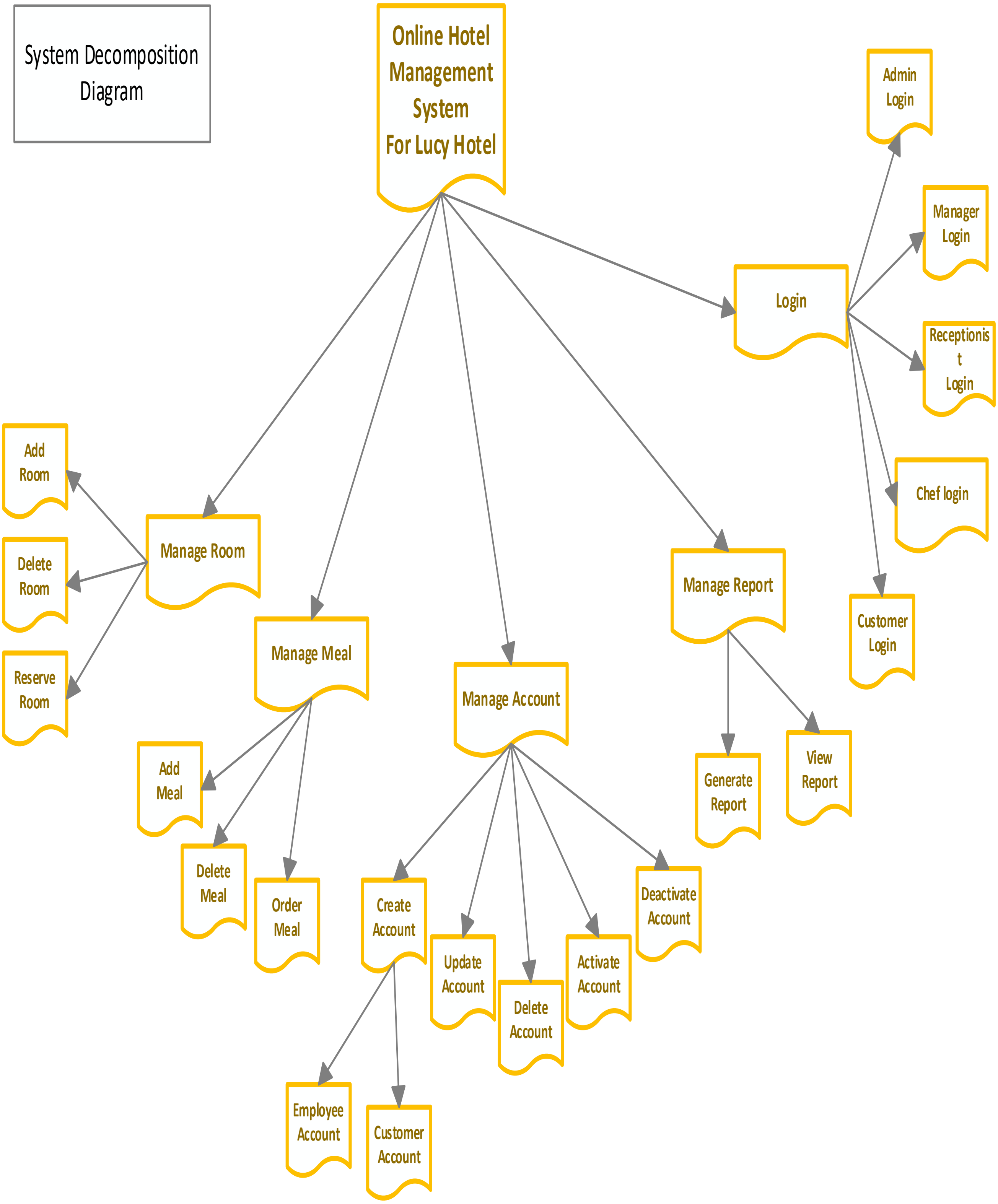


Figure 1. 31 System Decomposition Diagram

3.4.3 Component diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

So, from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So, the purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

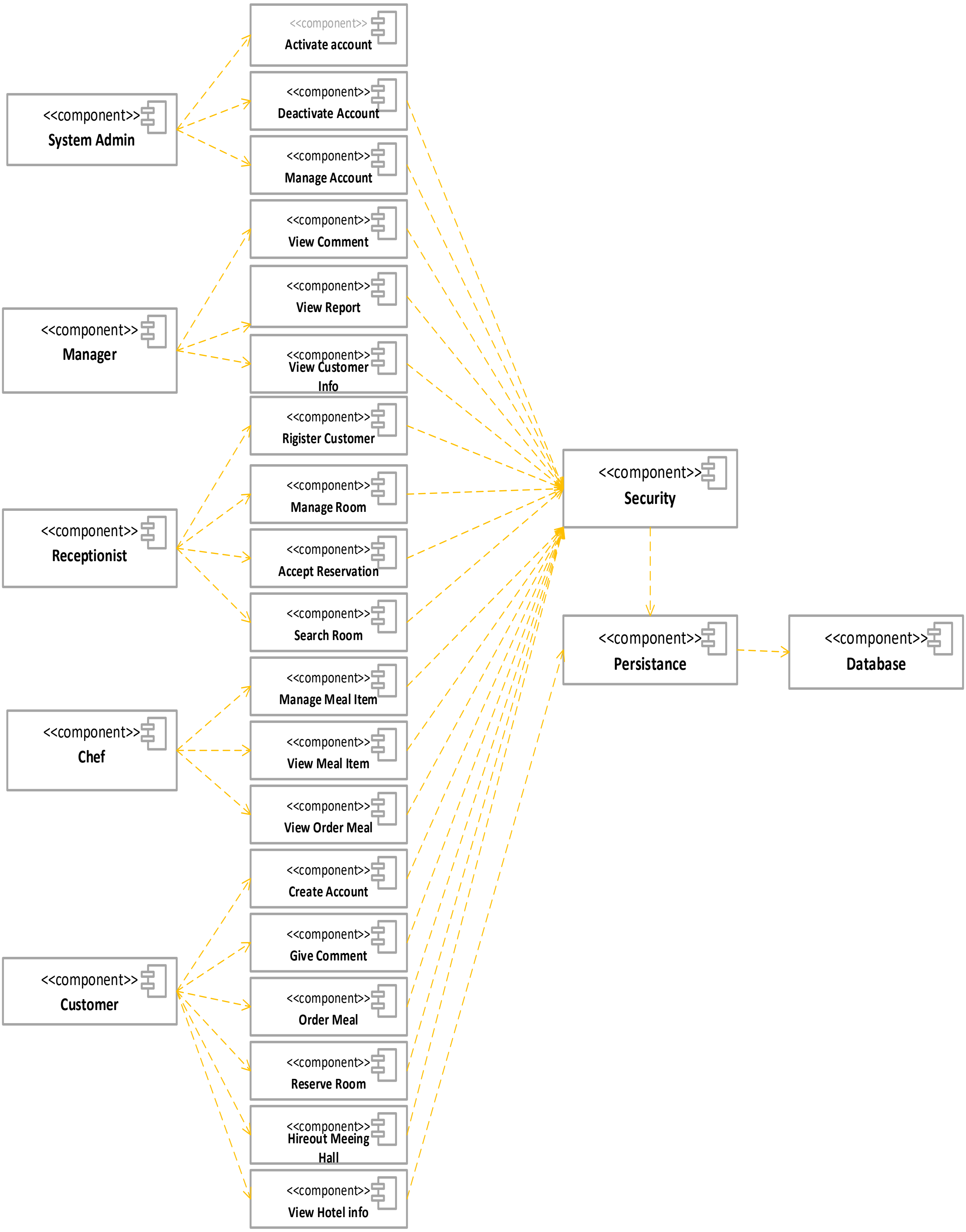


Figure 1. 32 Component diagram

3.4.4 Deployment diagram

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

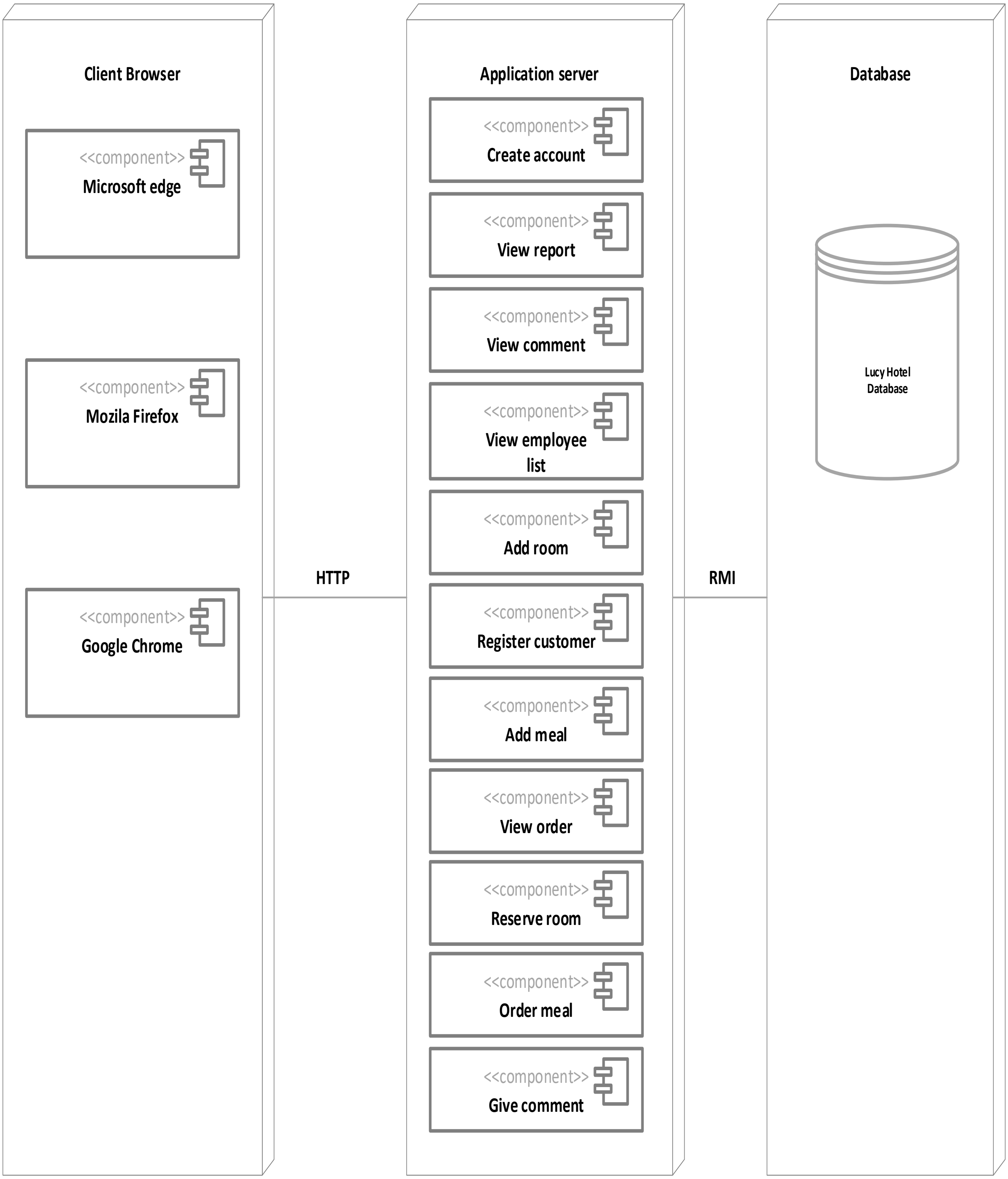


Figure 1. 33 Deployment Diagram

3.4.5 Persistence Modelling

Persistent Diagram is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. It can be thought of as the logical design of the base data structures used to store the data.

Mapping

In order to store information persistently we map objects into tables and the attributes into fields to the specific table based on the objects found on the system. Therefore, we identified four major tables that will be implemented on the selected DBMS. For this reason, the mapping of objects to tables is displayed as follows

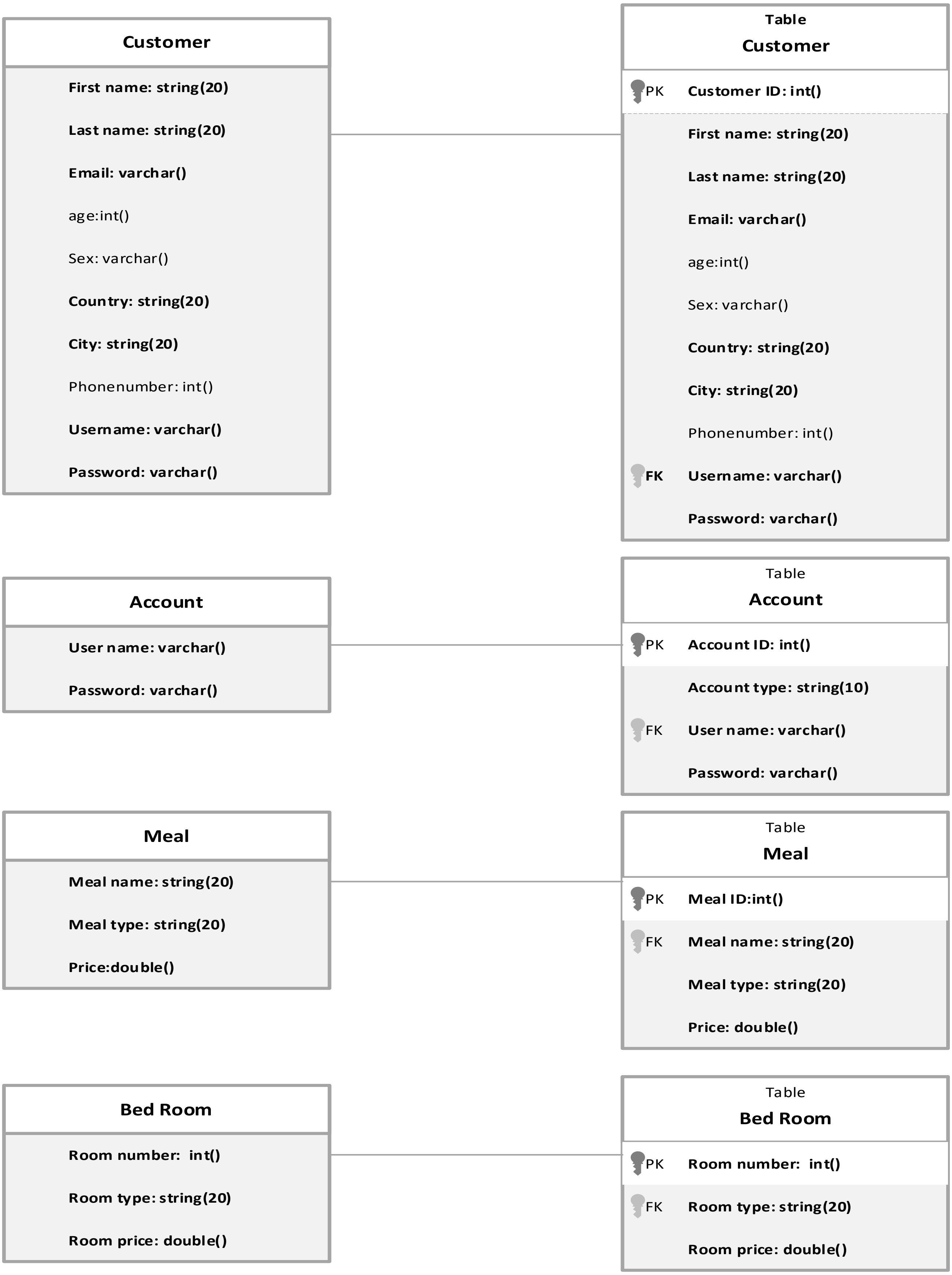


Figure 1. 34 Persistence modelling

3.4.6 Access Control and security

Access control is way of enabling & limiting access to a system or to physical or virtual resources according to access level. A control is a process by which users are denied access to the system or granted access and certain privileges to systems, resources or information. When the user can login, the user will be authenticated and then authorized access to the system.

Example: -

Administrator: Create account, login, activate account and so on.

Manager: Login, view report, view comment, etc.

Chef: Login, add meal, view available meal material, etc.

Receptionist: Login, add rom, delete room, view order and so on.

Customer: Login, book room, order meal, hire out meeting hall, and the like.

Security of the system are: -The system managed by admin to control and to manage user account. The system doesn’t allow anyone to access any service in the system except the authorized users.

Generally, the following table shows the actors that has the authority to access the system and what functionality will perform in that actor.

	System Administrator	**Manager**	**Receptionist**	**Store Manager**	**Customer**
Login	Yes	Yes	Yes	Yes	Yes
Create Account	Yes	No	No	No	Yes
Activate Account	Yes	No	No	No	No
Deactivate Account	Yes	No	No	No	No
Delete Account	Yes	No	No	No	Yes
Update Account	Yes	No	No	No	No
View Report	No	Yes	No	No	No
View Comment	No	Yes	No	No	No
View employee list	Yes	Yes	No	No	No
Search Customer	No	Yes	Yes	No	No
Add Room	No	No	Yes	No	No
Delete Room	No	No	Yes	No	No
Add Meal	No	No	No	Yes	No
Delete Meal	No	No	No	Yes	No
View Order	No	No	No	Yes	No
Book (Reserve) Room	No	No	No	No	Yes
Update Meal	No	No	No	Yes	No
Give Comment	No	No	No	No	Yes
Order Meal	No	No	No	No	Yes
Logout	Yes	Yes	Yes	Yes	Yes

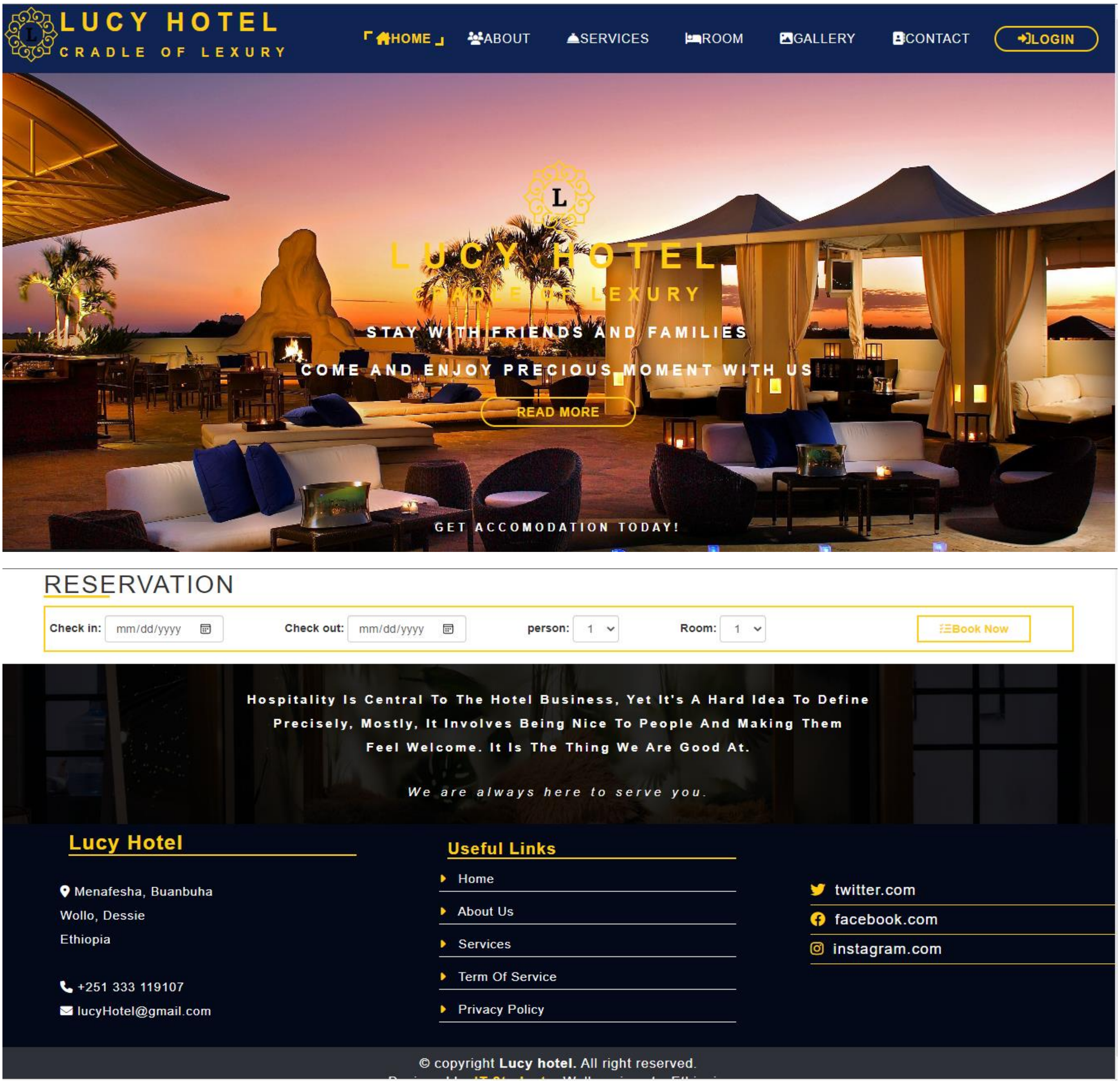
Table 1. 21 System access control

3.5 User-Interface Design

User interface design is the overall process of designing how a user will be able to interact with a system.

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals.

Home Page: This form contains some links which lead it to the concerned page, and if the user has an account he/she will directly go to concerned page by entering their username and password.



STAY WITH FRIENDS AND FAMILIES

COME AND ENJOY PRECIOUS MOMENT WITH US

GET ACCOMODATION TODAY!

Check in: mm/dd/yyyy Check out: mm/dd/yyyy person: 1 Room: 1

Hospitality Is Central To The Hotel Business, Yet It's A Hard Idea To Define Precisely, Mostly, It Involves Being Nice To People And Making Them Feel Welcome. It Is The Thing We Are Good At.

We are always here to serve you.

Menafesha, Buanbuha
Wollo, Dessie
Ethiopia

+251 333 119107
lucyHotel@gmail.com

© copyright **Lucy hotel.** All right reserved.

Figure 1. 35 user interface home page

Log In form:-this form found immediately following the home page. Home page appears as the site on which the system is deployed is opened. All Customer, Receptionist, Waiter and manager will have their own user name password. Those forms appeared using password and user name will not be accessible by other people except for those who have privilege. This website has to login form one form customer and one for reception, manager, chefs and admin.

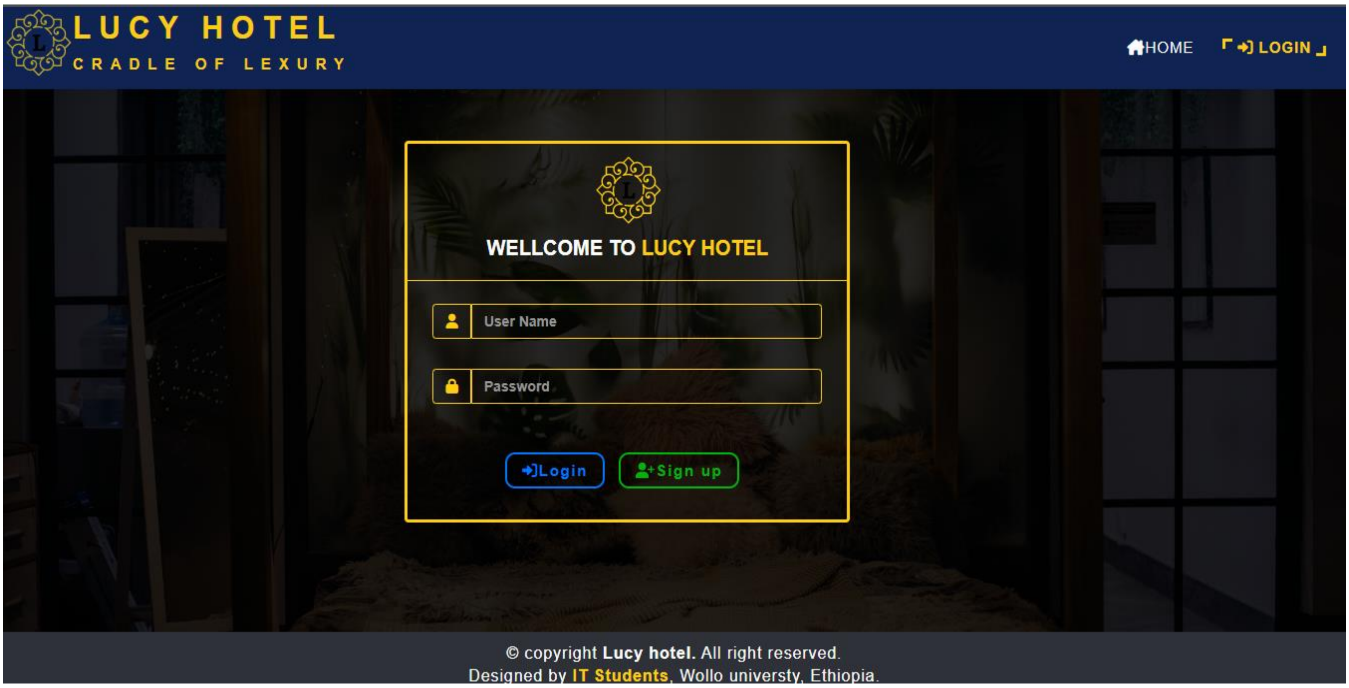

LUCY HOTEL
CRADLE OF LEXURY
HOME
LOGIN
WELLCOME TO LUCY HOTEL
User Name
Password
Login
Sign up
© copyright Lucy hotel. All right reserved.
Designed by IT Students, Wollo universty, Ethiopia.

Figure 1. 36 *customer login page*

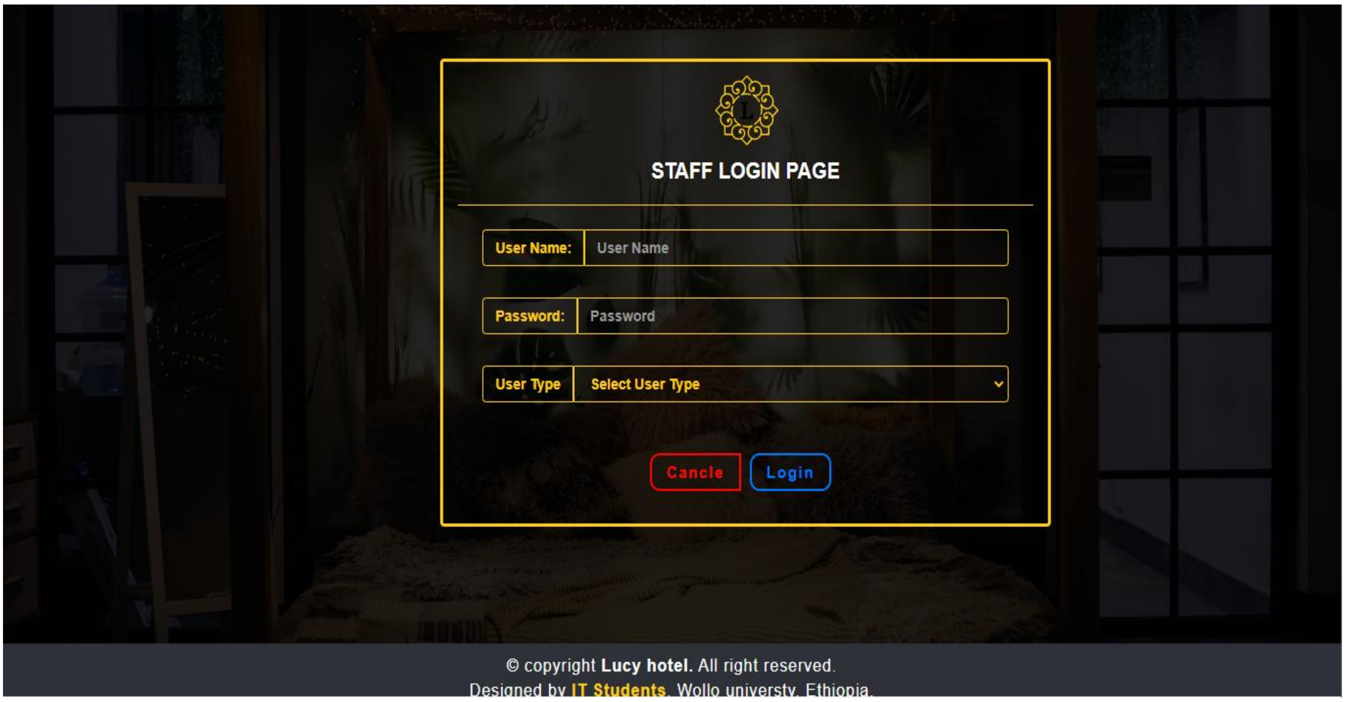
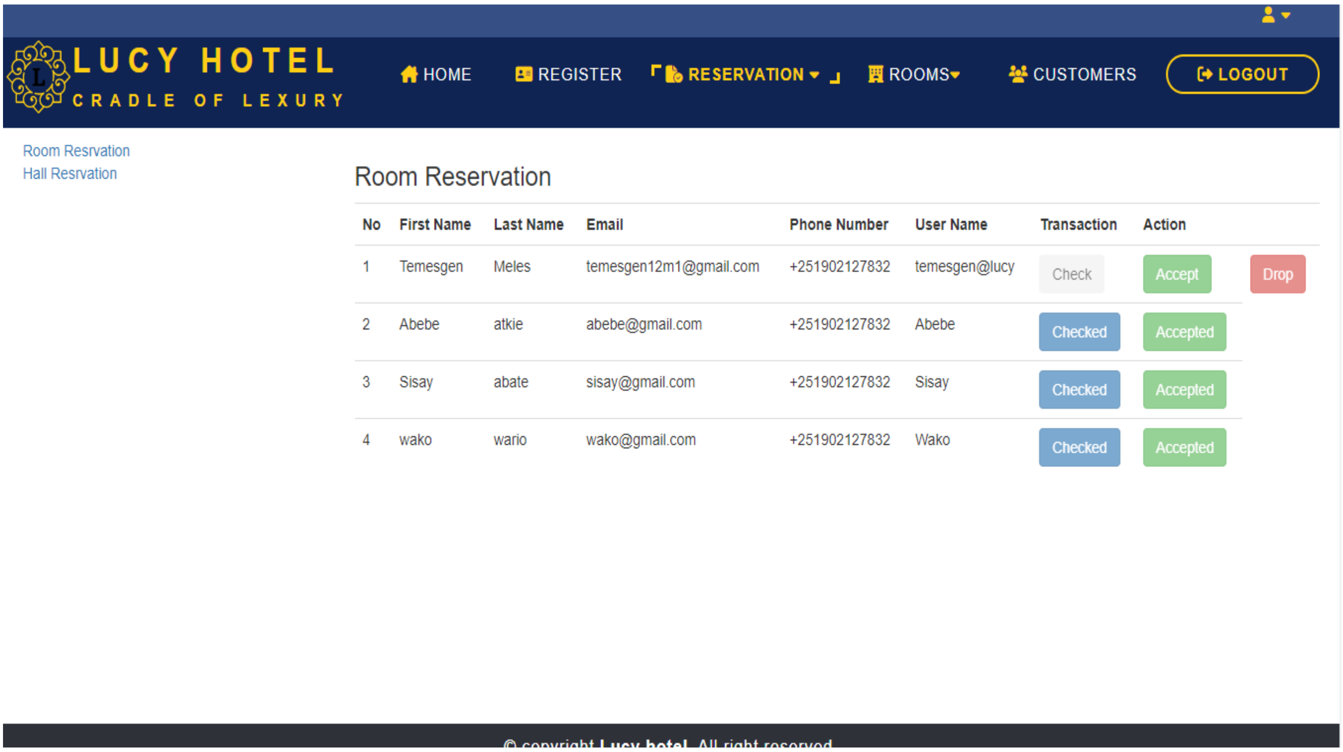

STAFF LOGIN PAGE
User Name:
User Name
Password:
Password
User Type
Select User Type
Cancle
Login
© copyright Lucy hotel. All right reserved.
Designed by IT Students, Wollo universty, Ethiopia.

Figure 1. 37 *staff login page*

View Reservation: This is view reservation page in this page the receptionist he/she can view customer information.

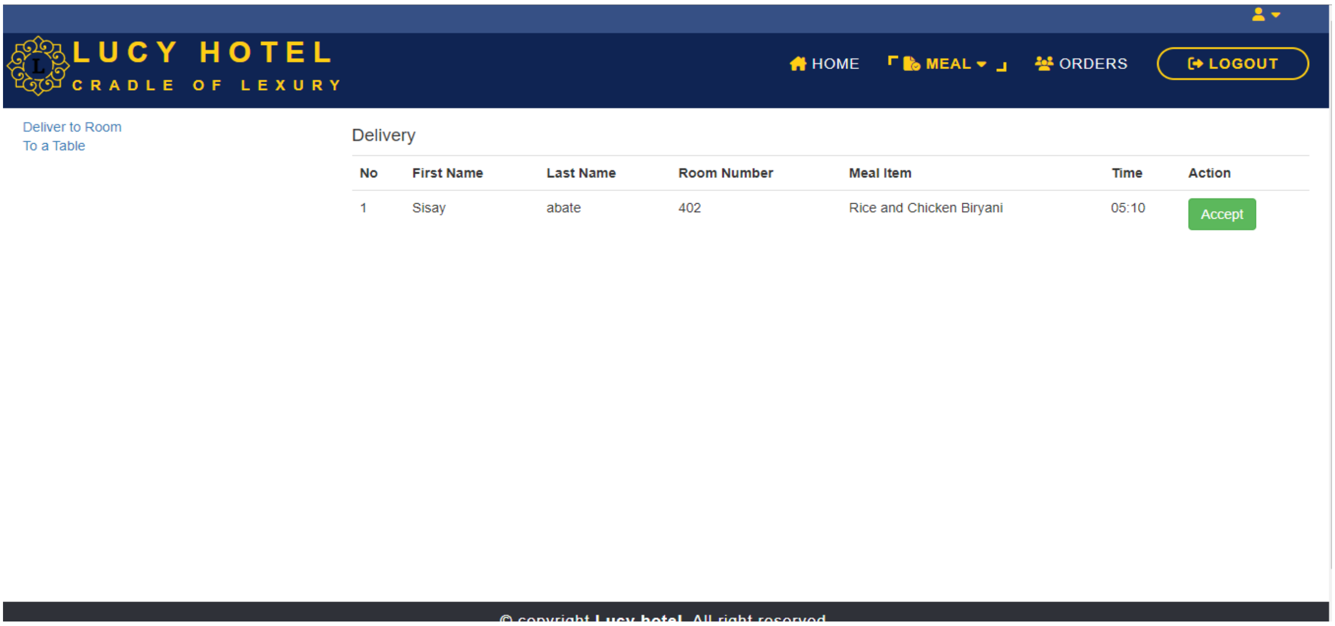


Room Reservation

No	First Name	Last Name	Email	Phone Number	User Name	Transaction	Action	
1	Temesgen	Meles	temesgen12m1@gmail.com	+251902127832	temesgen@lucy	Check	Accept	Drop
2	Abebe	atkie	abebe@gmail.com	+251902127832	Abebe	Checked	Accepted	
3	Sisay	abate	sisay@gmail.com	+251902127832	Sisay	Checked	Accepted	
4	wako	wario	wako@gmail.com	+251902127832	Wako	Checked	Accepted	

Figure 1. 38 *view reservation for reception*

View Order Information: This is view order page in this page the chef he/she can view order information.

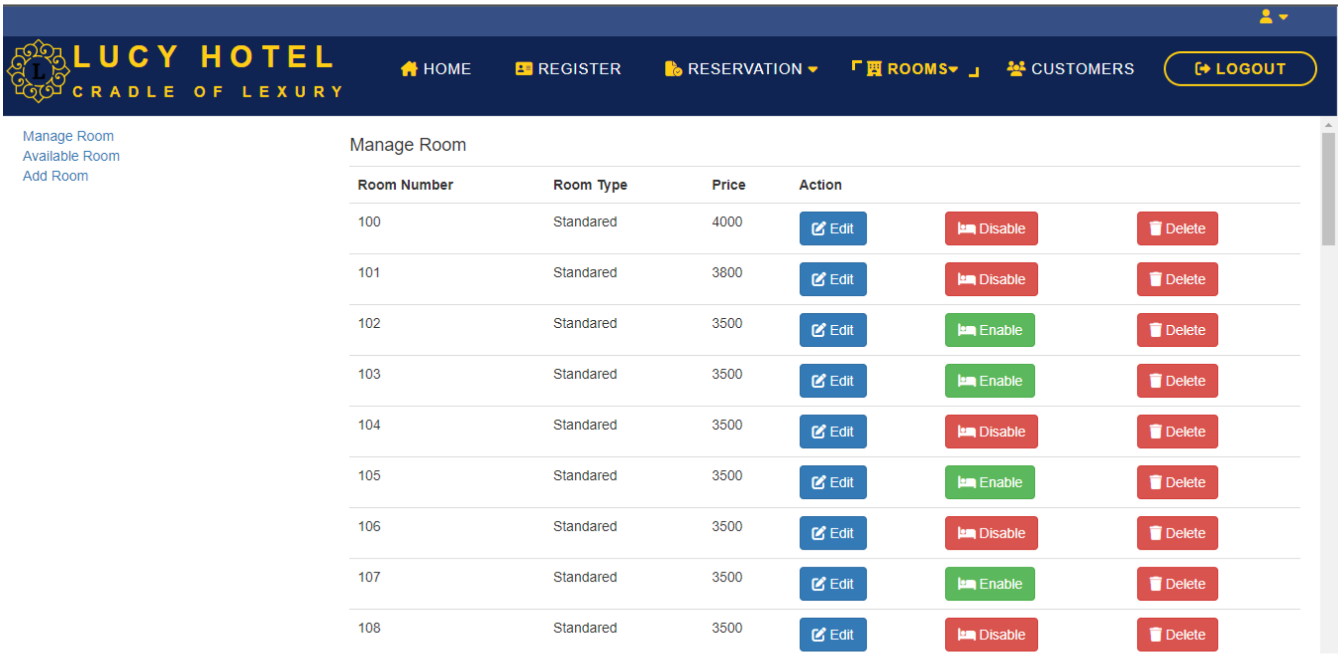


Delivery

No	First Name	Last Name	Room Number	Meal Item	Time	Action
1	Sisay	abate	402	Rice and Chicken Biryani	05:10	Accept

Figure 1. 39 *view order information for chefs*

Manage room Information: This is manage room page in this page the manager he/she can add new room, modify the room information and also deactivate the room if the has a certain item.



Manage Room

Room Number	Room Type	Price	Action		
100	Standared	4000	Edit	Disable	Delete
101	Standared	3800	Edit	Disable	Delete
102	Standared	3500	Edit	Enable	Delete
103	Standared	3500	Edit	Enable	Delete
104	Standared	3500	Edit	Disable	Delete
105	Standared	3500	Edit	Enable	Delete
106	Standared	3500	Edit	Disable	Delete
107	Standared	3500	Edit	Enable	Delete
108	Standared	3500	Edit	Disable	Delete

Figure 1. 40 *manage room information in reception page*

CHAPTER FOUR: IMPLEMENTATION AND TESTING

4.1 Implementation

The physical design specification created by the designers is turned in to working computer code by the programmer using PHP, HTML, Java script and CSS.

4.1.1 Sample code for login:

Login form HTML:

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Lucy Hotel </title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="mystyle.scss">
    <link rel="stylesheet" href="fontawesome/css/all.min.css">
    <link rel="stylesheet" href="loginform.css">
</head>
<body>
<header>
        <nav>
        <img src="image/lucy-hotel_logo3.png" alt="logo" class="my-logo-img">
                <a href="index.php" class="mylogo">lucy hotel<span>cradle of lexury</span></a>


                    <ul class="mymenu">
                        <li class="mymenu-item"><a href="index.php" class="mymenu-link"><i class="fa-sharp fa-solid fa-house-chimney"></i>Home</a></li>
                        <li class="mymenu-current"><a href="loginform.php" class="mymenu-link"><i class="fa-solid fa-right-to-bracket"></i> Login</a></li>
                    </ul>
        </nav>
    </header>
    <?php
    echo @$_SESSION['failed'];
    echo @$_SESSION['sign_up_msg'];
    ?>
    <div class="mycontainer">
        <div class="row">
            <div class="col-md-6 myform">
                <div class="panel panel-primary" style="width: 100%;">
                    <div class="panel-heading">
                        <img src="image/lucy-hotel_logo3.svg" alt=" lucy hotel logo">
                        <h4>Wellcome to <span>Lucy Hotel</span> </h4>
                    </div>
                    <div class="panel-body">
                        <form action="login.php" method="post">
                            <div class="input-group form-group">
                                <span class="input-group-addon"><i class="fa-solid fa-user"></i></span>
                                <input type="text" name="username" id="username" class="form-control" placeholder="User Name">
                            </div>
                            <div class="input-group form-group">
                                <span class="input-group-addon"><i class="fa-solid fa-lock"></i></span>
                                <input type="password" name="password" id="pwd" class="form-control" placeholder="Password">
                            </div>
                    </div>
                    <div class="panel-footer">
                            <button type="submit" name="login" class="button_login"><i class="fa-solid fa-right-to-bracket"></i>Login</button>
                            <a href="signup.php"><button type="button" class="button_signup"><i class="fa-solid fa-user-plus"></i>Sign up</button></a>



                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="footer">
        <div>&copy; copyright <strong>Lucy hotel.</strong> All right reserved.</div>
        <div>Designed by <a href="#"> IT Students</a>, Wollo universty, Ethiopia.</div>
    </div>
</body>
</html>
```

Form style sheet:

```
:root{
    --primary-color: #ffce14;
    --secondary-color: #0f2453;
    --white-color: white;
    --black-color: black;
    --blue-color: rgb(0, 119, 255);
    --green-color: rgb(2, 180, 11);
}
```

```
*{
   padding: 0px 0px;
   margin: 0px 0px;
   text-decoration: none;
   list-style: none;
   box-sizing: border-box;
}
.mycontainer{
   height: 80vh;
   background: linear-gradient(rgba(0, 0, 0, 0.85), rgba(0, 0, 0, 0.85)),
url("http://localhost/lucy hotel/image/hotel-bg.jpg") center/cover no-repeat;
}
.mycontainer .row{
   padding: 0px 0px;
   display: flex;
   justify-content: space-around;
   padding-top: 50px;
   padding-left: 150px;
}
.mycontainer .row .panel{
   border: 3px solid var(--primary-color);
   background-color: transparent;
}
.mycontainer .row .panel-heading{
  background-color: transparent;
  border-bottom: 1px solid var(--primary-color);

  text-align: center;

}

.mycontainer .row .panel-heading h4{

   text-transform: uppercase;

   font-size: 20px;

   font-weight: bold;

   text-align: center;

}

.mycontainer .row .panel-heading h4 span{

   color: var(--primary-color);

}

.mycontainer .row .panel-body input ,

.mycontainer .row .panel-body span{

   background-color: transparent;

   border: 1px solid var(--primary-color);

   color: var(--primary-color);

   font-weight: bold;

}

.mycontainer .row .panel-body input:focus{

   background-color: transparent;

   color: var(--primary-color);

   font-weight: bold;

}
```

```
.mycontainer .row .panel-body .form-group{

   padding: 7px 10px;

}

.mycontainer .row .panel-footer{

   background-color: transparent;

   border-top: none;

   text-align: center;

   margin-bottom: 20px;

}

.mycontainer .row .panel-footer .button_login{

   background-color: transparent;

   border: 2px solid var(--blue-color);

   padding: 5px 15px;

   color: var(--blue-color);

   letter-spacing: 1.7px;

   font-size: 15px;

   font-weight: bold;

   border-radius: 10px;

   margin-right: 10px;

}

.mycontainer .row .panel-footer .button_login:hover{

   color: var(--white-color);

   background-color: var(--blue-color);

}
```

```
.mycontainer .row .panel-footer .button_signup{

    background-color: transparent;

    border: 2px solid var(--green-color);

    padding: 5px 15px;

    color: var(--green-color);

    letter-spacing: 1.7px;

    font-size: 15px;

    font-weight: bold;

    border-radius: 10px;

    margin-right: 10px;

}

.mycontainer .row .panel-footer .button_signup:hover{

    color: var(--white-color);

    background-color: var(--green-color);

}

.footer{

    position: relative;

    padding-top: 10px;

    padding-bottom: 10px;

    font-size: 16px;

    letter-spacing: 0.9px;

    color: var(--white-color);

    background-color: #2f3138;

    text-align: center;
}
.footer a{
    text-decoration: none;
    color: var(--primary-color);
    font-weight: bold;
}
.footer a:hover{
    color: var(--primary-color);
    font-size: 17px;
    font-weight: bold;
}
```

Form PHP file:

```
<?php
session_start();
include('connection.php');

if (isset($_POST['login'])) {
    $uname=$_POST['username'];
    $pass=$_POST['password'];

    $query = "SELECT customer_ID,user_name,password from customer_information where user_name='$uname'";
    $result = mysqli_query($db,$query);
    $numrows = mysqli_num_rows($result);
    echo $numrows;
    if($numrows == 1){
        $row = mysqli_fetch_array($result);
        $_SESSION['ID'] = $row['customer_ID'];
        if ($row['password'] == $pass) {
            header('Location: http://localhost/lucy hotel/Customer/home.php');
        }
        else {
             $_SESSION['failed']  = "User name and password doesn't match";
             header('Location: http://localhost/lucy hotel/loginform.php');
        }

    }elseif ($numrows == 0) {
        $_SESSION['failed'] = "User name not found";
        header('Location: http://localhost/lucy hotel/loginform.php');
    }else{
        echo "Error found more than one account";
    }
}
?>
```

4.2 Choose Your Test Approach

Black box testing*:* -To test our system, the tester may use black box testing, if he/she has not enough time to check internal modules or codes. By looking only input /output or user interface, the tester can test our systems functionalities without looking the internal code. We used this testing technique for the following reasons: -

- This testing type is more effective on larger units of code
- Tester needs no knowledge of implementation, including specific programming languages
- Tester and programmer are independent of each other
- Tests are done from a user's point of view

White box testing: -

In this type of testing, skilled man in different programming languages tries to test the logic of our system. If the person who tests the system is not skilled, it is difficult to understand our systems functionality. If any failures occur while testing the system in all of the above testing methods, the team will take immediate correction where this fault occurred before jumping to next work. So, that it will meet the goal.

Unit Testing: -

It is a way of testing each module in the system independently. The group member has tested the major activities to accomplish the goal using different inputs, different login mechanisms and any methods of fault finding.

4.2.1 Functional Test Specifications

Test case specification

Log in

Test Case 1 user login		
Expectations = Successfully Logged in!		
Test Data = user name (invalid user name, Valid user name, Fill out the field) password (invalid password, valid password, Fill out the field)		
Steps to be Performed	**Data**	**Expected Results**

Empty user name and password and Click Login button	Username= ------------------
	Password= -------------------
		“Please fill out this field”

Invalid user name and password Click Login button	User name = wrong username Password= wrong password	“Sorry, wrong username or password please insert again”
Enter valid user name, empty password and Click Login button	User name = temesgen@lucy Password= ----	“please fill outthis field ”
All fields with valid data and Click Login button	All fields with valid data	"Successfully login thank you!"

Table 1. 22 Test case specification

Test Description: The login test is used to check whether anyone can access the system or only those registered Users have an access to the system

Test case Name: Login

Purpose of test: authorization test

Testing objects: **Login checking**

Test focus: correct username, Password validation

Test Process

1. **Starting condition**

❖ The user should enter username and password then click login button

❖ Go to the target page

2. **Expected result**

❖ At the time of sign-in, if the user name, password is valid target page will be displayed and accessible to the user.

3. **Failure Condition**

❖ If user enters random username and password, write the correct once message will be displayed. If password or user name is not enters, please enter your username or password message will be displayed.

CHAPTER FIVE: CONCLUSION AND RECOMMANDATION

5.1 Conclusion

As we come to end of our system development, it is time to reiterate the work previously have been done and check against our success indicators set at the beginning of the project to determine whether the endeavour was a success.

The project began by laying out the foundation that dictates the development process. This involved defining the system development methodology, identifying resource requirements, and setting the project schedule.

We performed requirements analysis to discover the needs of the new solution to proposed system. This phase consists of drawing out functional and non-function requirements of the system.

In analysis, we tried to model the new and proposed system using UML diagrams: use case diagrams, sequence diagrams, Class diagrams, Activity diagrams. Domain modelling was handled using Class Responsibility Collaboration (CRC) model, while we gave a sneak preview of what the new system might look and feel with our user interface design and documentation.

In design phase, we extended our work in analysis with more models. The class type architecture, Class diagrams in analysis were extended in design to step closer to implementation, while deployment and component diagrams were also drawn. We also used persistence modelling to have a feel of the database that would accompany the system.

A dawn side to this project was probably the persistence modelling we did not manage to undergo due to time constraint. Still, we are limited to persistence model was carried with real data obtained partially, in addition from the general overview of another institute spatially personnel information providing limited source from what we are required.

Our development team believes that we have at large accomplished what we set out for in the beginning of this project. We set to develop a working system prototype that would solve the problems in the beginning of our project. But before winding up, we would like to give our final word on how to go from here by recommending further improvement that could be made in the future.

5.2 Recommendations

Having seen the problem encountered in using the old system of hotel management system, we suggest that this Online Hotel Management System be adopted in hotels and staff that will be handling this software package be given training on computer based on how to use the new system.

From the various facts that constitute the project, the following recommendations have come to light:

- User must have a basic knowledge of computer.
- User must have a basic first-hand experience with online systems.
- User must have a basic skill on how to use browser software.
- Training user should be considering for best performance and efficiency of technology.
- Organization should have adequate computer facilities for the introduction of the new system.
- Behavioural change of the users of the system to adapt the new system.

Reference

From website

[1] https://en.wikipedia.org/wiki/Hotel

[2] https://en.wikipedia.org/wiki/Dessie

[3]. https://www.lucidchart.com/pages/use-case-diagram-for-hotel-management-system-UML

From book

[4]. Practical Software Development using UML book Timothy C. Lethbridge,5th edition.

[5] SharrafHussain (2011) Introduction to Unified Modelling Language (UML)