WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE:ONLINE HOTEL RESERVATION SYSTEM

A Project Submitted to the Department of Computer Science of Wolkite University in Partial Fulfillment for the Requirement for the Degree of Bachelor of science in Computer Science

Project advisor: Yetmwork T.

Wolkite University,Wolkite,Ethiopia
August 2021

DECLARATION

This is to declare that this project work which is done under the supervision of Miss.Yetmwork T. and having the title Hotel Reservation system for Soresa Hotel.

Date: _____

Group Members:

Full Name	Signature
1. Obsa Bekele	_____
2. Abinet Getachew	_____
3. Abdulkerim Mohammed	_____

APPROVAL FORM

This is to confirm that the project report entitled Hotel reservation system for Soresa Hotel submitted to **Wolkite University, College of Computing and Informatics Department of Computer Science** by: Obsa Bekele, Abinet Getachew, Abdulkerim Mohammed is approved for submission.

Advisor Name Signature

Date

Department Head Name Signature

Date

Examiner 1 Name Signature

Date

Examiner 2 Name Signature

Date

Examiner 3 Name Signature

Date

ACKNOWLEDGEMENT

The Almighty God, who gave us the insight and patience to accomplish this work. Special thanks for our Advisor Mss. Yetmwork T. for continuous support in every phase of the project and who continuously provides us her valuable advice to prepare this project document. Also thanks to the Soresa Hotel workers who gives for us a correct requirement for the system developed by our group members.Our appreciation also extends to our instructors who helps us to develop this project.Lastly we would like to acknowledge our friends for their moral and idea support.

ABSTRACT

In our project, on “Online Hotel Reservation for Soresa Hotel**”,** we have tried to show how the reservation in hotels is managed. This has been achieved by dividing project into various modules. Customer is proved with different services like checking in, checking out, booking, and editing entries or payment. If the customer wants he/she can cancel his/her booking. Enquiry about any customer or employee can be made either by customer Id or customer name. Enquiry about rooms available can also be made. It will generate reports for customer, employees (working in hotel) and Bill for customer is generated when the customer will check out from the hotel.Also we used Iterative development life cycle to develop our project and ASP.NET in order to make system secure, because of web based system.

Table of contents

List of tables

List of figure

Abbreviation and acronyms

- BR ……Business Rule
- MVC……..……Model View Controller
- GUI …….Graphical User Interface
- MYSQL……My structural Query Language
- OOA……..……Object Oriented Analysis
- OOD…….Object oriented Design
- SD…….Sequence Diagram
- UML…….Unified Modeling Language
- UI……..User Interface
- UC…….Use Case
- IIS……..Internet Information system
- CBE……..Commercial Bank of Ethiopia

CHAPTER ONE

1.1. Introduction

Technology spreads its wings in almost every walk of human life activity. Now a day its better if everything is done using new technology in order to fulfill the need of human being, organization, enterprise etc. As today's world there are many organizations and each organization need to be preferable, computable work on faster way in order to satisfy user interest i.e. they should have facilitate their activities in computerized way.The Hotel Industry like any other business opens up socioeconomic opportunities for both owner and customer. It has the function of providing hospitality services to customers.These customers can be travelers, foreigners, businessmen, tourists, visitors, etc. Customers are mostly constrained in trying to get a room to pass the night, as the usual practice is to look for a hotel when you have arrived in the particular location, walk in and find out whether there is a vacant room. In the case that there is no vacant room, you have to move to next closest hotel to enquire once more. You would be lucky to go and get a room booked for you. They are quick to serve those who walk in rather than those who may get access to them on phone to book a room. On other times too, if you have friends or family members in the area you want a room booked, they have to go and do the checking for you. There is no system in place that bonds the hotel and the customer that the customer has actually booked a room and for that matter he is guaranteed a room. This can make customers really stranded especially if it is getting late in the night. Now a days ICT (technology) makes life easier for the people. So we want to develop this reservation system to make reservation easier for the customers mainly and for the hotel.

1.2. Background of the organization

Soresa hotel is found in Wolkite,Gurage,Ethiopia which is 165km away from the capital Addis Abeba. The name soresa means rich. It was established in 2000e.c. by the owner Mr.Gizaw G/Michael with 20-30 employees. Currently the hotel has around 52 employees. 32 males and 20 females. Currently the hotel has 73 rooms which are three types (normal bed room, first class bed room, and double bed room) with different prices. First class and double bed room have shower(cold and hot), free internet access, telephone for room dinning, and free breakfast. The normal bed room has only bed and free breakfast. It was established to provide accommodation and

catering services for tourists, locals, foreigners, who are in wolkite and around to trade, rest, eat as well as those on vacation. The hotel have other special service called room service (it means that for example if the customer reserve the room by double bed room price 700birr, then they can use the normal bed room which is 250birr and they can use the rest of money 450birr for foods and drinks).this makes the hotel very marketable. Soresa Hotel uses manual reservation (booking) system.

1.3. Statement of the problem

In the existing system reservation and other services are done manually. So it wastes more time for the manager to manage the whole activities in the hotels, like how the customer to serve the room, and other services and even how to collect payments from individual customers, for all the old system has many problems and short comings which are listed below.

- The whole service of the hotel is done manually.
- Losing of data and information is simple because of insecurity.
- Management process of the hotel is very difficult and complex.
- There may be making mistake in calculating of the price of services.
- It consumes time, man power and economy.
- There is no data or information storage devise on the center.
- It is difficult to collect payments.
- The manual system is costly due to the use of paper works.
- It is difficult for the user to know which room is not reserved.
- Existing system require more labor force, workers or employees.
- Any enquiry to be made demanding feedback must usually be forwarded to the hotel in person.
- Sometimes management is given false reports concerning the work flow of the hotel.
- Employees also gives falsified pricing information to customer from time to time.
- Reports are not properly kept.
- Advertises depends on words of mouth.
- Customers cannot access hotel information.
- Data is viewed by unauthorized person.

1.4. Objectives of the project

The aim of our project is that in order solve problems that found in Soresa Hotel and to gate tangible and measurable out come by using the given time.

1.4.1. General objective

The main objective of this project is to design and develop web based online hotel reservation for Soresa Hotel.

1.4.2. Specific objectives

Specific objective of the project are the following:

- To change the manual reservation system in to advanced method of web based system.
- To reserve bed rooms and other services.
- To access the hotel information.
- To solve complicated problems that faced to reserve with the manual system.
- To reduce the time, cost and man power needed to job performance.
- To provide efficient service to the customers.
- For generation of reports.
- To solve data redundancy.
- To create user friendly environment.
- To keep data security and reliability.
- To advertise the hotel

1.5. Feasibility analysis

Feasibility study is used to investigate the proposed system in multiple dimensions. It deals in different aspect of either the system is visible or not.

1.5.1. Technical feasibility

Technical feasibility is the measure of practicality of the specific technical support, and the availability of technical resource and expertise to use the system. The proposed system can be easily maintained and repaired without requiring high experts or technical supports because the system will be developed by familiar programming languages. So the system is technically feasible.

1.5.2. Operational feasibility

Employees and customer are not happy due to having many problems for the existing system, like manually and takes time to do there action in order to solve this type of

problem we are going to do an automated system of online hotel reservation. After we will develop this online hotel reservation it will reduce problems of existing system such as it reduce work rate of employees and customers can easily access our hotel, and decrease many tedious activity. After we say this customer and employer will happy after we develop this system, so it is operational feasible.

1.5.3. Economic feasibility

The cost and benefit analysis has shown that cost that have incurred in developing the project is less than the benefits that the project is going to provide once it is developed. It is possible to get positive result by subtracting the cost break down of the current system from the previous system. Getting positive result informs the economic feasibility. Therefore, the system is economically feasible and has passed the feasibility test.

1.5.4. Schedule feasibility

The system after development may give efficient and effective service in a short period of time. And also the task is scheduled for effective of the system. Whatever the scarcity of the time given for the project, by the internal motivation and potential of the team members of the project, we surely expect the project will be completed on time.

1.6. Scope and limitation of the project

1.6.1. Scope of the project

- The system enables reservation for customer.
- Online payment with internet banking.
- Cancel the reservation.
- The system notifies qualities (facilities) of the hotel to the customer.
- The system generate reports about room reservation.
- The system displays food and drink menus.
- The system display tourism places around the gurage zone.
- The system displays the services the hotel give.
- The system update room information and reserved status.

1.6.2. Limitation of the project

- The system does not allow the customer to order food and drinks.

1.7. Significance of the project

- The system is significance in improving time management, time is very vital.
- The system reduce work load.
- The usage of the system makes work easier.
- The system increase the security of data.
- The system increase competency of the hotel.
- Customer can make reservation from their own place.
- Minimize cost operation.

1.8. Beneficiary of the project

- **For the customer**
- It reduce time and money because they can reserve from wherever they are.
- It helps them to know the quality of the hotel from their place.
- It helps for security of their.
- It enables to get fast and best service
- Payment are easier and faster.
- They can reserve 24/7.
- Instant answer to their availability inquiries.
- **For the hotel**
- Hotel would be competitive in wolkite.
- It reduces labor work.
- It helps for the security of the data.
- It helps to reduce the work load of the manager.
- It makes the auditing easier.
- It helps to promote the hotel.
- Increase hotel revenue.
- **For the developers team**
- Create work opportunity.
- Increase knowledge.
- Initiate to develop other new systems.

1.9. Methodology of the project

There are many techniques of data collection. Among them the team used the following methods for collecting data.

1.9.1. Data gathering methodologies

We have used different mechanism in order to get a sufficient information about the existing system.

Interview

To get basic information and background information about the existing system, the team member has interviewed the manager of the hotel. The interview is conducted face to face and by telephone.

Observation

Another method we used for data gathering was observation. It helps us to get real information how the organization performs its function, and this helps to strength the data gathered through interview.

Document analysis

The other method we used for data gathering is analysis the documents that used for existing system.

1.9.2. System analysis and design

Among the different methodologies available we plan to use the object oriented design methodology for the development of our system.

Object Oriented Analysis (OOA): During this phase we will look at the problem domain, and with the aim of producing a conceptual model of the information that exists the area which will be analyzed. And this Model the functions of the system (use case modelling), the objects and also the relationship between them and finally model the behaviour of the objects.

Object Oriented Design (OOD): During this phase Model object 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 imposed to our system format, so that we will use this phase to refine the use case model to reflect the implementation environment.

1.9.3. System development model

Iterative model

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is

complete. The iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

1.9.4. System testing methodology

As we used the iterative system development model we have to use all (unit, integration, system, and acceptance) testing methodology.

Unit testing

While coding we performs some testes on the unit of the program, to know if it is error free.

Help us to decide that the individual unit of the program working as required.

Integration testing

It helps us to know if the unit integrated together would also work without error.

System testing

- **Functionality testing**

Testing all functionality of the system relative to the requirement.

- **Performance testing**

Testing the effectiveness and average time taken by the system to do the desired task.

Load testing and stress testing when the system is put under high user and data load under various environment condition.

- **Security and portability testing**

Security testing: security testing involves the testing of software, in order to identify any flow and gaps from security and vulnerability point of view. The main aspects of security testing are.

- Authentication.
- Availability.
- Input checking and validation.
- Software data is secure and etc.

Portability testing: portability testing of software with intended that it should be re-usable and can be removed from another software as well. Strategies that can be used for portability testing are building executable and run software on different platform. Portability testing can be considered as one of the sub parts of system

testing, as this testing type includes overall testing with respect to its usage over different environment. Computer hardware, operating system and browsers are the major focus of portability testing.

Acceptance testing

Acceptance testing, is a testing technique performed to determine whether or not the software system has meet the requirement specification. The main purpose of this test is to evaluate the systems compliance with the business requirements and verify if it it has meet the required criteria for delivery to end user.

1.9.5. Development tools and technologies

Tools used to design the system requirement documentation and implement.

We have used different features of development tools and in front end and back end in order to do a well functional project.

1.9.5.1. Front end technologies

- ✓ Asp.Net MVC

  Why Asp.net because of:

- ➢ Easy and friction-less testability
- ➢ Full control over your HTML, JavaScript , and URLs
- ➢ A new presentation option for ASP.Net
- ➢ A simpler way to program Asp.Net
- ➢ Clear separation of logic: Model, View, Controller
- ➢ Test-Driven Development
- ➢ Support for parallel development

- ✓ JavaScript and Html

  1. We used JavaScript to load the content in database or from the page
  2. Since we are developing web based system for that Html must need.

1.9.5.2. Back end technologies

- ✓ MySQL Server

  1. We used to store data in database because of we are developing dynamic web page. And to backup information.

1.9.5.3. Documentation and modelling tools

- MS word: for documentation.
- Power point: for presentation.
- Edrawmax: for UML diagram.

1.10. Document Organization

Chapter one contains about the problem of statements, objectives, methodology, importance of the project and others. Chapter two contains about existing system of drawback, functions and user. Chapter three contains proposed system of functional and non-functional requirements. Chapter four contains about system models such as class diagram, sequence diagrams, state chart, use case diagrams, use case description and others. Chapter five contains of solution part which are design of system to simplify tasks of problems such as subsystem decomposition, hardware/software mapping, packages and some pseudo codes

CHAPTER TWO

2.1. Introduction to existing system

The current Soresa hotel Room reservation system is more of manual where everyone has to make reservations by presenting himself/herself physically to the hotel. As stated in the first chapter, these types of reservation has drawn backs like unwanted time waste for both the hotel and the customer, unstructured customer data recording and manual billing system. Though the communication between the customer and the hotel is geographically apart, there is no best way as such (reserving online) which decreases workloads in both sides (hotel and customer).Having a system that creates a common platform for communication of the hotel and customers would be helpful to realize the vision of the hotel.

2.2. User of existing system

Users of existing system are:-

- ✓ Manager
- ✓ Receptionist
- ✓ customer

2.3. Major function of the existing system

The major Function of existing system:-

- ✓ Provide room reservation service to the clients
- ✓ Provide food and drink service to the clients
- ✓ Provide internet access
- ✓ Generate report

2.4. Form and other document of the existing system

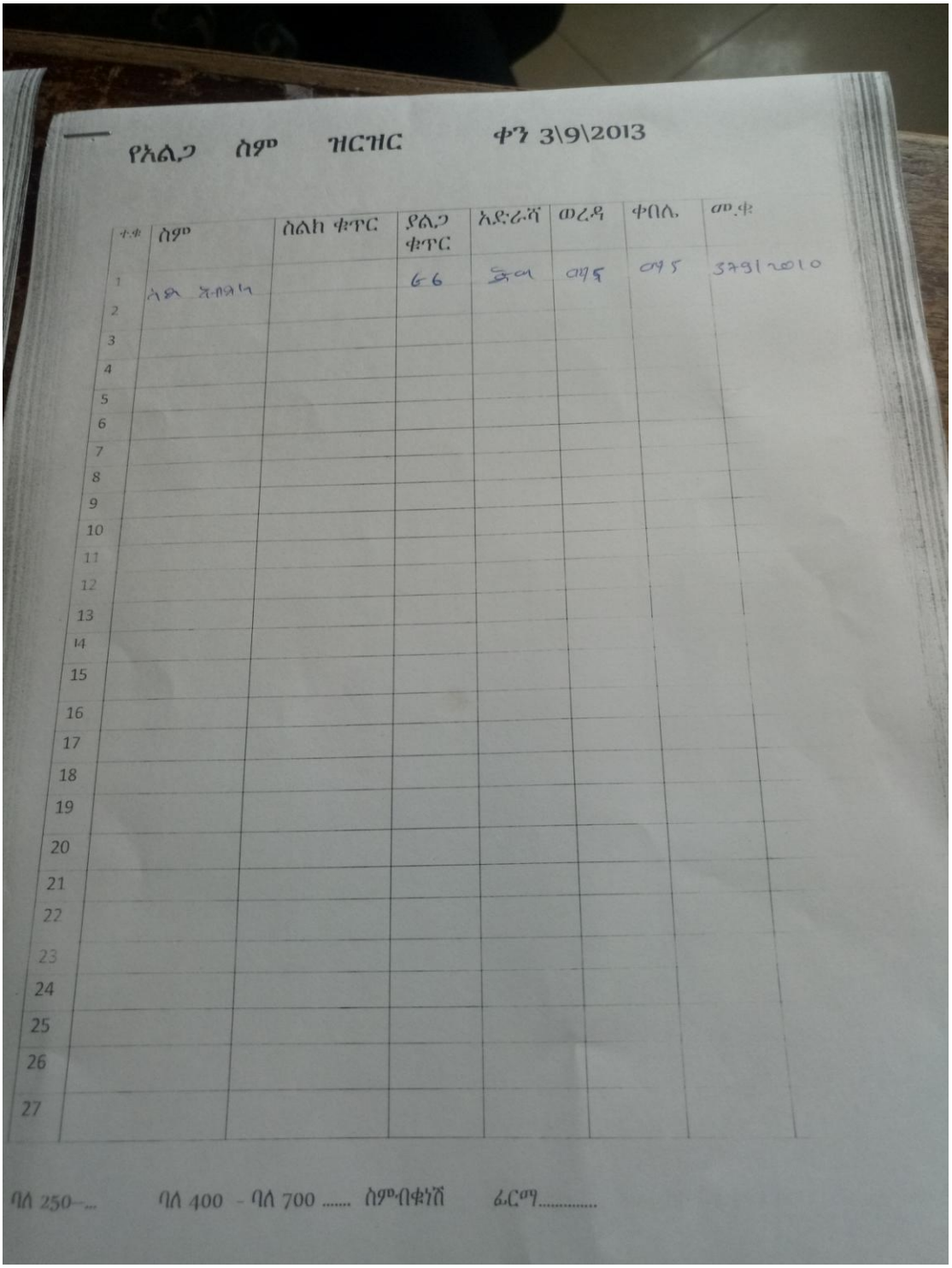

Figure 2. 1: Forms of booked room

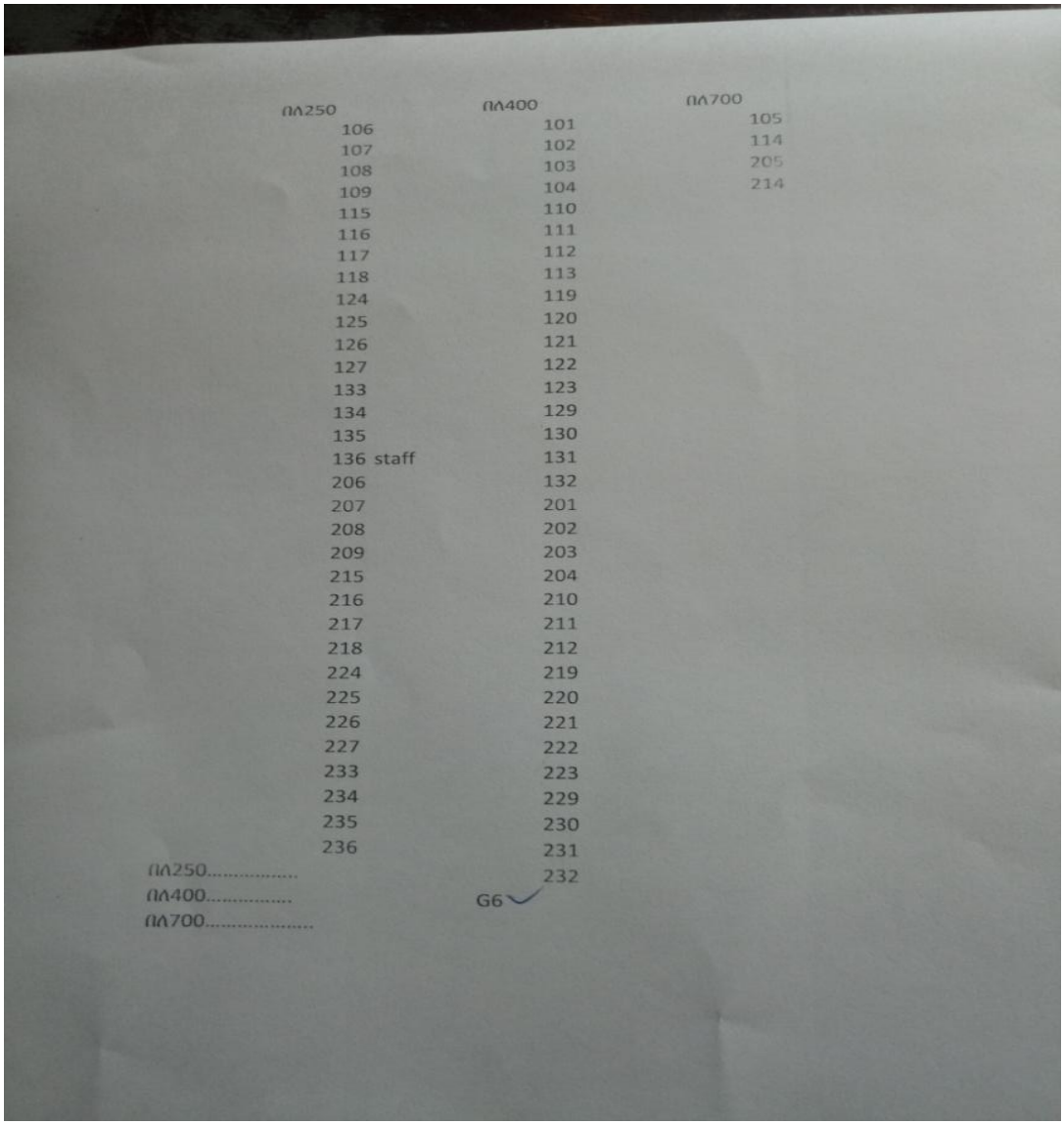

Figure 2. 2: reserved room report

2.5. Draw back of the existing system

Everything is done manually due to lack of computerized system and lack of any means of introducing the hotel service (absence of website).this brings the following problems.

- ✓ Limited number of customers
- ✓ Data redundancy
- ✓ Uses large number of man power
- ✓ Delay of customer service
- ✓ No security of data
- ✓ Use manual reservation of bed rooms
- ✓ User cannot easily access the hotel service

- ✓ Brings high cost because of using man power and phone
- ✓ Editing of data become a tedious job
- ✓ Paper wastage in using manual system
- ✓ Required more time to complete user requirement

If customer wants to reserve bed, first call a phone or talk face to face with receptionist or hotel manager. Then if there is free bedroom fill form and pay.

2.6. Business rule of the existing system

Business rule are principles, requirements and polices that must be fulfilled and obligated in order the system will function properly and effectively. The business rules that must be considered for this project are described below. Soresa hotel reservation system has its own business rule to give service for the customer. The rules included are:

BR1. Guests should register using an identification (id) document

BR2. Guests should have to pay in advance or deposit money for the room reserved

BR3. Guests are requested not to bring food and drinks from outsides

BR4. Room service is available up on requests

BR5. Guests should lock the door and hand over the key to reception

BR6. Additional payment should be paid if the user extends day out.

BR7. The receptionist should give better services to customer.

BR8. Guests should note that listening to loud music will disturb the other

BR9. Guests are not allowed wash clothes in the room

2.7. Organizational structure of the system

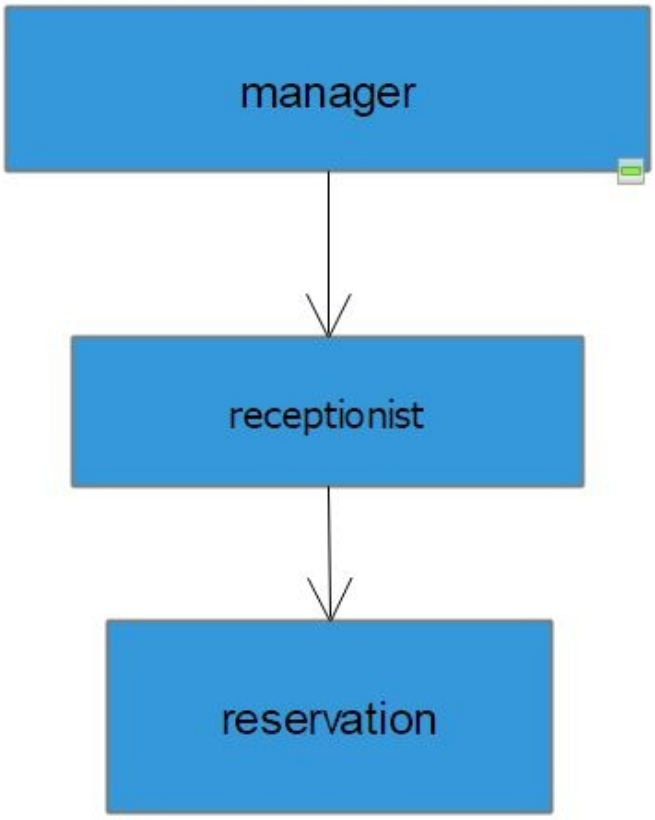

Figure 2. 3: Organizational structure of the existing system

CHAPTER THREE

3.1. The proposed system

Web based hotel reservation system is a system that provide the reservation with a modern system, which improved the problems of the current system by developing the new developed system. The team intended to do this by computerizing the current system and making it readily available to both customers and hotel alike.

The proposed system is capable of providing high security of data, capability of organizing all information in easy way of recording and accessing items information by its well organized user-friendly interface. Generally the proposed system improved the performance of the existing system and reduce this problems, time wastage, Bring data security, data inconsistency, Poor quality service delivery, and reduces wastage of paper.

3.2. Functional Requirements

Functional requirements define the fundamental actions that system must perform. Functional requirements are the intended behaviors of the system.

Functional requirements describe the relations between the system and the user or the environment. Here the ‘relations’ means the direct or indirect interactions between the user and the system. Hence the system has the following stakeholders with different requirements.

- ✓ The system allows the manager to generate report.
- ✓ The system allows the manager to add, terminate employees.
- ✓ The system allow the manager to update hotel profile.
- ✓ The system allows the manager to see comments.
- ✓ The system allows the receptionist to add, update and delete room.
- ✓ The system allow the receptionist to see for available room.
- ✓ The system allow the customer to see for available room.
- ✓ The system allows the customer to reserve room.
- ✓ The system allows the customer to cancel reservation.
- ✓ The system allow the customer to pay online.
- ✓ The system allows the receptionist to generate report.
- ✓ The system allows the administrator to create, update and deactivate account.
- ✓ The system allows the customer to write fed back for the hotel.

- ✓ The system allows the customer to get verification message after payment.

3.3. Non-Functional Requirements

Non-Functional requirement explains and describes requirements that support the main of the system that should have but they are not part of the system functionalities. Generally non- functional requirements describe the quality of the system. The following lists states the non-functional requirements.

3.3.1. User interface and human factors

System we will develop support both novae and expert user which means moderate system to support both level of knowledge. We will do such thing by providing button, menus, and others interactive button in user friendly manner.

As a human factors, there are many different persons which have different feeling and attitude, so that we will develop our systems user interface by selecting best color which many will people supports and use. This makes our system very interesting because users will uses the system without frustration for long period of time.

3.3.2. Hardware consideration

System we developed was compatible with any hardware like windows 7, 8, 10 and mobile phone by making responsive interface to applicable in mobile phone.

3.3.3. Security issue

The system must be secure because of the frame work we used make system itself secure. MVC is framework in asp.net (Model View Controller), users first interact with view and no direct communication with model but between them there was controller which is controlling unwanted or invalid request to the model.

Other, we used MD5 hash algorithm to make password encrypted and no one see or access another user password. And also we proved role based user account to prevent unauthorized users. This all mechanisms makes our system secure.

3.3.4. Performance consideration

The system performance will be highly responsive because of by query optimizing, normalizing tables, and making client side validation which is very important to check user's data validation before navigating to model or database. Also ASP.NET have its own web optimization for enhancing the performance.

3.3.5. Error handling and validation

Exception handling is the process of responding to the occurrence of exceptional conditions requiring special processing. Exception handling is important in any application. In ASP.NET we can handle exceptions in the following two ways:

- ✓ Try-catch-finally block at method level
- ✓ Using Application Error

3.3.6 Quality Issues

MVC is a software architecture pattern which follows the separation of concerns method. In this model .Net applications are divided into three interconnected parts which are called Model, View, and Controller. So each of these parts can be developed, tested in relative isolation and also combined to create a very robust application.

3.3.7. Physical environment

This article introduces how to set up server environment for MVC application on local development machine. As a web application hosted on a server accessible by end user so, it is good practice that we develop application in the same live environment on a local system so that we don't need to do more changes at the time it goes live. An MVC application may host and test one of the two web servers.

IIS Express

Its default internal web server in Visual Studio is typically used to build and run our application during development for us to test and debug codes. It is a lightweight, self-contained version of IIS optimized for developers. IIS Express makes it easy to use the most current version of IIS to develop and test websites. It has all the core capabilities of IIS 7 and above.

IIS Web Server

It comes built-into Windows. It provides server environment that is closest to what the live site will run under, and it is practical for us to install and work with IIS on our development machine.This article will walk us through how to publish and host our ASP.NET MVC application in our local IIS Web Server and debug hosted application in Visual Studio.

CHAPTER FOUR

4. System analysis

This chapter focuses on analyzing the proposed system. It explains the System analysis by designing use case diagrams, use case documentation for each use case identified in the use case diagram, sequence diagram, activity diagram, analysis level class diagram or conceptual Modeling.

4.1 System model

Use cases are used during requirements elicitation and analysis to represent the functionality of the system. It describes what a system does from the viewpoint of an external observer. A use case describes a function provided by the system that yields a visible result for an actor. An actor describes any entity that interacts with the system.

4.1.1 Actor identification

Actor classes are used to model and represent roles for "users" of a system, including human users and other systems. Actors are denoted as stick person icons. In our proposed system there are four actors which participating with system. The following are the actors of the proposed system

- ✓ **Manager**: A person who manage overall activities in the hotel.
- ✓ **Receptionist**: A person who register, reserve customer or clients who come to reserve room and take payment from the clients for the reservation.
- ✓ **Customers**: A person who use the system to reserve, or to order food and beverages.
- ✓ **System administrator**: A person who manage account

4.1.2 Use case identification

A use case represents a function that the system performs. Alternatively, a use case can be thought of as a goal that some actor can achieve with the system. Identifying use case uses to visualize the overall system functions and the actors interacting with use cases. By concerning this concept our project identifies the following use cases.

Customer

- ✓ The system allows Customers to reserve room.
- ✓ The system allows Customers to cancel reserved room.

- ✓ The system allows Customers to see for available room.
- ✓ The system allows Customers to leave comment as a feedback for hotel manager.
- ✓ The system allows Customers to Visit Hotel on the web site.
- ✓ The system allows customers to pay for used service.

Manager

- ✓ The system allows the Manger to view comment.
- ✓ The system allows the Manger to generate report.
- ✓ The system allows the Manager to login in the system.
- ✓ The system allows the Manager to logout from the system.
- ✓ The system allows the Manager to upload hotel profile.
- ✓ The system allows the Manager to record employee information.
- ✓ The system allows the Manager to terminate employee.

System Administrator

- ✓ The system allows the System admin to create account.
- ✓ The system allows the System admin to deactivate account.
- ✓ The system allows the System admin to add account.

Receptionist

- ✓ The system allows the Receptionist to cancel reserved room.
- ✓ The system allows the Receptionist to update reserved room.
- ✓ The system allows the Receptionist to see for available room.
- ✓ The system allows the Receptionist to see payment for reservation.
- ✓ The system allows the Receptionist to upload available rooms.

4.1.3 Use case diagram

- ✓ **Boundary**: This defines the system of interest in relation to the world around it.
- ✓ **Actors:** usually individuals involved with the system defined according to their roles.
- ✓ **Use cases:** are the specific roles played by the actors within the system.
- ✓ The **relationship**s between and among the actors and the use cases.

Our use case diagram looks as follow

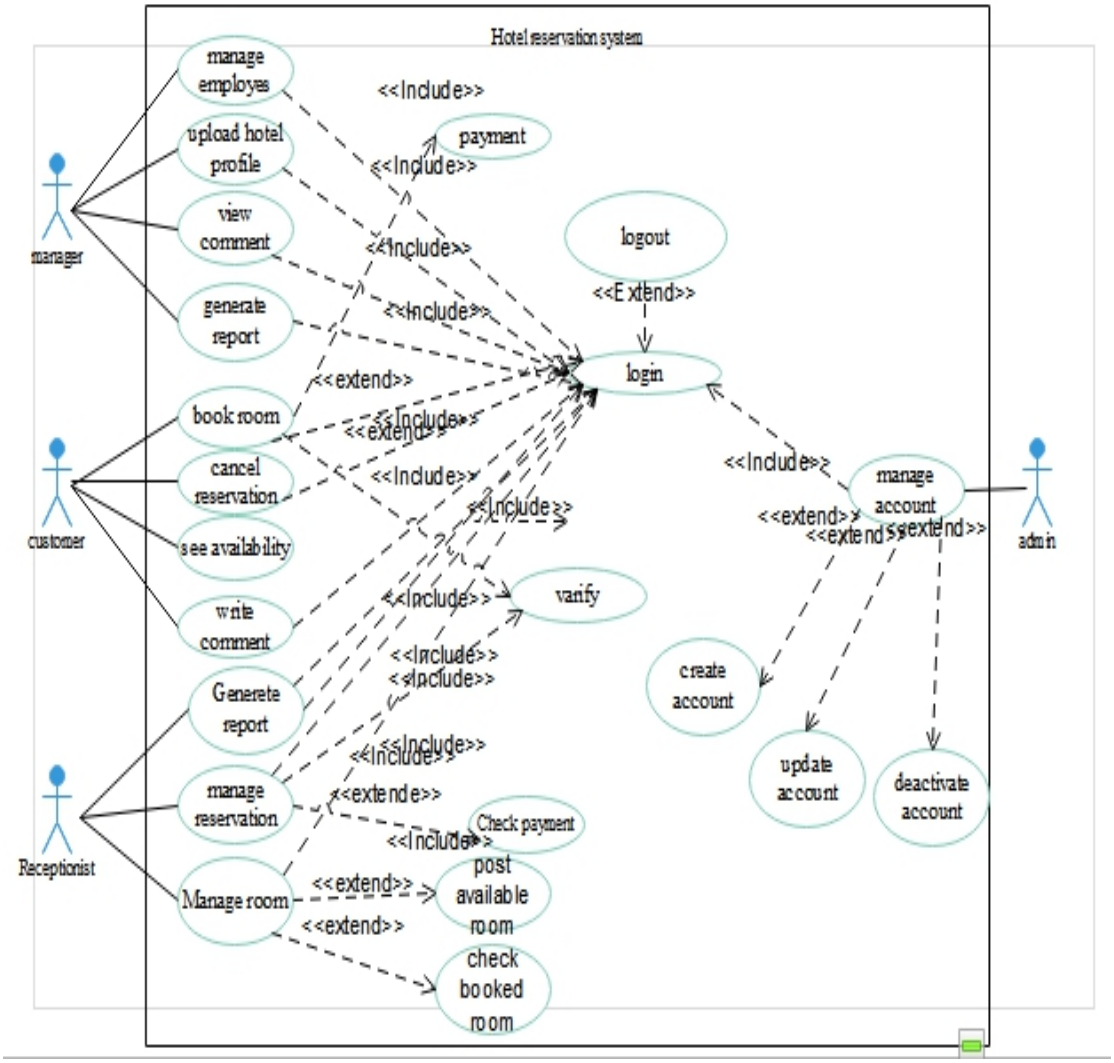

Figure 4. 1:use case diagram

4.1.4 Use case description

Table:4. 1:Login use case description

Use case name	**Login**	
Actor	Manager, Receptionist, system admin, customer	
Description:	The user enters authorized username and password in order to access the system.	
Precondition	The user should have username and password and account.	
Basic course of action	**User action**	**System response**
	1. The user opens the 'login' page. 3. The user enters the username and password on the form. 4.The user clicks the login button. 8. End use case.	2.The system will display a'login' form that accepts username and password from the user. 5.The system searches the username from the database and matches it with the password. 6.If correct the system opens the authorized page.
Alternative course of action	7. If the username and password are incorrect go to step 2, 3, 4 and 5 of the basic course of action.	

Table:4. 2: Logout use case description

Use case name	**Logout**	
Actor	Manager, Receptionist, system admin	
Description:	When the user logouts when he/she wants to exit from the system.	
Precondition	The user should login first.	
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 login form for the user.

Table:4. 3:Generate report use case description

Use case name	**Generate report**	
Actor	Manager, Receptionist	
Description:	The user can see the hotels overall report and can generate different information from the report.	
Precondition	Report should be prepared by the system.	
Basic course of action	**User action**	**System response**
	1.The user will login to the system. 3. The user opens the report page. 4. The user retrieves different information from the report. 5. End use case.	2.The system will display a report page that helps the user to generate information on it.

Table:4. 4:Add room use case description

Use case name	**Add room**	
Actor	Receptionist	
Description:	Receptionist will add new room with new attributes to the system.	
Basic course of action	**User action**	**System response**
	1.Receptionist will login to the system. 2.Receptionist will open the ‘add room’ page.	3.The system will display the ‘add room’ form with fields such as: - Room name. - Room type.

	4.Receptionist will fill the form. 8. End of use case.	- Room price. - Room image. 5.The system will check all fields of the form are filled or not. 6.If all fields are filled correct show successful message.
Alternative course of action	7. If the form is not filled correctly go to step 3, 4 and 5 of the basic course of action.	

Table:4. 5: delete room use case description

Use case name	**Delete room**	
Actor	Receptionist	
Description:	Receptionist will delete the room from the database.	
Basic course of action	**User action**	**System response**
	1. Receptionist will login to the system. 2. Receptionist will open the 'delete room' page. 4. Receptionist will enter the information of the room to be deleted. 7. Receptionist selects the room from the result box and click delete button. 10. End use case.	3.The system will display the 'delete room' form that helps Receptionist to choose the room to be deleted. 5. The system searches for the room that the receptionist wants to delete. 6.If the room that the receptionist wants to delete is found the system displays the room's information on the result box. 8.The system will delete the room from the rooms table. 9.The system will display a successful message.

Table:4. 6:Update room use case description

Use case name	**Update room**	
Actor	Receptionist	
Description:	Receptionist updates the room's information.	
Basic course of action	**User action**	**System response**
	1.Receptionist will login to the system. 2.Receptionist will open the 'update room' page. 4.Receptionist will enter the information of the room to be updated. 8.Receptionist selects the room from the result box and click 'update information' button. 10.Receptionist will change the room's information. 11.Receptionist will click save button. 14. End use case.	3. The system will display the 'update room' form that helps the Receptionist to choose the room to be updated. 5. The system searches for the room that the Receptionist wants to update. 6.If the room that the Receptionist wants to update is found the system displays the room's information on the result box. 9.The system will display a form that contains the selected rooms' information. 12.If all form fields filled correctly the system will display a successful message.
Alternative course of action	7. If the room is not found go back to steps 3, 4 and 5 of the basic course of action. 13. If form not filled correctly go to step 9, 10 and 11 of the basic course of action.	

Table:4. 7: Reserve room use case description

Use case name	**Book reservation**	
Actor	Customer	
Description:	The customer reserves room online, so do the receptionist.	
Precondition	The customer should have internet access or connection.	
Basic course of action		

	User action	**System response**
	1.The user opens the reservation page. 2. The user click 'reserve room' button. 4.User fills the form. 5.User click reserve button 9.End use case	3.The system will display a reservation form that contain the following -First name -Last name -Address -Room Type -Duration for reservation 6. The system checks all the form fields have filled out. 7. If the form filled correctly the system display payment pages then customer clicks pay button on payment page to pay through payment gateway or pay required money through CBE birr.
Alternative course of action	8. If the form is not filled correctly go back to step 3, 4, 5 and 6 of basic course of action.	

Table:4. 8:Check availability use case description

Use case name	**Check availability**
Actor(s)	Customers ,receptionist
Description	For reservation purpose if there is free space
pre-condition	Receptionist should have an account and login with his account.
Post condition	Customer will order room Receptionist make required decision.
Basic course of action	The customer opens availability page and system will display the page. receptionist should first login and click see list of available

	reservation.
Alternate course action	If no free bed room the system displays a message to the customer.

Table:4. 9: Feedback use case description

Use case name	**Feedback**
Actor(s)	Customer
Description	For giving suggestion or comment
Precondition	Customer fills the Feedback form.
Post condition	Customer has received a message from the system.
Basic course of action.	Customer clicks the send button. System receive request and send the feedback form to customer. Customer fills the feedback form and submits. System receives feedback form. System sends thank you message to customer.

Table:4. 10: Change hotel profile use case description

Use case name	**Change the hotel's profile**	
Actor	Manager	
Description:	Manager will edit the content of the hotels website pages.	
Basic course of action	**User action**	**System response**

	1.Manager will login to the system. 2.Manager will open the web pages with an editor tool. 3.Manager will change the information on the page. 4. Manager will save the page. 5. End use case.	

Table:4. 11:View customer profile

Use Case name	**View Customer profile**
Actor(s)	Manager
Description	Manager can access customers profile, reservation and details of the customers
pre-condition	Manager has password to log in.
Post condition	Manager can access customer profile.
Basic course of action	System processes retrieve customer information from database and display to the manager.
Alternative course of action	If there is an error with login nothing happens.

Table:4. 12:Manage account use case description

Use case name	**Manage account**
Actor(s)	System admin
Description	In order to create, update account, deactivate
Precondition	System admin should login into the system
Post condition	Managing account of employee
Basic course of action	1. System admin click sign up form to register.

	2. System receives the request and sends the registration form to System admin. 3. System admin completes the form and click at proceed button. 4. System verifies the information in the form, check the validity of the information. if the information is not valid, error massage display if valid, Successful massage display.
Alternative course of action	1. System admin select a duplicated username or invalid address or miss some information in registration form: a. System will send a notification 2. System admin clicks cancel or close webpage: System terminates the connection with user due to some errors.

Table:4. 13:Add employee use case description

Use case name	**Add employee**	
Actor	Manager	
Description:	In order to hire employees	
Basic course of action	**User action**	**System response**
	1. The system administrator will login to the system. 2. The system administrator will open the 'add employees' page. 4.The system administrator will fill the form. 8. End of use case.	3.The system will display the 'add employees' form with fields such as: - Employee name. - Employee ID. -Employee age. -Employee address. 5.The system will check all fields of the form are filled or not. 6.If all fields are filled correctly show successful message.
Alternative course of action	7. If the form is not filled correctly go to step 3, 4 and 5 of the basic course of action.	

Table:4. 14:Terminate employee use case description

Use case name	**Terminate employee**	
Actor	Manager	
Description:	Manager would delete the employee information when they terminate.	
Basic course of action	**User action**	**System response**

	1. Manager will login to the system. 2. Manager will open the 'terminate employee' page. 4.Manager will enter the information of the employee to be terminated. 8.Manager selects the employee from the result box and click terminate button. 11. End use case.	3.The system will display the 'terminate employee' form that helps Manager to choose the employee to be terminated. 5.The system searches for the employee that manager wants to terminate. 6. If the employee that manager wants to terminate is found the system displays the employee's information on the result box. 9.The system will send the employee's information to the terminated table and delete it from the employee table. 10.The system will display a successful message.
Alternative course of action	7. If the employee is not found go back to steps 3, 4 and 5 of the basic course of action.	

4.1.5 Use case scenarios

Scenario name: login

Participant actor User

If User wants to login to the system first run the system then the system display home page next User click login link from home page. After that the system display login form then User fill login form and click login button and home page is display next User perform his authenticated operation and logout.

Scenario name: Reservation or booking

Participant actor Customer

If customer wants to reserve room first she/he would have CBE account/hello cash account and run the system then the system display home page next customer open reservation page then click on book now button. After that system display booking form then customer fill all required form including check in and check out date, then system calculate and display total amount of she will stay. Then after customer pay calculated price through CBE/hello cash or transfer money to hotel.

Scenario name: Manage reservation

Participant actor Receptionist

Mr. X as user if she wants to manage reservation first he wants to run system, then system display home page and then he must login first, after that system would displays reservation page, then he would clicks on reservation details link, then system display all reserved details of customer. Then after that he would book the required rooms. And he must to be check CBE/hello cash account if customer pay or not, if pay reserve room, else he cannot reserve.

4.2 Object model

The object model visualizes the elements in software application in terms of objects. An object is real world elements in an object-oriented environment that may have a physical or a conceptual existence. Object can modeled according to the needs of the application.

4.2.1 Class diagram

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modelling the domain-specific data structure to detailed design of the target system. With the share model facilities, you can reuse your class model in the interaction diagram for modelling the detailed design of the dynamic behaviour. The From Diagram allows you to generate diagram automatically with user-defined scope.

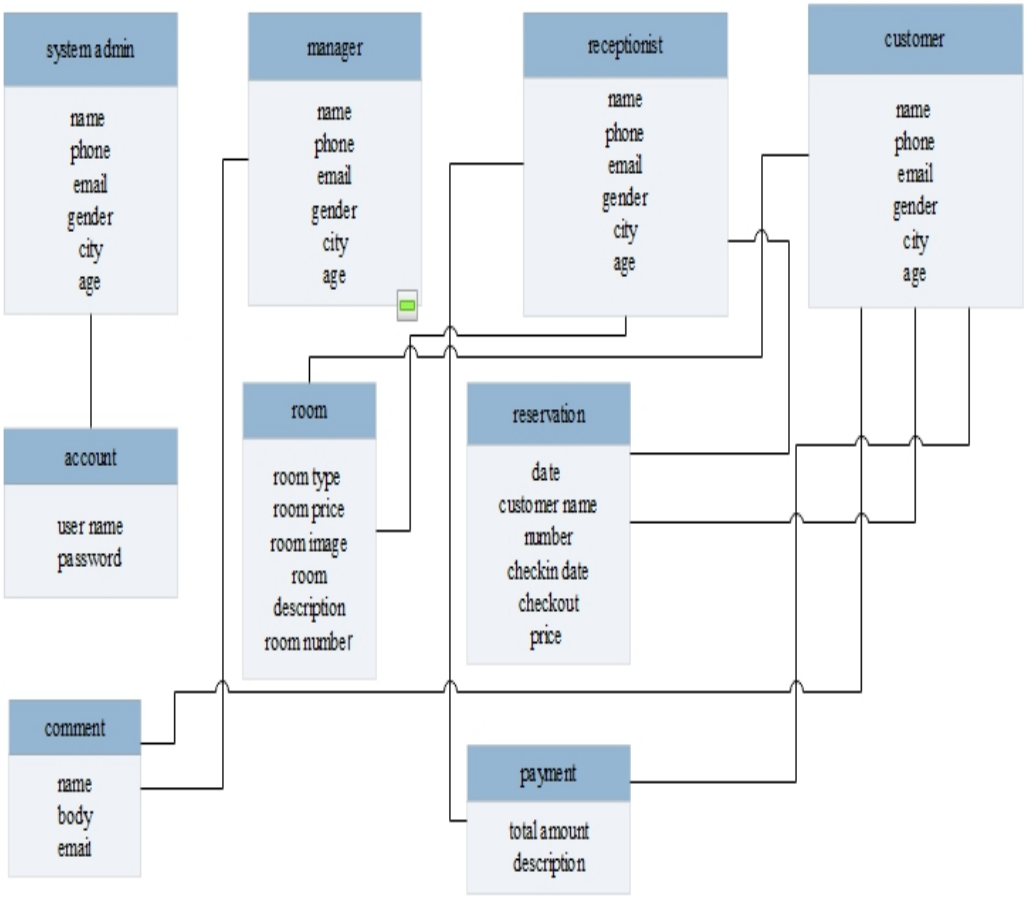

Figure 4. 2:class diagram

4.4.2. Data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Customer_id	int	20	Primary key	NOT NULL
Customer name	Varchar	40	-	NOT NULL
Gender	varchar	20	-	NOT NULL
Email address	varchar	20	-	NOT NULL
Phone number	varchar	20	-	NOT NULL
Age	int	20	-	NOT NULL
Region	varchar	20	-	NOT NULL

Figure 4. 3: Customer data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Manager_id	int	20	Primary key	NOT NULL
Manager name	Varchar	40	-	NOT NULL
Gender	varchar	20	-	NOT NULL
Email address	varchar	20	-	NOT NULL
Phone number	varchar	20	-	NOT NULL
Age	int	20	-	NOT NULL
Region	varchar	20	-	NOT NULL

Figure 4. 4: Manager Data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Room_id	int	20	Primary key	NOT NULL
Room name	varchar	20		NOT NULL
Room type	varchar	20		NOT NULL
Room price	float	20	-	NOT NULL

Figure 4. 5: Room Data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Resrvation_id	int	20	Primary key	NOT NULL
Customer_id	int	20	Foriegn key	NOT NULL
Room_id	int	20	Foriegn key	NOT NULL
Ckeck in date	Date	20	-	NOT NULL
Ckeck out date	Date	20	-	NOT NULL
Price	int	20	-	NOT NULL

Figure 4. 6: Reservation Data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Feedbak id	int	20	Primary key	NOT NULL
name	varchar	20		NOT NULL
Comment	varchar	20		NOT NULL

Figure 4. 7: Feedback Data dictionary

Attributes	Data type	Data size	Key constraints	Constraints
Account id	int	20	Primary key	NOT NULL
User name	varchar	20		NOT NULL
Password	varchar	20		NOT NULL
Status	bool	10	-	NOT NULL

Figure 4. 8: Account Data dictionary

4.3 Dynamic model

In every system, objects do not just sit idle; they interact with one another by passing messages. In the UML, the dynamic aspects of a system is modeled using interactions

4.3.1 Sequence diagram

Is an interaction diagram that emphasizes the time ordering of messages

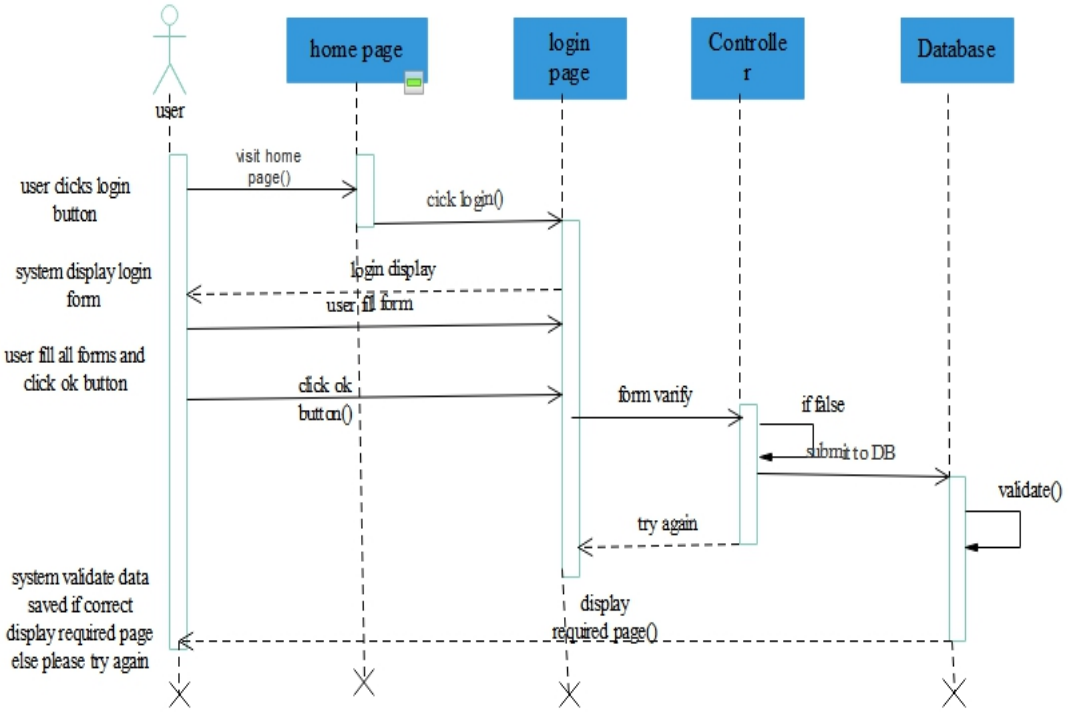

Figure 4. 9:login sequence diagram

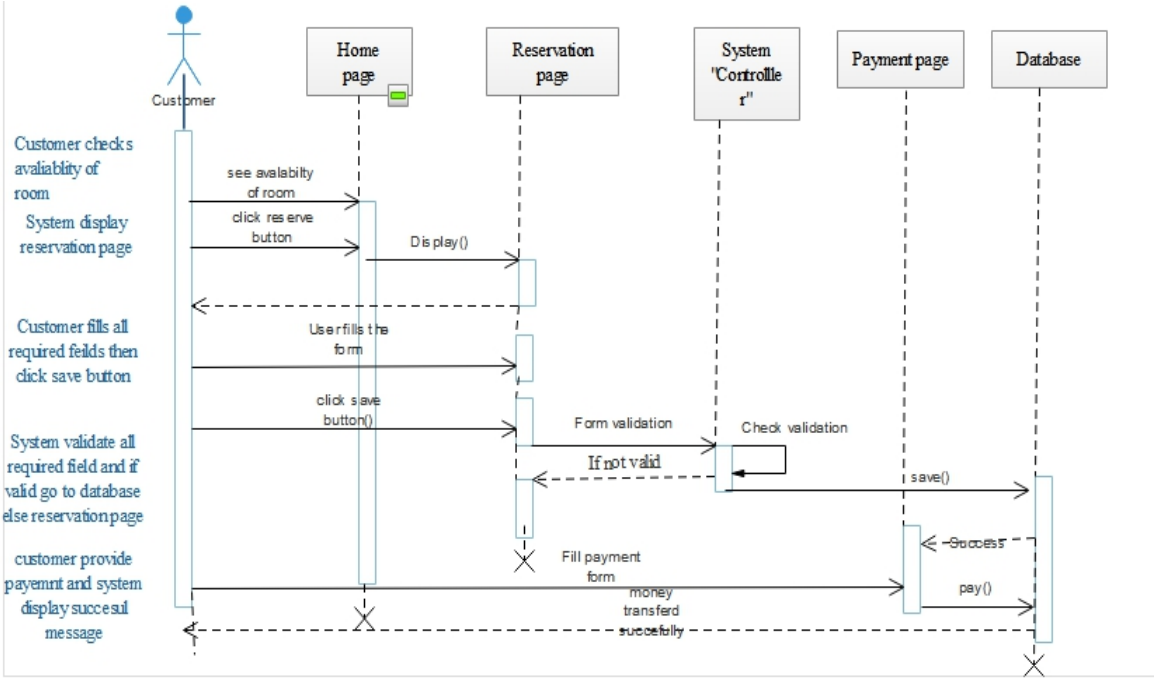

Figure 4. 10: reservation sequence diagram

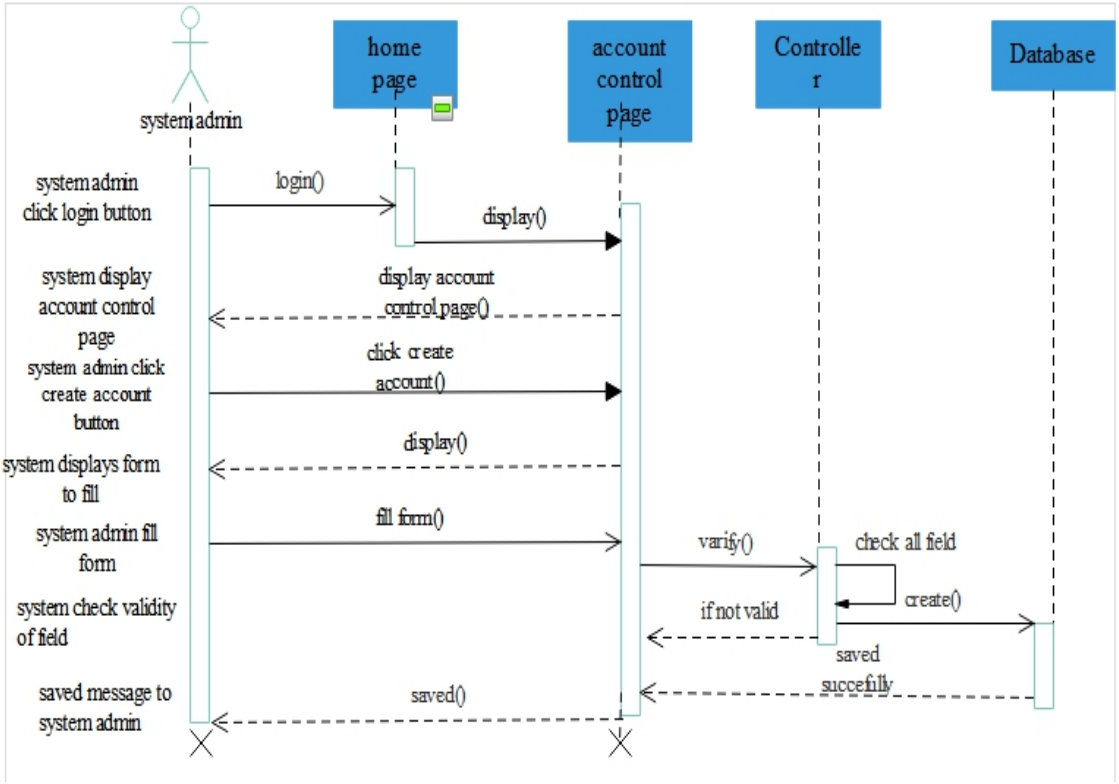

Figure 4. 11: Create account sequence diagram

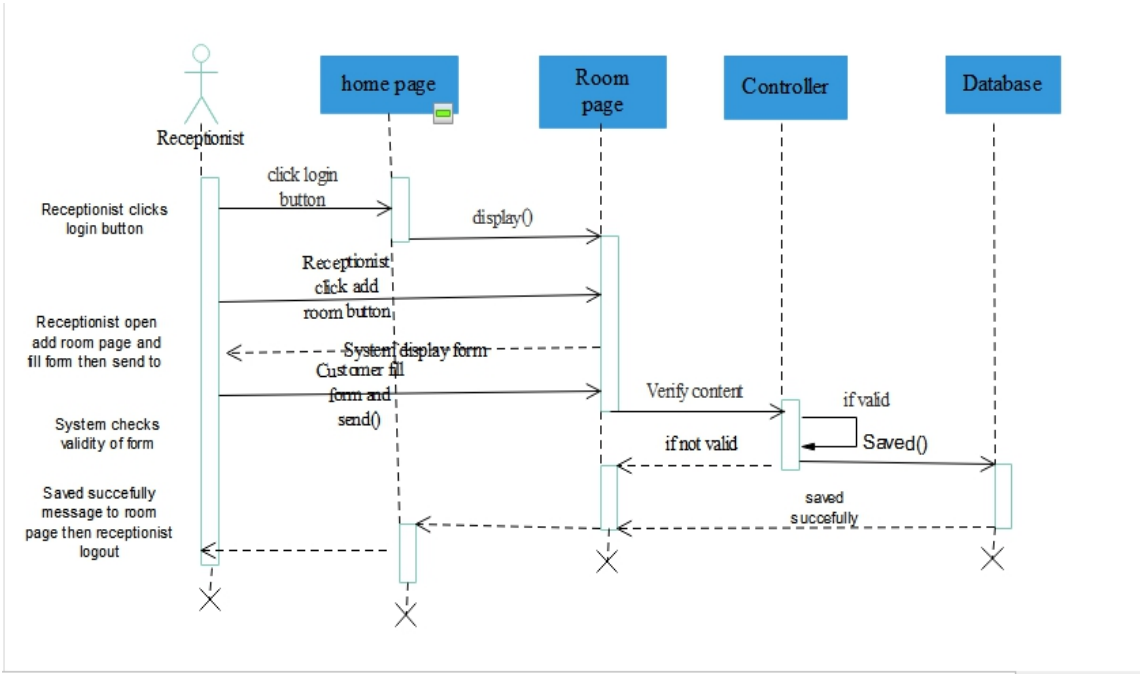

Figure 4. 12:Add room sequence diagram

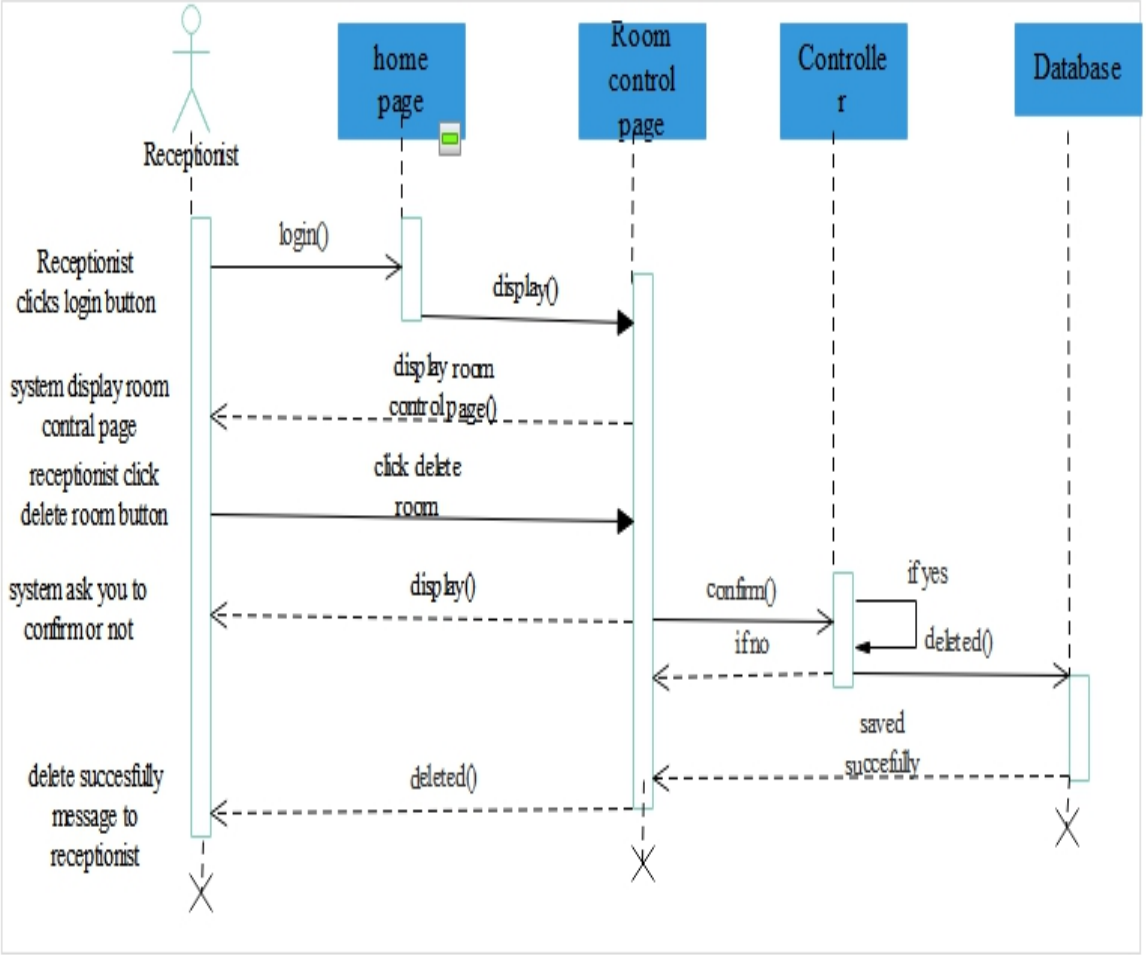

Figure 4. 13: Delete room sequence diagram

4.3.2 Activity diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice. In its basic form an activity diagram is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow.

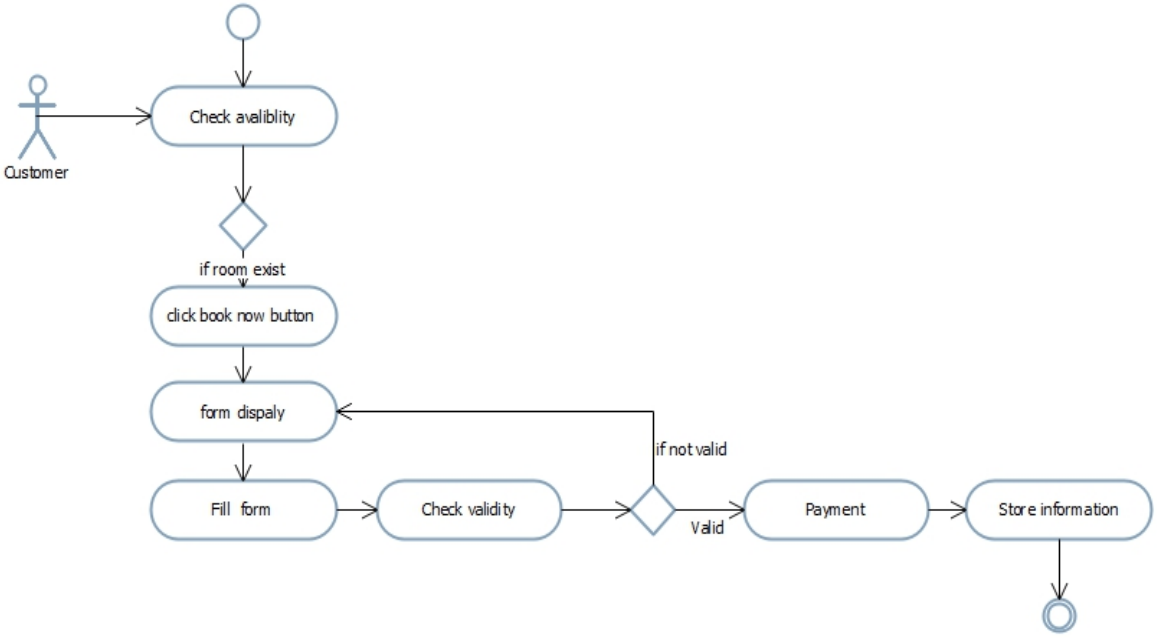

Figure 4. 14:Reservation activity diagram

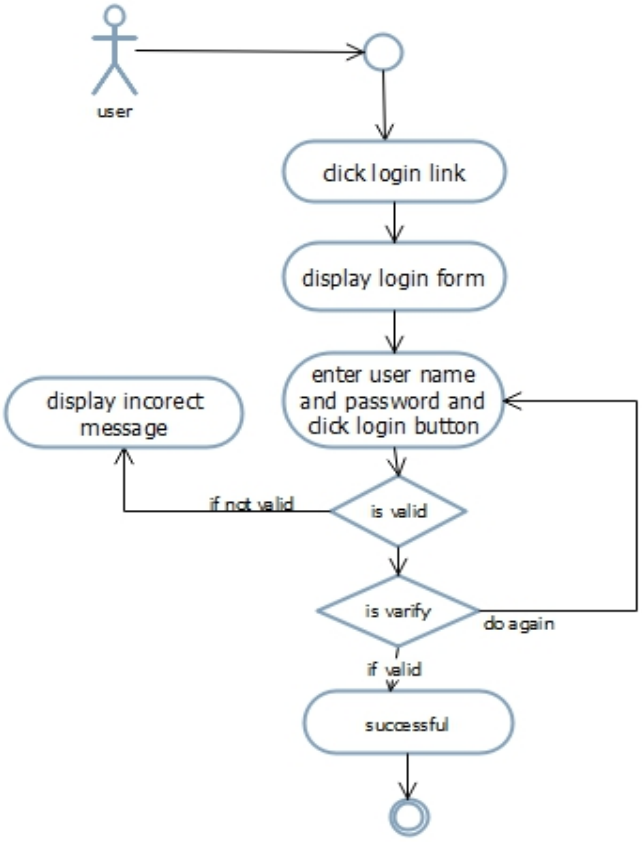

Figure 4. 15: Login activity diagram

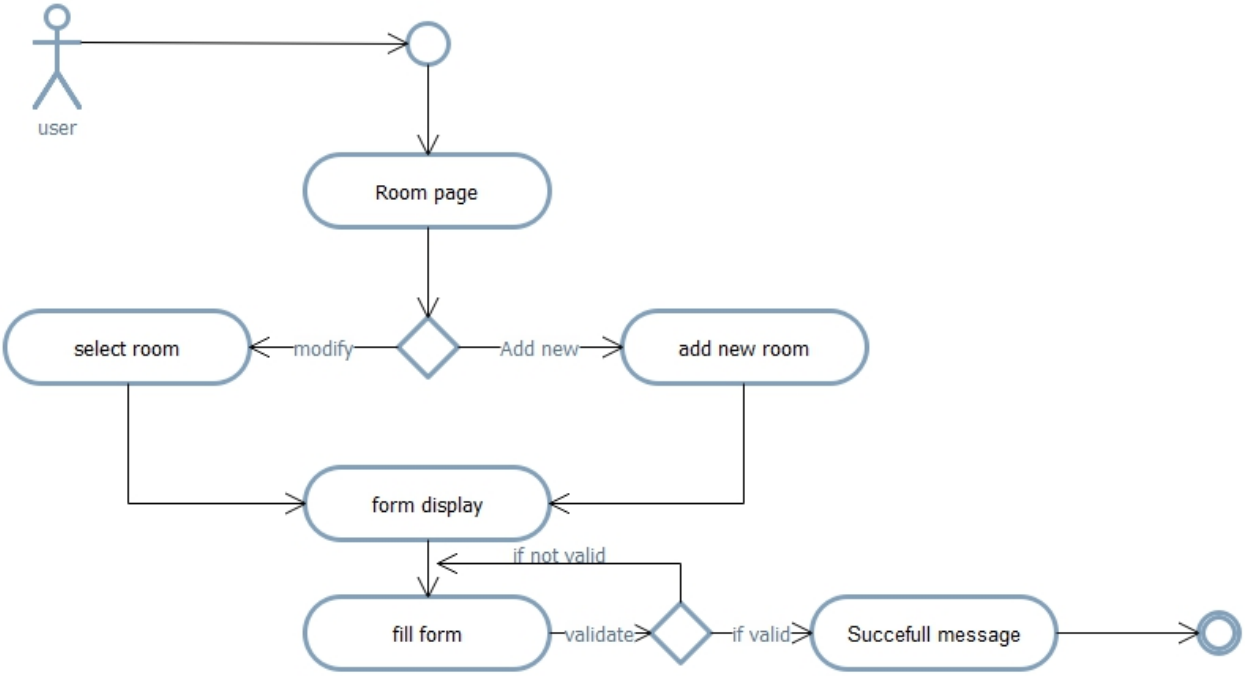

Figure 4. 16:Add, update room activity diagram

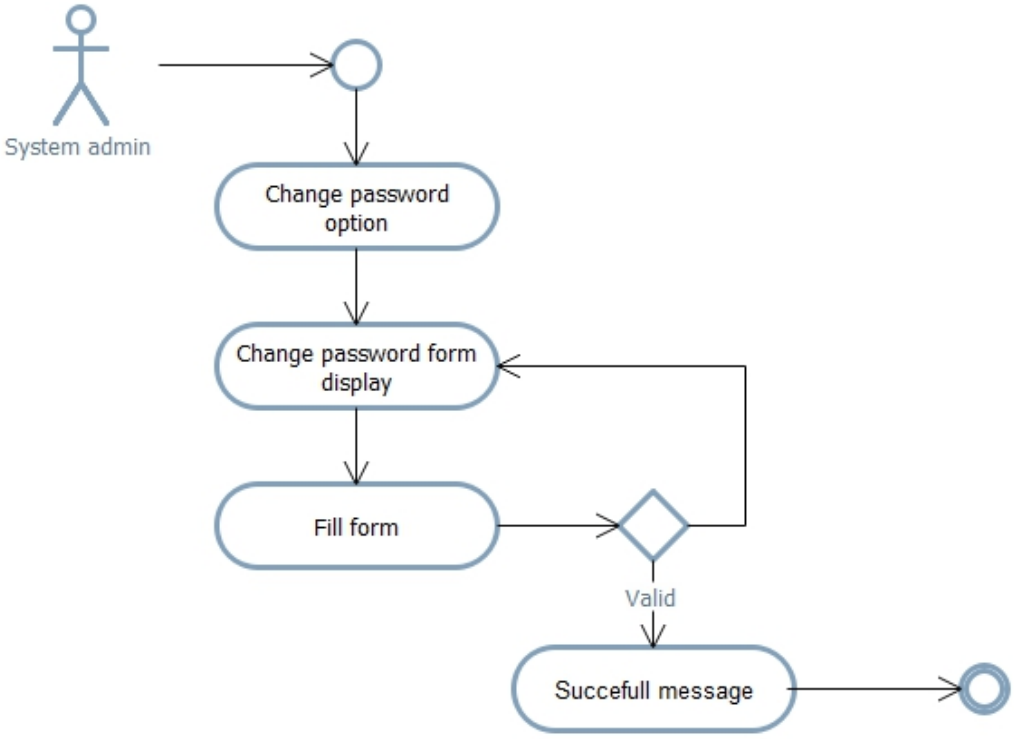

Figure 4. 17:Change password activity diagram

4.3.3. State chart

A state chart diagram is a view of a state machine that models the changing behavior of a state. State chart diagrams show the various states that an object goes through, as well as the events that cause a transition from one state to another.

The common model elements that state chart diagrams contain are:

- ✓ States
- ✓ Start and end states
- ✓ Transitions

A state represents a condition during the life of an object during which it satisfies some condition or waits for some event. Start and end states represent the beginning or ending of a process

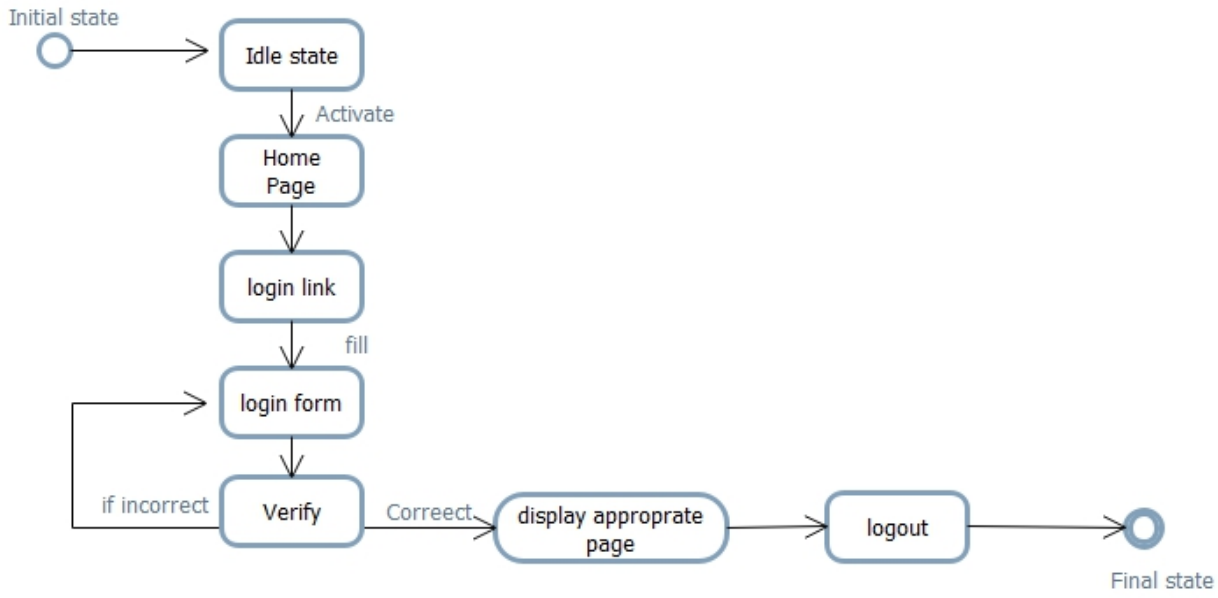

Figure 4. 18: State chart diagram for login

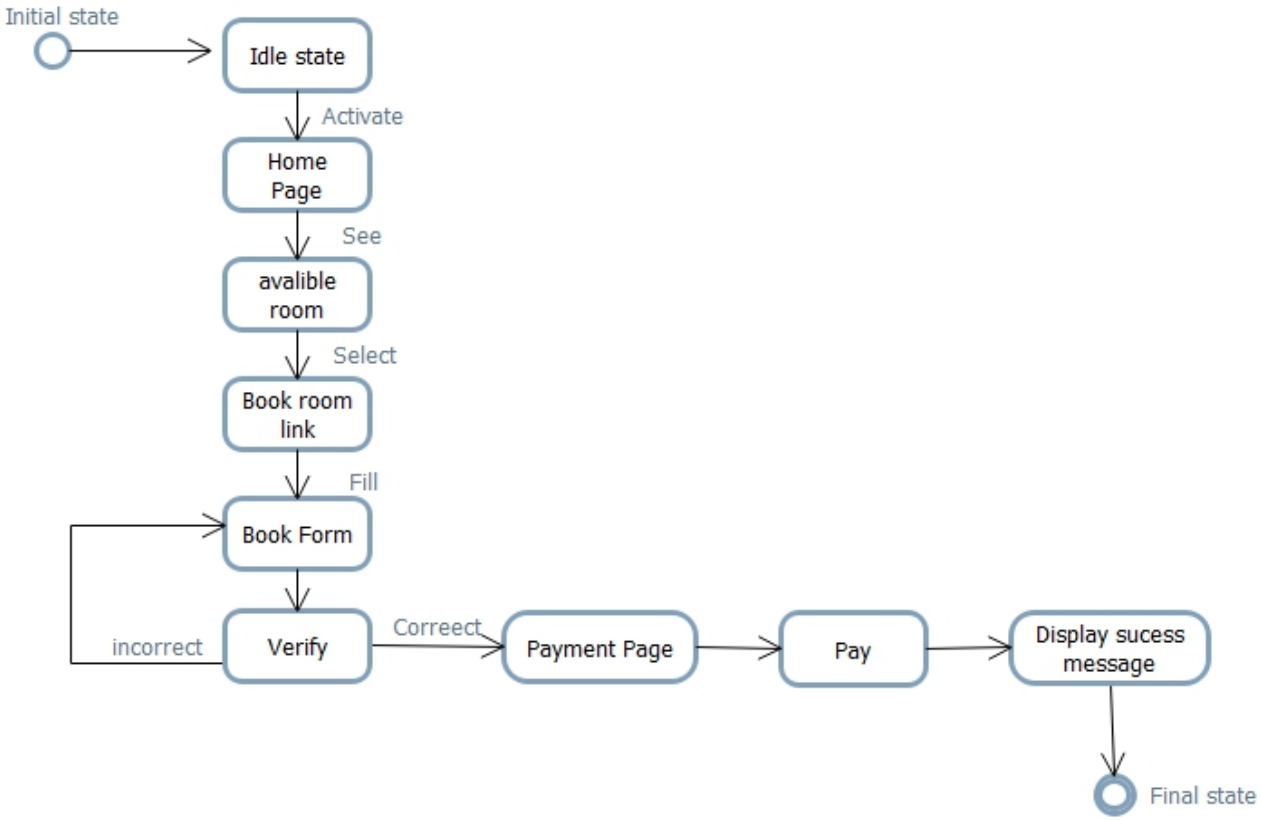

Figure 4. 19: State chart diagram for Reservation

CHAPTER FIVE

5. System design

5.1 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.

5.2 Design goals

The objectives of designing are to model a system with high quality. Implementing of high quality system depends on the nature of the design created by the designer. If one wants to make changes to the system after it has been put in to operation depends on the quality of the system design. So if the system is designed perfectly, it will be easy to make changes to it. The goal of the system design is to manage complexity by dividing the system in to manageable pieces. Some of the goals are listed below.

- ✓ **Modifiability**: The separation of model view and controller makes it easier for system easy to manage test and modify.
- ✓ **Flexibility:** The system should be changeable to suit new condition or situation.
- ✓ **Efficiency:** The system must do what it supposed to do efficiently without the problem.
- ✓ **Performance :**

- Response time of system must be good
- System MVC has its own web optimization to increase performance and we designed well database to easily and fast access so we increase performance.

✓ **Security design goals**

- Protect the confidentiality of data
- Preserve the integrity of data
- Promote the availability of data for authorized use assigning roles to each users.

5.3 Proposed system architecture

The proposed system is expected to replace the existing manual system by an automated system. The proposed system architecture is MVC or Model view controller.

MVC is a software architecture pattern which follows the separation of concerns method. In this model .Net applications are divided into three interconnected parts which are called Model, View, and Controller.

The goal of the MVC pattern is that each of these parts can be developed, tested in relative isolation and also combined to create a very robust application.

Models

Model objects are the parts of the application that implement the logic for the application domain data. Often, model objects retrieved and store model state in database.

Controllers

Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render in the browser.

Views

Views are the components that display the application's user interface (UI), typically this UI is created from the model data.

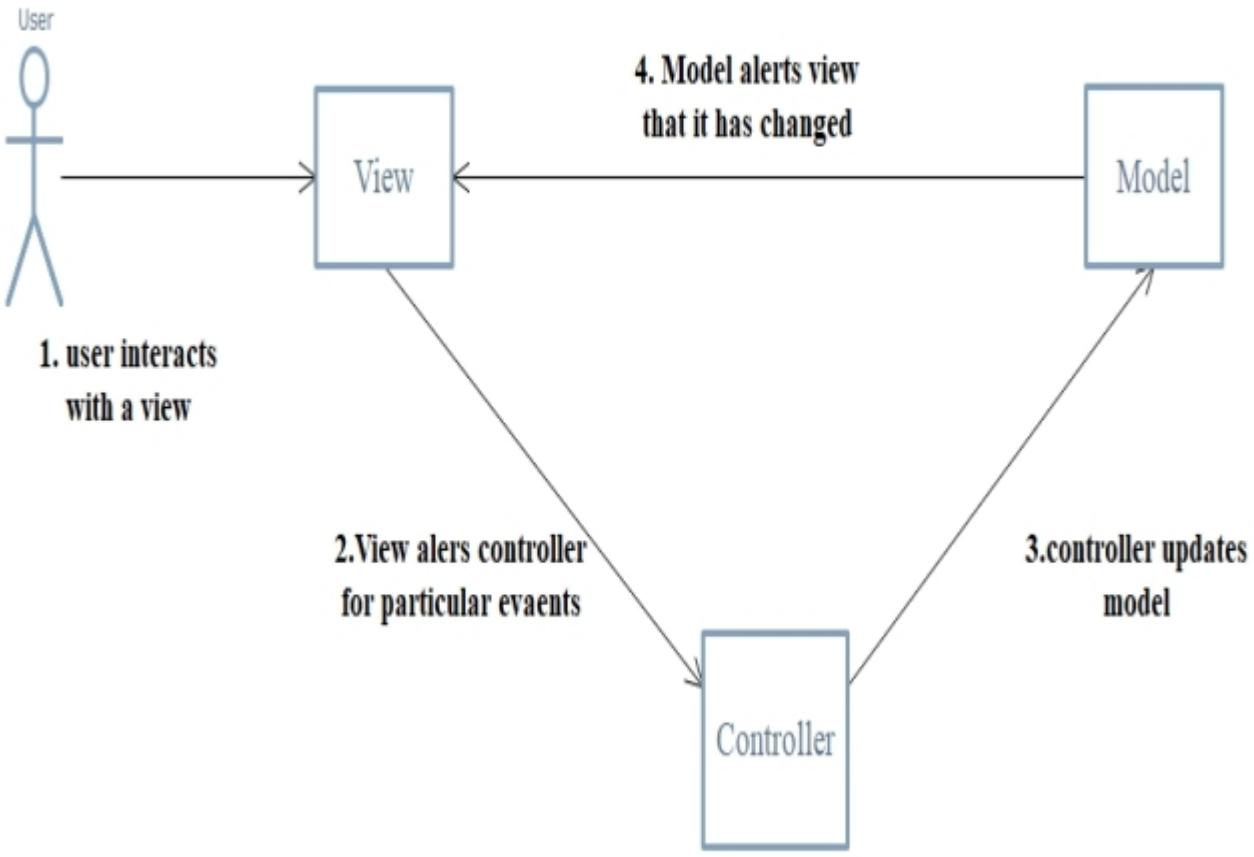

Figure 5. 1: System architecture

5.3.1 Subsystem decomposition and description

System decomposition refers to the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program, and maintain. To reduce the complexity of the solution domain, we decompose our system into simpler parts, called subsystems, which are made of a number of solution domain classes. In the case of complex subsystems, we recursively apply this principle and decompose a sub- system into simpler subsystems. We decompose our system in to four (4) subsystems. These are:

System administrator subsystem: - This subsystem perform this action

- ✓ Create account
- ✓ Deactivate account
- ✓ Update account

Manager subsystem: - This subsystem perform this action

- ✓ Register employee
- ✓ Generate report
- ✓ Update hotel profile
- ✓ Delete employee
- ✓ Update employee
- ✓ See feedback

Customer subsystem: - This subsystem perform this action

- ✓ Reserve room
- ✓ Cancel reservation
- ✓ Provide payment
- ✓ Check availability of room
- ✓ Provide feedback

Receptionist subsystem: - This subsystem perform this action

- ✓ See lists of reservation
- ✓ Post available rooms
- ✓ Generate report
- ✓ Check payment of reservation
- ✓ Add, Update, Delete rooms

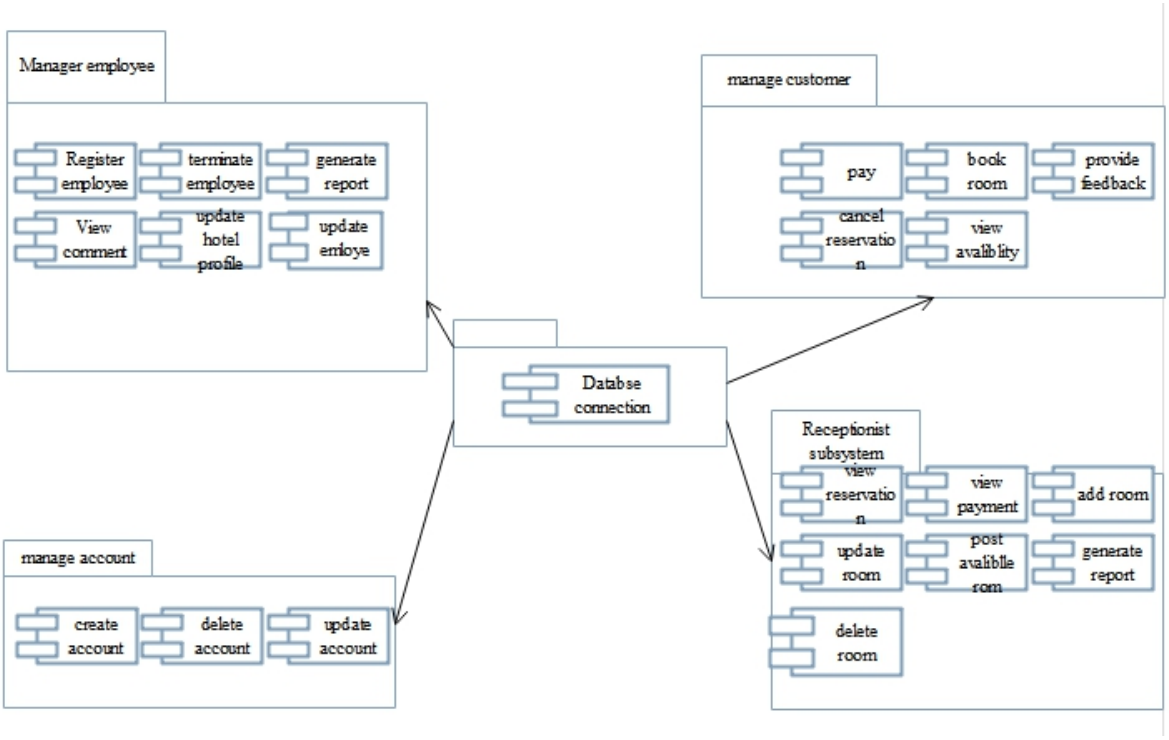

Figure 5. 2: Subsystem decomposition

5.3.2 Hardware/Software mapping (Deployment diagrams)

Deployment diagrams model the physical architecture of a system, and it shows the relationships between the software and hardware components in the system and the physical distribution of the processing.

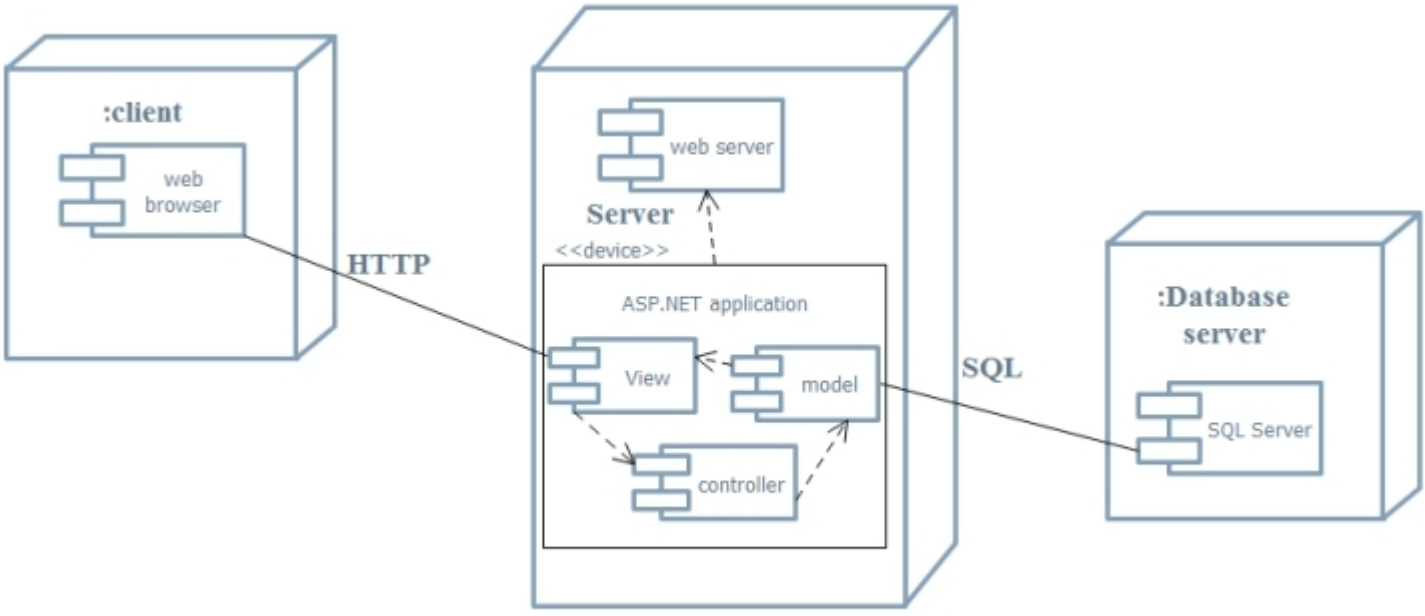

Figure 5. 3:Hardware / software mapping

5.3.3 Detailed class diagram

Class diagrams are used to describe the structure of this system. Classes are abstractions that specify the common structure and behavior of a set of objects in the new system. Class diagram which going to be designed is describing the system in terms of objects, classes, attributes, operations and their association once the team has identified the association and the corresponding multiplicity of association of the objects of the proposed new system, the next step will be to draw the class diagram. The next sub topic shows the class diagram of the new system. Class diagram is the building block of the system that we develop. It shows all the objects and tells us how they are interrelated. Class diagrams are the most popular UML diagrams used by the object oriented community. It describes the objects in a system and their relationships. Class diagram consists of attributes and functions.

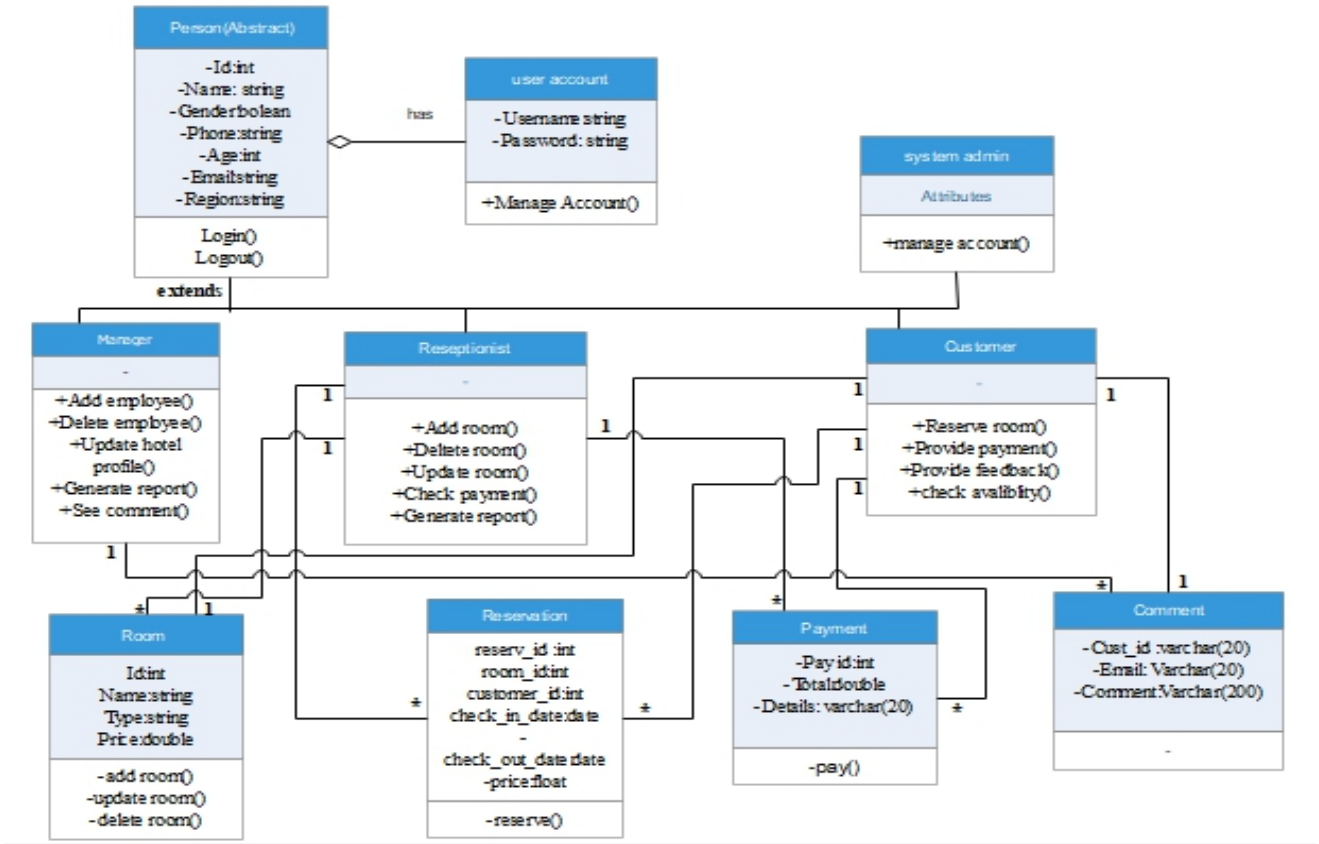

Figure 5. 4: System class diagram

5.3.4 Persistence data management

Persistent data management is basically used to represent the design of the database.

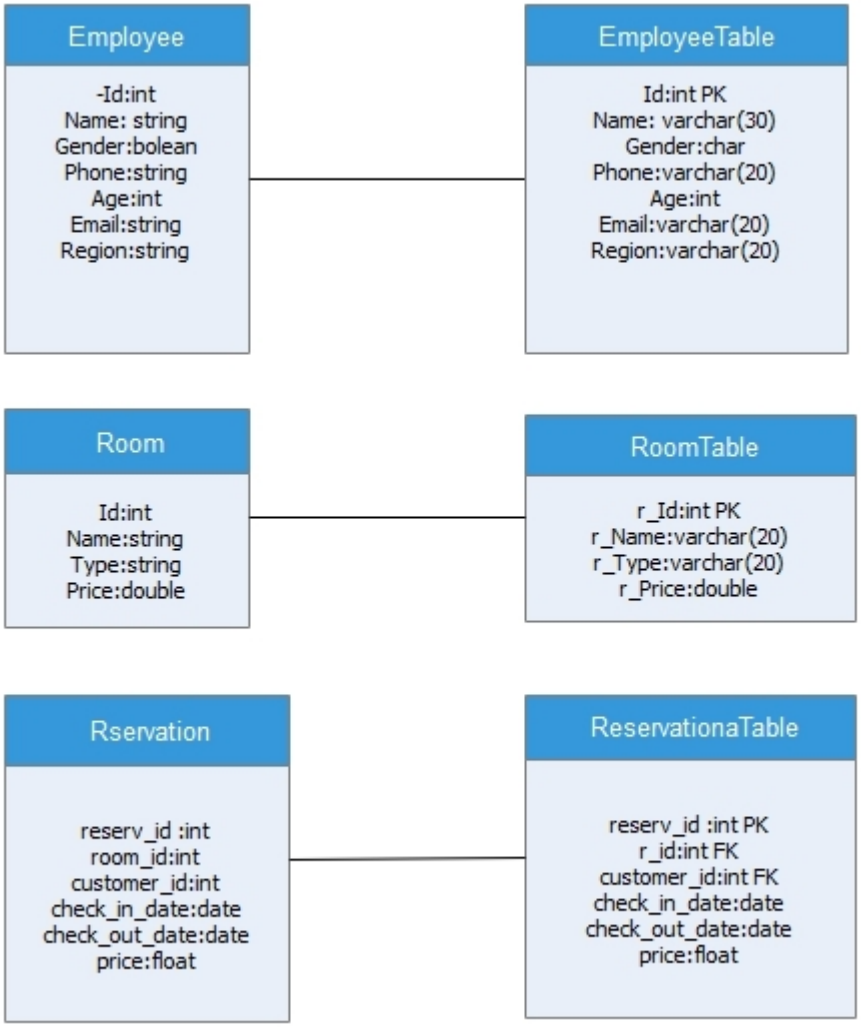

Figure 5. 5: persistence data management

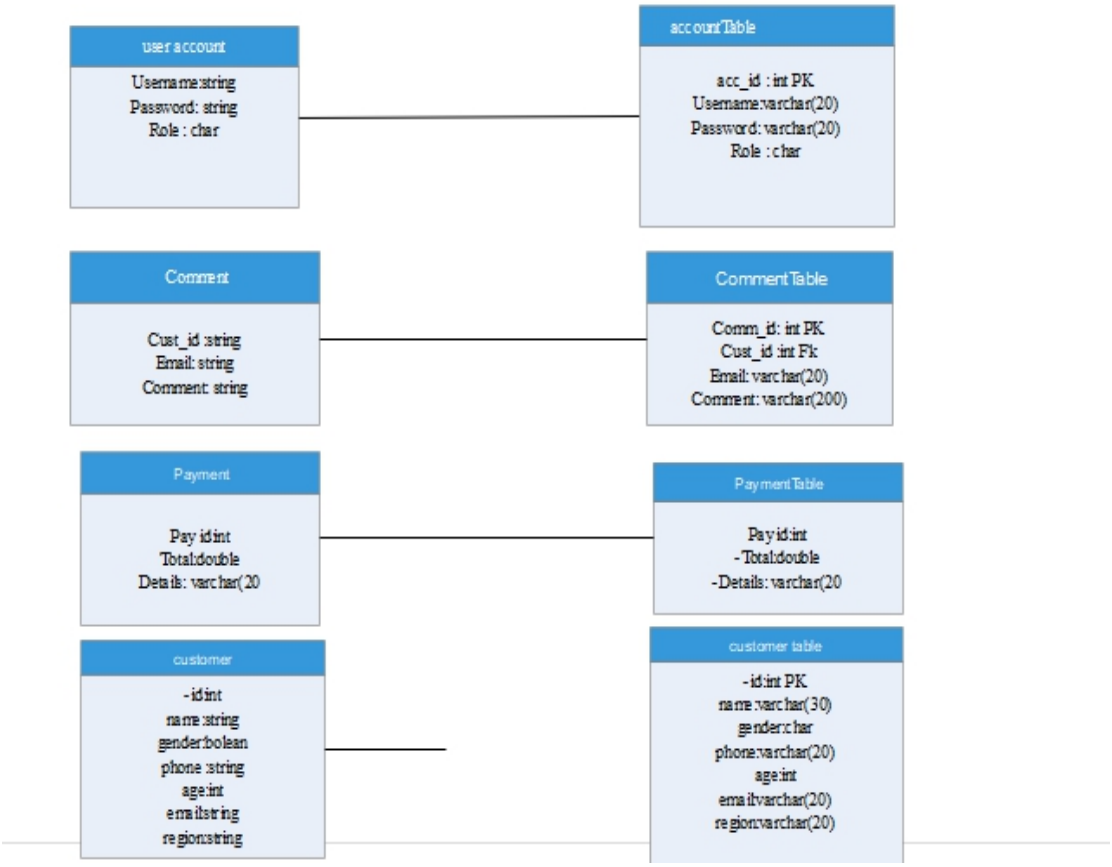

Figure 5. 6: persistence data management

5.3.5 Access control and security

Access control and security is very important for our system. Good security requires physical access control, reliable personal, trust worthily

Soresa hotel reservation System is centrally managed by the Manager or the Administrator of the System. That is for all Customers the Administrator of the Hotel has to control the profile of each customer account. This is useful in terms of accessibility of the system and security. In term of accessibility means the system will not be accessed by unauthorized users to maintain the system security. To make the system more secure username and password will be saved in an encrypted format. When each user logs in to the system they will see part of the database that will only related to their authority. Session and cookies also used in this system for the security of browser access.

In our proposed system, there are four (4) actors that have different access to different functionality and data. Here is the table that describes the access control of each actor.

Functionality	System Admin	Manager	Receptionist	Customer
Login()	✓	✓	✓	
Reserve room()	✗	✗	✗	✓
Check payment()	✗	✓	✓	✗
Manage account()	✓	✗	✗	✗
View coomen()	✗	✓	✗	✗
Manage room()	✗	—	✓	✗
Payment()				✓
Generate report()	✓	✓	✓	

Figure 5. 7:Global access table

5.4 Packages

Package diagram is UML structure diagram which shows packages and dependencies between the packages. Package diagram enables you to gain a high level understanding of the collaboration among model elements through analyzing the relationships among their parent package.

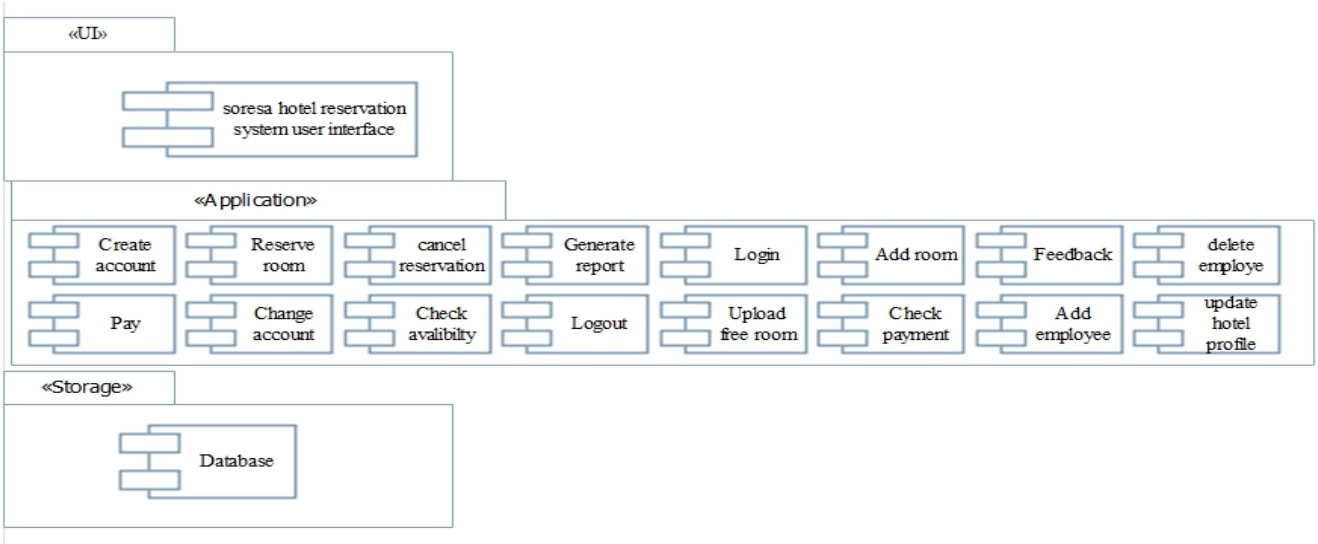

Figure 5. 8:Package diagram

5.5 Algorithm Design

Pseudo code: It's simply an implementation of an algorithm in the form of annotations and informative text written in plain English. It has no syntax like any of the programming language and thus can't be compiled or interpreted by the computer. Improves the readability of any approach. It's one of the best approaches to start implementation of an algorithm. The main goal of a pseudo code is to explain what

exactly each line of a program should do, hence making the code construction phase easier for the programmer

#Pseudocode: A Login Form module

##Entities

Email field: input type = email, placeholder: “obsilala@gmail.com”

Password field: input type = password, placeholder: “Password”

Password confirmation field: input type = password, placeholder: “Password again”

Login submit: value: “Login”, default state, disabled

##Login Flow/Logic

###On leave focus of email field

IF email is blank

Error message: “Please enter an email address.”

ELSE IF email field value is not a valid email address

Error message: “This doesn’t look like an email address. Please try again.”

On leave focus of password

IF password is not sufficiently strong

Error message: “Please replace with a stronger password.”

On leave focus of password confirmation

IF password confirmation does not match password

Error message: “Please replace with a stronger password.”

On leave focus of email, password or password confirmation

IF email AND password AND password confirmation all contain valid values

Enable Login Submit

Pseudo code: Reservation

1. Home page displayed
2. See availability of room
3. Customer checks availability of room
4. System display reservation page
5. Customer click reserve button
6. Customer fills all required fields then click save button the system check validation
7. System validate all required field and
8. If the fill form is valid go to database
9. Else back to reservation page
10. If the customer provide payment and system display successful message
11. Else the payment not provide successful message displayed

5.6 User Interface Design

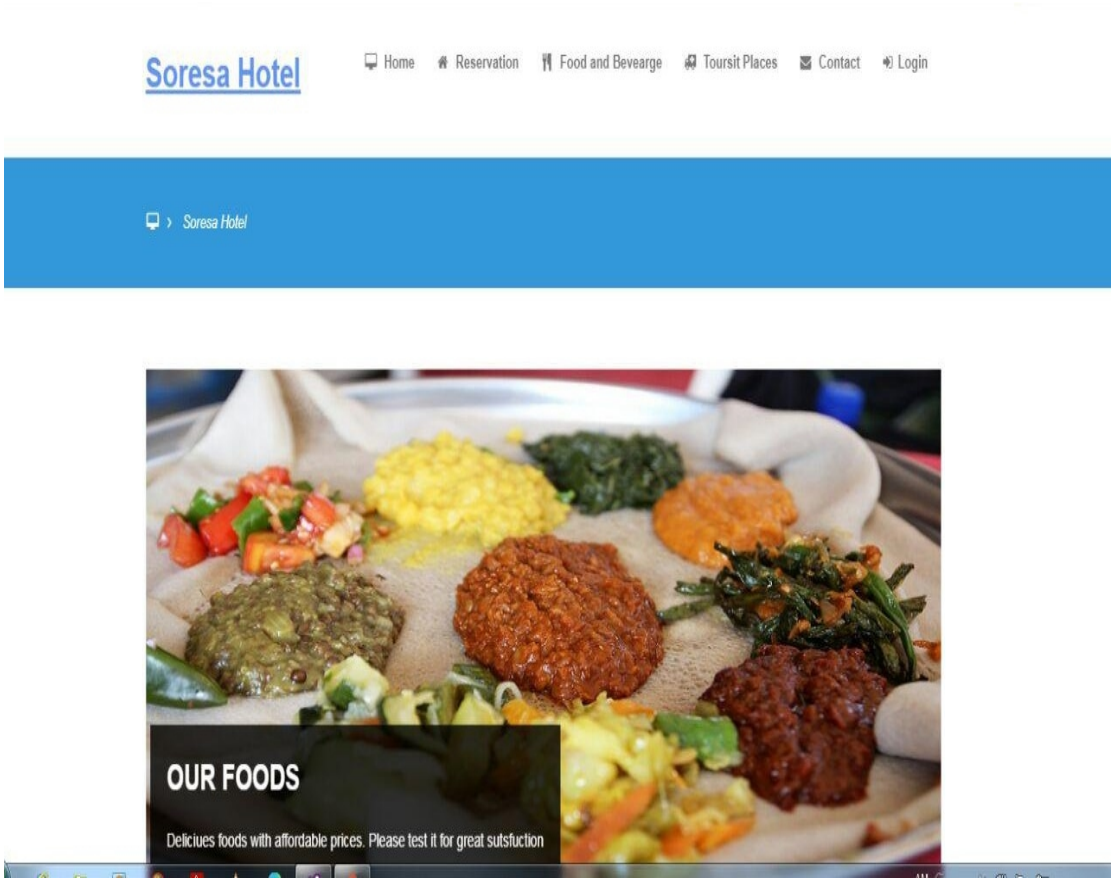

Figure 5. 9:homepage

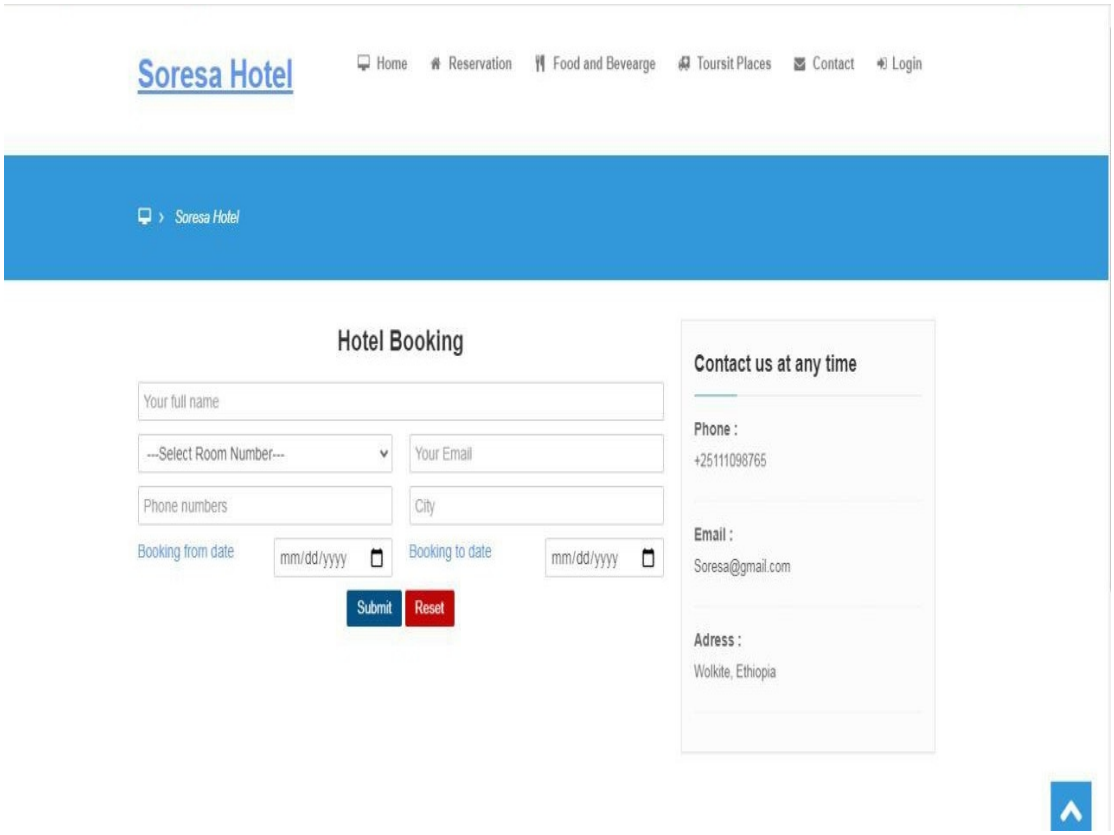

Figure 5. 10: reservation Page

CHAPTER SIX

6. Implementation and testing

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

6.1 Implementation of the database

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

6.2 Implementation of the class diagram

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

6.3 Configuration of the Application Server

We use ASP.NET server because ASP.NET is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purpose. Everything you needed is to set up a web server – server application (Apache), database and scripting language ASP.NET works equally well on Linux,Mac and Windows. Since it is suitable and the function we listed above we use ASP.NET application server.

6.3 Configuration of application security

Our system called Soresa Hotel online reservation system is an reservation issue validates all the inputs by returning error message and suggesting to try again when invalid input occurred. We implement encryption for user password by MD5 when the system admin creates a user account for manager, receptionist and others create their account or the user changes their password system encryption the password. The system has its security principles that control unauthorized authentication.

6.4 Implementation of user interface

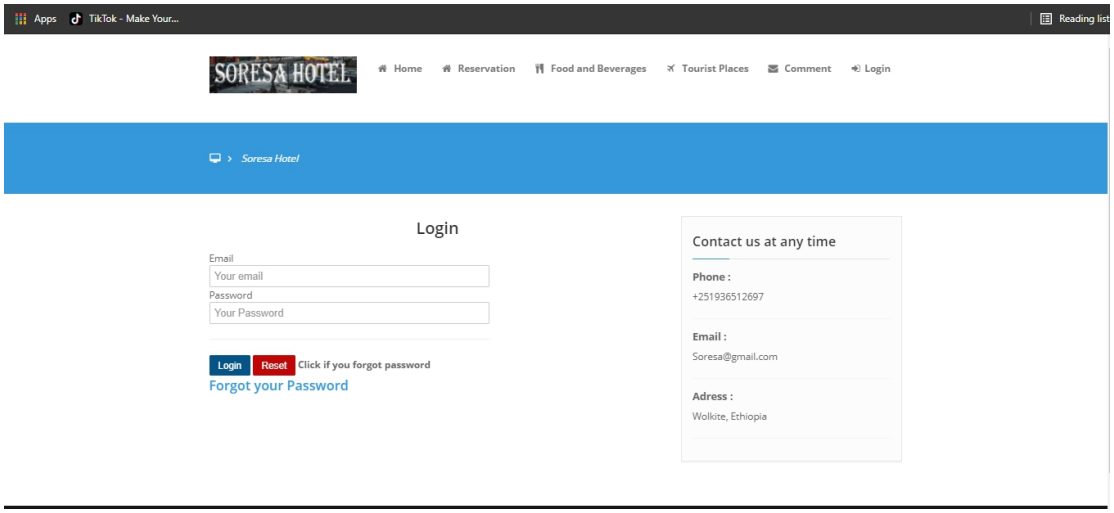

Figure 6.1:Login

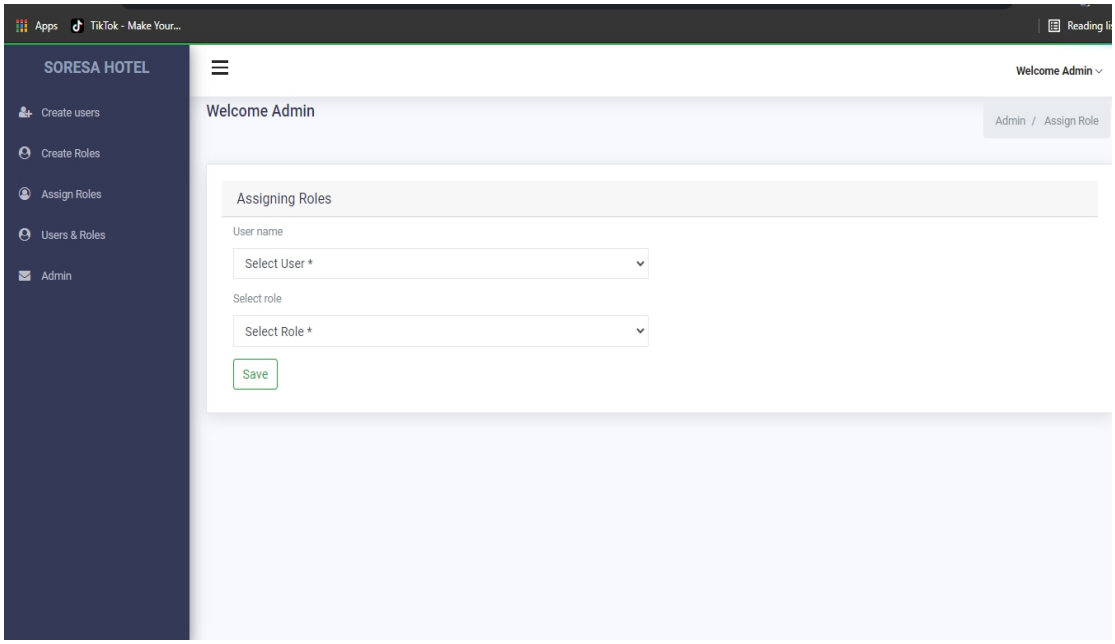

Figure 6. 2:Create User and assign Role

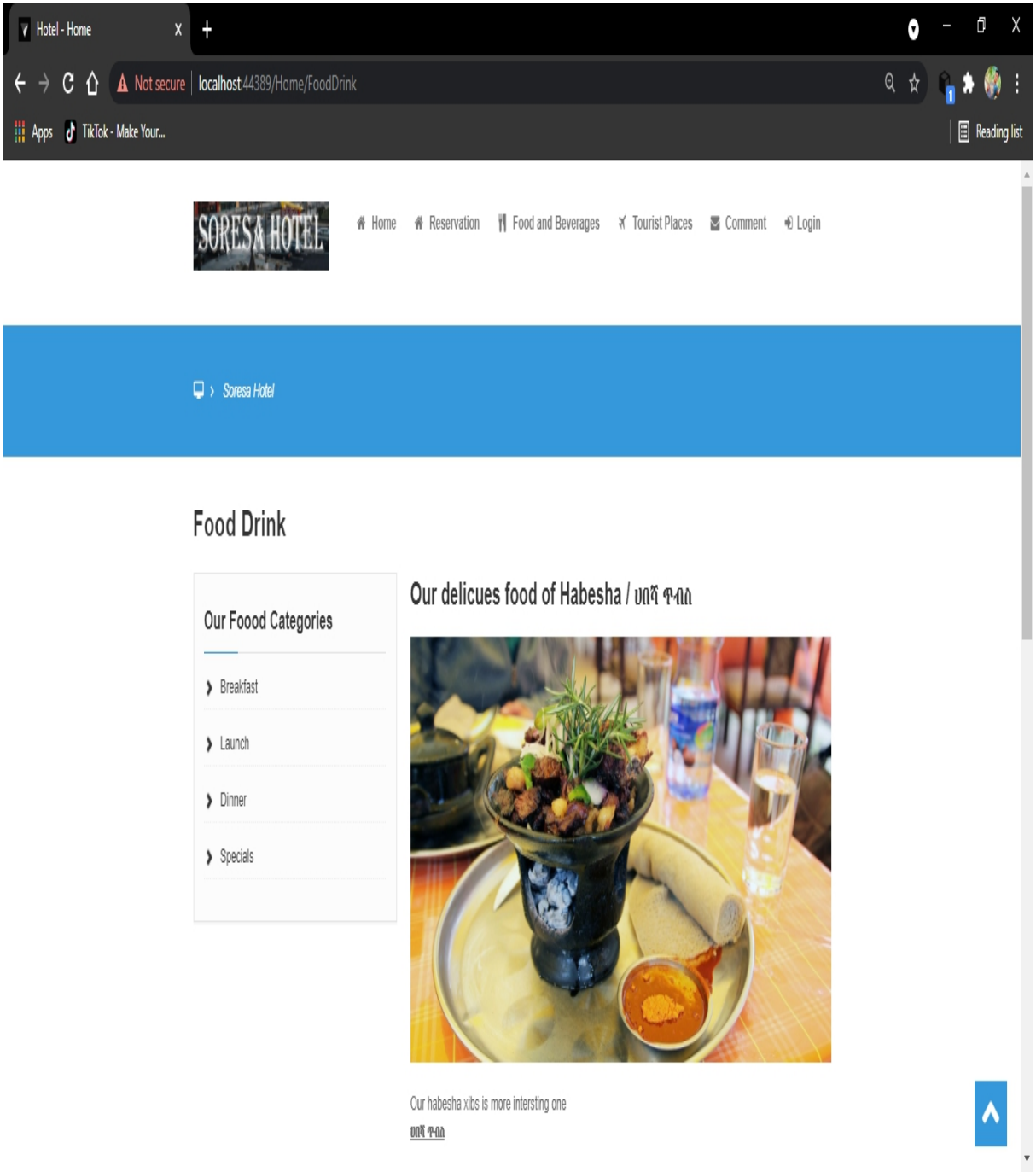

Figure 6. 3:user interface

6.6 Testing

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

6.6.1 Testing criteria

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

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

6.6.2 Test case

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

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

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

Input	Expected Result	Actual outcome	Pass/Fail
Valid User Name and Valid Password	The user logs to the system successfully	The user logs to the system successfully	Pass
If inserted only user name, Without Password	The system displays an error message “please enter password”	The system displays an error message “please enter password”	Fail
If the user name or password Text Box or both are not filled	The system displays an error message “please fill user name and password”	The system displays an error message “please fill user name and password”	Fail

Table 6. 1:Test case

6.6.3 unit testing

The main objective of unit testing to ensure that each individual part is working well and as it’s supposed to work. The entire system will only be able to work well if the

individual part are working well. Unit testing is preformed by software developers themselves. In this level of testing process, the OHR system developers test different sub procedures, functions and tested by applying the black and white box testing.

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

6.6.4 System Testing

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

Since we have tested the overall features of our systems and we have conducted testing activities to check if there are errors as well as to correct them as not to occur after or later.

The goal of the system testing phase is to evaluate all what we have been implemented in the previous development phase and to find errors in order to make corrections on the errors and these can not occur again

6.6.5 Integration Testing

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

6.6.6 performance testing

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

CHAPTER SEVEN

7.Conclusion and Recommendation

7.1 Conclusion

The main purpose of this project is to establish a long lasting and effective communication between different users and also to introduce various features regarding Soresa Hotel room reservation system.

An effort has been made to study Soresa Hotel room reservation system as partial fulfillment of BSC degree in computer science. In doing the study the team has tried to follow object oriented system analysis and design methodology.

Since the success and failure of any system depends on gathering the right information through different fact-finding techniques and user involvements, the team has made the best effort to gather requirements. After a detail review and study of the existing system of Soresa Hotel room reservation system models have been designed to reflect the new system that are suppose to solve problems.

In order to solve different problems existed the team has tried to propose a solution that at least reduce the existed problems and model the proposed system using different tools and methodologies. The team believe the different tools and techniques has helped us a lot in capturing real user requirements and model the right system for the users for their day to day transactions. Thus it should have the precedence in know-how and experience in collecting, processing and utilizing information.

This project also focuses on online room reservation and payment, but it has a little bit difficult to implement.

7.2 Recommendation

Since we are now living in a world that is led by technology and technology results, we need more and more applications to familiarize ourselves and also come up with the fast advancing technology. Thus, as we are beginner website developers, we recommend that other website designers, beginners or professionals, to create more dynamic pages that are very user friendly, more secure and also introduce the community as a whole to be familiar with the current technology. Online payment is difficult to the security purpose; we hope that this project will create some initiation for those people who wants to develop online payment.

REFERENCES

1. Final WKU reviewed industrial project guideline.
2. Object oriented software engineering pdf.
3. Professor LonnineD.Bentlley “System Analysis and Design Methods” Simon Bennet, Steve McRobb and Ray Farmer: Object oriented system analysis and design
4. Curescu, Object Oriented Analysis and Design andSoftware Development Process, England: Addison- Wesley., 2006.

APPENDIX

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using CheckProject.Models;
using PagedList;
using PagedList.Mvc;

namespace CheckProject.Controllers
{
    public class BookController : Controller
    {
        private asterDBEntities3 db = new asterDBEntities3();

        // GET: Book
        public ActionResult Index(string search, int? i)
        {
            List<BookTbl> listOfProducts = db.BookTbls.ToList();

            return View(db.BookTbls.Where(x => x.bstatus.StartsWith(search) ||
search == null).ToList().ToPagedList(i ?? 1, 4));
            //    return View(db.RoomTbls.ToList());
        }

        // GET: Book/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            BookTbl bookTbl = db.BookTbls.Find(id);
            if (bookTbl == null)
            {
                return HttpNotFound();
            }
            return View(bookTbl);
        }

        // GET: Book/Create
        public ActionResult Create()
        {
           // ViewBag.r_id = new SelectList(db.RoomTbls, "r_id", "rno");
```

```
            ViewBag.rid = new SelectList(db.RoomTbls.Where(m => m.rstatus ==
"free").ToList(), "r_id", "rno");
            BookTbl tt = new BookTbl();
            tt.Bfrom = DateTime.Now;
            tt.Bto = DateTime.Now.AddDays(1);
            return View(tt);
        }

        // POST: Book/Create
        // To protect from overposting attacks, please enable the specific properties
you want to bind to, for
        // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"b_id,r_id,fullname,emailAddress,city,phone,Bfrom,Bto,total,bstatus")] BookTbl
bookTbl)
        {
            int numberofDay = Convert.ToInt32((bookTbl.Bto - bookTbl.Bfrom));
            RoomTbl objRoom = db.RoomTbls.Single(model => model.r_id ==
bookTbl.r_id);
            int RoomPrice = Convert.ToInt32(objRoom.rprice);
            int TotalPrice = RoomPrice * numberofDay;

            if (ModelState.IsValid)
            {
                bookTbl.bstatus = "Waiting";
                bookTbl.total = TotalPrice;
                db.BookTbls.Add(bookTbl);
                db.SaveChanges();

                objRoom.rstatus = "Reserved";
                db.SaveChanges();
                ViewBag.message = "Success..";
                return RedirectToAction("Home");
            }

            ViewBag.RID = new SelectList(db.RoomTbls.Where(m => m.rstatus
== "Free").ToList(), "r_id", "rno", bookTbl.r_id);
            return View(bookTbl);

            //ViewBag.r_id = new SelectList(db.RoomTbls, "r_id", "rno",
bookTbl.r_id);
            //return View(bookTbl);
        }

        // GET: Book/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
```

```
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            BookTbl bookTbl = db.BookTbls.Find(id);
            if (bookTbl == null)
            {
                return HttpNotFound();
            }
            ViewBag.r_id = new SelectList(db.RoomTbls, "r_id", "rno",
bookTbl.r_id);
            return View(bookTbl);
        }

        // POST: Book/Edit/5
        // To protect from overposting attacks, please enable the specific properties
you want to bind to, for
        // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
"b_id,r_id,fullname,emailAddress,city,phone,Bfrom,Bto,total,bstatus")] BookTbl
bookTbl)
        {
            if (ModelState.IsValid)
            {
                bookTbl.bstatus = "Booked";
                db.Entry(bookTbl).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.r_id = new SelectList(db.RoomTbls, "r_id", "rno",
bookTbl.r_id);
            return View(bookTbl);
        }

        // GET: Book/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            BookTbl bookTbl = db.BookTbls.Find(id);
            if (bookTbl == null)
            {
                return HttpNotFound();
            }
            return View(bookTbl);
        }
```

```
        // POST: Book/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {

            BookTbl bookTbl = db.BookTbls.Find(id);
            db.BookTbls.Remove(bookTbl);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```