

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTEMENT OF COMPUTER SCIENCE

Project Title: Web-Based Absiniya Lounge Service System in WKU.

Project Team Members

Name of students	ID
1. Merga Tulu _____	CIR/067/11
2. Edosa Negaro _____	CIR/040/11
3. Motu Getachew_____	CIR/074/11

Advisor Mr.Melaku Kore

Wolkite University, Wolkite, Ethiopia.

Submission date. March .15/2022

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE:WEB_BASED ABSINIYA LOUNGE SERVICE SYSTEM IN WKU.

SUBMITED TO DEPARTMENT OF COMPUTER SCIENCE

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

BY

NAME OF STUDENT ID

1. MERGA TULU ----------CIR/067/11
2. EDOSA NEGARO ----------CIR/040/11
3. MOTU GETACHEW----------CIR/074/11

Advisor Mr.Melaku Kore

Wolkite University, Wolkite, Ethiopia.

Submission date. June. 15/2022

DECLARATION

This is to declare that this project work which is done under the supervision of **Melaku Kore**

(MSc) and having the title **Web-based absiniya lounge service system in WKU** is the sole contribution of: Merga Tulu, Edosa Negaro and Motu Getachew No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Members:

Full Name Signature

1. Merga Tulu Abebe _____

2. Edosa Negaro

_____

3. Motu Getachew

_____

Approval Form

This is to confirm that the project report entitled Web-based absiniya lounge service system in WKU submitted to Wolkite University, College of Computing and Informatics Department of computer science by: Merga Tulu, Edosa Negaro and Motu Getachew 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

Since time is of the essence of all, we give our deepest thanks to Almighty God, who has given us the opportunity to make our beginning so successful and to see the results of our efforts. There are individuals who support and motivated us in this project. For your supports we would like to express our sincere gratitude to Melaku Kore (MSc), our Advisor, who has been by our side as a big brother, who has been closely following our day-to-day work, and who has worked tirelessly for the success of this project. Sisay Tigistu (MSc), the Head of the Computer Science Department, for being our strength and giving us the courage to work hard with us.we appreciated and acknowledge you for being such supportive. We would like to acknowledge the Wolkite University lounge manager for their cooperation in providing important information on the business rule of the existing system and its major functional requirements as well as on its activities when we need it. In general we strongly acknowledge those of you who supports us from the beginning to end.

Table of content page

Figure ix

Table x

LIST OF ABBREVIATIONS xii

ABSTRACT xiii

CHAPTER ONE 1

1 INTRODUCTION 1

1.1. Background Information of the Organization 2

1.2. Statement of the problem 2

1.3. Objectives 3

1.3.1. General Objective 3

1.3.2 .Specific Objectives 3

1.4. Feasibility Study of the Project 3

1.4.1. Operational feasibility 4

1.4.2. Technical feasibility 4

1.4.3. Economic feasibility 4

1.4.3.1. Intangible benefits 4

1.4.3.2. Tangible benefit 4

1.6. Scope of the project 5

1.7 .Limitation of the project 6

1.8. Significance of the project 6

1.9 .Beneficiary of the projects 6

1.10. Methodologies 7

1.10.1. Data / requirement collection method 7

1.10.2. System Analysis and Design 7
1.10.3 .System development model 8
1.10.4. System Testing Methodology 8
1.10.4.1. Unit Testing 9
1.10.4.2. Integration Testing 9
1.10.4.3. System Testing 9
1.10.4.4. Acceptance Testing 9
1.11. System Development Tool 10
1.11.3. Case Tools 11
1.11.4. Documentation tools 11
1.11.5. Hardware Requirements: 11
1.14. Organization of the project 11
CHAPTER TWO 12
2 DESCRIPTION OF THE EXISTING SYSTEM 12
2.1 Introduction of Existing System 12
2.2. Users of Existing System 13
2.3 .Major Functions of the Existing System 14
2.4. Forms and Other Documents of the Existing Systems 14
2.4.1. Forms Used 14
2.4.2. Reports Generated 15
2.5. Drawbacks of the Existing System 15
2.6. Business rules 14
CHAPTER THREE 15
3. PROPOSED SYSTEM 16
3.1. Functional Requirements 17

3.2. Non Functional Requirements 19
3.2.1. User Interface and Human Factors 19
3.2.2. Hardware Consideration 19
3.2.3. Security Issues 20
3.2.4. Performance Consideration 20
3.2.5. Error Handling and Validation 21
3.2.6. Quality Issues 21
3.2.7. Backup and Recovery 22
3.2.8. Physical Environment 23
3.2.9 Resource Issues 23
3.2.10. Documentation 24
CHAPTER FOUR 24
SYSTEM ANALYSIS 24
4.1. System Model 24
4.1.2 Use Case Diagram of the system 25
4.3.1.2. Use Case Description 27
4.2.3 Use case Scenario 38
4.3. Object Model 41
4.3.1. Class Diagram 41
4.3.2. Data Dictionary 43
4.5.1. Sequence Diagram 48
4.4.2. Activity Diagram 52
4.4.3. State Chart Diagram 54

CHAPTER FIVE 52
5. SYSTEM DESIGN 52
5.1. Design Goals 52
5.2. Current System Architecture 53
5.3. Proposed System Architecture 53
5.3.1. Subsystem Decomposition and Description 55
5.3.1.2. Component Diagram 56
5.3.2. Hardware/Software mapping 57
5.3.4. Detailed Class Diagram 59
5.3.5. Persistent Data Management 60
5.3.6. Access Control and Security 61
5.4. Packages 62
5.5. Algorithm Design 62

5.6. User Interface Design 65
CHAPTER SIX 67
6 . IMPLEMENTATION AND TESTING 67
6.1. Implementation of the Database 67
6.2. Implementation of the class diagram 67
6.3. Configuration of the application server 68
6.4. Configuration of application security 68
6.5. Implementation of user interface 68
6.6. Testing the system 69
6.6.1 test case 69
6.1.2 Unit Testing 71
6.1.3 Integration Testing 70
6.1.4 System Testing 70
6.1.5 Acceptance Testing 71
CHAPTER SEVEN 73
7 CONCLUSION AND RECOMMENDATION 73
7.1 Conclusion 73
7.2 Recommendation 75
References 76

Figure

Figure 1. 1 ticket for non-cafe student 14

Figure 4. 2 Class diagram 43

Figure 4. 3 Sequence diagram for users login 49

Figure 4. 4customer registration 50

Figure 4. 5 Sequence diagram for manage account 51

Figure 4. 6 Sequence diagram for order 52

Figure 4. 7 Activity diagram for users login 53

Figure 4. 8 Activity diagram for report generate 54

Figure 4. 9 State chart diagram login for users 55

Figure 4. 10 State chart diagram for user's registration 56

Figure 5. 1 Architecture for web application of Absiniya lounge service system 60

Figure 5. 2 Component diagram for Absiniya lounge service system 63

Figure 5. 3 Hardware/Software mapping diagram for Absiniya lounge service system 65

Figure 5. 4 Detailed class diagram 67

Figure 5. 5 Persistent data management system diagram 68

Figure 5. 6 Package diagram 70

Figure 5. 7 User interface for Homepage 65

Figure 5. 8 User interface for login 66

Table

Table 4. 1 Login 27

Table 4. 2 Register customers 28

Table 4. 3 Register Employee 29

Table 4. 4 view comments 30

Table 4. 5Create account. 31

Table 4. 6 Active users. 32

Table 4. 7 Deactivate users. 33

Table 4. 8 View food menus. 34

Table 4. 9 Order 35

Table 4. 10 add food menus 36

Table 4. 11 Update food menus. 37

Table 4. 12 Employee data dictionary 43

Table 4. 13 Admin data dictionary 43

Table 4. 14 Manager Data dictionary 44

Table 4. 15 Cashier data dictionary 44

Table 4. 16 Barista data dictionary 44

Table 4. 17 Menu item data dictionary 45

Table 4. 18 Kitchen data dictionary. 45

Table 4. 19 payment data dictionary 46

Table 4. 20customer data dictionary 46

Table 4. 21 Order data dictionary 46

Table 4. 22 Waiter data dictionary 47

Table 4. 23 comment data dictionary 47

Table 4. 24 Order data dictionary 47

Table 4. 25 Notification data dictionary 48

Table 6. 1 test case to create account 69

LIST OF ABBREVIATIONS

DBMS ---------- ----------Database Management System.

CSS ----------Cascading Style Sheets.

Ms-word:----------Microsoft Word.

My-SQL----------My Structured Query Language.

OOSAD----------Object Oriented System Analysis and Design

PHP----------Hypertext Preprocessor.

SNNPR:----------Southern Nations, Nationalities and Peoples Region.

UML----------Unified Modeling Language.

Xamp Server: ----------Windows, Apache, MYSQL, PHP.

ICT ----------Information Communications Technology.

WKU---------- ---------- --Wolkite University.

RAM ----------Random Access Memory.

BR---------- ----------Business Rule.

SD----------Sequence Diagram.

ABSTRACT

The principal objective of this project is to demonstrate and automate the capability in Absiniya Lounge to conduct effective, automated, accurately,and inexpensively. The system handles customers' online order, view order histories, customer registration, receive customer order request, display payment information and many other functions.The general objective of the project is to design and implement a web based lounge service system for Absiniya Lounge. For the success of this project, the group will use many methods of data collection like interviewing the staffs of the Absiniya Lounge workers. System also includes the use of hardware and software's like operating system, DBMS, application software and the likes. DBMS for the purpose of database implementation, Application software has to develop user and administrative interface. From this project, we build automatically effective and efficient system for Absiniya Lounge. Simply users have a bank account on local database and can order online foods based on food menus. The system automatically deduct from customers balance to the lounge bank account balance

CHAPTER ONE

1 INTRODUCTION

Information and communication technology has a great effect especially in the developed countries. But when we see the developing countries still they are using traditional system of data communication, data storage and data movement & transfer so this influences the development of the country as well as their peoples.

Information and Communication Technologies (ICT), which is the convergence of information processing and information transmission technologies, have profound effect on every aspect of a society.

ICT has been changing and affecting communication, trade, business making, culture, education, service provision, entertainment, research and other developmental activities of present society. In a nutshell, ICT has been affecting the way people live and work.so lounges are one business area which needs ICT to computerize all of activities inside the lounge. So we motivated to develop the system for our university.

The project is developing because; many lounges have a lot difficult to manage the business such as customer ordering and reservation table. If the customer book an order and later wants to cancel the order, he is permitted to do this only within a specific time period. In this system the group would develop a Web based Lounge Service System that solves the challenges of Abyssinia Lounge in WKU.

The system handles customers' online order, view order histories, customer registration, receive customer order request, display payment information and many other functions. For the success of this project, the group will use many methods of data collection like interviewing the staffs of the Absiniya Lounge workersSystem also includes the use of hardware and software's like operating system, DBMS, application software.

1.1. Background Information of the Organization

Wolkite University (WKU) is one of the third generation universities in Ethiopia. The University is cited in the SNNR regional state, Gurage Zone, about 150 km west of the capital. WKU is giving different services to the students and other staff members. Among those services, students' Absiniya Lounge is the one that provide food service for students. This Absiniya Lounge was established in 2004 E.C. together when the university is established.

The purpose of this lounge is to give food service for Absiniya lounge. This lounge gives services like soft drink, hot drink, food etc. for students (it includes non-cafe students) and academic staffs. These lounge works its activities like report generation, food ordering and search mechanism are by manually.

By using manual customer ordering it is difficult for the waiter to keep the correct customer information and may lose the customer information. The customer is also given the facility to view the status of the order to determine if it is ready.

1.2. Statement of the problem

- Nowadays, Absiniya Lounge in WKU manage their business by manual especially take customer ordering and waiter takes the customer ordering by manual system with using paper. It takes 5 to 10 minutes to book the order and waiter book the order on paper so there is probability of lost and duplication of customer information. Lounge management system puts the order in a queue with specific priority according to time and quantity, and then a chef is assigned for the specific order to complete it.
- It leads to wastage of paper and also it requires reprinting of all menu cards ticket for user. Also, in many cases for small change to be making in menu card it is not convenient to print all menu cards again and again. Simply saying that the menu card once printed can't be changed. After some days, the menu card lost its worthy look and

attractiveness.As a result, the current system (manual system) is not effective and efficient to use anymore because there is no current system that save, manage and monitor the lounge waiter information, menu information, customer ordering in formation and generate report well.

- There is face-to-face contact between customers and lounge workers.

1.3. Objectives

1.3.1. General Objective

The general objective of the project is to design and implement a web based lounge service system for Absiniya Lounge.

1.3.2 .Specific Objectives

In order to achieve the general objective of this project, the following specific objectives were addressed:

- To analyze the problems in the existing (manual system) Absiniya Lounge.
- To design the system.
- To model the system.
- To implement the system
- To testing the system.

1.4. Feasibility Study of the Project

Accessing feasibility means answering question to the utility and viability of the system that is going to be developed. We have analyzed the feasibility of the system in terms of the following:-

- Operation Feasibility
- Technical Feasibility
- Economic Feasibility

1.4.1. Operational feasibility

The lounge service management system which we will develop is operationally feasible by giving training in deployment of the system and making them familiar with the proposed system. Since we are using graphical user interface in order to accept inputs and generates different outputs from the system. So it becomes easy to the user to use and familiar with the proposed system. In addition to this the system has a user manual and help and support facilities to them.

1.4.2. Technical feasibility

In technical feasibility, we notified that our new system can be implemented with current technology and the user has enough experience using that technology. Technical feasibility addresses three main things:

- Is the technology practical
- Do we currently pass the necessary technology?
- The ability to do on the technologies
- Our system is technically feasible.

1.4.3. Economic feasibility

Economic feasibility is the process of identifying the financial benefits and costs associating with the project being develop.

1.4.3.1. Intangible benefits

It refers to items that cannot be measurable in terms of money. The following Intangible benefits are as follow: Minimizing the workload of the employee (kitchen), Give moral satisfaction for the user of the system, Reduce Resource Consumption increase security and Increase Management flexibility.

1.4.3.2. Tangible benefit

Is a benefit that can easily be measurable? Our system will provide tangible benefits such as:

- Cost reduction
- Improvement in management
- Planning and control.
- Increase sales.

1.6. Scope of the project

The scope of this project is to develop web based lounge service system for Absiniya Lounge. There are different kinds of lounge service system in the Wolkite University. But this project targets to do web based Absiniya Lounge service system. The system contains modules that can handle users' and candidates' registration system, including cart module. The project will cover the following activities:-

- Allow feedback from any user.
- Generation of daily, weekly and monthly sale reports.
- The system calculate profit.
- The system give priority for golden customer.
- The system allow to print receipt.
- The system give recommendation to customer.
- **Payment** This module is used for the money transfer and getaways between the customer and cashier.When the customer start to make a order the system checks their balance and if it has enough for the price of that product they will go ahead or else they don't. Process the instance payment after every order: if consumers validate the product or it will be paid some of it when they are ordered from farmers for trust. And the whole will be paid after every transaction like delivery and validity of product and arrival checked and validated.The customer have already the menu then he/she knows his/her balance before ordering.

1.7 .Limitation of the project

- Our system didn't support local Language like Amharic and Guragigna. Because there is insufficient time to study and implement it.
- The system does not cover the payroll system of staffs in the Lounge.

1.8. Significance of the project

The main benefit of this system as it is computerized web-based system:

- It increase quality services than the manual one at the right time and ensure the users satisfaction.
- It saves the customer's time when they want to access service from the lounges.
- The customers uses the system safely and gives comfort for them.
- It initiates and increases employee confidence to do their task properly.
- It makes the lounges efficient and more profitable in a short period of time, as it follows fully automated system and has many customers.
- It increases performance of the lounges.

1.9 .Beneficiary of the projects

The first beneficiaries of these proposed systems are customers, employees and our group member. We have knowledge on how to access information from the lounges. In addition, solve the problems of customers to get full access from the lounges.

- **For managers**:
- ✓ It will decrease work load and increase security.
- ✓ Increase Speed of activity.
- ✓ Increase the numbers of users.
- **For students:**
- ✓ It will help them to use their time wisely.
- ✓ He/she can see the type of food online.

- **For lounge owner:**

✓ It will increase profit, manage the service easily manage waiter as well as cashier.

✓ This system require less number of workers in the lounge.

1.10. Methodologies

1.10.1. Data / requirement collection method

We used different methods of data collection methods to gather information from both primary and secondary source those are: -

- **Interview** –we asked the general managers of Absiniya Lounge about the overall activities of the lounge.
- **Document analysis** – to get more information about the Absiniya lounges, we analyzed the written document which contains all about the six lounges like their food menus, reports.
- **Observation** –Assessing and analyzing overall system that has been carried out observing the current working system, we observed the current working of Absiniya Lounges and how their system (manual system) work and we conducted a physical observation. And how data are handled and information is kept in the systems. Generally, the above data collection tools describe mainly focused at manual-based activities. Due to this reason, the group members decided to change manual based activities into computerized system.

1.10.2. System Analysis and Design

System Analysis: is the process of breakdown an entire system into module, analyzing each module separately, and determining the relationship between them. In system development process we use system modeling i.e. creating model of the system. System modeling is the process of creating a model of system by analyzing and organizing the system elements. This

is the phase where deeply understanding of the existing system problems and finding alternative solution. Models are: Class diagram, Activity diagram, Sequence diagram and Use case diagram (G. Booch, 1994)

1.10.3 .System development model

System development method is a frame work that is used to structure, plan and control the process of developing a system so to develop this project we are going to select iterative system development method. The reason why we are going to select this model: it is unusual to design a complete project once and instead of starting with fully known requirements, project teams implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Therefore, to design this project we will required to review and redesign in each phase iteratively to meet user requirements.

1.10.4. System Testing Methodology

The goal of utilizing numerous testing methodologies in our project development process is to make sure our software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

The key to releasing high quality software that can be easily adopted by our end users is to build a robust testing framework that implements functional software testing methodologies. Those testing are:

1.10.4.1. Unit Testing

Unit testing is the first level of testing and is often performed by our members. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

1.10.4.2. Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected. (i.e., the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either project developers or independent testers as well as manual tests.

1.10.4.3. System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of our software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

1.10.4.4. Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of our software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users for beta testing along with those of your team. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns

1.11. System Development Tool

To develop the proposed system, we would use different software and hardware components.

1.11.1. Front end: the front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS).

- **HTML:** Hypertext Markup Language is the core of a website, providing the overall design and functionality. It used to design the user interface.
- **CSS: Cascading** style sheets give developers a flexible, precise way to create attractive, interactive website designs.
- **JavaScript:** This event-based language is useful for creating dynamic elements on static HTML web pages.

1.11.2. Back end: The back-end, also called the server side, consists of the server which provides data on request,the application which channels it, and the database which organizes the information.

- **PHP:** PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages and it can be run in any platform.
- **XAMPP 7.3 Server**: To create, store and retrieve files to/from the database.

- **MYSQL: -**It is the database management system tool for adding, accessing and processing data in a database.
- **Web server:** APACHE: -It is an HTTP web server notable for playing a key role in the initial growth of the World Wide Web and the most popular web server to host and handle HTTP request from client machines to the web server.

1.11.3. Case Tools

- Microsoft Visio 2003: it is one of the appropriate tools for use UML modeling.

1.11.4. Documentation tools

- MS-Word 2019: in order to document the project.

1.11.5. Hardware Requirements:

- Desktop and Laptop computer: To work documentation, implementation and access internet available.
- RAM Size 4GB
- Hard disk size- 465.75GB
- Flash disk size-8,16GB.

1.14. Organization of the project

In these project documents we are discussing system details in each chapter. Before we begin to write chapter one the document includes acknowledgment, list of acronym, abstracts, table of contents, list of figure, and list of tables. The first chapter of this document includes the Introduction part of the project which includes sub topics like background, existing system study, proposed system, general objective, and specific objective, significant of the system, scope and methodologies and tools. The second chapter of the document includes system feature like existing system, problem of existing system, proposed system, requirement analysis like functional requirement system use case, use case documentation, business rules, user interface prototype, activity diagram, sequence diagram, analysis class model logical

model, nonfunctional requirement and system requirement like software requirement and hard ware requirement.

The third chapter of this document describes system design architecture diagram such as component diagram and deployment diagram, detail design like design class diagram and persistent model, user interface design and access control and security. The fourth chapter of this document describes as implementation and testing chapter five .System Design. Chapter six . Implementation and the last chapter includes conclusions and recommendations of the project.

CHAPTER TWO

2 DESCRIPTION OF THE EXISTING SYSTEM

2.1 Introduction of Existing System

The Current system is the manual system that needs intensive human labor, consume time. Customers order by using tickets which contains thicket number and food type and

registration of customers (especially non-cafe students) are recorded manually. If they want to update these files and records, the searching activity is very tedious and boring. Here there is no legacy system in this lounge so we cannot show some demos or scenarios on the old system. So, we are trying to show how the manual system is working in this lounge. Student going to the lounge and stand in front of the cashier and orders the food to the cashier then the cashier receives birr from student and gives bono to student then student go to the machine worker and gives the bono to him then the machine worker will give the food to student. Generally the existing system uses manual way of implementing tasks and information sharing. This is inconvenience in terms of providing integrated service

2.2. Users of Existing System

There are different users in the existing system:-

1. Customers - customers are the system users which can get service from the lounge

- ❖ They view the type of food from menu and order.
- ❖ Make payment in cash to cashier.
- ❖ If a student is non-cafe he/she may buy ticket.

2. Cashier:-cashier is a person which is responsible for:-

- ❖ Give ticket for the students/customers.
- ❖ Report the sold food daily in paper.

3. Managers:-

- ❖ Register employee and other employees.

4. Waiters:- is persons which deliver the foods and have the following duties:

- ❖ Call customers/students.
- ❖ Deliver the food to the students/customers

5. Chef: are persons which are preparing foods.

- ❖ View order histories and prepare food based on the type of ordered food.

6. Barista: are persons which are preparing tea and coffee.

- View order histories and prepare tea and coffee. Based on the type of order.

2.3 .Major Functions of the Existing System

The existing system at Abyssinia lounges services uses manual system. The first activities of existing system is:-

- Prepare ticket for non-cafe students in the form of paper.
- The existing system also upload daily food menus and collecting payment from non-cafe students and any user's in cash.

2.4. Forms and Other Documents of the Existing Systems

2.4.1. Forms Used

- In this lounge the cashier also gives ticket for non-cafe students in the form of paper.

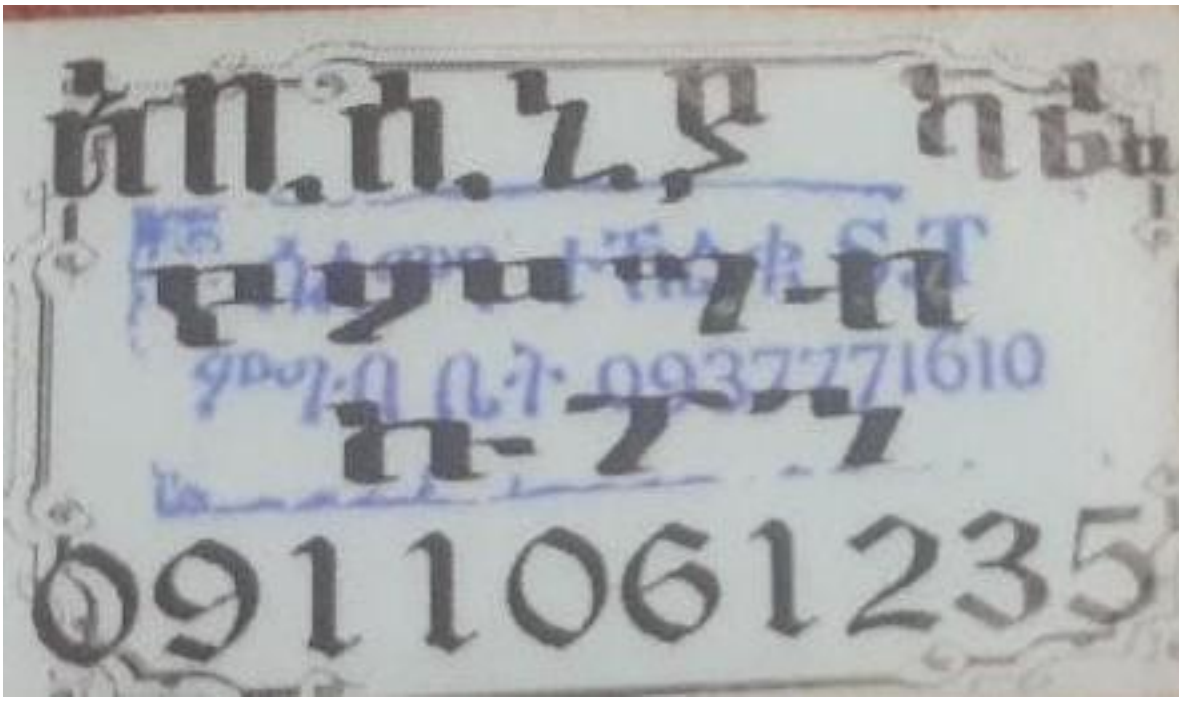


Figure 1. 1 ticket for non-cafe student

2.4.2. Reports Generated

The report can be generated by Cashier in all daily activity in the lounge is stored on the temporary table and the manager of the lounges view this report.

2.5. Drawbacks of the Existing System

1. Information Problem

- ❖ Due to manual collecting of data, there are a redundant records and inconsistency Problems.
- ❖ The inaccuracy of data and information may produce.
- ❖ Poor flow of information between managers and cashiers.
- ❖ Lack of a well-organized database system.

2. Registration related problem

- ❖ The food registration is done manually. This, in turn, will result in poor techniques for handling the data. The Data can be lost or redundantly registered.
- ❖ Relying on a manual ordering process or a paper-based system will never improve the business processes and can jeopardize your business growth.

3. Efficiency Problem

- ❖ The efficiency of the existing system is not optimal, because ordering of takes much more time.
- ❖ It is also labor-intensive process.

2.6. Business rules

Business rule identifications for manual or existing system are:-

- **BR1:-** For non-café students: they are customers of the lounges those gets services from the lounges day to day(at breakfast, lunch and dinner time) and permanent users until agreements of non-café registration disclosed.
- **BR2:-** One manager manages many lounge members.
- **BR3:-** Non-cafe students must have ticket which shows a member of this lounge.
- **BR4**:- Customer cannot pay with a credit card.
- **BR5:**- Many customers can make many orders.
  - **BR6**:-All meals in a single order must be paid for using the same payment method.
  - **BR7:**-The user can order the food based on uploaded daily food menus.
  - **BR8:**-Order price is calculated as the sum of each food item price times the quantity of the ordered food.

CHAPTER THREE

3. PROPOSED SYSTEM

The system is to be developed web-based lounge service system for Abyssinia Lounge. Which used to replace the existing manual system and the new proposed system is aimed to have a login interface for all customers(non-café students and another students who are not non-cafe), manager, system admin, cashier, waiter and chef, by inserting valid user name and password will log on to the system. The proposed system works over networked environment. After log in users view daily food menus and based on daily menus order the food, view order history, give comment and logout. The proposed system have two database the one is that of main database of the lounge and another is bank database which used to show the bank transaction locally. The users of the system have account number and balance in the

bank database, then when customers view food menus and order the food, the system automatically deducts money from customers balance and transfer the deducted value to the lounge account number. Generally the new system will eliminate/reduce problems related to time accuracy, speed, security, and complexity in information searching, accessing and retrieving. The proposed system will enable to retrieve, insert, delete and modify needed information simply for the admin. It includes an attractive user interface that facilitates fast access to the database and generates reports easily

3.1. Functional Requirements

Functional requirements describe the interactions between the system and its environment independent of its implementation. The environment includes the user and any other external system with which the system interacts. Functional requirements that must be included in the system are:

1. **System admin:** The system allows the admin to
   - ❖ Manage account.
   - ❖ Restore Database.
2. **Manager :**The system allows the manager to
   - ❖ Register customers.
   - ❖ Register employee.
   - ❖ View customers.
   - ❖ View food menus.
   - ❖ View report.
   - ❖ Send notification
   - ❖ Manage schedule.
   - ❖ Manage account.
3. **Customer :**The system allows students to
   - ❖ Order based on menus,

- View menu item.
- Give comments,
- Make payment.
- Update account.

4. **Cashier :**The system allows the cashier to
   - View menu item.
   - Check payments.
   - Receive notification sent from manager.
   - View comment.
   - Update account.
5. **Chef:**The system allows the chef to
   - View menu item.
   - View order histories.
   - Prepare order.
   - Manage menu item.
   - Update account.

6. Waiter : The system allows the waiter to
   - View menu item.
   - View order histories.
   - Take order.
   - Manage menu item.
   - Update account.
   - Receive notification sent from manager.

7. Barista: The system allows the barista
- View menu item.
- View order histories.
- Prepare order.

- Manage menu item.
- Update account.

3.2. Non Functional Requirements

Non-functional requirements (also known as quality requirements), are requirements which impose constraints on the design or implementation (such as performance requirements, security, or reliability)

The following are the main lists of the system's non-functional requirements:

3.2.1. User Interface and Human Factors

The system can easily be accessed by an authorized person. This accessibility can be achieved through using user name and password. The dialogue with the system will be easy to follow due to user friendly button which is very easy to communicate and users can to navigate their way through various screens. The system will have graphical user interface which suitable for any kind of user by using easily used buttons and safe web colors with more attractive to stay on the website. Generally it will include the necessary features for each user with a user friendly and attractive interface. The system should have to provide interactive and easily manageable user interface for the users. The user interface of online lounge service system is:

- Simple to use or interact with system.
- No need the expertise level of the users but give training at once to users.

3.2.2. Hardware Consideration

The hardware that is consumed by the new system is basically a computer machine that is used for all the operation of the system. The Software product to be developed should run on existing standard computers. The system will be portable that can run on any type of computer and it supports any type of browsers. This Website opens in any personal computer supporting windows 7, Windows8, Windows10 and other smart phone.

3.2.3. Security Issues

The system is much secured based on the username and password for all user activity. Nobody can access the system without the authorized person. Passwords that are not visible cannot be accessed by anyone because the passwords are encrypted by MD5 encryption. The MD5 algorithm is a very simple and easy-to-follow manner, and it offers much more assurance of data security.

3.2.4. Performance Consideration

There might be many users accessing to the web server simultaneously. As an online food order tool performance shouldn't be affected much and response time for submitted page should be less than a minute. The performance of our system is also measured in terms of load time and response time. 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 and it support concurrent user at a time.

- **Response Time**
- Upon request for user inquiry the system under normal condition should display results as quickly as possible.
- **Processing Time**
- Since the system is developing with efficient programming language and database upon request for user's activities the system under normal condition should process the request as quickly as possible.
- **Work ability**
- The system should be suitable for all users. It should be accurate in performing its functions and secured enough from attacks by external users.
- **Correctness:** The system provides correct response to the correct request.

- **Concurrent Processing:** Because the proposed system is web based can execute multiple users.
- **Concurrency:** The system should support multiple accesses of users. It should give service to multiple users concurrently.

3.2.5. Error Handling and Validation

Errors could rise from users and from the system. Errors that occurred from the wrong doing of users will be handled by appropriate exception handling mechanisms. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error and should display an error message if the user inputted the characters that are mismatched to corresponding data types.

Incorrect input: the system handles many exceptions like inserting empty string to the database and inserting the food that is not exist and display an appropriate message for each error.

Login error: the system shall handle an attempt to login with incorrect username and password and display error message. To overcome those error, there are different error handler we have to use. From those some of them are: Set Error Handler, Trigger an Error, Error Logging.

3.2.6. Quality Issues

The quality issue is a defect, deficiency, or a significant variation in products expected appearance or performance. Create a quality issue quickly with only essential information such as Organization, Severity, and Source. We use the quality issue edit page to add more information such as work flow, proposed dispositions, and descriptions. There are different mechanism that show the quality of our project from those some of them are:

Availability: The system should have to be functional at any given time (the system is available in 24 hours a day and seven days a week).There is no delay in the availability of any information, whatever needed, can be captured very quickly and easily by using more than one server. The server should be always on to be available. The system shall support the use of multiple users at a time and understandable by all users.

Maintainability: The System should be easily maintainable in case of problems and gives consistent service at all times without fluctuation until there is internet connection or local connection and should be testable.

Reliability: The proposed system will minimize crash during its run time, since more than one user could use the system simultaneously. The system should be reliable and matured enough in giving its service. It should have a fault tolerance mechanism in which it can recover fast from problems that may occur.

3.2.7. Backup and Recovery

In order to prevent data loss in case of system failure, the result of customers that were order food till then have to be saved in the database, for the system to resume the counting process on database . The system should be capable of gracefully recovering from earlier crashes and continuing the cart process. Here we will make back up and restore database easily using full backup. The basic idea is to dump database table structure and data into a SQL file for backing up. While restoring, the SQL queries dumped into the file are executed one after another. Typically, backup administration tasks include the following:

- Planning and testing responses to different kinds of failure.
- Configuring the database environment for backup and recovery.
- Setting up the backup schedule.
- Monitoring the backup and recovery environment.

- Troubleshooting backup problems.
- Recovering from data loss if need arises.

3.2.8. Physical Environment

The proposed system we are going to develop can be affected by the physical environment when a natural disaster occurs. On the other hand external conditions such as weather condition will not have any effect on the performance of the system. The new system will be affected by internal physical environment like attack by viruses, worms and the like. To overcome this problems the team member develop the backup (recovery) system to protect the data when the system damage. As we know the system is web-based, so in order to use the system their existing local area network in the Wolkite university and use host IP addresses connect to the internet.

3.2.9 Resource Issues

With our project, there are limitations and risks that need to be addressed to ensure our project's ultimate success. The three primary constraints that our group should be familiar with are time, scope, and cost.

- **Time constraint:** The time constraint refers to the project's schedule for completion, including the deadlines for each phase of our project, as well as the date for rollout the final deliverable.
- **Scope constraint:** The scope of our project defines has specific goals, deliverable, features, and functions, in addition to the tasks required to complete the project.
- **Cost constraint:** The cost of our project, often dubbed the project's budget, comprises all of the financial resources needed to complete our project on time, in its predetermined scope. We kept in mind that cost does not just mean money for

materials it encompasses costs for labor, vendors, quality control, and other factors, as well.

3.2.10. Documentation

This online lounge service systems will provide two types of system documentation. These are internal and external documentation. System documentation addresses programmers, system developers, owners, and users. Programmers include those who are currently working on the project as well as those who give support and maintain the system including the system administrator.Internal Documentation: This contains program source code.External Documentation: This also contains, use case models, the class and interaction diagrams (activity, sequence, and state diagrams).

CHAPTER FOUR

SYSTEM ANALYSIS

4.1. System Model

Object-Oriented System Analysis (OOSA) looks at the problem domain, with the aim of producing a conceptual model of the information that exists in the area being analyzed. The result of object-oriented analysis is a description of what the system is functionally required to do, in the form of a conceptual model. That will typically be presented as a set of use cases, activity diagrams, and Class responsibility collaborator. The purpose of object oriented analysis is to develop a model that describes computer software (Website) as it works to satisfy a set of customer defined requirements. (A.Hoffer, 2000)

4.1.1. Use Case Model

A use case model is a model of how different types of users interact with the system to solve a problem. The most important model elements are: use cases, actors and the relationship between them.

Use case components:

- **Actor:** is a person, or external system that plays a role in one or more interaction with the system. And represented with:
- **Use case**: describes a sequence of actions that provides something of measurable value to an actor and is drawn as a horizontal ellipse.
- **System boundary**: indicates the scope of the system project. Anything within the box represent functionalities in side in scope.
- There different actors are there in Abyssinia Lounge Service System.Those are described as follows.

4.1.2 Use Case Diagram of the system

Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from an external point of view. 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 (e.g., a user, another system, the system's physical environment). The identification of actors and use cases results in the definition of the boundary of the system that is, in differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system. As use case is a list of steps, typically defining interaction between a role or actor and the system to achieve a goal. The use cases of our system is defined as the following:-

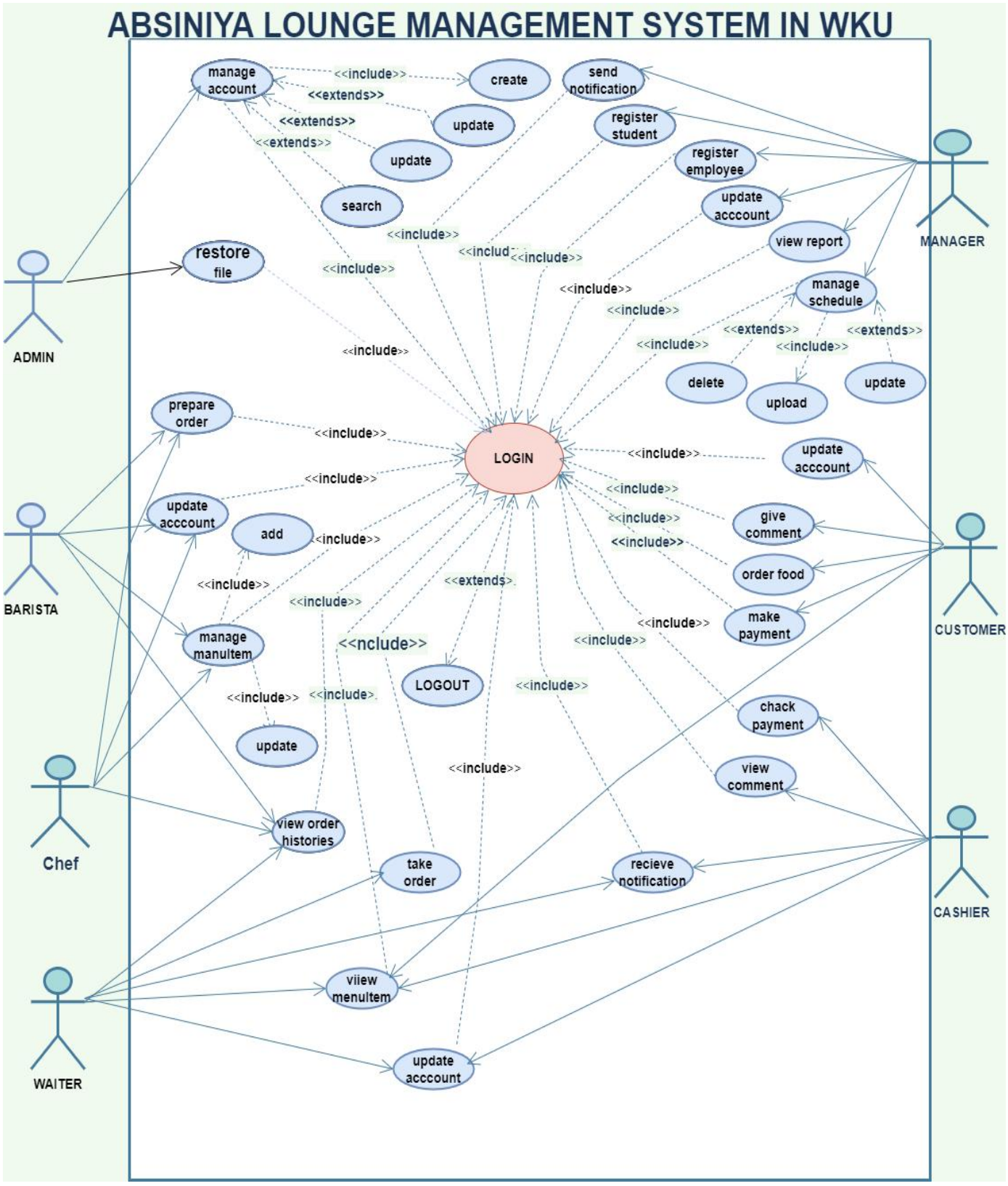

ABSINIYA LOUNGE MANAGEMENT SYSTEM IN WKU
manage account
create
send notification
update
register student
update
register employee
search
update acccount
view report
restore file
manage schedule
delete
upload
update
prepare order
LOGIN
update acccount
update acccount
add
give comment
order food
make payment
manage manuItem
LOGOUT
chack payment
update
view comment
view order histories
take order
recieve notification
viiew menuItem
update acccount
ADMIN
BARISTA
Chef
WAITER
MANAGER
CUSTOMER
CASHIER
<<include>>
<<extends>>
<<nclude>>

Figure 4. 1 Use case diagram

4.3.1.2. Use Case Description

As use case is a list of steps, typically defining interaction between a role or actor and the system to achieve a goal some of the use cases of our system are the following:-

Table 4. 1 Login

Use case Name	Login	
Use case number	UC-01	
Actor	Admin, Manager, Customers, Cashier, Waiter, Chef and Barista.	
Description:	Users are authenticated and taken to their own user interface based on the given privilege	
Pre-conditions	Users must have a valid user name and password.	
Post condition	The user is authenticated and taken to his/her own user interface.	
Basic course of action	**User action**	**System response**
	1. The user opens the home page. 2. The user clicks the login menu. 4. The user enter correct username and password. 5. The user clicks on the login button.	3. The system will display the login form. 6. The system will check the user name and password is entered correctly. 7. if valid display message “successfully login”

		8. User action performed and the wanted page displayed. 9. End-use case
Alternative course of action	If 7 is not true. The System display Error Message and return back the user to step 4.	

Table 4. 2 Register customers

Use case Name	Register customers	
Use case number	UC-02	
Actor	Manager	
Description:	Register customers	
Pre-conditions	The Manager must login into the system	
Post condition	Customers registered.	
Basic course of action	**User action**	**System response**
	1. The manager clicks on the register customers menu. 3. The manager fills the registration form. 4. The user clicks on the register button.	2. The system will display a registration form.

		5. The system will check the filled form. 6. If all necessary form is filled,display“successfully registered message” 7. End use case
Alternative course of action	If 6 is not true registration will fail and nothing changed Return to step 3 of the flow of events.	
Special requirement	Registration information should been entered correctly	

Table 4. 3 Register Employee

Use case Name	Register Employee	
Use case number	UC-03	
Actor	Manager	
Description:	Register Employee	
Pre-conditions	The Manager must login into the system	
Post condition	Employee registered.	
Basic course of action	**User action**	**System response**
	1. The manager clicks on the register button 3. The manager fills the registration form. 4. The user clicks on the register button	2. The system will display a registration form.

		5. The system will check the filled form. 6. If all necessary form is filled,display“succe ssfully registered message” 7. End use case
Alternative course of action	If 6 is not true registration will fail and nothing changed return to step 3 of the flow of events.	
Special requirement	Registration information should been entered correctly	

Table 4. 4 view comments

Use case name	View comments.	
Use case number	UC-05	
Actor	Cashier	
Description:	View comments	
Pre-conditions	The Cashier must login into the system	
Post condition	Comment viewed.	
Basic course of action	**User action**	**System response**
	1. Click on ‘view comment button. 3. The cashier read the comment.	2. Click on ‘view comment button

	4. Cashier give feedback by clicking give feedback button.	5. The system checks entered d feedback and send to users. 6. End use case.
Alternative course of action	If 6 is not true registration will fail and nothing changed return to step 3 of the flow of events.	
Exit condition	cashier logout from the system	

Table 4. 5 Create account.

Use case Name	Create account	
Use case number	UC-06	
Actor	System administer	
Description:	The administrator creates an account for Manager, Customers, Cashier, Waiter, Chef and Barista.	
Pre-conditions	System Administrator Login to the system	
Scenario	Admin create account for all types of users and Each user can enter their page by their username and password.	
Post condition	Account Created Successfully	
Basic course	**User action**	**System response**

of action	1. The system admin click on The create account button. 3. The system admin fills the form. 4. The system admin must enter Eight character password and username. 5. The system admin clicks on create ount button.	2. The system will display the user account creation form. 6. The system will check the filled form. 7. The system will display "account created successful message". 8. End use case
Alternative course of action	If admin misses the required field or entered wrong data, the system notifies "the missed field or wrong data" and the process continues from step 3 and 4.	

Table 4. 6 Active users.

Use case name	Active users.
Use case number	UC-07
actor	System Administrator
Description:	The administrator creates an account for Manager, Customers, Cashier, Waiter, Chef and Barista.

Pre-conditions	System Administrator Login to the system	
Post condition	User activated Successfully	
Basic course action	**User Action**	**System Action**
	1. Open user profile. 3. click ‘active button’	2. The system displays user’s information table. 4. The system checks the withdrawal users. 5. If user appear displayed “activated successfully message”. 6. Use case end.
Alternative course of action	If the user not appear the process continues from step 2 and display “the user doesn't exist”.	
Exit condition	System administrator may logout from the system.	
Special requirement	The registered user have its own access privilege.	

Table 4. 7 Deactivate users.

Use case name	Deactivate users.
Use case number	UC-08
actor	System Administrator
Description:	The administrator Deactivate an account for Manager, Customers, Cashier, Waiter, Chef and Barista.

Pre-conditions	System Administrator Login to the system	
Post condition	User Deactivated Successfully.	
Basic course action	**User Action**	**System Action**
	1. Open user profile. 3. Click deactivate button.	2. The system displays user's information table. 4. The system checks the withdrawal users. 5. If user appear displayed "deactivated successfully message". 6. Use case end.
Alternative course of action	If the user not appear the process continues from step 2 and display "the user doesn't exist".	
Exit condition	System administrator may logout from the system	
Special requirement	The registered user have its own access privilege	

Table 4. 8 View food menus.

Use case name	View food menus.	
Use case number	UC-09	
Participation actors	Customer, Waiter, Barista, Chef , Cashier	
Pre-condition	Open the web page and login in his/her account	
Post condition	Users view food menu.	
Basic course Action	**User Action**	**System Action**
	1. Users click 'food menus button. 3.click on 'view button'	2. The system displays daily food menus. 4. The system checks available foods. daily food menus,

		5. If food available display the found food picture with cost. 7. Use case end.
Alternative course of action	If the food not appear the process continues from step 2 and display “the food doesn't exist”.	
Exit condition	User may logout from the system	

Table 4. 9 Order

Use case name	Order	
Use case number	UC-10	
Participation actors	Customer.	
Goal	To order based on daily food menus that uploaded by the cashier.	
Entry condition	Open the web page and login in his/her account	
Basic course Action	**User Action**	**System Action**
	1. Users click ‘food menus button. 3.click on ‘view button’	2. The system displays daily food menus. 4. The system checks available daily food menus. 5. If food available display the found

	7. click on ‘order button’ 9. The customer choose quantity. 10. The customer click on pay button. 12. If the customer pay in cash. He click cash button 13. If the customer pay in bank. He click bank button.	food picture with cost. 6. The system display order histories. 8 .The system displays quantity and cost. 11. The system displays payment option. 14. The system check payment. If paid the system display “the food is order successfully” 7. Use case end.
Alternative action	If 4 and 14 are not true order will fail and nothing changed return to step 1 of the flow of events.	
Post condition	Users must be pay before order the food.	
Special requirement	Non-café students must be deposit money from the lounge account and normal users must be registered correctly.	

Table 4. 10 add food menus

Use case name	Add food menus.	
Use case number	UC-11	
Participation actor	Cashier.	
Goal	To add food menus.	
Entry condition	Open the web page and login in his/her account	
Basic course Action	**User Action**	**System Action**

	1 Clicks on food registration. 3. Fill Registration form. 4. Click on register button.	2. System display the form. 5. Check the validity of the data inserted 6 if all necessary form is filled the system display "successfully registered" message. 7 use case end
Alternative action	If 6 is not true registration will fail and nothing changed return to step 3 of the flow of events.	
Post condition	The food will be registered on the database	
Special requirement	Registration information should been entered correctly	

Table 4. 11 Update food menus.

Use case name	Update food menus.	
Use case number	UC-11s	
Participation actor	Cashier.	
Entry condition	Cashier login to the system and open existing report on the registration food menus.	
Goal	To update the data which registered before	
Basic course action	**User action**	**System action**
	1. Click on update button 3 Insert the id no.to be updated detail. 4. Change the new value that the form must have. 5. Click on update button	2. system display update form 6.system check validity 7.If valid "successfully update "message will displayed

		8 use case end
Alternative action	If 6 is no true the field is missed then will not update or it stay at it is	
Post condition	Save the updated data	
Exit condition	Cashier logout from the system	

4.2.3 Use case Scenario

A scenario is an instance of a use case describing a concrete set of actions. It describes any relevant aspects of the context in which the user is working with the system, including what information the user has on hand when beginning to use the System. It also describes for the user when and where they are working, why they are using the system, and what they need the system to do for them.

1. Scenario name: login

Participant actor: User

Flow of events for login scenario are listed below:

If users want to login to the system the first step will be enter in to the system Home Page then he/she should have login Password and user name, if he/she clicked to the login link on the home page then the system display login form display then fill the username and password.

- If he/she is not an authenticated user, he/she must enter a valid user name and password.
- Then when the user have entering a valid username and password he/she can perform the operation they need to perform and finally logout from the system.

2. Scenario name: Manage account

Participant actor: Admin

Flow of events for manage account information scenario are listed below:

- If the Admin wants to create account for user first run the system, then the system displays the system home page and then must login first by fill the username and password click the login button.
- If the Username and password are correct the home page displays and click the create account button.
- Then the Admin create account and do other operation like update and restore after complete the operation the register officer must be logout from the system.
- If a username and password are not correct must reenter the correct username and password.

3. Scenario name: Customer registration

Participant actor: Manager

Flow of events for Customer registration Scenario are listed below:

- If the manager wants to Customer registration first run the system and click login button then fill username and password.
- If the Username and password are correct the system displays the home page then click view user's information button.
- After that manager chose the Customer registration and fill the form for registration after complete the operation the manager must be logout from the system.
- If a username and password are not correct must reenter the correct username and password.

4. Scenario name: Manage food menu.

Participant actor: Cashier

Flow of events for manage food menu **s**cenario are listed below:

- If cashier wants to manage food menu first run the system and fill the username and password then click the login button.
- If the Username and password are correct the home page displays then click manage food menu button.
- Then system display add food, update food, view order history, check payment, and generate report. Then click the one if he/she need and perform the operation.
- If a username and password are not correct must reenter the correct username and password.

5. Scenario name: Order.

Participant actor: Customer.

Flow of events for Order scenario are listed below:

- If Customer wants to Order first run the system and fill the username and password then click the login button.
- If the Username and password are correct the home page displays then click Order button.
- Then system display view order history, if he/she click on view order history he/ she come to cart page and order. Then he/she perform the operation.
- If a username and password are not correct must reenter the correct username and password.

6. Scenario name: Take Order.

Participant actor: Waiter

Flow of events for deliver food scenario are listed below:

- If Customer wants to take order first run the system and fill the username and password then click the login button.
- If the Username and password are correct the home page displays then click take order button.

- Then system display view order history, if he/she click on view order history he/ she come to cart page and call customer. Then he/she perform the operation.
- If a username and password are not correct must reenter the correct username and password.

4.3. Object Model

- An object model helps describe or define a software/system in terms of objects and classes. It is a visual representation of a system's objects, actions, and associated attributes.

4.3.1. Class Diagram

Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design. You can use class diagrams to visualize, specify, and document structural Features in our models.

Absiniya lounge class diagram

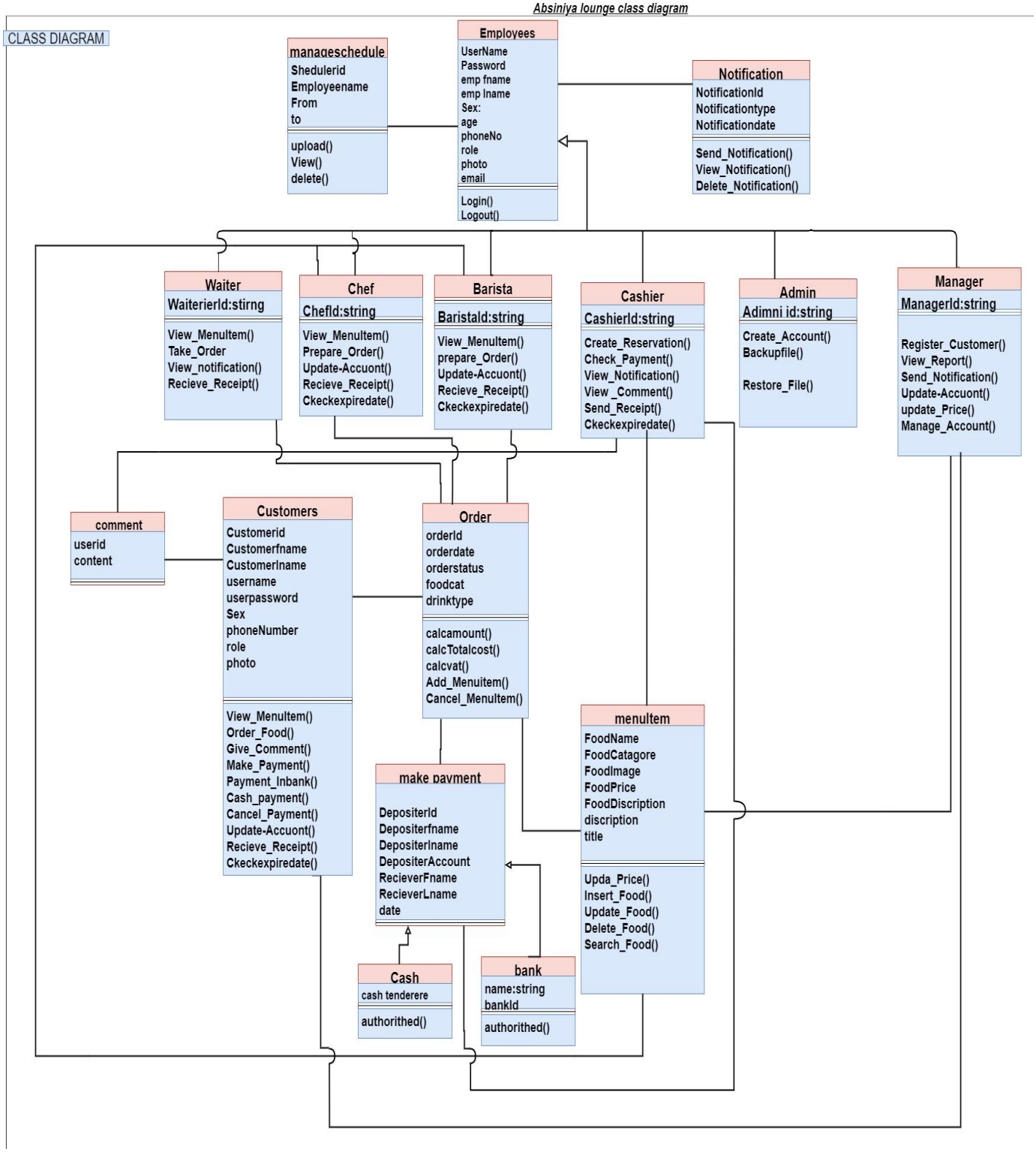


Figure 4. 2 Class diagram

4.3.2. Data Dictionary

Data dictionaries are used to provide detailed information about the contents of a data-set or database, such as the names of measured variables, their data types or formats, and text descriptions. We are going to mention it by using tabular form.

Table 4. 12 Employee data dictionary

Attributes	Data type	Data size	Constraint
Emp-id	String	15	Primary key
Emp-fname	String	20	Not Null
Emp_fname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null
Username	string	20	Not Null
Password	string	20	Not Null

Table 4. 13 Admin data dictionary

Attributes	Data type	Data size	Constraint
Admin_id	String	15	Primary key
Admin_fname	String	20	Not Null
Admin_fname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 14 Manager Data dictionary

Attributes	Data type	Data size	Constraint
ManagerId	String	15	Primary key
Manager_fname	String	20	Not Null
Manager_lname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 15 Cashier data dictionary

Attributes	Data type	Data size	Constraint
Cashier _id	String	15	Primary key
Cashier_fname	String	20	Not Null
Cashier_lname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 16 Barista data dictionary

Attributes	Data type	Data size	Constraint
Barista _id	String	15	Primary key
Barista _fame	String	20	Not Null

Barista _lame	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 17 Menu item data dictionary

Attributes	Data type	Data size	Constraint
Food_Id	String	15	Primary key
Food Name	String	20	Not Null
FoodCatagore	String	20	Not Null
Food Image	Long text	Max	Not Null
Food Price	Int	10	Not Null
FoodDiscription	String	20	Not Null
Food title	string	20	

Table 4. 18 Kitchen data dictionary.

Attributes	Data type	Data size	Constraint
Kitchen_id	String	15	Primary key
Kitchen_fname	String	20	Not Null
Kitchen_lname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 19 payment data dictionary

Attributes	Data type	Data size	Constraint
Depositer_id	String	15	Primary key
Depositer_fname	String	20	Not Null
Depositer_lname	String	20	Not Null
Depositer_Account	String	20	Not Null
Reciever_Fname	String	10	Not Null
Reciever_Lname	String	20	Not Null
Reciever_Account	string	max	Not Null
Phone number	int	20	
Sex	string	20	
Role	string	20	
Photo	Long text	max	

Table 4. 20 customer data dictionary

Attributes	Data type	Data size	Constraint
Customer_id	String	15	Primary key
Customer_fname	String	20	Not Null
Customer_lname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 21 Order data dictionary

Attributes	Data type	Data size	Constraint
Order_id	String	15	Primary key
Food _name	String	20	Not Null
Quantity	String	20	Not Null
Price	String	20	Not Null
Sender_account _no	int	10	Not Null
pay with	String	20	Not Null
Photo	Long text	Max	Not Null
Reciever_accou nt_no	int	20	Not Null
Date	date	10	

Table 4.22 Waiter data dictionary

Attributes	Data type	Data size	Constraint
Waiter_id	String	15	Primary key
Waiter_fname	String	20	Not Null
Waiter_lname	String	20	Not Null
Phone number	Int	20	Not Null
Sex	String	10	Not Null
Role	String	20	Not Null
Photo	Long text	Max	Not Null

Table 4. 23 comment data dictionary

Attributes	Data type	Data size	Constraint
User id	String	15	Primary key
Userfname	Date	20	Not Null
userlname	String	20	Not Null
content	string	45	Not Null

Table 4. 24 Order data dictionary

Attributes	Data type	Data size	Constraint
Order id	String	15	Primary key
Order-Date	Date	20	Not Null
OrderStatus	String	20	Not Null
Foodcatagories	string	45	Not Null
Drinktype	string	20	Not Null
Order-Amount	int	15	Not Null
Price	int	20	Not Null

Table 4. 25 Notification data dictionary

Attributes	Data type	Data size	Constraint
notification_ id	String	15	Primary key
Notification date	Time	20	Not Null
Number of customer	String	20	Not Null
Date	Date	20	Not Null

4.5.1. Sequence Diagram

Sequence diagrams show the interaction between participating objects in a given use case. They are helpful to identify the missing objects that are not identified in the other analysis object models

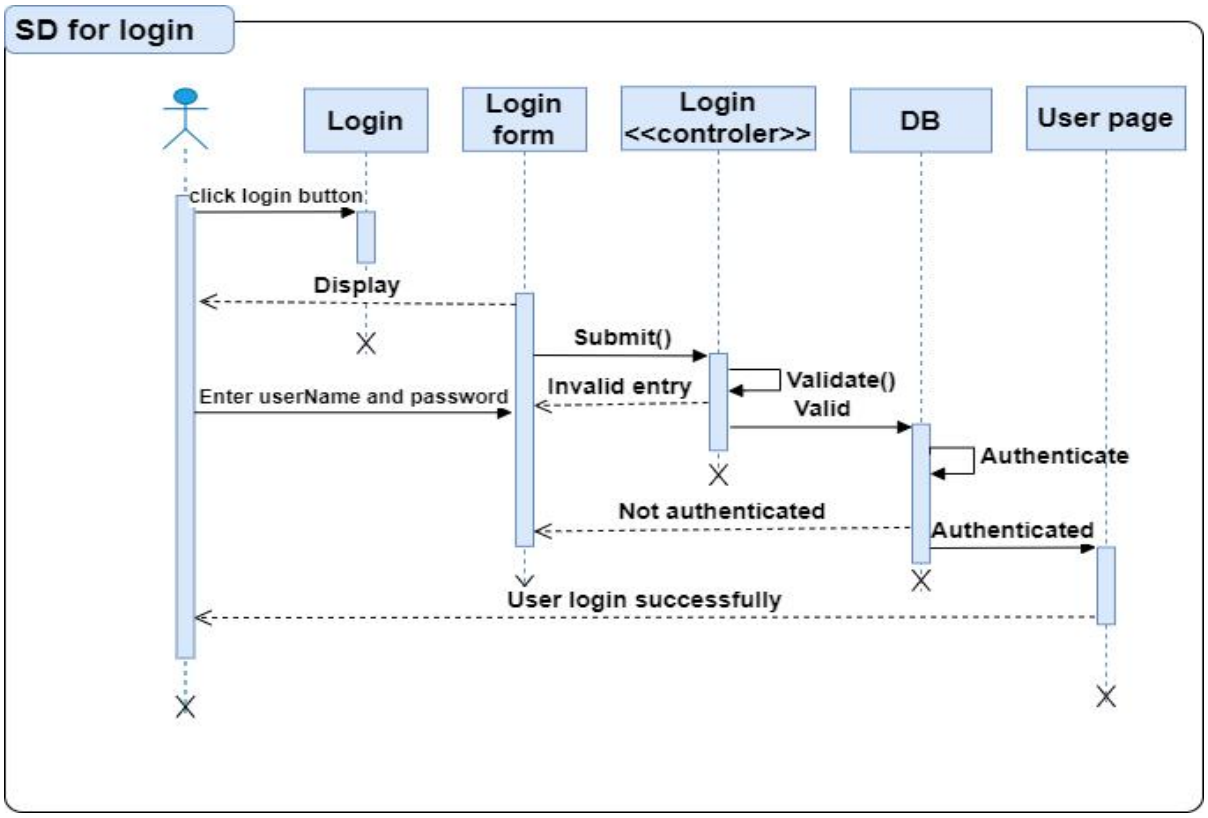


Figure 4. 3 Sequence diagram for users login

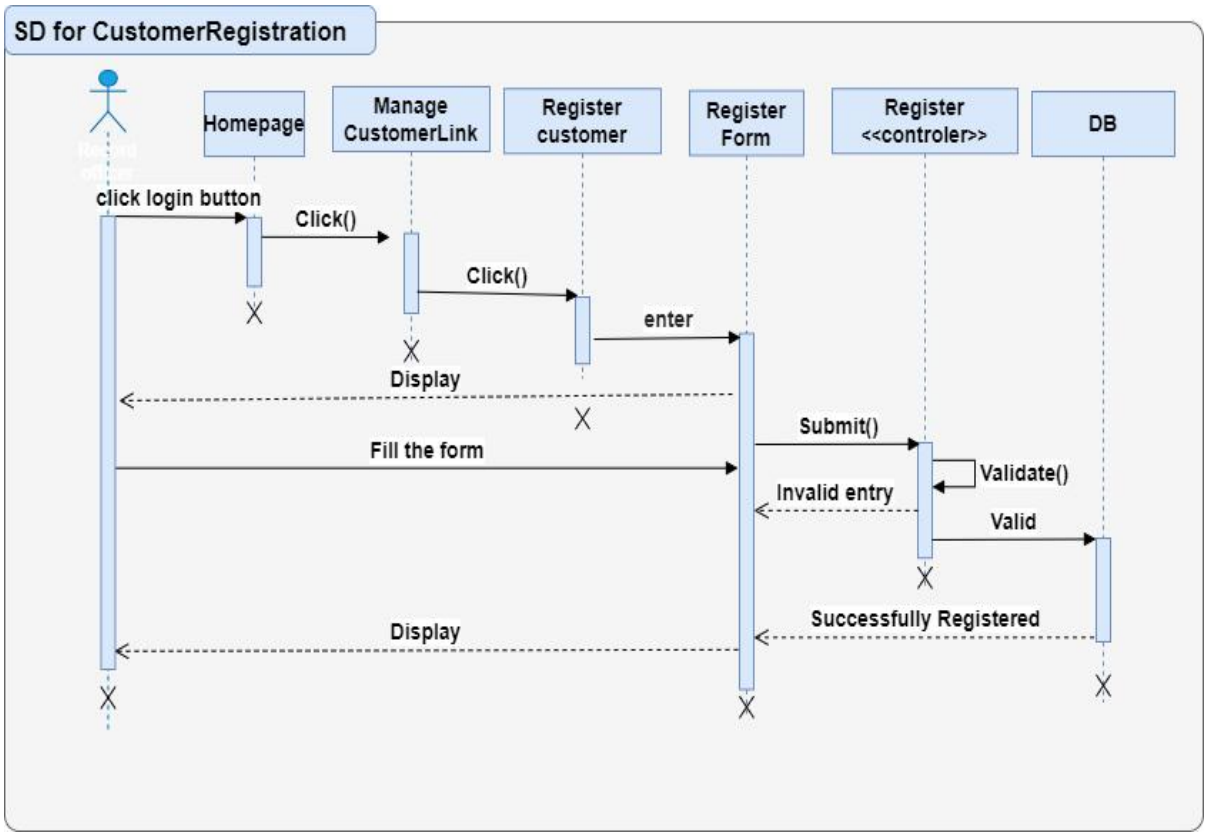


Figure 4. 4customer registration

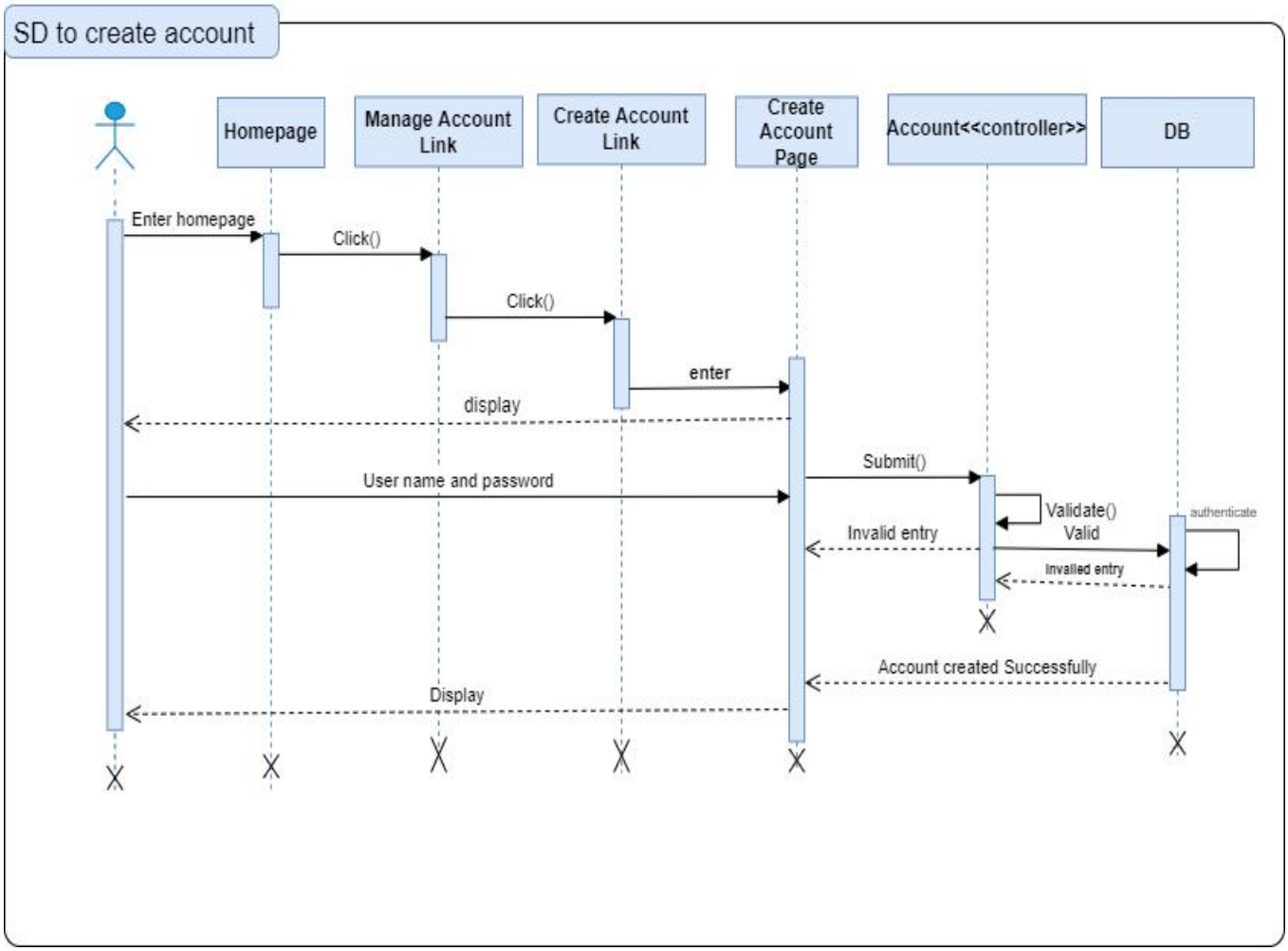


Figure 4. 5 Sequence diagram for manage account

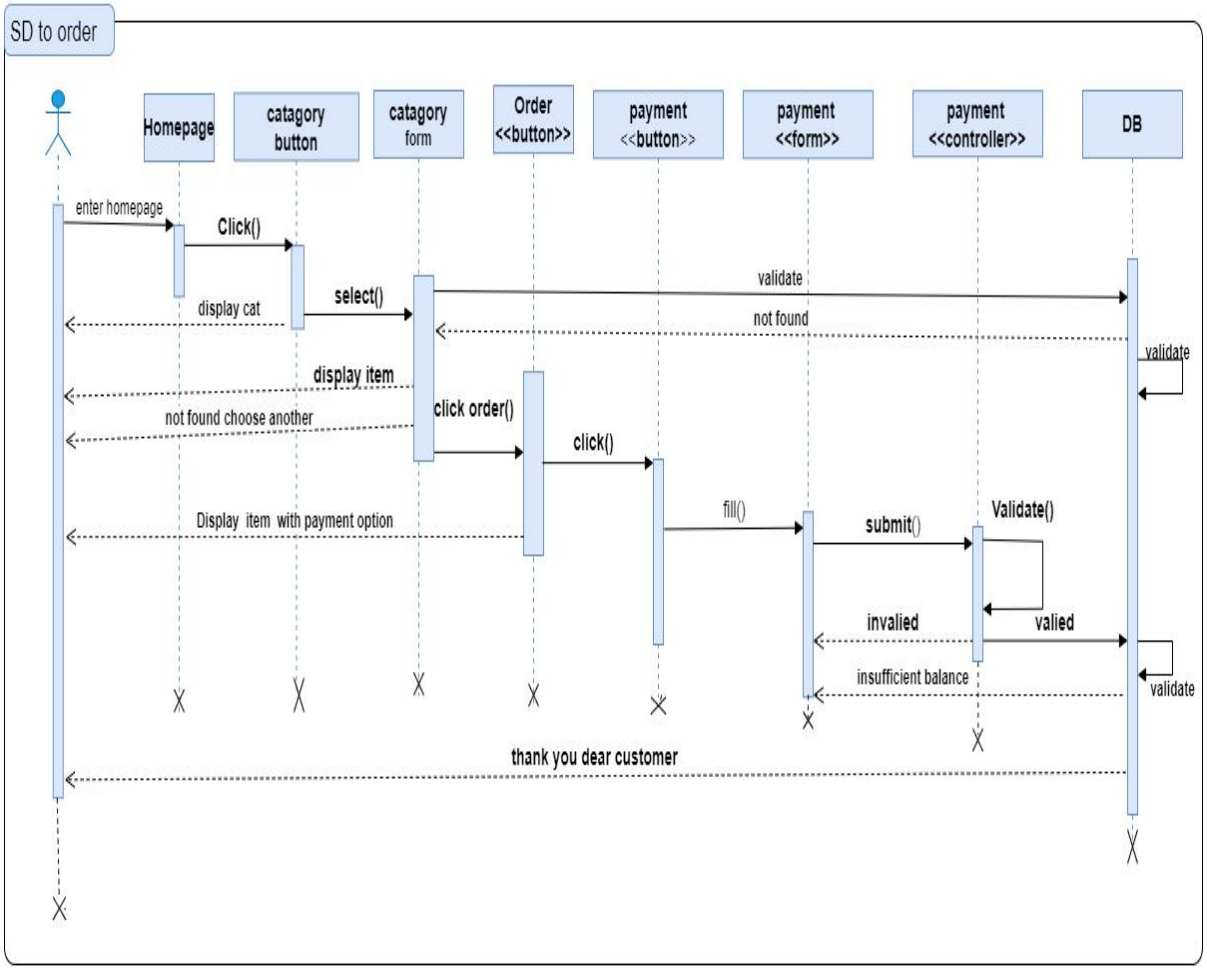


Figure 4. 6 Sequence diagram for order

4.4.2. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. So use to describe the flow of activities and a series of actions in a system similar to a flowchart or a data flow diagram.

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.Some of the activity diagram for our project is listed below.

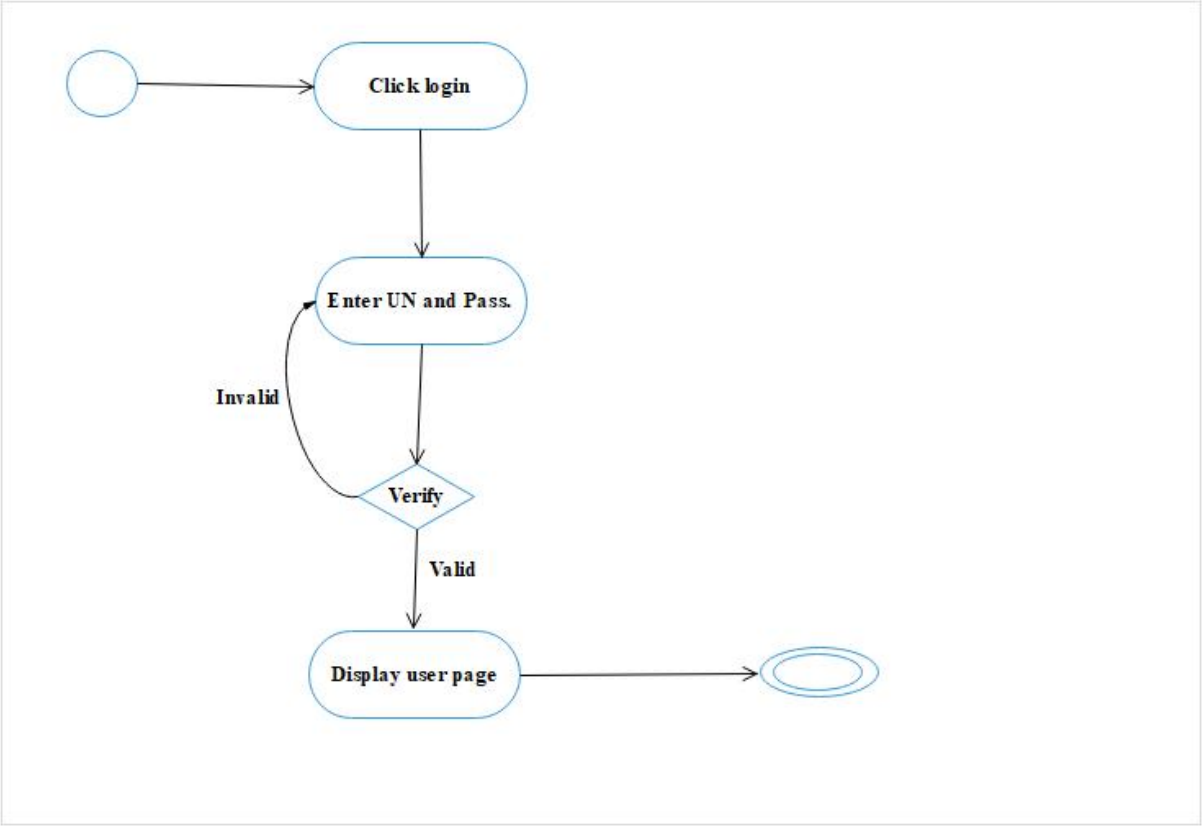


Figure 4. 7 Activity diagram for users login

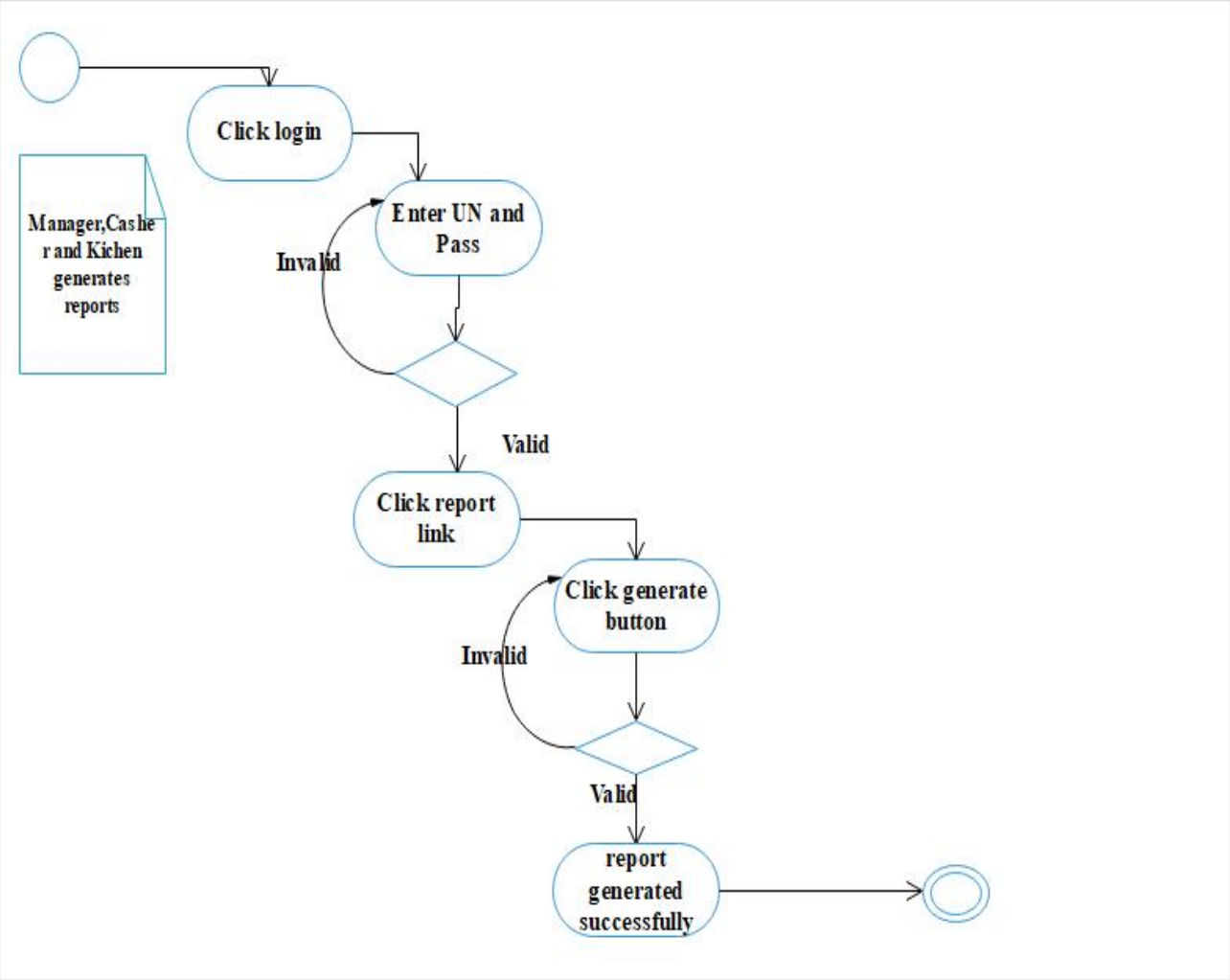


Figure 4. 8 Activity diagram for report generate

4.4.3. State Chart Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as **State machines** and **State-chart Diagrams**. These terms are often used interchangeably. So simply, a state diagram is used

to model the dynamic behavior of a class. In response to time and changing external stimuli. 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, end states and Transitions.

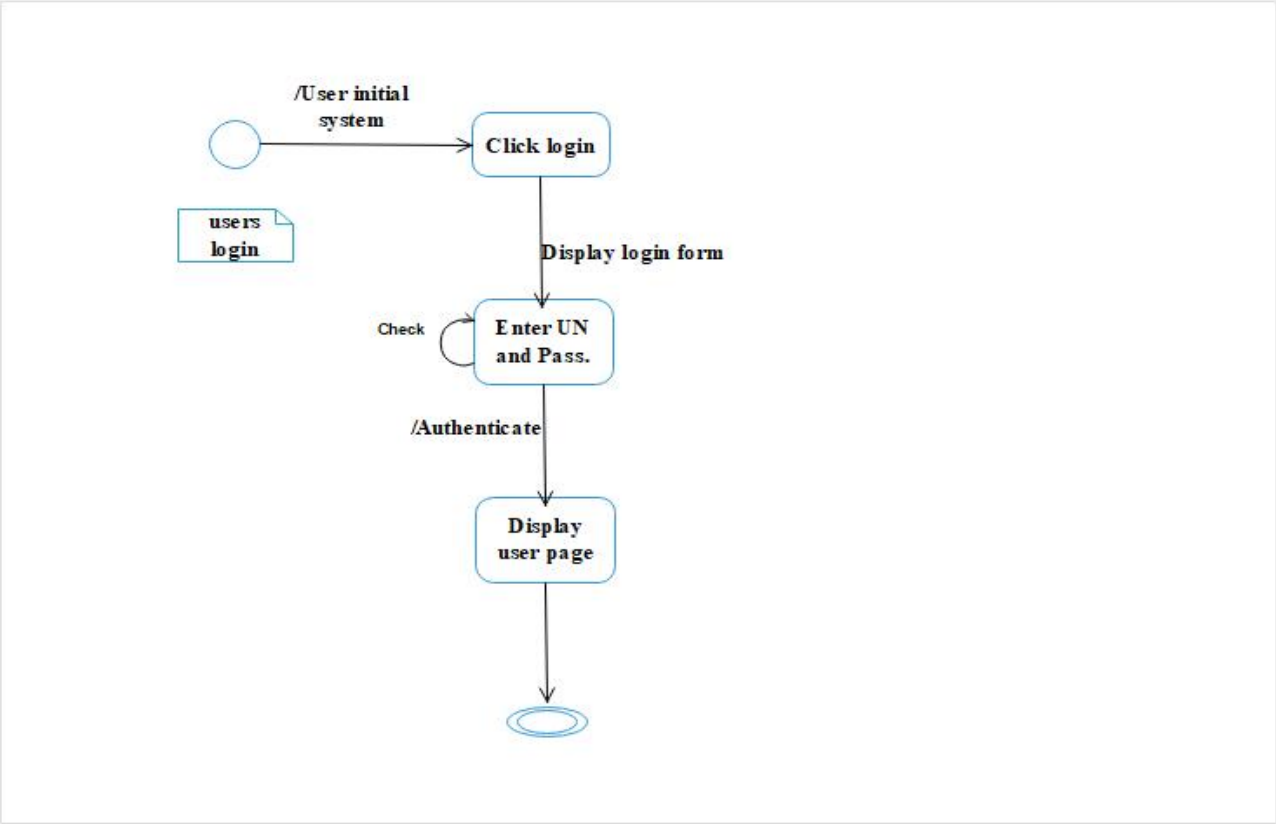


Figure 4. 9 State chart diagram login for users

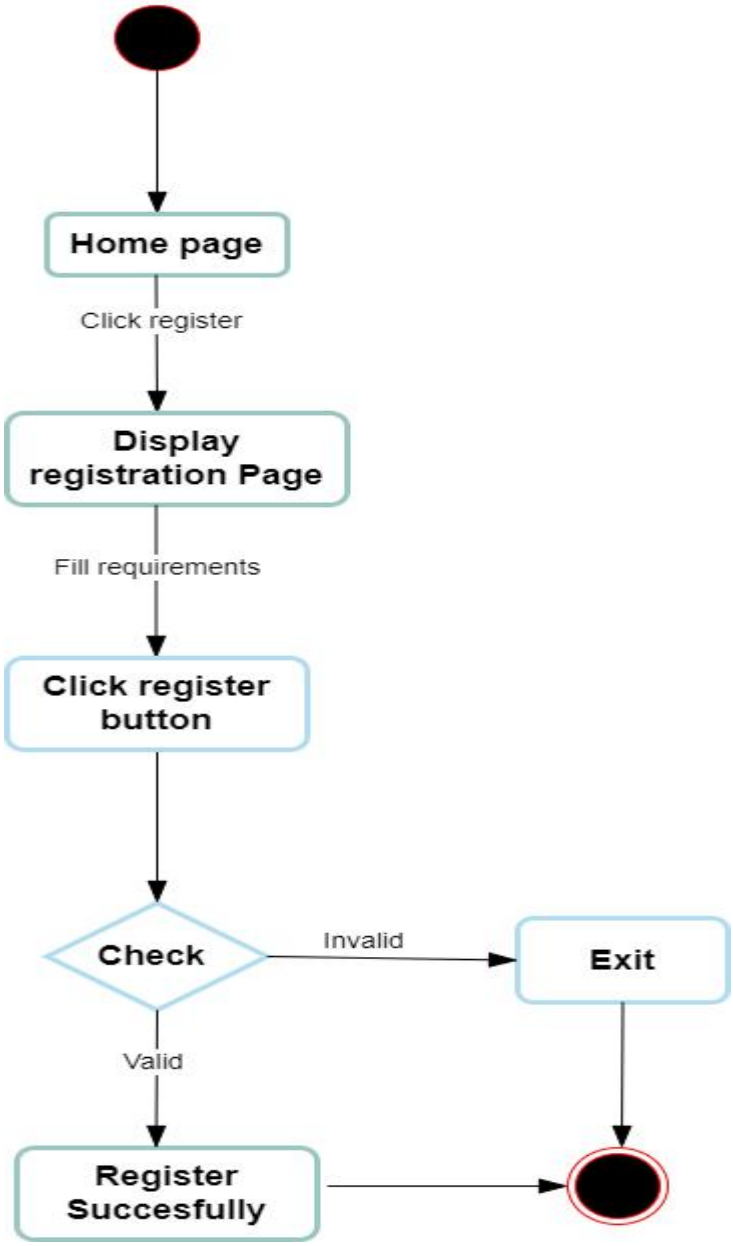


Figure 4. 10 State chart diagram for user's registration

CHAPTER FIVE

5. SYSTEM DESIGN

System design is the process and focuses on decomposing the system into manageable parts. During requirements elicitation and analysis, we concentrated on the purpose and the functionality of the system. During system design we focus on the processes, data structures, and software and hardware components necessary to implement it. The challenge of system design is that many conflicting criteria and constraints need to be met when decomposing the system. Design goals of the proposed system are inferred from nonfunctional requirements and elicited from clients. It will help to make better decision when trade- offs are needed (Lonnie.D.Bantlely, 2004).

5.1. Design Goals

- **User Interface**: - The interface is simple to understand, easy to use and user friendly interface.
- **Security:** - The system is with session controlled way to permit URL based system access it. The system must confidentially be controlled from being accessed by any no authorized users or outsiders.
- **Error Handling:** - The system provides error handling mechanism on system interacts with data base and on input accepter fields. The system handles the errors in efficient manner. It can tolerate to wrong inputs and prompts the users to correct the inputs.
- **Effectiveness:** - It achieved objectives and the extent to which targeted problems are solved.
- **Maintainability:** - Our system should be capable of being maintained in the future if needed.

- **User Interface and Human Factors**: - The interfaces are attractive, user-friendly and the user wants the interface to be graphical interface and to have hypertext and hyperlink menus.
- **Performance Consideration:** - The system will be reasonably fast in order to access the required information easily. Moreover, since the system will be web-based application, it has to demand minimal memory and reasonable processing power so that any user can access it with available resources. Our system should be functioning or operating in well man and also.
- **Quality Issues:** - For each action the user performs, the system must prompt to confirm the action that drop down to our system. Also, Reliability of the system will be good. The recovery methods will be simple and fast enough to process user wants.
- **Backup and Recovery:** - To make our project safer from different risks and attacks we use backup for our project files using External Hard Disk and Flash Disk, in case suddenly the computer that contain the file is damaged we can easily recover the data.
- **Physical Environment:** - The physical environment can affect the proposed system we are going to develop when a natural disaster occurs. On the other hand, external conditions such as weather condition will not have any effect on the performance of the system. To protect the system from attacks like viruses and worms the team members develop the backup (recovery) system to protect the data when the system damage.
- **Documentation: -** After the overall development process will be proposed we prepare the user manual template for our system.

5.2. Current System Architecture

The existing system of Absiniya lounge service system in WKU is not web based and hence there is no current software architecture that will be considered. As a result, we only describe the software architecture of the proposed system.

5.3. Proposed System Architecture

In this section we design the conceptual model of the Absiniya lounge service system. We use client/server architecture. The system uses browser to interact with web server using http. The client side server interface and the database server is where all information is stored and can be accessed by authorized parties. The project team selected three tier system architecture because Three-tier architecture allows any one of the three tiers to be upgraded or replaced independently. The reason why three-tier architecture is selected for this project is the following:

- It gives us the ability to update the technology stack of one tier, without affecting other areas of the application.
- It allows for different development teams to each work on their own areas of expertise. Today's developers are more likely to have deep competency in one area, like coding the front end of an application, instead of working on the full stack.
- **Scalability:** - You are able to scale the application in and out. A separate backed tier, for example, allows you to deploy to a variety of databases instead of being locked into one particular technology. It also allows you to scale up by adding multiple web servers.
- **Reliability: -** It adds reliability and more independence of the underlying servers or services.
- **Maintainability**: - It provides an ease of maintenance of the code base, managing presentation code and business logic separately. So that a change to business logic does not affect the presentation layer.

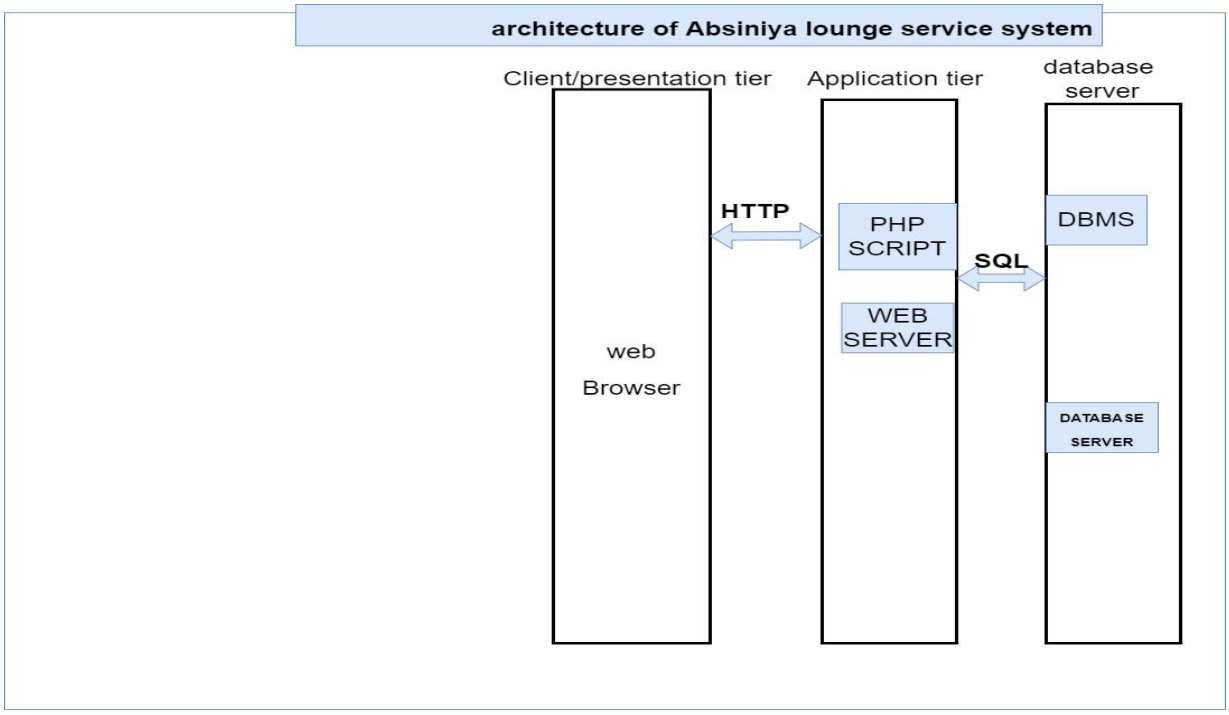


Figure 5. 1 Architecture for web application of Absiniya lounge service system

5.3.1. Subsystem Decomposition and Description

Subsystem decomposition is the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program, and maintain.

1. Registration management subsystem: In this subsystem managing of information regard to register and perform the following task:

- ❖ Employee registration
- ❖ Customer registration
- ❖ Food registration

2. Notification management subsystem: In this subsystem managing of information with regard to new things that post from the user is performed.

- ❖ View notification.
- ❖ Check payment.
- ❖ View comment.
- ❖ View report.
- ❖ View food menu.
- ❖ View order history.

3. Data manipulation management subsystem: In this subsystem managing of information with regard to data stored before is performed.

- ❖ Search record.
- ❖ Update price
- ❖ Update food.
- ❖ Cancel payment.
- ❖ Cancel menu item.
- ❖ Change reservation.
- ❖ Update account.
- ❖ Delete account.

4. Registration account management subsystem: In this subsystem all record will be registered which are:

- ❖ Create account.
- ❖ Active user.
- ❖ Inactive user.

5. Admin subsystem: In this subsystem all the record information are generate to the customer such as:

- ❖ Give comment.

- ❖ Sent notification
- ❖ Order
- ❖ Make payment.

5.3.1.2. Component Diagram

In this Diagram components of the system will be wired showing that there is relation among components, management of the system, database and operations performed on databases some security issue. This in some extent shows which component or objects will be accessed by whom and what type of security infrastructures it is using. The diagram is draw as the following:

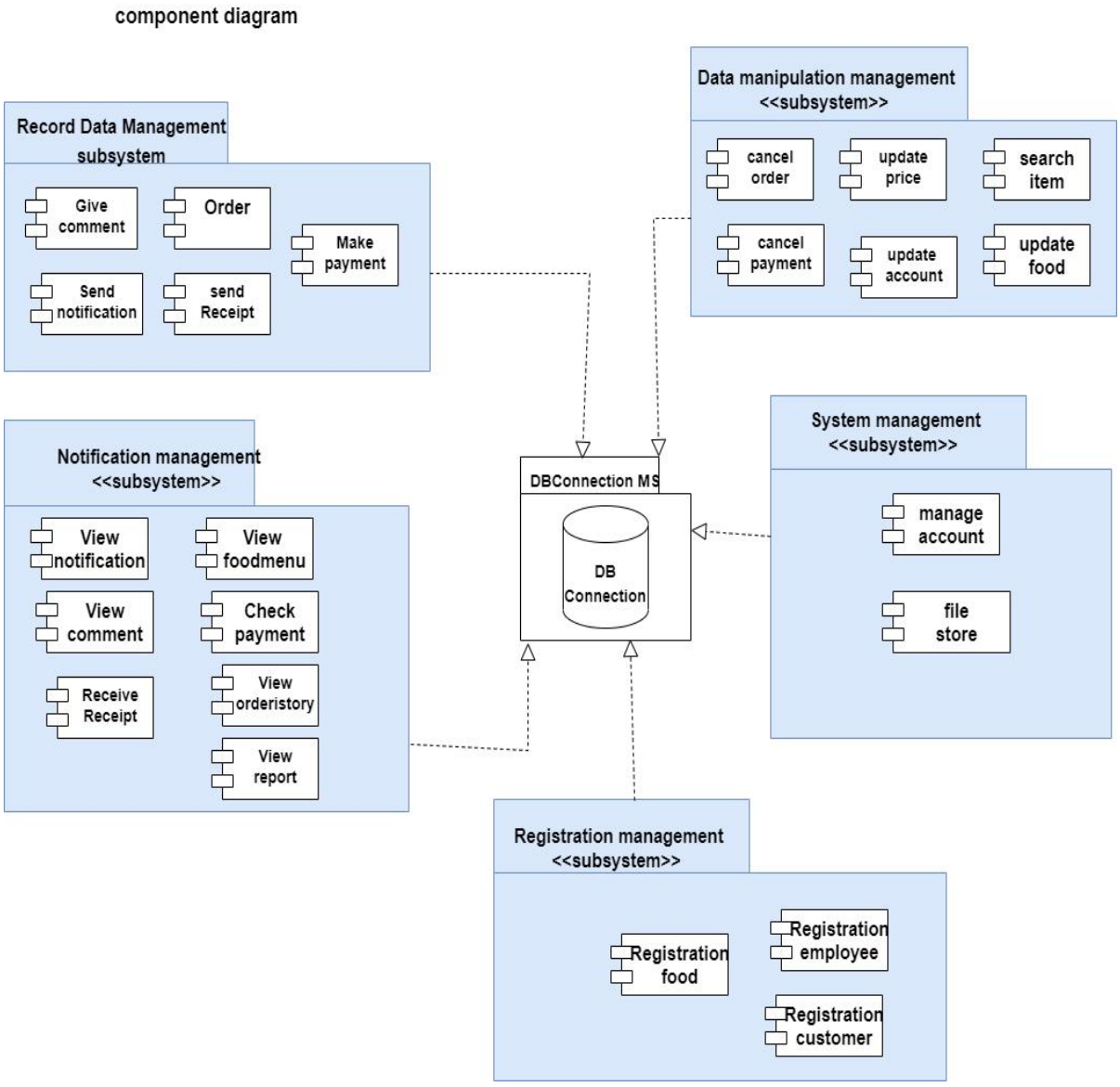


Figure 5. 2 Component diagram for Absiniya lounge service system

5.3.2. Hardware/Software mapping

Deployment modeling is used to for the purpose of showing the hardware of the system, the software that is installed in the hardware and also the middleware that is used to connect the disparate machines to each other. It also shows the relationship the hard ware and software and how these components are work together in the machine. This diagram is given as follows:

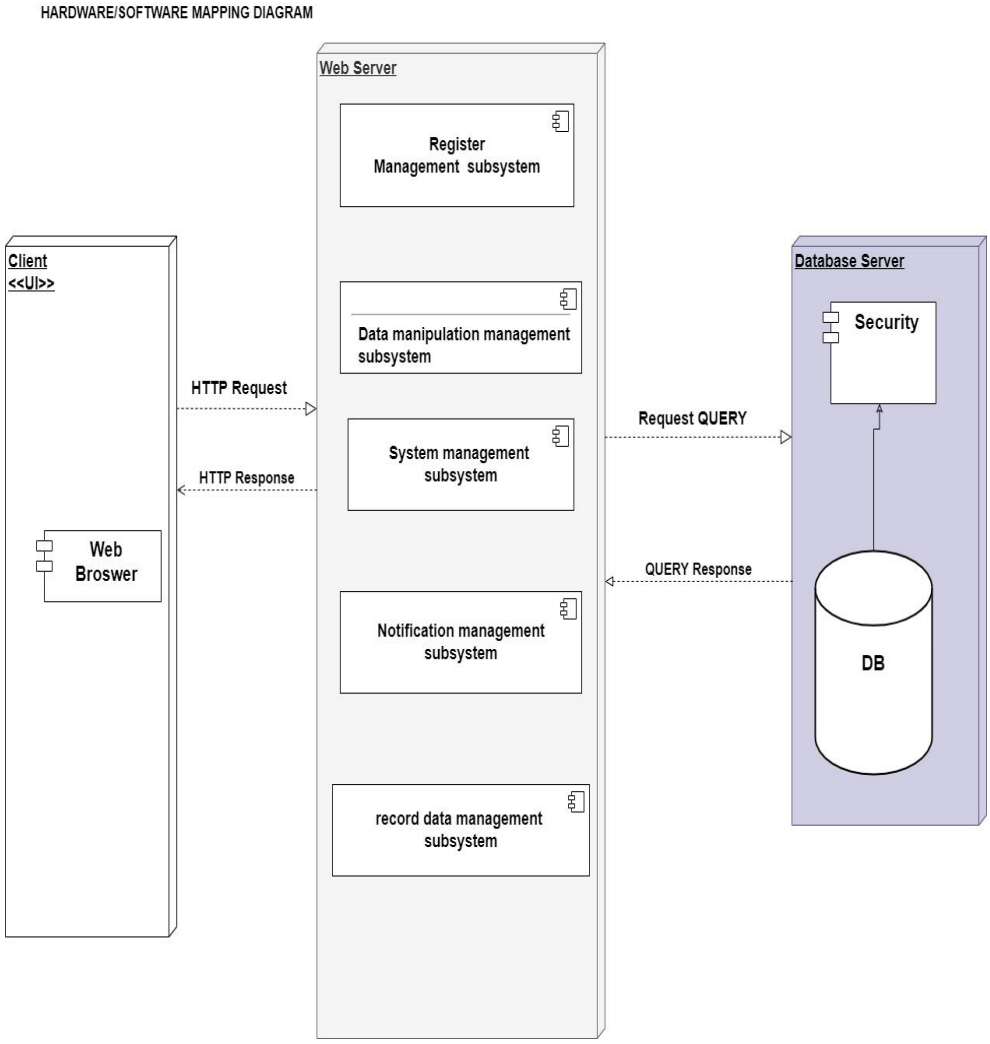


Figure 5. 3 Hardware/Software mapping diagram for Absiniya lounge service system

5.3.4. Detailed Class Diagram

Absiniya lounge class diagram

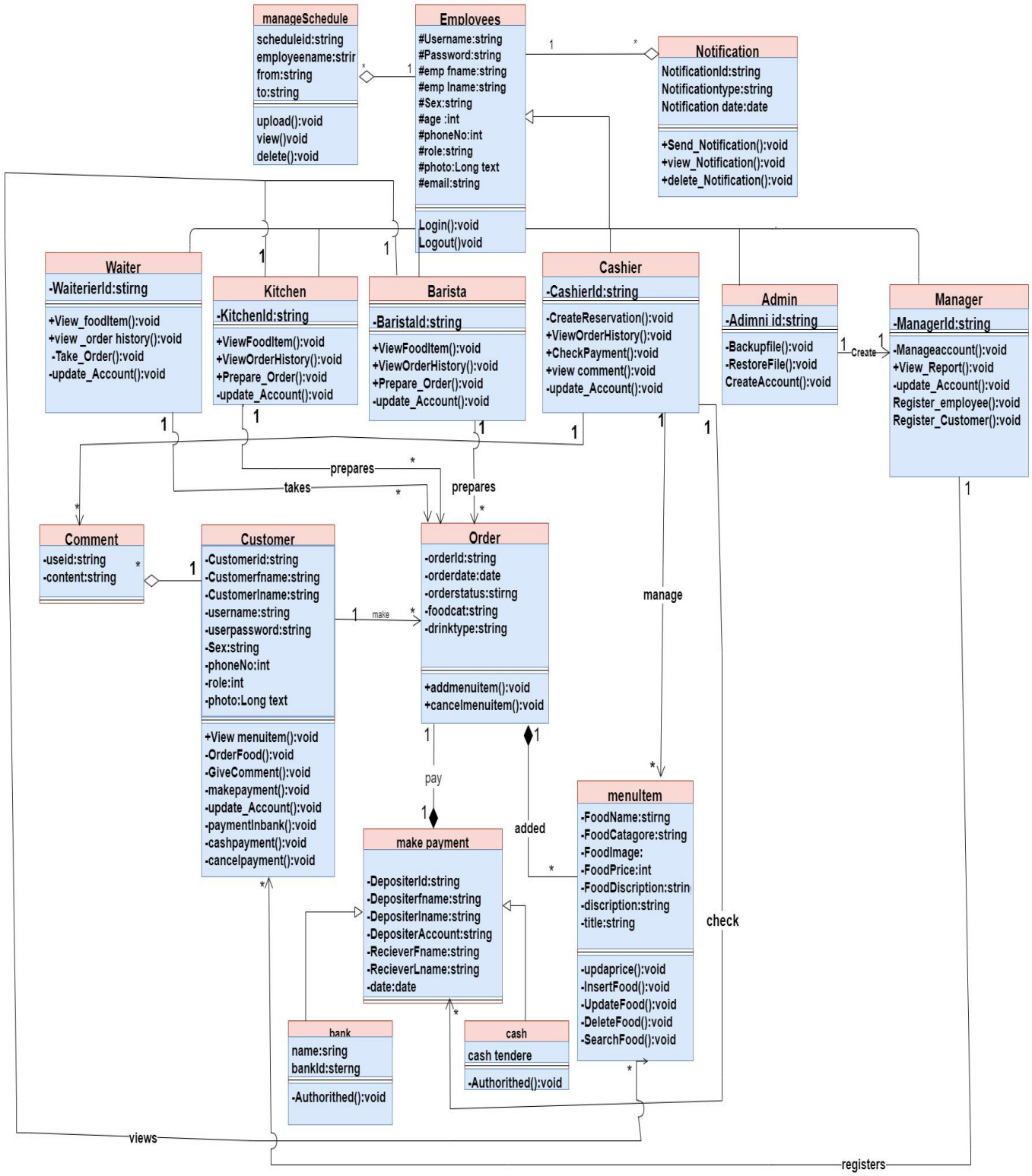


Figure 5. 4 Detailed class diagram

5.3.5. Persistent Data Management

In this we describe how the persistent data stored by the system and the data management infrastructure required for it. Those data must be stored permanently that the system get required information when it needs.

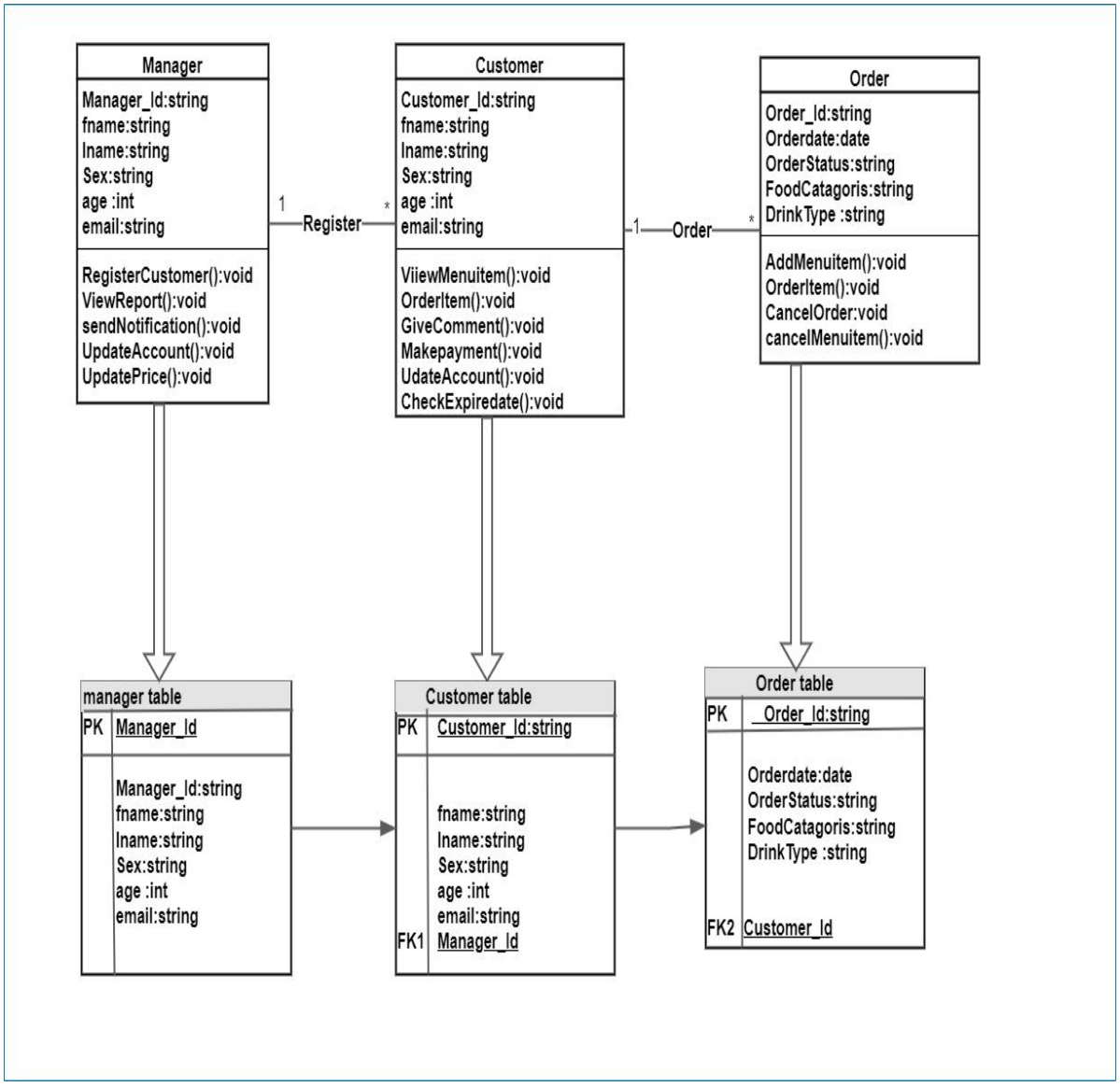


Figure 5. 5 Persistent data management system diagram

5.3.6. Access Control and Security

The Absiniya lounge service management system is used by many users. Authentication is the first security that protects our system from illegitimate users. So, in the system we have users like system administrator, Manager, Customers and Cashier. The user has different access right.

Table 5. 1 Access Control and Security

No	Activities	Actor names						
		Admin	Manager	Customers	Cashier	Waiter	Kitchen	barista
1	Login	✓	✓	✓	✓	✓	✓	✓
2	Create user account	✓						
3	Activate account	✓						
4	deactivate account	✓						
5	Restore database file	✓						
6	Register customers		✓					
7	View customer information		✓					
8	View order histories		✓	✓	✓	✓	✓	
9	View reports		✓		✓		✓	
10	View comment		✓					✓
11	Prepare order						✓	✓
12	Take order					✓		
13	View food menus			✓	✓	✓	✓	
14	Order food based on			✓				

	the menus							
15	Make payment			✓				
16	Send notification		✓					✓
17	View notification			✓	✓	✓	✓	✓
18	Check payment				✓			

5.4. Packages

In this section we design the package diagram of the system which depicts the dependencies between the packages that make up a model.

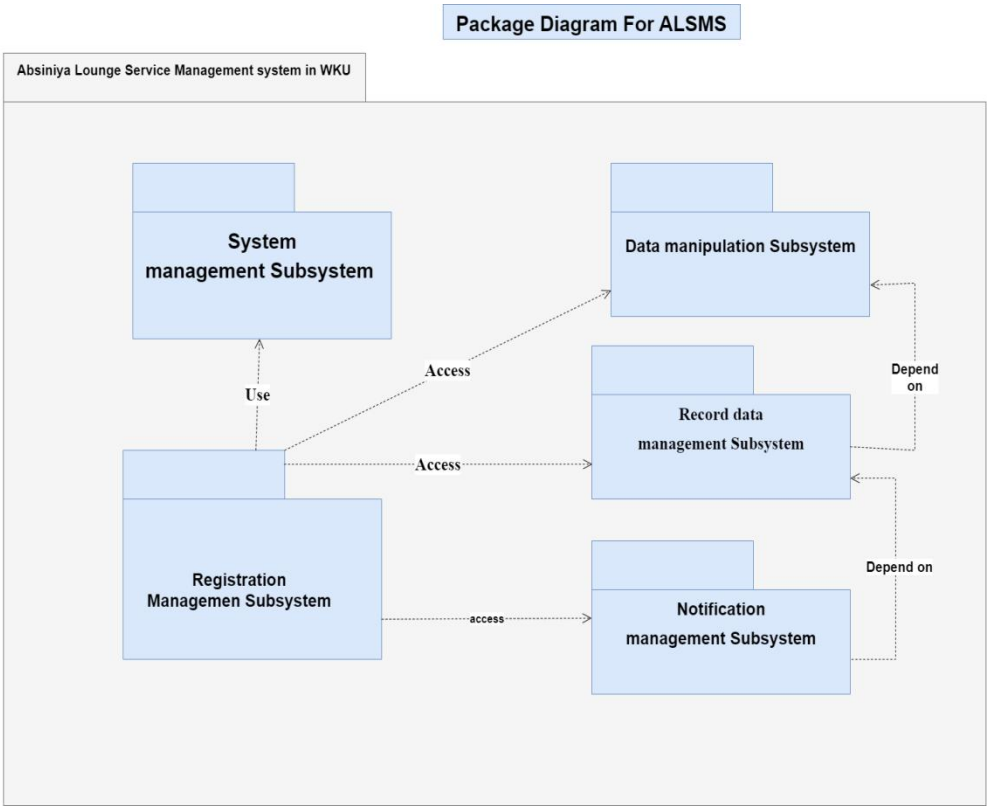


Figure 5. 6 Package diagram

5.5. Algorithm Design

Algorithm design is identified and incorporated into many solution theories of operation research, such as dynamic programming techniques for designing and implementation. This design part is important in the coding part of implementation. Some of the algorithms are listed below.

Algorithm for login

Begin
If user click login button
Login form
Fill form
If (valid)
Generate SQL select queries
Connect to database
Check query within database
If information in database
Display page
Else
Display error message
Else
Display error message (Ask user refill the form)
End if
End if

End if
End

Algorithm use for create user account

Begin
If admin click create user account
The system register form
Then admin fill the form
If (valid)
Send information to database
If information exist on database
Display the account already exist
Else
Display message account successful create
Else
Display error message
End if
End if
End if
End

5.6. User Interface Design

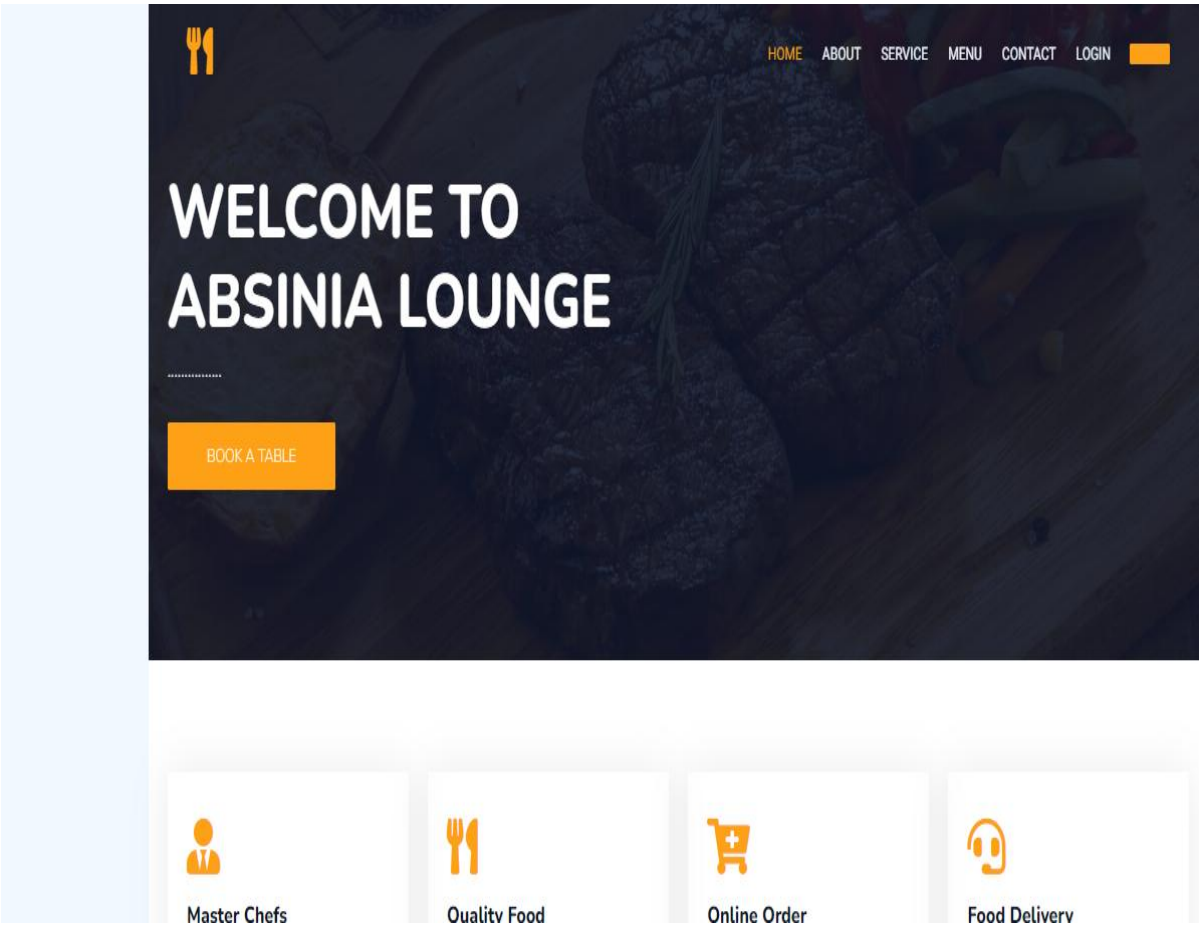


Figure 5. 7 User interface for Homepage

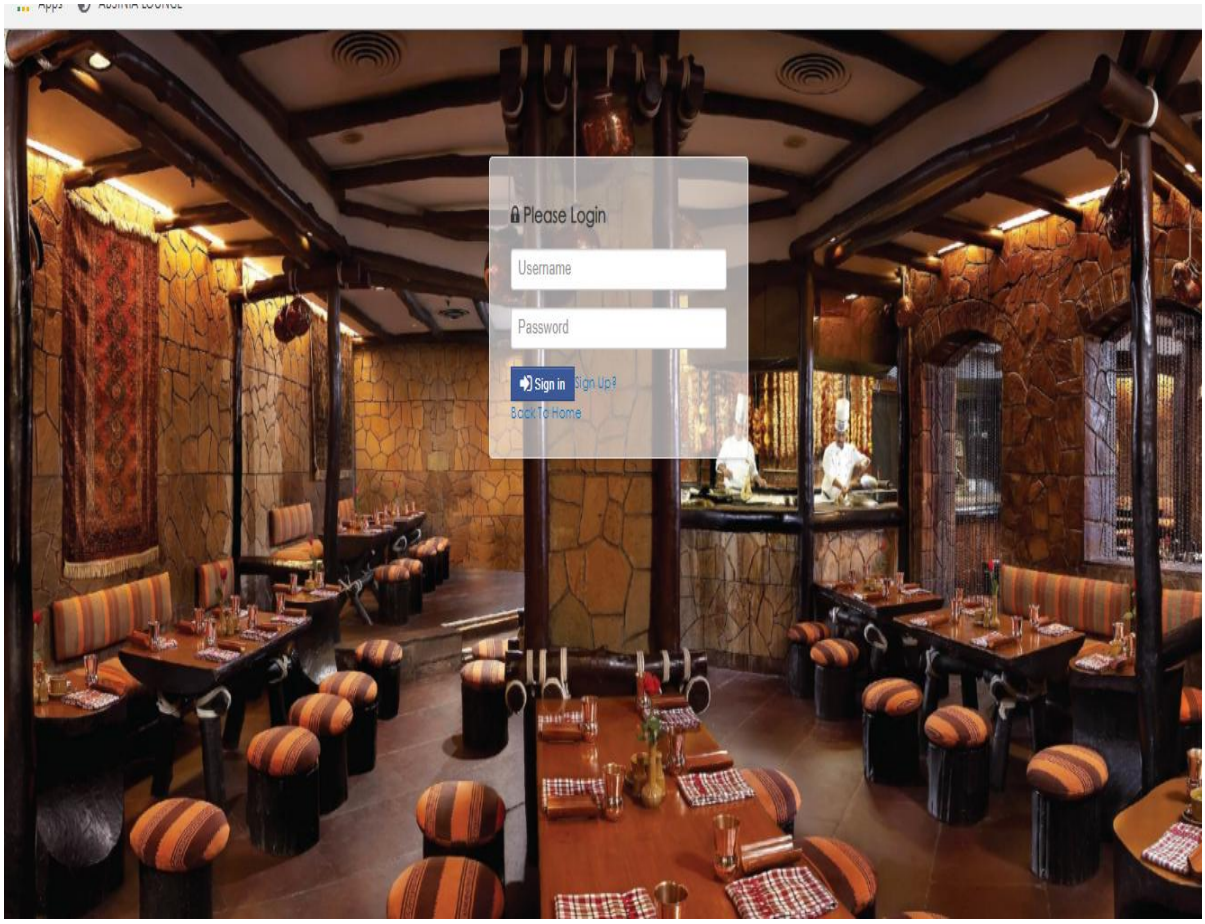


Figure 5. 8 User interface for login

CHAPTER SIX

6 . IMPLEMENTATION AND TESTING

6.1. Implementation of the Database

In this project MYSQL database management system used for the implementation of the database. Because MYSQL is an open-source object-relational system which applies SQL language and it is most popular database. It allows to store large and sophisticated data safely and help developers to build the most complex applications, run administrative tasks and create integral environments. It is more secure, cheap and scalable.

Using MYSQL database management system helped us to perform the following activities:

- We created all tables which were identified and shown as persistent model in the design document with their primary keys, foreign keys, check constraints and unique constraints.
- We configured a schedule of database backup aromatically in every week.
- we take appropriate level of security to our data base .

6.2. Implementation of the class diagram

The implementation of the system is based on the design of the class diagram and the business logic with the targeted programing language. The attributes are properly defined with their data types and access visibilities (modifiers). Methods are also defined appropriately with their return types, parameters, and data types including access visibility. The document of this project is well organized and documented based on the document design.

6.3. Configuration of the application server

In this project, XAMPP is used as an application server because it is a free, open-source framework and for the purpose of developmental speed and performance as well as for security purposes. It offers interactive shortcuts for full integration with the system databases.

- Folders and files are properly organized.
- The application server properly starts and shutdowns.
- Its configuration enables the remote access of the system.

6.4. Configuration of application security

The configuration of application security server is well handled for the security purposes. Security issues are handled properly.

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

6.5. Implementation of user interface

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

- It reduces memory loads of the users.
- It is consistent and interactive.

- It is easy and clear to use.

6.6. Testing the system

The functionality of the developed system must be tested because it has to provide its intended goal or functions. The following features are tested:-

- Input and output function
- Subsystem communication
- Database and application server communication
- Databases and user interface interactions
- Validations and authentication
- Security mechanisms also tested in this system.

6.6.1 test case

Sample Test cases

Registration Form having following fields:

- First Name
- Last Name
- Username
- Password
- Phone Number
- Gender

Required fields are: All fields.

6. 1 test case to create account

Test Case1: Register user account	
Test case objective : To create account	
Test case description: admin enters basic info of the user, then presses create button. Client program contacts with server, server contacts with the database, and database checks for create and sends message to the administrator.	
Requirements Verified: Yes	
Test Environment: Apache MySQL server must be in running state, Database Should contain appropriate table and link must be established between server and client program.	
Test Environment: Apache MySQL server must be in running state, Database Should contain appropriate table and link must be established between server and client program.	
Actions	Expected Results
The administrator should login and press the create menu.	Displays registration page

Administrator should enter user information	Displays success message of registration.
If some fields are not filled the system display to fill the fields again.	

6.1.2 Unit Testing

Unit test is a way of testing each of the system functionality independently. Accordingly, we have tested each one of the major activities and the rest accompanying activities independently using different user input, different login mechanisms and any technique of fault finding so that an incorrect functioning of the activities was corrected at the right time.

6.1.3 Integration Testing

In an integration testing we integrate and test all components of the system modules that a program comprises for testing purpose. Tasks that we have performed under Integration testing.

- Prepare integration test plans.
- Identify integration test objectives.
- Identify integration test acceptance criteria.
- Perform the tests according to our plan.
- Document the test results.

6.1.4 System Testing

In this systemwh have tested the entire system as a whole with all forms, code, modules. In this we tested all the functionalities in the System. All errors in the forms, functions, modules have been tested. Finally, we ensured that the entire integrated software system meets the desired requirements. It tests a configuration to ensure known and predictable results. We

apply System testing to insure the functional and non-functional requirements correctness, performance, accuracy, function validations, and security have been met correctly.

6.1.5 Acceptance Testing

Acceptance testing is a level of the software testing where a system is tested for acceptability. The system tested with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. Similar to a unit test, an acceptance test generally has a binary result, pass or fail. A failure suggests, though does not prove, the presence of a defect in the product.

CHAPTER SEVEN

7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

As a system come up to the stage of completion, it's time to literate the work has been done and check against the developed system success indicator which was set at the beginning of the system. It's the moment to determine whether the attempt is successful as per the predetermined objectives. The project begins with Identifying, defining the problem and adopting an appropriate system development methodology, identifying resource requirements, and setting project schedule. In analysis, we tried to model the new and proposed system using UML diagrams: use case diagrams, sequence diagrams, class diagrams. The new system might look and feel with greater comfort and desired objectivity. In design phase, we extended work analysis with more models. The class type architecture, class diagrams in analysis were extended in design to step closer to the implementation. In general; during the various phases of the project, we have learnt a lot in multidisciplinary technologies required to be embed with this system and these technologies would be useful in our future techno-economic career. We have applied the best efforts towards the achievement of our desired objectives of the proposed system. We tried our best to design a better architecture of the proposed web based lounge service system for Absiniya lounge in Wolkite university. We conclude that we have almost achieved objectives for the promising solutions for problems to Absiniya lounge.

Generally: - This web based lounge service system for Absiniya lounge in Wolkite university performs or cover some limitation of manual based system like order, paymnet, report, and other like searching food data and updating data are previously performs by manually but, Our system Performs this task based on computerized with effective and fast way.

7.2 Recommendation

According to scope of our project the team develops web based application. Because of the time constraint we may have limitation which should be consideration in, but in the feature the team believes that this system should be fully operationally by adding some functionality that are not included in the system.We also want to recommend this project can be expanded and more automated with additional functionalities by integrating with many new technologies. If any organization, group or individual is interested in re-making and improving this web based lounge service system project we as a group recommend them to include the following features need to be incorporated in any further revision and extension attempt.

Generate financial analysis periodically.

References

A.Hoffer, J. (2000). *modern system analysis and design six edition.* Belem: A. Jeffrey.

bass, B. (1999). *softeware Architecture in practice.* weley: Addision.

G. Booch. (1994). *object oreinted analysis and design with application.* redwood city: benjaima/cumming.

Lonnie.D.Bantlely. (2004). *system analysis and design for global enterprise.* Bejing: Lonnie.