Online Food Ordering System For Huts

Thesis submitted to the Institute of Information Technology, Quaid-I-Azam University, Islamabad, for the partial fulfillment of the degree of Master of Science in Information Technology.

By

Nayab Aziz

Supervised By

Madam Robina Rashid

Institute Of Information Technology

Quaid I Azam University, Islamabad

Session 2019-2021

ACKNOWLEDGEMENT

All praise to Almighty Allah for giving me the courage to complete this project within the specified time. I express my gratitude to my kind supervisor Miss Robina Rashid who kept my morale high by her appreciation and motivation and guided and taught me different techniques. She was always available whenever I consult her. Her valuable suggestion significantly improved this work and without her precious guidance I would never be able to complete such a project. I am extremely fortunate to have her as supervisor.

I would like to acknowledge the support of my family member, I owe all my achievements to my truly, sincere and most loving parents, Brother, and Sisters. They mean a lot to me.

I am also thankful to all the worthy teachers and staff members of Institute of information technology, Quaid-I-Azam University Islamabad. At the end I would like to extend thanks to all my friends, class fellows who have helped me in this project.

Thank You So much!

Nayab Aziz

DEDICATION

I would like to dedicate this thesis to my Parents, my respected teacher **Miss Robina Rashid** and my Friends due to their backing and courage; I’m able to complete this perplexing task.

DECLARATION

I Nayab Aziz Reg No. 01161911005 Student of Master of Information Technology; Department of Information Technology, Quaid-I-Azam university Islamabad, solemnly declare that the data quoted in this project title "Online Food Ordering System for Huts" is based on my original work and has not yet been submitted or published elsewhere.

Nayab Aziz - 01161911005 ________________________

I certify that the mentioned above Student of Master of Information Technology; Department of Information Technology, Quaid-I-Azam university Islamabad, worked under my supervision and the above stated declaration is true to the best of my knowledge.

Madam Robina Rashid ________________________

Department of Information
Technology, Quaid-I-Azam
University Islamabad.

Abstract

The Food ordering System in this document has been designed to provide online food ordering facility to Quaid-I-Azam University students and Faculty members. So, that they can order food from their favorite huts(located within university region) in their respective departments.

The system, which is highly customizable, allow the huts owner/manager to easily manage the menu, themselves through a very intuitive graphical interface. The website, which is the only component seen by the huts customers, is then built dynamically based on the current state of the system, So any changes made are reflected in real time. Visitors, to the site ,once registered ,are then able to navigate menu, add one or more food item to their shopping cart with just few clicks, greatly simplifying the ordering process. Back in the Huts Manager dashboard, placed orders are promptly retrieved and displayed in an easily readable format for efficient processing.

The purpose of this document is to provide in-depth descriptions of design and implementation details of the system, as well as description of all available functionality and plans for evolutions.

TABLE OF CONTENTS

Acknowledgement I

Abstract I

Chapter 1: Introduction 1-5

1.1 Introduction 2

1.2 Background And History 2-3

1.3 Objective 3

1.4 Scope 3

1.5 Modules 3-4

Admin 4

Hut Managers 4

Cart 4

Registered Users 4

Guest Users 4

1.6 Limitations Current Manual System 4

1.7 Advantages Of Online System Over Physical System 4-5

1.8 Limitations Of Proposed System 5

Chapter 2: Requirements 6-10

2.1 Requirement 7

2.1.1 Requirement Elicitation 7

2.1.2 Requirement Analysis 7

2.1.3 Requirement Specification 7

2.1.4 Requirement Validation 7-8

2.2 Functional Requirements 8-9
2.2.1 Users 8

2.2.2 User Registration 8

2.2.3 Add.. 8

2.2.4 Update 8

2.2.5 Delete 8

2.2.7 Login 9

2.2.8 Logout 9

2.3 Non-Functional Requirements 9

2.3.1 Reliability 9

2.3.2 Implementation 9

2.3.3 Availability...9

2.3.4 Portability 9

2.3.5 Usability..9

2.3.6 Security 9

2.4 Interface Requirements.. 10

2.4.1 Hardware Interface 10

2.4.2 Software Interface 10

Chapter3: System Analysis And Design.. 11-33

3.1 Introduction 12

3.1.1 Analysis 12

3.1.2 Design 12

3.2 Component Overview 15

3.3 UML Diagrams.. 15

3.3.1 Behavioral Diagrams 15

3.3.1.1 Use Case Diagram 15-25

3.3.1.2 Activity Diagram 26-27

3.3.1.3 Sequence Diagram 28-29

3.3.2 Structural Diagrams 30

3.3.2.1 Class Diagram 30-31

3.4 Relational Database Model 32-33

3.4.1 Relational Model Concepts 33

Chapter 4: Relation Schema..34-35

4.1 Introduction 35

Chapter 5: Software Tools and Technologies 36-40

5.1 Introduction 37

5.2 Tools And Technologies 37

5.2.1 Microsoft Visual Studio Code 37-38

5.2.2 WAMP...38

5.2.3 Microsoft Visio...38

5.2.4 PHP...38-39

5.2.5 HTML...39

5.2.6 MySQL 39

5.2.7 JavaScript 39

5.2.8 Bootstrap 40

5.2.9 CSS...40

Chapter 6: User Interface...41-46

6.1 Introduction 42

Chapter 7: Testing..47-55

7.1 Introduction 48

7.2 Testing 48

7.2.1 Software Testing Levels 48

7.2.1.1 Module Testing...49

7.2.1.2 Unit Testing.. 49

7.2.1.3 Subsystem Testing.. 49

7.2.1.4 System Testing 49

7.2.1.5 Acceptance Testing 49

7.3 Software Testing Types 49

7.3.1 Manual Testing.. 49

7.3.2 Automation Testing 49

7.4 Testing Approaches 49

7.4.1 White-box Testing 50

7.4.2 Black-box Testing 50

7.5 Evaluation.. 55

7.5.1 Purposes.. 55

Chapter 8: Conclusion and Future Enhancement... 56-58

8.1 Problem during coding 57

8.2 Summary 57

8.3 Enhancement 57-58

8.3.1 Evaluation of the system 58

8.3.2 Future Enhancements 58

References 59

.

Project in brief

Project Title:	Online Food Ordering System For Huts
System Used:	Intel(R) Core(TM) i3
Operating System:	Microsoft windows 10
Development Tool:	Visual Studio Code 2019
Language:	PHP, HTML, CSS and JavaScript
Start Date:	20th September 2020
End Date:	30th May 2021

Chapter #01

Introduction To Online Food Ordering System For Huts

1.1 Introduction

The purpose of this project is to design a website for students and faculty members of Quaid-I-Azam university where they can easily order food from their university huts in their respective departments. Quaid-I-Azam University is on 1700 acres(or 6.km^2) campus which is very large in area, So one department is far away from another department and also the university's huts are far away for student of many departments. So, if a student wants to eat food from his/her favorite huts which is far away from his/her department then either he/she has to get a taxi or walk for 15 to 20 mins to reach his/her destination and if the weather is hot or rainy so it will become very difficult for him/her to walk for 15-20 mins to eat food from his/her favorite huts.

In this technological world, everyone prefers technical system over manual system. So, one of the main advantages of this project is that the customers(either students or faculty members) can order food easily while sitting in their departments without spending extra money on taxi nor walking for 15-20mins to get food from their favorite huts.

The system also greatly lightens the load on hut's end, as the entire process of taking orders is automated. So, the key benefit of this system is that it simplifies the process of food ordering for both the customers(students and faculty members) and the hut's owners.

1.2 Background and History

The concept of food ordering system has been dated back to as old as Ancient Rome in the early 753 B.C. – 476 A.D. Ancient Romans loved the concept of convenient meals as much as we do in the modern era, and they often ordered their fast food restaurants Thermopolium. They served meals at a counter out of large clay pots that stored and kept the food hot, similarly to how we serve fast food today.

The concept of food ordering system became extremely during the 1950s. Due to the fact that most families would prefer going home for dinner and that almost every family had a television at home. Most families would prefer to eat their meals while watching their favorite television shows, this results in a drastic decrease of customers in restaurants. Fast food restaurants boosted their business by putting up their advertisements on television, this introduced the concept of ordering fast food from home and fast food delivery service.

The first online food order was a pizza from Pizza Hut in 1994. The online food ordering market has increased in the U.S with 40 percent of U.S adults having ordered their food online once. The online food ordering market includes foods prepared by restaurants, prepared by independent people, and groceries being ordered online and then picked up or delivered.

The first online food ordering service, Worldwide Waiter(not known as waiter.com), was founded in 1995. The site originally serviced only northern California, later expanding to several additional cities in the United States.

By the late 2000s, major pizza chains had created their own mobile applications and started doing 20-30 percent of their business online. With increased smartphone penetration, and the growth of both Uber and the sharing economy, food delivery startups started to receive more attention.

1.3 Objective

The main purpose of designing this website is to provide platform to the students and faculty members of Quaid-I-Azam university so that they can order food online from their favorite university's huts while sitting in their departments.

1.4 Scope

By using this system, the students and faculty members of Quaid I Azam university can easily order food online without spending extra money on taxi nor wasting time by walking 15-20 mins to get food from their favorite huts in their specified delivery address.

The system also helps huts managers to develop healthy customer relationships by providing good services. The system enables huts managers to update and make changes to their food and beverage list information based on the orders placed and the orders completed.

1.5 Modules

A module is a separate unit of any software or hardware who can manage, control and use the system. Modules of the proposed project are:

- **Admin:** Admin module is for the admin of the website. In this module admin can add huts, users. He login from the backend and manage database and view orders.

- **Hut Managers:** This module is for the huts owner of the website. In this module each huts manager can add menu, delete menu and manage orders.

- **Cart:** The cart module is accessed by the registered users of this website. Via this module, the registered users can add their food item and proceed to checkout. The food items that the user add in the cart will remain in the cart until the user logout from the system.

- **Visitors/Guest Users:** The guest user module is that module which is accessed by the visitor of this website. Via this module, a user can see the huts, view their menu, contact to the admin of the website. This module is only for the visitors.

- **Registered Users:** Registered users can login into the system by providing login credentials and can view menu, add items in their cart and see all the details in their shopping cart before checking out.

1.6 Limitations Current Manual System

Some limitations/disadvantages of current manual systems are:

- The cost of having a kitchen and serving area. In addition to the space and equipment costs, there are additional utility costs.
- There is more trash created with a manual food services and thud the cost for removal is increased.
- If the all the tables/benches are full then the customer doesn’t get a clean place to eat.
- More labor is required to operate manual food service, but if a food service contractor is chosen this becomes their issues.

1.7 Advantages of Online System over Physical System

- Makes the ordering process easier.

- You can monitor your expenses incurred in real-time.

- You can easily know your regular customers are by using analytics and insights provided by a robust online ordering system.

- You don’t need to go outside, and you can easily order your food which will save time and energy and extra money which will use in travelling.

- You can order food any time you want as it mostly provide services 24/7.

- Online system provides more accuracy in food orders.

1.8 Limitations of Proposed System

They are also few limitations of the current proposed system, these are:

- Require data/Wi-Fi to get online.

- While using internet there is a chance that virus and worms may affect our system badly.

- It is good technique to provide online food ordering facility but there is a chance that the user’s data may be hacked by someone.

- Error stays online FOREVER.

- The system that we are creating is accessed only by those people who are currently part of Quaid I Azam University.

Chapter #02

Requirements For Online Food Ordering System

2.1 Requirements

A requirement is a specification or want that must be met or satisfied within a certain timeframe. The requirements for a system are the descriptions of the services that a system should provide; requirements reflect the needs of customers for a system that serves a certain purpose.
Requirements are dividing into different categories.

2.1.1 Requirements Elicitation

Before requirements can be analyzed, modeled, or specified they must be gathered through an elicitation process. Requirements elicitation is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders. The practice is also sometimes referred to as "requirement gathering". Requirements elicitation involves meeting with stakeholders of different kinds to discover information about the proposed system.

2.1.2 Requirements Analysis

In requirements analysis we determine the needs or conditions to meet the new or altered product or project. The requirements should be documented, actionable, measurable, testable, related to identified needs or opportunities.

2.1.3 Requirements Specifications

Requirements specification is the process of writing down the user and system requirements in a requirements document. The user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent. The user requirements for a system should describe the functional and nonfunctional requirements so that they are understandable by system users who don’t have detailed technical knowledge

2.1.4 Requirements Validations

Requirements validation is the process of checking that requirements define the system that the customer really wants. It overlaps with elicitation and analysis, as it is concerned with finding problems with the requirements.

During the requirements validation process, different types of checks should be carried out on the requirements in the requirements document. These checks include:

- Validity checks
- Consistency checks
- Completeness checks
- Realism check
- Verifiability

2.2 Functional Requirements

Functional requirements are features or functions that developer must implement to enable the users to accomplish their tasks. Functional requirements describe what a system should do.

2.2.1 Users

They are the unregistered users called visitors who can visit the website and can view huts, their menu and contact to the admin.

2.2.2 User Registration

Registration is allowed to the users who are not registered yet(unregistered users) and after completion of this function they can view menu and order food items from different huts.

2.2.3 Add

Admin can add huts, users.

Huts manager can add menu and manage orders.

Registered users can add food item in their shopping cart and proceed to checkout.

2.2.4 Delete

Admin can delete huts, registered users.

Huts manager can delete menu.

Registered users can delete/remove food items from their shopping cart before checkout.

2.2.5 Update

Admin can update or change huts and users information .

Huts manager can change/update menu.

2.2.6 Login

By providing login details admin/huts owner/registered users can login into the system according to their privileges.

2.2.7 Logout

After completing his/her task he/she will be logout from the system.

2.3 Non-Functional Requirements

Non-functional requirements elaborate performance characteristics of a system. They are also known as quality attributes.

2.3.1 Reliability

The system should provide storage of all databases with automatic switchover. All information should be reach to admin without an error.

2.3.2 Implementation

Implementation of the system using different languages like HTML, CSS, JavaScript and PHP is used for database connectivity. Database is a part of MySQL.

2.3.3 Availability

The system will be available according to proposed schedule so the user can access it easily.

2.3.4 Portability

The end user is fully portable. The user can use this system on any OS using any web browser. The system will be able to run on nay device.

2.3.5 Usability

The website is designed for user friendly environment and ease of use.

2.3.6 Security

The system's confidential information is accessed and managed only by authenticated administrators on server side.

2.4 Interface Requirements

It specifies hardware, software, or database elements with which a system or component must interface**.**

2.4.1 Hardware Interface

All hardware's that are required for system to connect internet will be hardware interface like MODEM, WAN and LAN.

Table 2.1: Hardware Interface

System	32/64-bit operating system
RAM	4GB or above
Processor	1.9GHz or higher
Internet	4Mbps or above

2.4.2 Software Interface

Software interfaces are the languages, codes and messages that programs use to communicate with each other and to the hardware.

Table 2.2: Software Interface

Operating System	Microsoft Windows 7/Windows 8/Windows 10
Environment	Any web browser like chrome, Mozilla, internet Explorer etc.
Web Server	WAMP
Internet	4Mbps or above
Coding Platform	Microsoft Visual Studio Code
Diagramming Tool	Microsoft Visio

Chapter #03

System Analysis And Design

3.1 Introduction

Systems development is a systematic process deal with understanding and specifying in detail what system should do and how the components of the system work together. It includes planning, analysis, design, deployment, and maintenance. Here, I will primarily focus on:

- Analysis
- Design

3.1.1 Analysis

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. Analysis specifies **what the system should do.**

It is conducted to study a system or its parts in order to identify its objectives. It is a problem-solving technique which is used to improve the system and ensures that all the components of the system work efficiently to accomplish their purpose.

3.1.2 Design

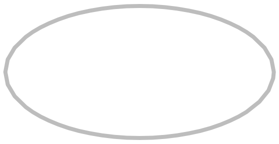
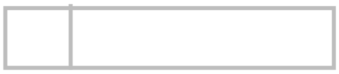
A design is a plan or specification for the construction of an object or system or specification in the form of a prototype, product or process. System Design focuses on how to accomplish the objective of the system.

It is used to create a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements.

Data Flow Diagram

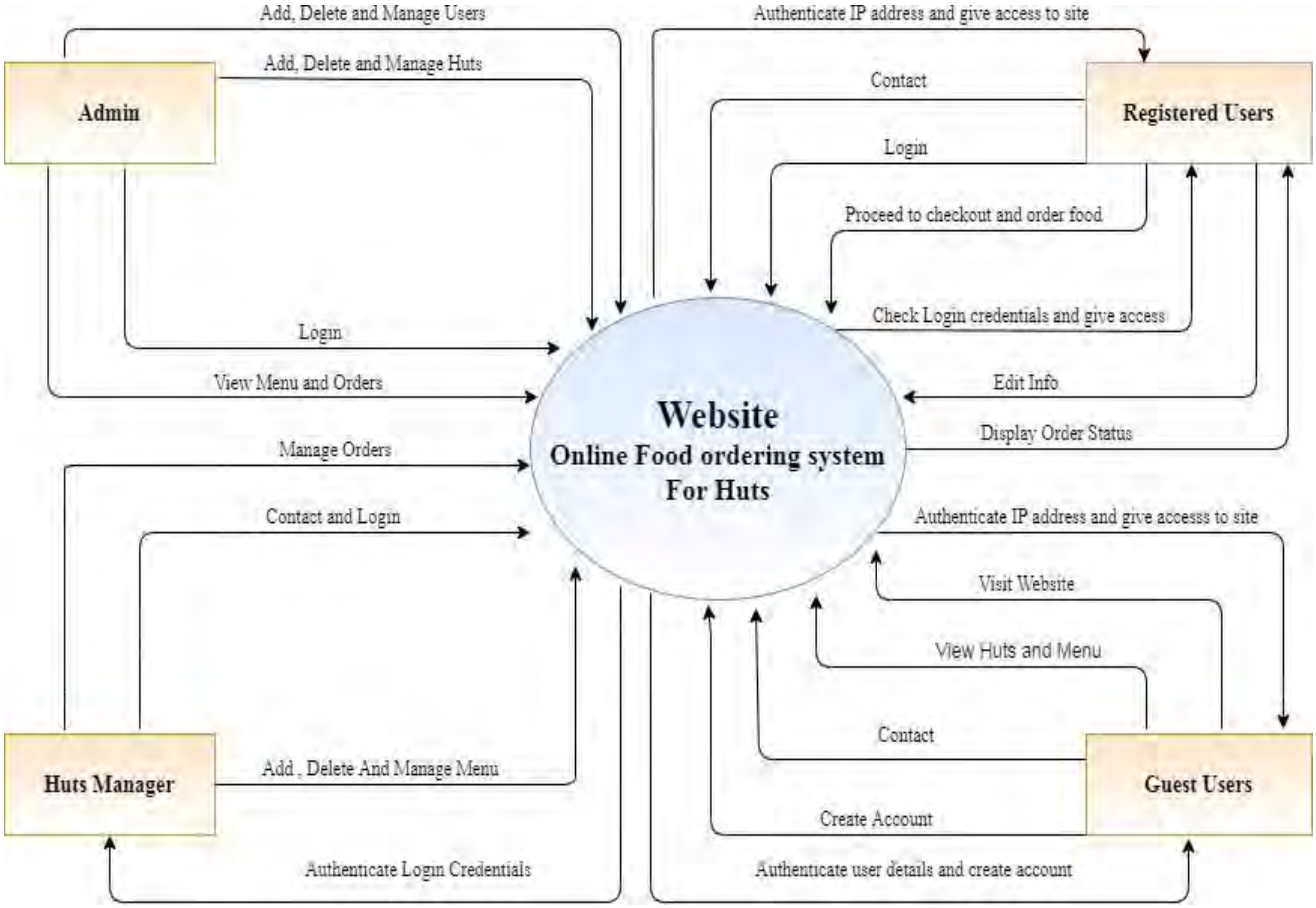
Data Flow diagram is a graphical representation of flow of data throughout the information system. Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs.

Table 3.1 Data Flow Diagram

Name & Role	Notation
Process Transform incoming dataflow to output dataflow.	
Data Flow They are pipeline through which packet of information flow	

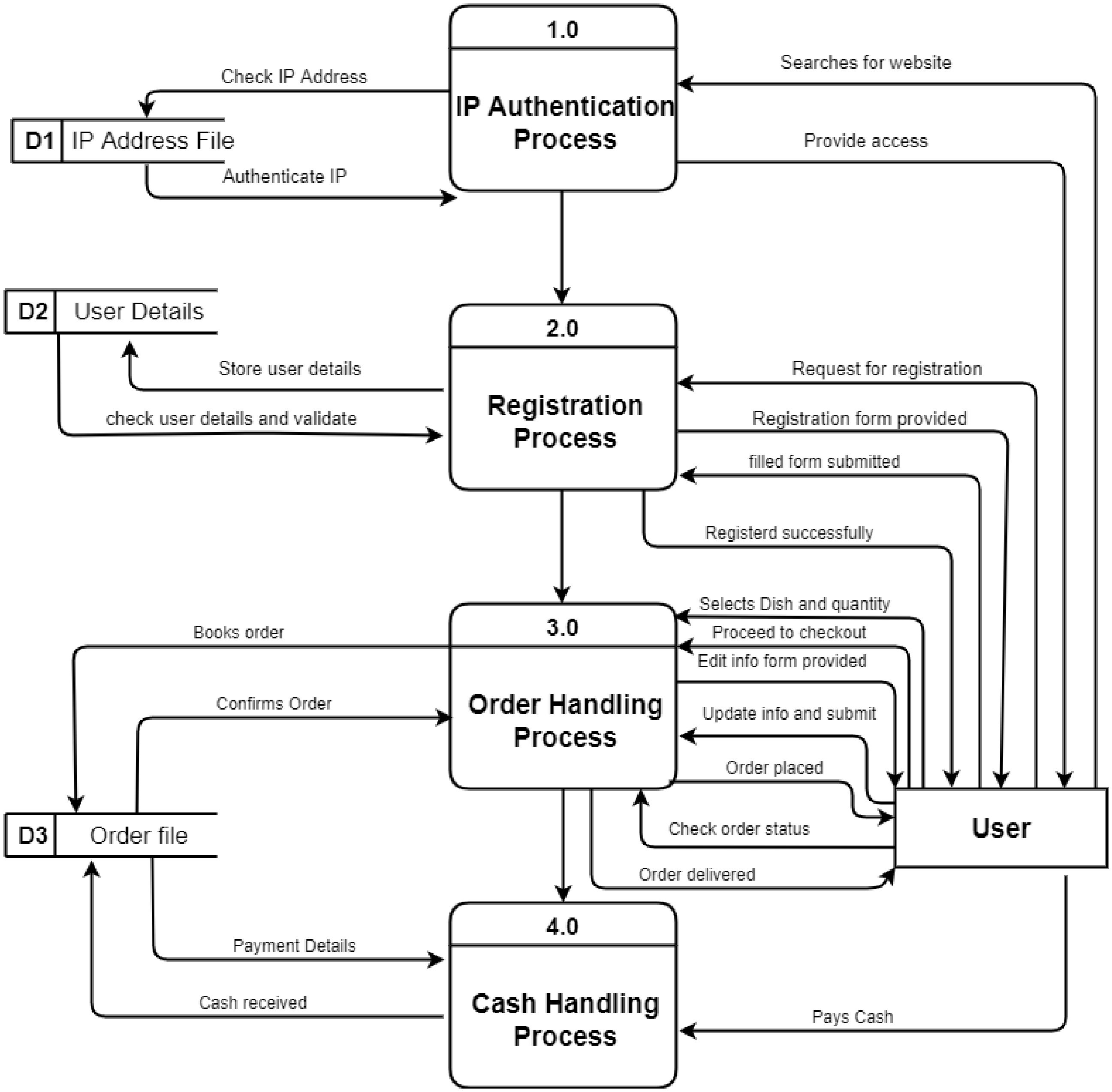
Datastore Repositories of data into the system.	→
External Entities They are object outside the system with which the system communicates	▭

“Context level Data flow Diagram of Food Ordering System for Huts”



Admin
Add, Delete and Manage Users
Add, Delete and Manage Huts
Login
View Menu and Orders
Huts Manager
Manage Orders
Contact and Login
Add , Delete And Manage Menu
Authenticate Login Credentials
Website
Online Food ordering system
For Huts
Registered Users
Authenticate IP address and give access to site
Contact
Login
Proceed to checkout and order food
Check Login credentials and give access
Edit Info
Display Order Status
Guest Users
Authenticate IP address and give accesss to site
Visit Website
View Huts and Menu
Contact
Create Account
Authenticate user details and create account

“Level 1 Data flow Diagram of Food Ordering System for Huts”



3.2 Component Overview

It is an online site for students and faculty members of Quaid I Azam University where they can easily order food from different university's huts in order to save their time and money. The main functionality of the system is described in following UML diagrams.

3.3 UML Diagram

A UML diagram is a diagram based on the UML (**Unified Modeling Language)** with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

There are two types of UML diagrams:

- Behavioral Diagrams
- Structural Diagrams

3.3.1 Behavioral Diagrams

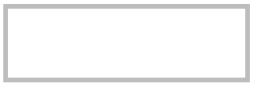
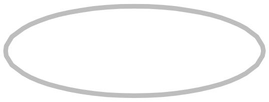
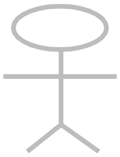
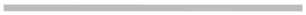
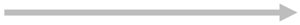
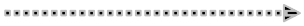
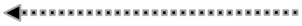
It shows how the system behaves and interacts with itself and other entities (users, other systems). They show how data moves through the system and how objects communicate with each other.

3.3.1.1 Use Case Diagram

Use cases are a set of actions, services, and functions that the system needs to perform. Use case diagrams model the functionality of a system using actors and use cases. A "system" is something being developed or operated, and “actors" are people or entities operating under defined roles within the system.

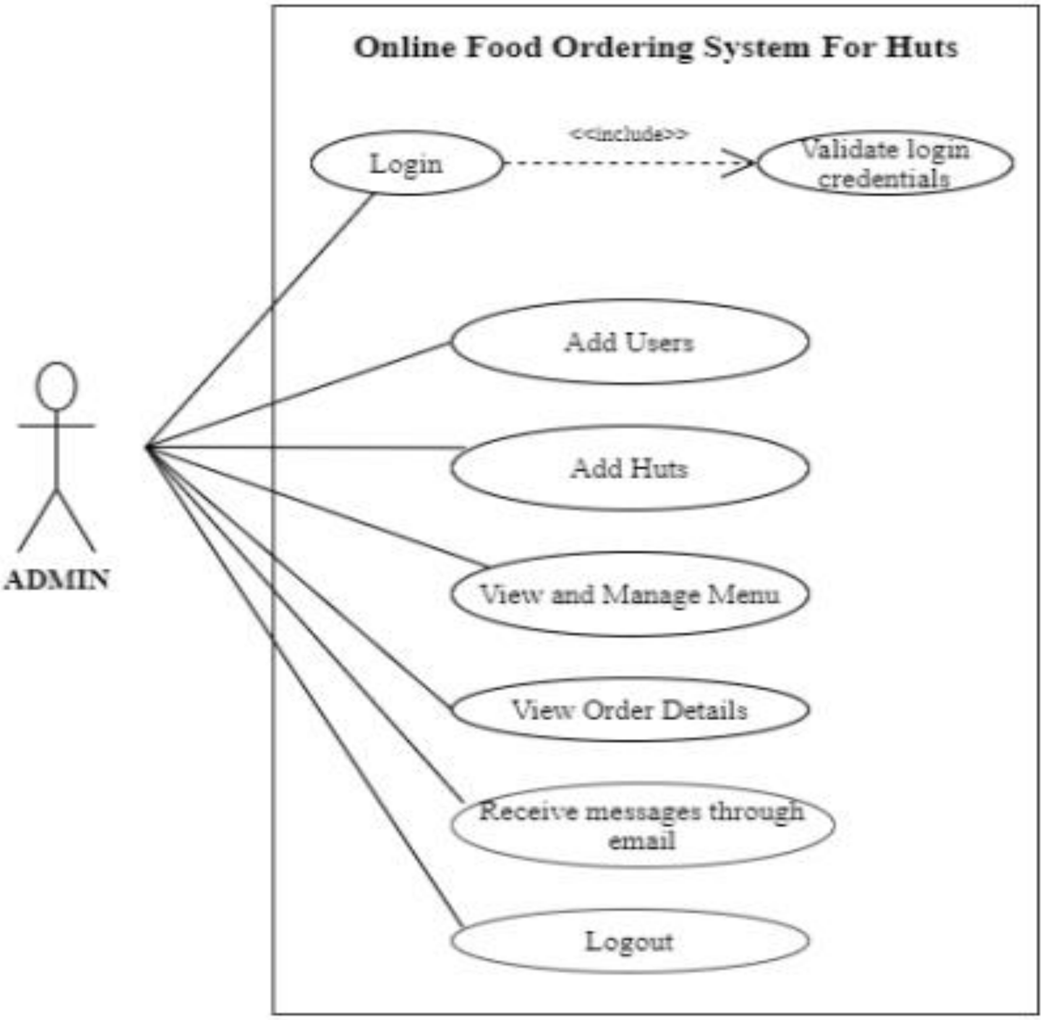
Use case Diagram Notations

Table 3.2 Use Case Diagram Notations

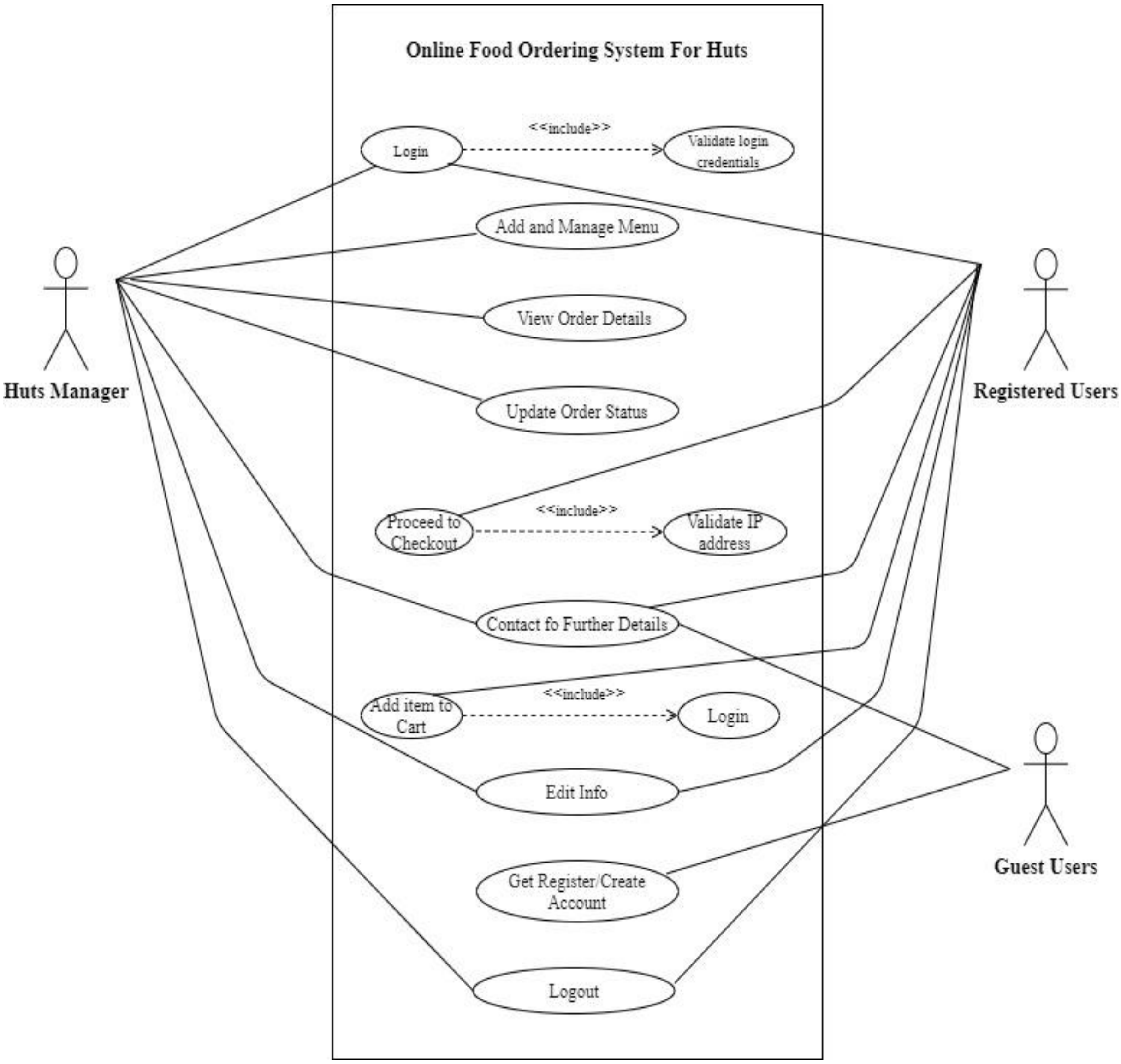
Name & Role	Notation
System It is a rectangular shape system boundary contains use cases in it and actors are place outside the system.	
Use case It is an oval shape which represents the function of system.	
Actor Actors are the users of a system. They are represented with the actor stereotype.	
Associations A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.	
Generalization Generalization of actor means that one actor can inherit role of the other actor. The descendant inherits all the use cases of ancestors.	
Stereotype Relationships **a.<<include>>** It's an implicit function. It is used when base use case is incomplete without the included use case. The included use case is mandatory and not optional. **b.<<extend>>** It's an explicit function as the name applies it extends the base use case and adds more functionality to the system. Sometimes used and called and sometimes not.	 (a)  (b)

“Use Case Diagram of Food Ordering System for Huts”

(a) : Use Case diagram of Admin


Online Food Ordering System For Huts
<<include>>
Login
Validate login credentials
Add Users
Add Huts
View and Manage Menu
View Order Details
Receive messages through email
Logout
ADMIN

(b) : Use Case diagram of Users



Use case ID	**01**
Use case Name	**Login**
Actor	**Admin, Registered Users, Huts Manager**
Description	**Login to his/her account by providing username and password.**
Pre-condition	**Huts Managers, Registered Users and Admin should have internet connection and must be connected to the system.**
Post-condition	**Huts Manager, Registered Users and Admin can access their accounts.**
Basic Flow	**Huts manager, registered users and admin should visit the login page and enter their username, password then system validates username and password and then they logged into the system.**
Exceptional Flow	**Huts manager, registered users and admin cannot log in to the system if invalid data is entered.**

Use case ID	**02**
Use case Name	**Add Users**
Actor	**Admin**
Description	**Admin can Add, Delete and Edit Users.**
Pre-condition	**Successful login**
Post-condition	**Successfully Users Added.**
Basic Flow	**After login admin is provided with option to click on add users and by providing necessary details and clicking on save button, users will be added, and admin can also edit and delete users by clicking on view users option.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**03**
Use case Name	**Add Huts**
Actor	**Admin**
Description	**Admin can Add, Delete and Edit Huts.**
Pre-condition	**Successful login**
Post-condition	**Successfully Huts Added.**
Basic Flow	**After login admin is provided with option to click on add huts and by providing necessary details and clicking on save button, huts will be created, and admin can also edit and delete huts by clicking on view huts option.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**04**
Use case Name	**View and Manage Menu**
Actor	**Admin, Huts Manager**
Description	**Admin and Huts Managers can view menu and can also manage them by performing, Delete and Edit functions.**
Pre-condition	**Successful login**
Post-condition	**Successfully view and manage.**
Basic Flow	**After login both admin and huts managers are provided with option to click on view menu and then they can view menu, also on that page Huts manager can edit and delete menu but admin can only delete menu.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**05**
Use case Name	**View Orders Details**
Actor	**Admin, Huts Managers**
Description	**Admin and Huts Managers can view the orders detail that are ordered from registered users from huts. The huts manager can only view those orders which are ordered from their huts, but the admin can view all orders which are ordered from different huts.**
Pre-condition	**Successful login**
Post-condition	**Successfully view orders detail.**
Basic Flow	**After login admin, huts managers are provided with option to click on view orders and on that page, they can view all the orders that are ordered from registered users from different huts.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**06**
Use case Name	**Receive messages through Email**
Actor	**Admin**
Description	**Admin can get all the messages that are sent by the customers and huts manager.**
Pre-condition	**Admin must provide valid email address**
Post-condition	**Successfully received messages.**
Basic Flow	**Admin can login to any of his/her account like Gmail , Hotmail which he provided on the contact form and after login admin can receive the messages if the email, he/her provided is valid.**
Exceptional Flow	**Admin cannot receive messages because the email he/her provided is invalid.**

Use case ID	**07**
Use case Name	**Add Menu**
Actor	**Huts Manager**
Description	**Huts Managers can add menu according to their available food items.**
Pre-condition	**Successful login**
Post-condition	**Successfully add menu.**
Basic Flow	**After login huts managers are provided with option to click on add menu and by providing necessary details and clicking on save button, menu will be added.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**08**
Use case Name	**Update Orders Status**
Actor	**Huts Managers**
Description	**Huts Managers can view the orders detail that are ordered from registered users from their huts and update order status.**
Pre-condition	**Successful login**
Post-condition	**Successfully update orders status.**
Basic Flow	**After login huts managers are provided with option to click on manage orders and on that page, they can view all the orders that are ordered from registered users from different their huts and they can update order status by clicking on take action button.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**09**
Use case Name	**Proceed to Checkout**
Actor	**Registered users**
Description	**Registered users can proceed to checkout if their device IP address matches with one of the Quaid I Azam university departments Wi-Fi IP addresses**
Pre-condition	**Visit the menu page**
Post-condition	**Successfully proceed to order now page.**
Basic Flow	**The registered users add item in the cart and click on checkout button when the user device IP address matches with one of the Quaid I Azam university departments Wi-Fi IP addresses, the system will redirect to order now page.**
Exceptional Flow	**IP address doesn't match**

Use case ID	**10**
Use case Name	**Contact for further details**
Actor	**Huts Manager, Registered users, Guest users.**
Description	**All the users contact to admin if they face any problem.**
Pre-condition	**Must visit the website and must provide valid information.**
Post-condition	**Successfully send message to admin.**
Basic Flow	**After visit website all the users can contact to admin by clicking on contact us button and after providing valid information, they can send their message by clicking the send button.**
Exceptional Flow	**Provide invalid information.**

Use case ID	**11**
Use case Name	**Add Item to Cart**
Actor	**Registered Users**
Description	**Registered users can add item in their cart.**
Pre-condition	**Successful login**
Post-condition	**Successfully add item in the cart.**
Basic Flow	**After login registered users can add item in their cart by clicking on huts page and then select menu and quantity of that menu.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**12**
Use case Name	**Edit Info**
Actor	**Huts Managers, Registered users.**
Description	**Both the user can edit their basic info(like phone number, Email etc.)**
Pre-condition	**Successful login**
Post-condition	**Successfully edit info.**
Basic Flow	**After login both huts managers and registered users are provided with option to click on edit info and by entering valid information and clicking on update button, the information of both of them will be updated.**
Exceptional Flow	**Unsuccessful Operation.**

Use case ID	**13**
Use case Name	**Get Register/ Create Account**
Actor	**Guest users**
Description	**Guest users visit the website and register themselves if they want to order food.**
Pre-condition	**Should have internet connection and must visit the website.**
Post-condition	**Successfully register.**
Basic Flow	**Visit the website, click on "sign up" option and provides required details to create an account.**
Exceptional Flow	**Registration unsuccessful because of invalid data entered.**

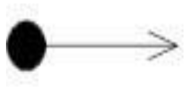
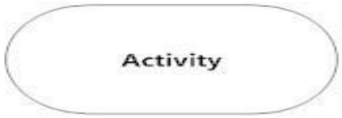
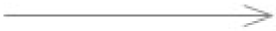
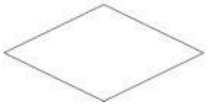
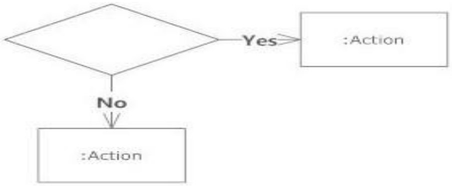
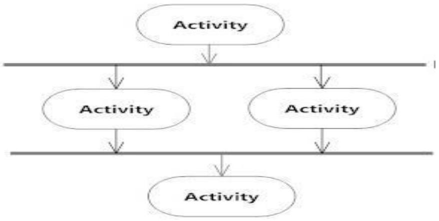
Use case ID	**14**
Use case Name	**Logout**
Actor	**Admin, Huts Manager, Registered users.**
Description	**Logout from his/her account by clicking on logout button.**
Pre-condition	**Must have logged into the account.**
Post-condition	**Destroy their session when they logout.**
Basic Flow	**Clicks on logout button.**
Exceptional Flow	**If user doesn't click on logout button, session will not be destroyed.**

3.3.1.2 Activity Diagram

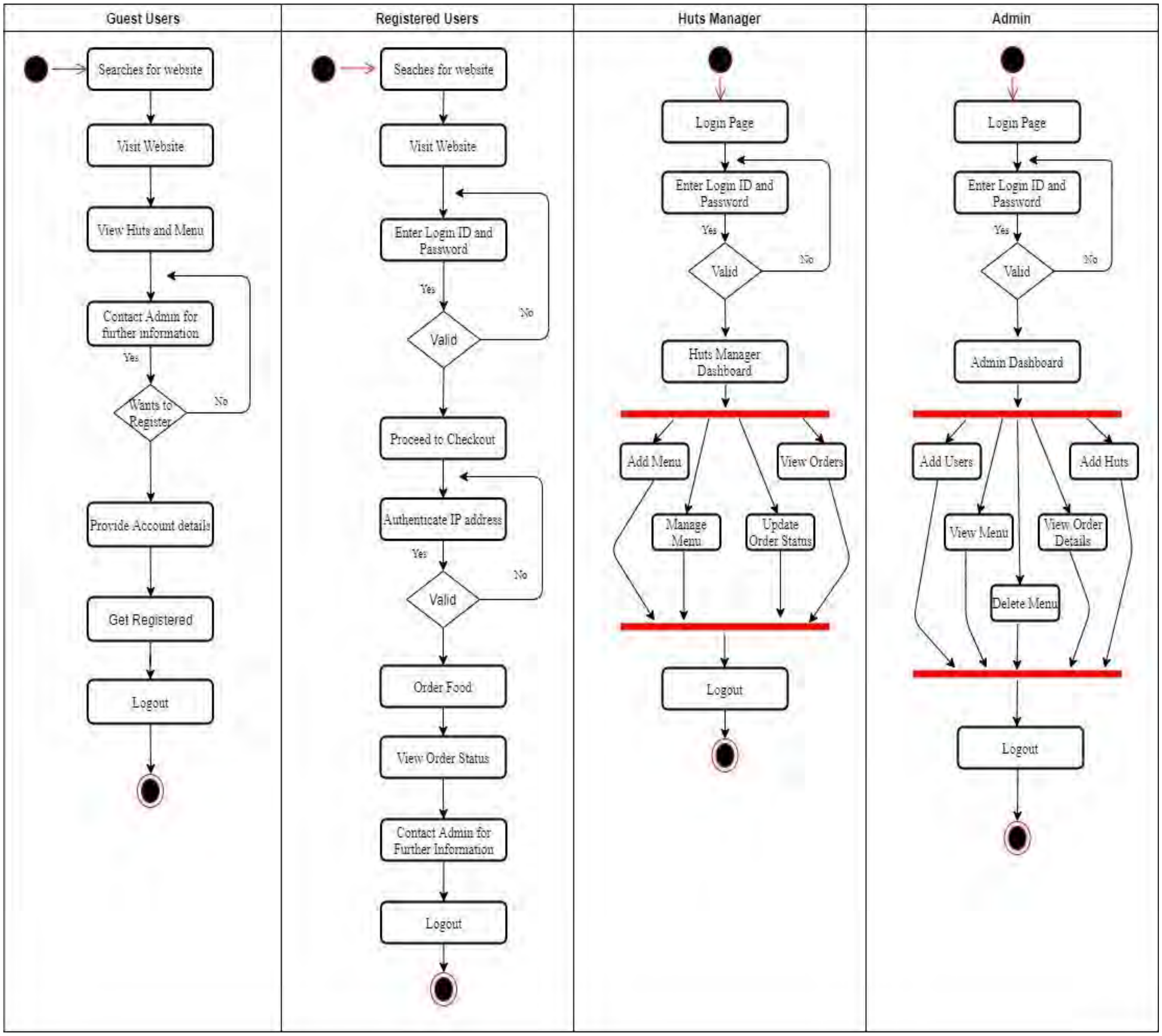
We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case.

Activity Diagram Notations

Table 3.3 Activity Diagram Notations

Name	Notation
Initial state or start point A small filled circle followed by an arrow represents the initial start point.	
Activity or Action State An action state represents the non-interruptible action of objects.	
Action Flow Action flow also called edges and paths illustrate the transitions from one action state to another.	
Decisions and Branching A diamond represents a decision with alternate paths	
Guards In UML, guard are a statement written next to a decision diamond that must be true before moving next to the next activity.	
Synchronization A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.	

“Activity Diagram of Food Ordering System for Huts”

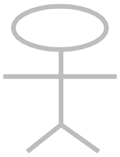
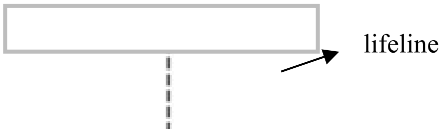
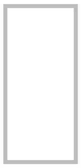
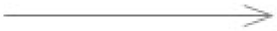
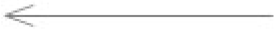
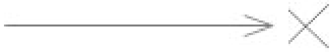


3.3.1.3 Sequence Diagram

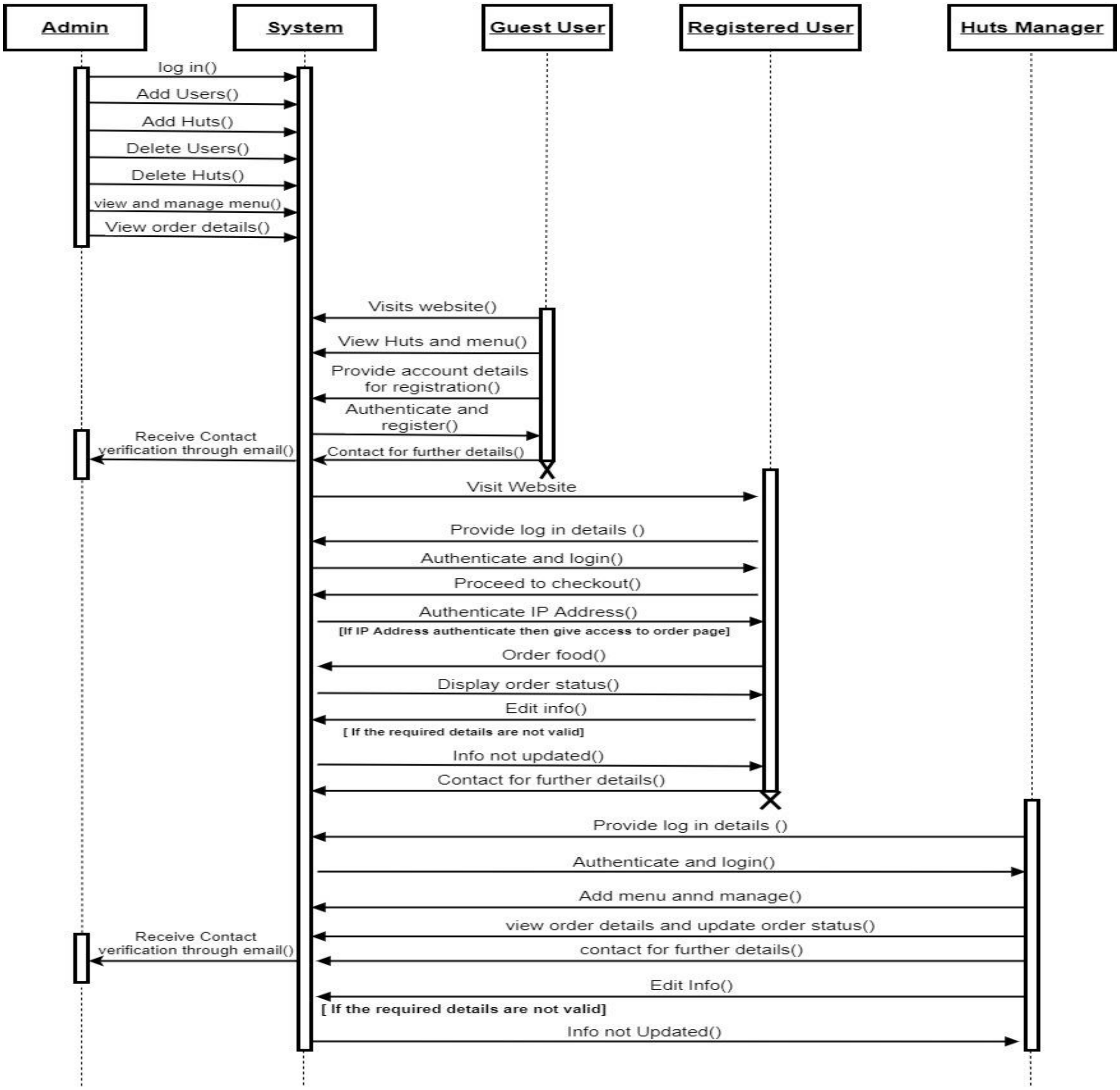
A sequence diagram simply depicts interaction between objects in a sequential order.

Sequence Diagram Notations

Table 3.4 Sequence Diagram Notations

Name & Role	Notation
Actor It is a type of role played by an entity that interacts with the subject.	
Lifeline A lifeline represents an individual participant in the Interaction.	
Activations A thin rectangle on a lifeline represents the period during which an element is performing an operation.	
Call Message A message defines a particular communication between lifelines of an Interaction.	
Return Message It represents the pass of information back to the caller of a corresponded former message.	
Destroy Message It represents the request of destroying the lifecycle of target lifeline.	

"Sequence Diagram of Food Ordering System for Huts"


Admin
System
Guest User
Registered User
Huts Manager
log in()
Add Users()
Add Huts()
Delete Users()
Delete Huts()
view and manage menu()
View order details()
Visits website()
View Huts and menu()
Provide account details for registration()
Authenticate and register()
Receive Contact verification through email()
Contact for further details()
Visit Website
Provide log in details ()
Authenticate and login()
Proceed to checkout()
Authenticate IP Address()
[If IP Address authenticate then give access to order page]
Order food()
Display order status()
Edit info()
[If the required details are not valid]
Info not updated()
Contact for further details()
Provide log in details ()
Authenticate and login()
Add menu annd manage()
view order details and update order status()
Receive Contact verification through email()
contact for further details()
Edit Info()
[If the required details are not valid]
Info not Updated()

3.3.2 Structural Diagrams

It shows the thing in the system classes, objects, packages or modules, physical nodes, components and interfaces.

3.3.2.1 Class Diagram

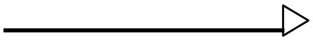
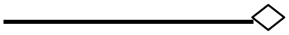
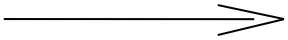
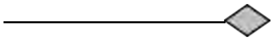
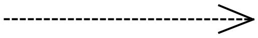
It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations or methods, and the relationships among objects.

Class Diagram Notations

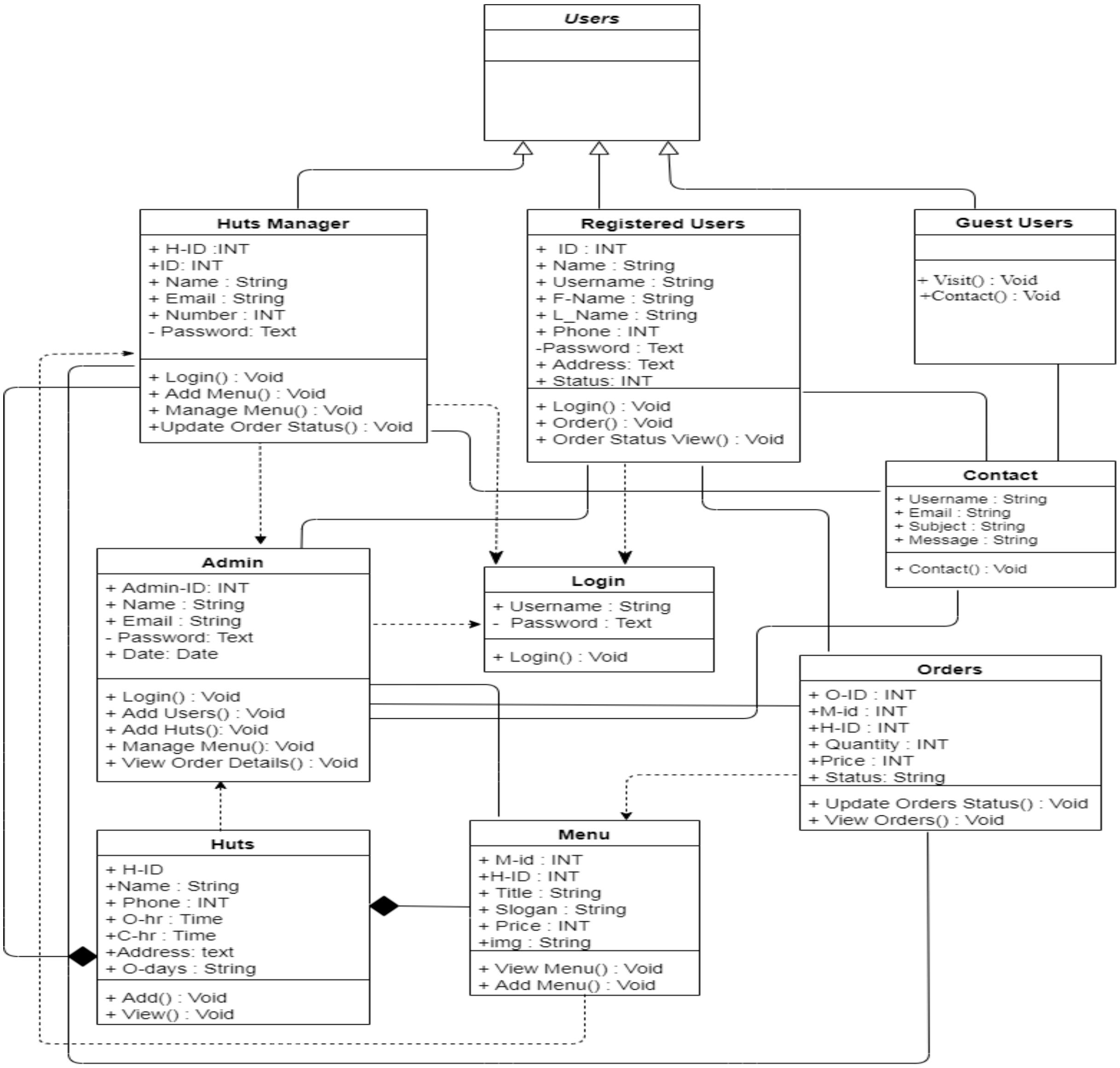
A class notation consists of three parts:

- **Class Name:** the name of class appears in the first partition.
- **Class Attribute:** Attributes are shown in the second partition.
- **Class Operation:** Operation are shown in the third partition. They are servicing the class provides.

Class Relationships

- Inheritance or Generalization 
- Aggregation 
- Simple Association 
- Composition 
- Dependent 

<u>“Class Diagram of Food Ordering System for Huts”</u>



3.4 Relational Database Model

The relational model represents the database as a collection of relations. A relation is nothing but a table of values. Tables consist of rows and columns.

3.4.1 Relational Model Concepts

- **Attribute:** Attributes are the properties which define a relation. e.g., Student, Roll no, Name, etc.

- **Tables:** In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns.

- **Tuple:** It is nothing but a single row of a table, which contains a single record.

- **Relation Schema:** A relation schema represents the name of the relation with its attributes.

- **Degree:** The total number of attributes which are in a relation is called the degree of the relation.

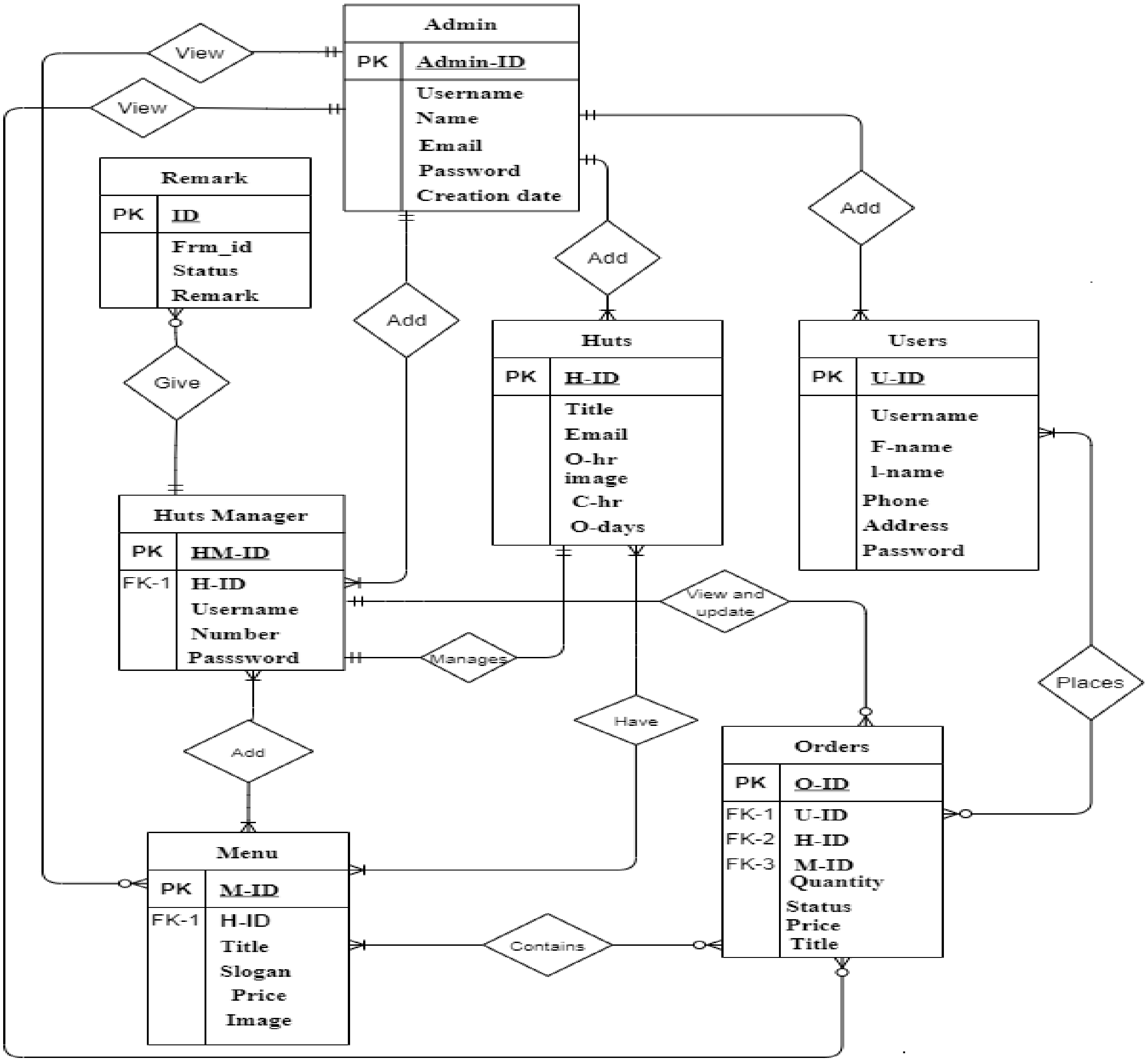
- **Cardinality:** Total number of rows present in the table**.**

- **Column:** The column represents the set of values for a specific attribute.

- **Relation instance**: Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.

- **Relation key**: Every row has one, or multiple attributes, which is called relation key.

“Entity Relationship Diagram of Food Ordering System for Huts”



Chapter #04

Relation Schema

4.1 Introduction

A relation schema for a database of how data is structured. It can be graphic, illustration or another kind of chart used by programmers to understand how each table is laid out, including the columns and the type of data they hold and how tables connect.

A database schema usually specifies which columns are primary keys in tables and which other columns have special constraint such as being required to have unique values in each record. It also usually specifies which columns in which tables contain references to data in other tables, often by including primary keys from other table records so that row can be easily joined.

Table 4.1: Relation Schema

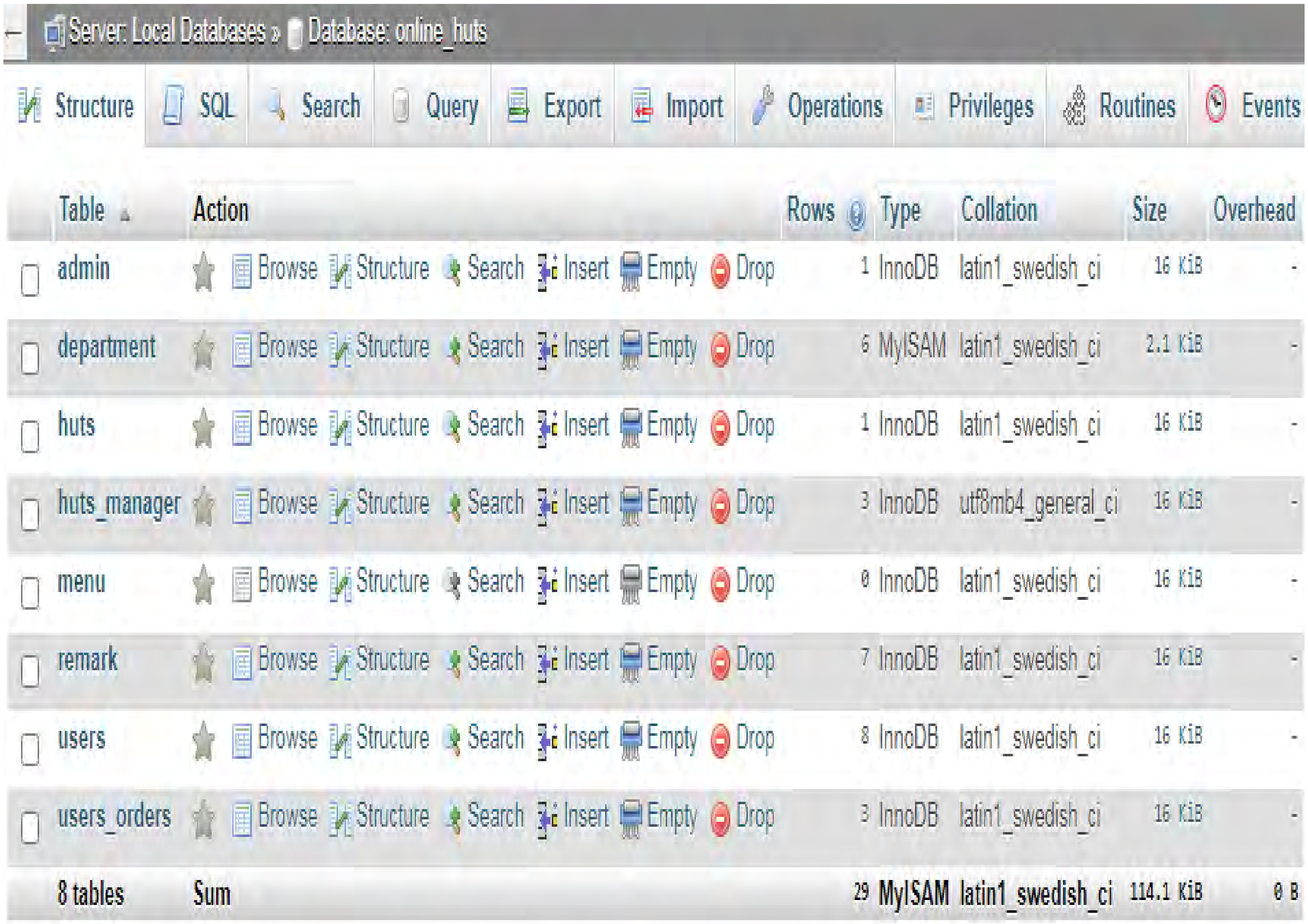


	Table	Action	Rows	Type	Collation	Size	Overhead
□	admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	department	★ Browse Structure Search Insert Empty Drop	6	MyISAM	latin1_swedish_ci	2.1 KiB	-
□	huts	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	huts_manager	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16 KiB	-
□	menu	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
□	remark	★ Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KiB	-
□	users	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
□	users_orders	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
	8 tables	Sum	29	MyISAM	latin1_swedish_ci	114.1 KiB	0 B

Chapter #05

Software Tools And Technologies

5.1 Introduction

Subsequently completing the design stage, we step forward to implementation phase to transform our design into an executable and working product. Implementation is basically grasping the technical requirements of the system and then deploying them using various tools and technologies available at large extent in the world of internet.

As we know that this system is web based, so it is needed to decide which tools and technologies we have to opt for system development. In this chapter we will discuss about all possible options necessary for development and deployment of the system and which programming languages and tools we have used throughout the implementation phase.

5.2 Tool and Technologies

It may not be a tough task for an experienced developer to choose from a never-ending list of tools and technologies, but for a beginner having no experience in the field, surely it becomes a grim task. Having some knowledge and research about various technologies, we will now be using different kind of software's, languages, web browser, servers and database management tools for the development of this system.

Tool that I will be using for developing this system are Microsoft Visual Code 2019, tool through which our database is designed and managed is WAMP, Windows 2007 or above as operating system, WAMP as server and Microsoft Visio as diagraming tool. To make it compatible with environment in which it is running, accessible to all users, making fast load times, achieving Browser Consistency, Effective Navigation, user friendly interface, Good Error Handling, and user-friendly color contrasts; I have chosen distinguished languages of their time like PHP, HTML, JavaScript, Bootstrap, and SQL etc. This system will be used on laptops, desktops etc. having a good internet connection.

5.2.1 Microsoft Visual Studio Code

The Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most widespread developer environment tool, with 50.7% of 87,317 respondents claiming to use it.

5.2.2 WAMP

WAMP Stands for Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP).It was created by Romain Bourdon and is often used for web development and internal testing but may also be used to serve live websites.

The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used to run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet.

WAMP also includes MySQL and PHP, which are two of the most common technologies used for creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.

5.2.3 Microsoft Visio

Microsoft Office Visio is a diagramming and vector graphics application and is part of the Microsoft Office family. It was initially hosted in 1992 by the Shapeware Corporation. Later on, it was developed by Microsoft in 2000.

Microsoft released their latest version v16.0; Standard and Professional in 2019.

5.2.4 PHP

It was originally created by Rasmus Lerdorf in 1994 and now it is produced by The PHP Group.

PHP initially stood for Personal Home Page, but now it stands for the PHP: Hypertext Preprocessor.

It is a widely used open source general-purpose scripting language that is specifically apposite for web development and can be implanted into HTML.

5.2.5 HTML

HTML was developed by WHATWG in 1993. It is the standard markup language for making Web pages. It stands for Hyper Text Markup Language and styles the structure of a Web page which consists of a series of HTML elements.

These elements tell the browser how to display the content and are represented by different tags. These tags label pieces of content such as "heading", "paragraph", and “table. Browsers use these tags to render the content of the page.

5.2.6 MySQL

MySQL is an open-source relational database management(RDBMS). It’s name is a combination of “My”, the name of co-founder Michael Widenius’s daughter and “SQL”

stands for Structured Query Language. SQL is used to converse with a database. According to ANSI, it is the standard language for relational database management systems. SQL statements are used to carry out tasks such as update data on a database or retrieve data from a database.

Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. The standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" are mostly used to achieve nearly all tasks that one may need to do with a database.

5.2.7 JavaScript

It was designed by Brendan Eich and developed by Netscape Communications Corporation, Mozilla Foundation, Ecma International in December 4, 1995. It is frequently shortened as JS and it is a high level, interpreted programming language having curly-bracket syntax. Along with HTML and CSS, JavaScript is one of the essential technologies of the WWW.

JavaScript empowers interactive web pages that's why it is an important part of web application. The enormous majority of websites use it and main web browsers have a dedicated JavaScript engine to execute it.

5.2.8 Bootstrap

Bootstrap is a free front-end framework for faster and less difficult web development which includes HTML and CSS based design templates for design, forms, buttons, tables, navigation, modals, image containers and many other, as well as optional JavaScript plugins.

Bootstrap additionally offers the ability to easily create responsive designs which is; creating those web sites that automatically adjust themselves to look good on all devices, from small phones to large desktops.

5.2.9 CSS

Cascading Style Sheets(CSS) is a style sheet language used for describing the appearance of a document written in a markup language like HTML. CSS is a bedrock technology of the World Wide Web, alongside HTML and JavaScript.

It was developed by World Wide Web in Consortium on December 17, 1996. CSS is designed to enable the parting of appearance and content, including layout, colors, and fonts.

Chapter #06

User Interface

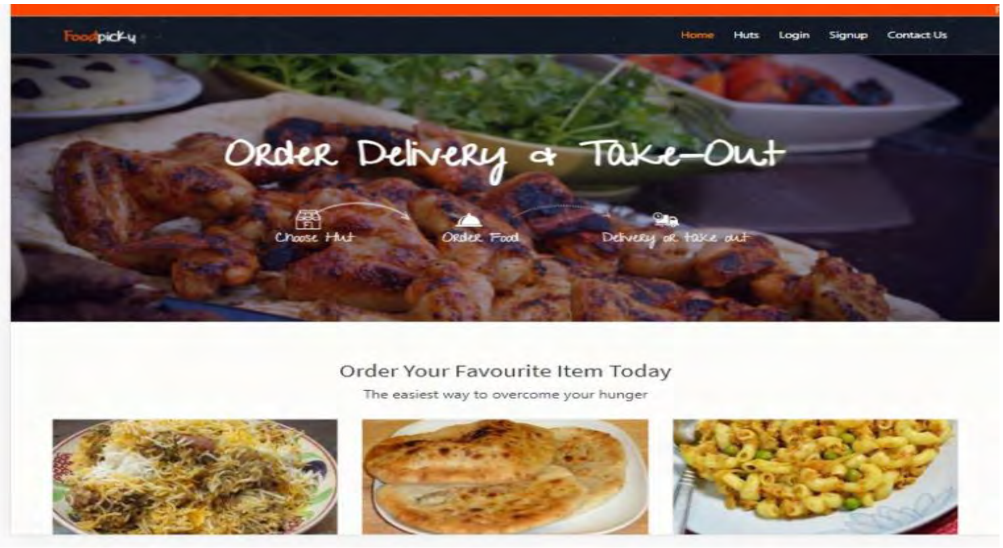
6.1 Introduction

User interface(UI) is everything designed into an information device with which a human being may interact -- including display screen, keyboard, mouse, light pen, the appearance of a desktop, illuminated characters, help messages, and how an application program or a Web site invites interaction and responds to it. In early computers, there was very little user interface except for a few buttons at an operator's console. The user interface was largely in the form of punched card input and report output.

Later, a user was provided the ability to interact with a computer online and the user interface was a nearly blank display screen with a command line, a keyboard, and a set of commands and computer responses that were exchanged. This command line interface led to one in which menus (list of choices written in text) predominated. And, finally, the graphical user interface (GUI) arrived, originating mainly in Xerox's Palo Alto Research Centre, adopted and enhanced by Apple Computer, and finally effectively standardized by Microsoft in its Windows operating systems. The user interface can arguably include the total "user experience," which may include the aesthetic appearance of the device, response time, and the content that is presented to the user within the context of the user interface.

For the website design purpose, user the interface’s framework can be shown below:

Fig 6.1 Home Page

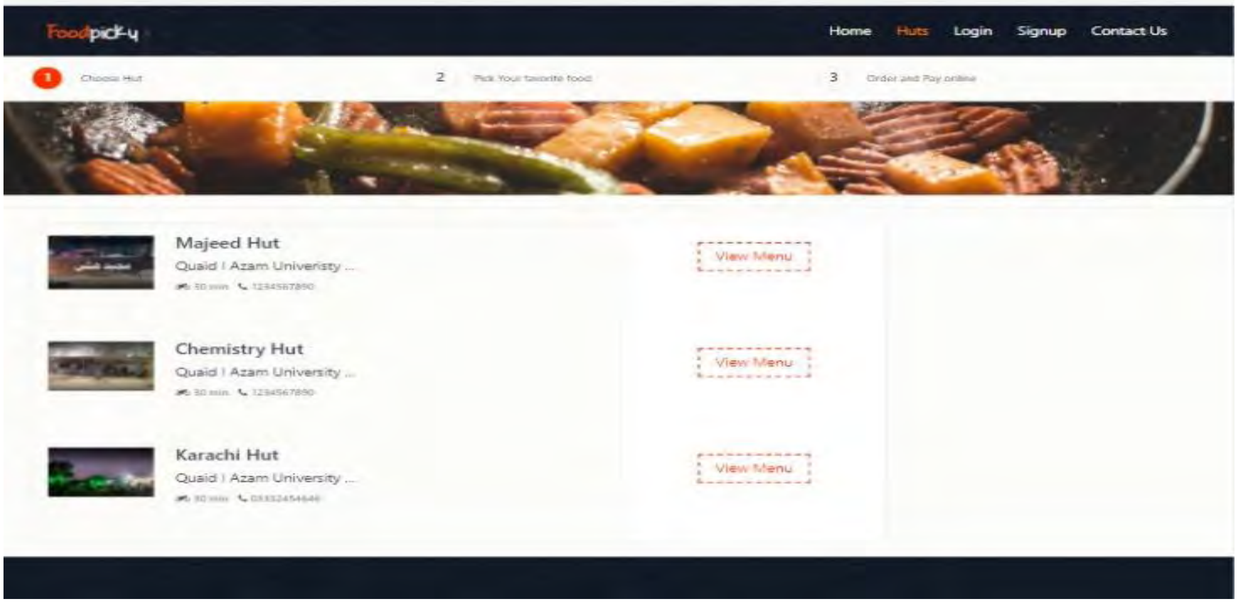


Order Delivery & Take-Out

Order Your Favourite Item Today

The easiest way to overcome your hunger

Fig 6.2 Huts Page

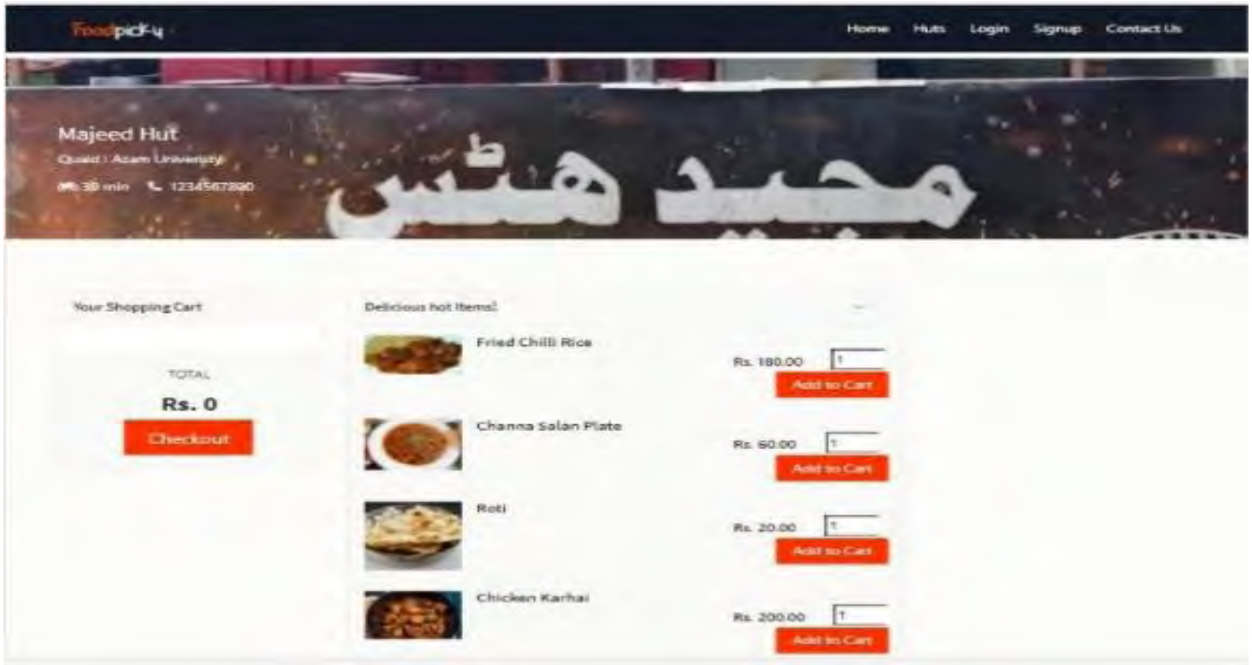


Majeed Hut
Quaid I Azam Univeristy ...
30 min ☎ 1234567890

Chemistry Hut
Quaid I Azam University ...
30 min ☎ 1234567890

Karachi Hut
Quaid I Azam University ...
30 min ☎ 03332454646

Fig 6.3 Menu Page



Majeed Hut
Quaid I Azam Univerpty
30 min ☎ 1234567890

Your Shopping Cart

TOTAL
Rs. 0

Delicious hot Items!

Fried Chilli Rice	Rs. 180.00	1
Channa Salan Plate	Rs. 60.00	1
Roti	Rs. 20.00	1
Chicken Karhai	Rs. 200.00	1

Fig 6.4 User login Page

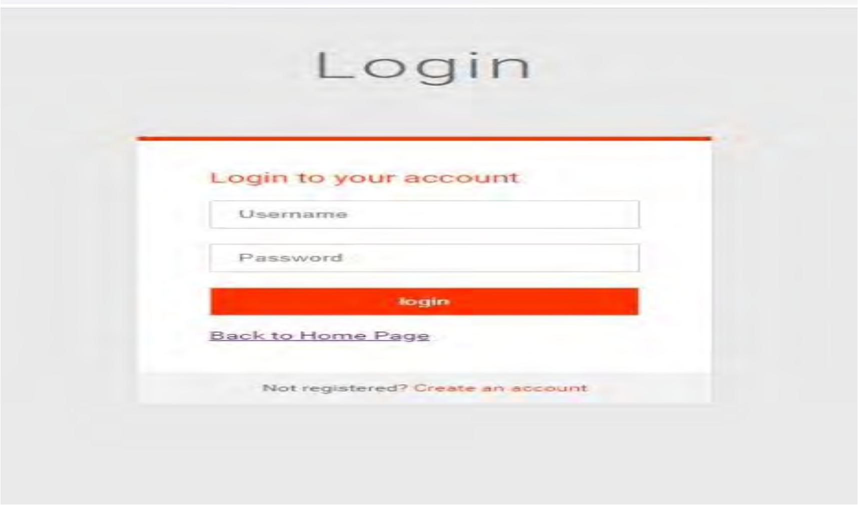
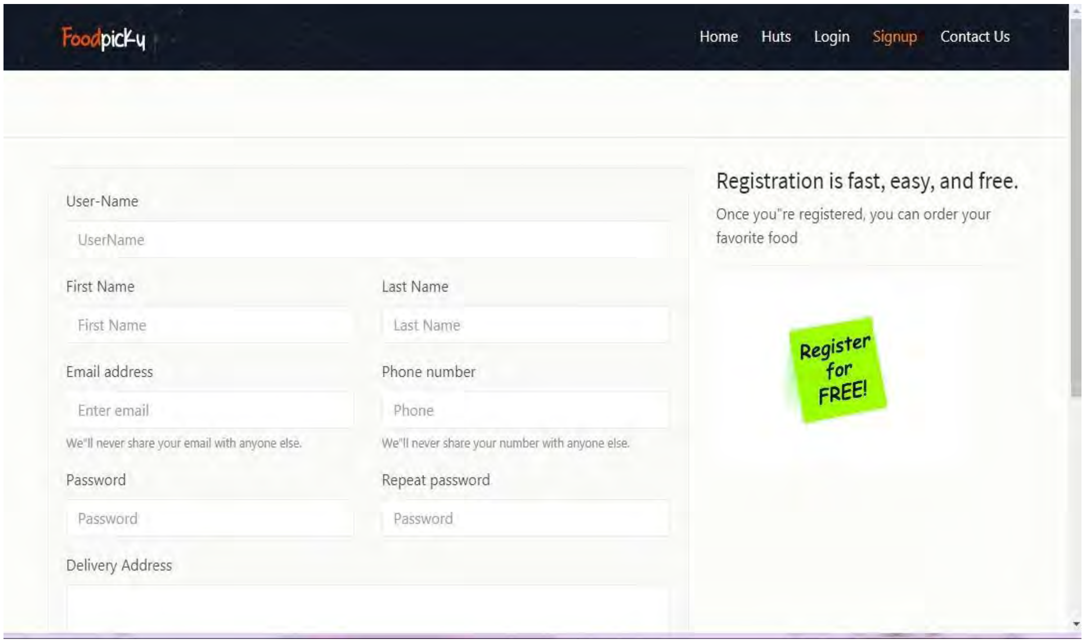


Fig 6.5 Registration Page



We"ll never share your email with anyone else.

We"ll never share your number with anyone else.

Registration is fast, easy, and free.

Once you"re registered, you can order your favorite food

Fig 6.6 Huts Manager Dashboard

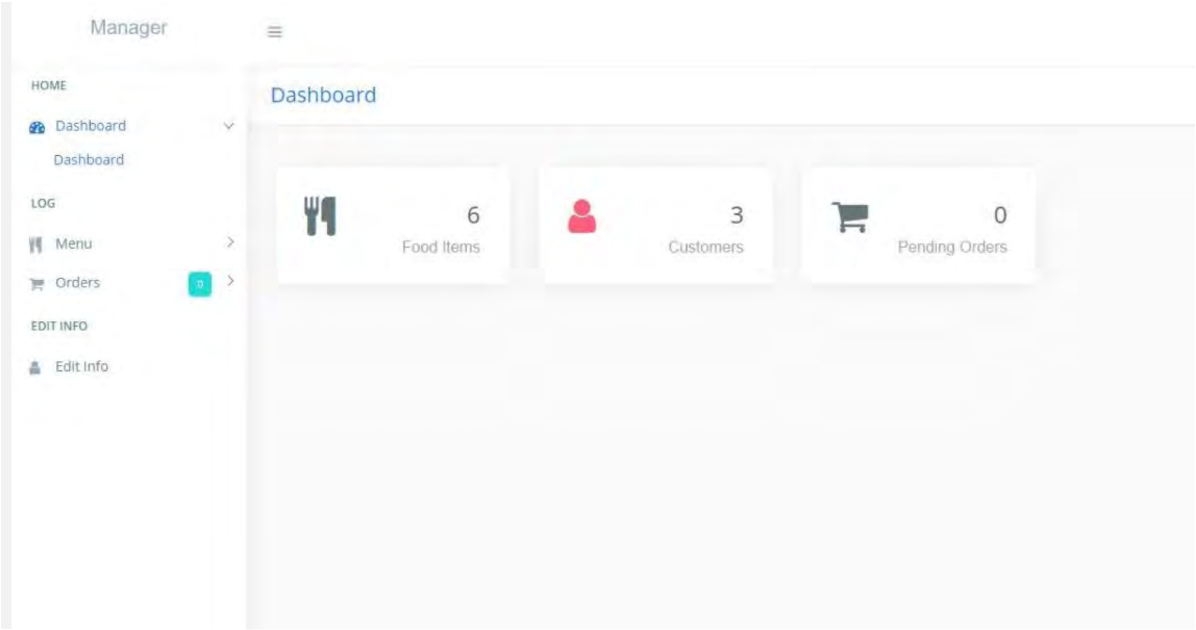


Fig 6.7 Admin Dashboard

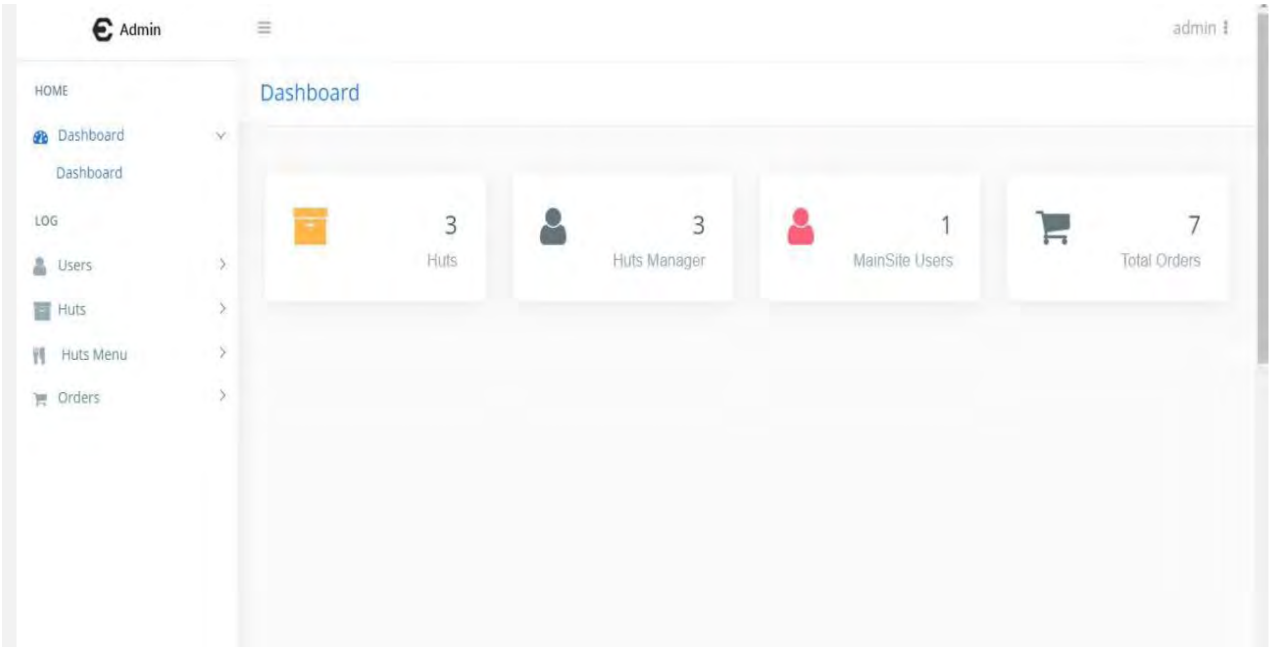
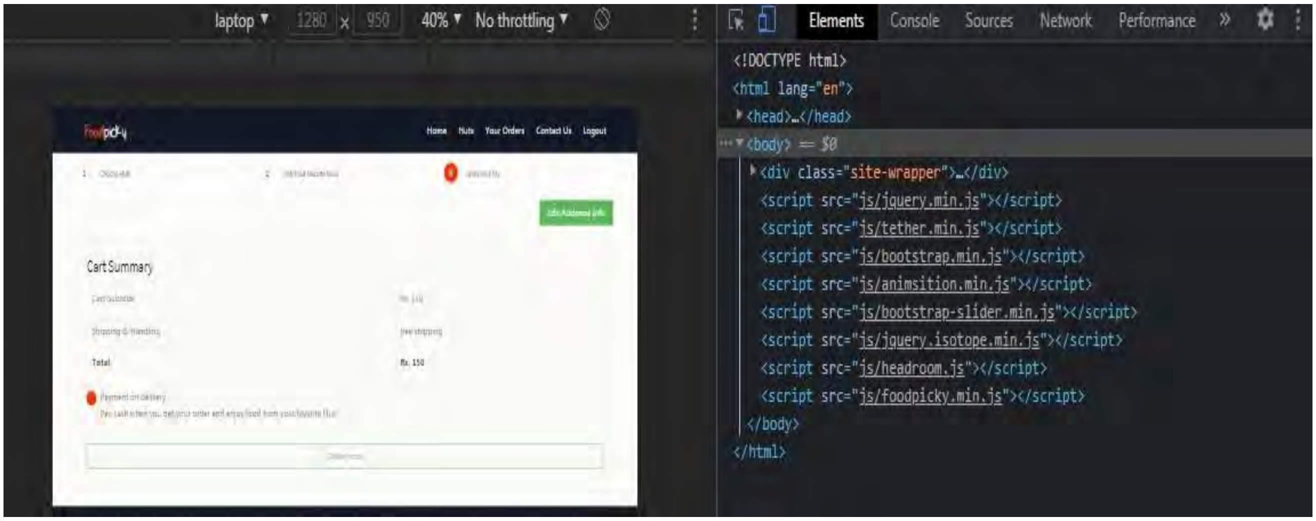
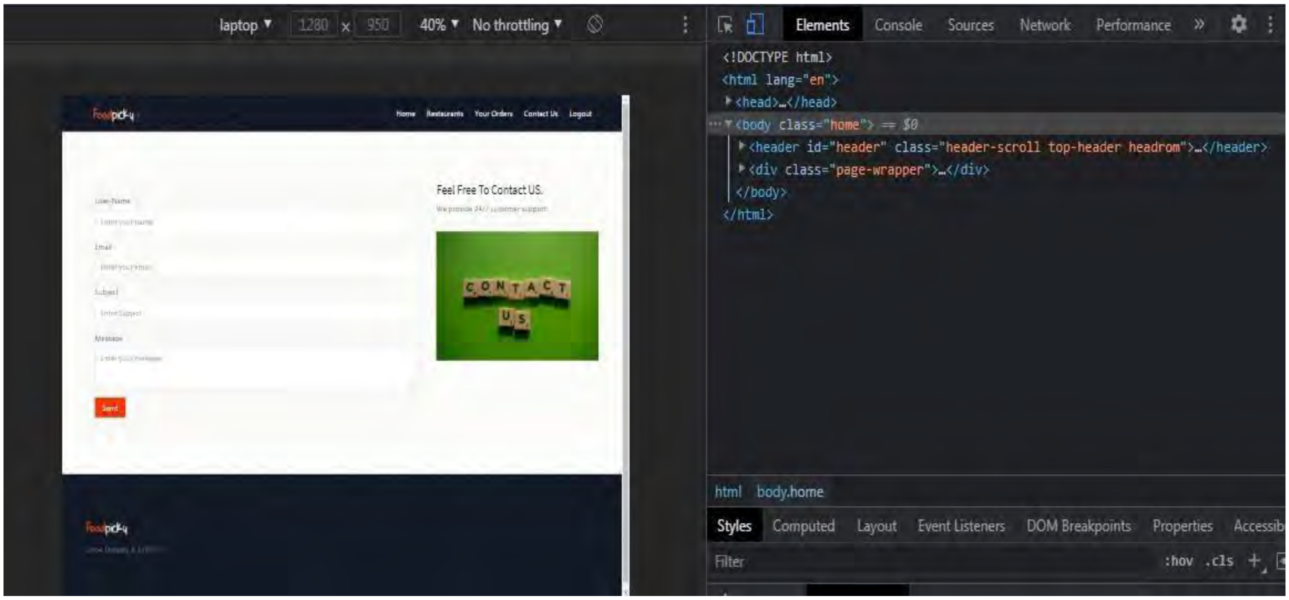


Fig 6.8 Order Page



```
<!DOCTYPE html>
<html lang="en">
▸<head>…</head>
▾<body> == $0
  ▸<div class="site-wrapper">…</div>
   <script src="js/jquery.min.js"></script>
   <script src="js/tether.min.js"></script>
   <script src="js/bootstrap.min.js"></script>
   <script src="js/animsition.min.js"></script>
   <script src="js/bootstrap-slider.min.js"></script>
   <script src="js/jquery.isotope.min.js"></script>
   <script src="js/headroom.js"></script>
   <script src="js/foodpicky.min.js"></script>
  </body>
</html>
```

Fig 6.9 Contact Us Page



```
<!DOCTYPE html>
<html lang="en">
▸<head>…</head>
▾<body class="home"> == $0
  ▸<header id="header" class="header-scroll top-header headrom">…</header>
  ▸<div class="page-wrapper">…</div>
  </body>
</html>
```

Chapter #07

Testing And Evolution

7.1 Introduction

This chapter describes the testing and evaluation of the website developed. Section 6.2 reviews the testing that took place at three levels. Software testing refers to the testing of the produced courseware. It looks at questions such as: whether the web pages are generated correctly, whether all the links work. Section 6.3 covers the evaluation phase of the system.

It reports on the implementation and whether it works properly or not. The system is then evaluated from a software point of view.

7.2 Testing

This section covers the testing of the template. It deals with concept testing, software testing and the testing of the courseware itself. The concept testing asks the question as whether the developed website properly works.

7.2.1 Software Testing Levels

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification.

Software testing involves both verification and validation. Verification involves checking that the program conforms to its specification, while validation involves checking that the program as implemented meets the expectations of the user. Static checking techniques include program inspections and analysis. Dynamic techniques (tests) involve exercising the system. Accordingly, the verification stages employ what in the technical lingo are known as the white box testing techniques whilst the validation stage uses black box testing techniques.

The testing process normally has five stages. Firstly, individual units are tested in unit testing. Module testing tests modules (usually a collection of dependent units). Sub-system testing tests collections of modules and often exposes sub-system interface mismatches. System testing tests the system as a whole and finally, there is (user) acceptance testing.

7.2.1.1 Module Testing

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. Module testing is largely a white box oriented.

Module Testing is recommended because:

- Probability of identifying errors or bugs on smaller chunks of program becomes higher.
- Multiple modules can be tested simultaneously and hence supports parallel testing.
- Complexity of testing can be easily managed.

7.2.1.2 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

7.2.1.3 Subsystem Testing

Subsystem verification testing is performed as a prelude to system testing. It is performed in the operational environment using installed system hardware and software.

Testing at the subsystem level should be performed:

- When different developers, vendors, or contractors have been responsible for delivering stand-alone subsystems.
- When the complete functionality of a subsystem could not be tested at a lower level because it had not been fully integrated with the necessary communication infrastructure.
- When it was previously impossible to connect to the field for the testing phase.

7.2.1.4 System Testing

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

7.2.1.5 Acceptance Testing

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

7.3 Software Testing Types

7.3.1 Manual Testing

Manual testing is the procedure of testing the software by hand to find the faults. Tester should have knowledge of the viewpoint of end-users to confirm that all features are working as stated in the software requirement document.

7.3.2 Automation Testing

Automation testing is the procedure of testing the software using an automatic tool to discover faults. In this procedure, tester execute executes the test scripts and generate the test outcomes automatically by using the various tools.

7.4 Testing Approaches

The approaches for testing are:

- White Box Testing

- Black Box Testing

7.4.1 White Box Testing

It is also called Glass Box, Clear Box, and Structural Testing. This testing is grounded on application's internal code structure. In white-box testing, a core viewpoint of the system as well as programming skills are used to design test cases. This testing is generally done at the unit level.

7.4.2 Black Box Testing

It is also called behavioral, Specification-Based or Input-Output Testing. In this testing process testers assess the functionality of the software under tests deprived of watching at the internal code structure of the software.

Table 7.1: Test Case of User Registration

Test Case ID	TC-1
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Registration
Description	Check whether user is registered or not.
Procedure	Enter all the required details and click the register button.
Expected Result	User should register successfully.
Actual Result	User registered successfully.
Status	Success.

Table 7.2: Test Case of User Login

Test Case ID	TC-2
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Login
Description	Check whether user is login or not.
Procedure	Enter username and password and click the login button.
Expected Result	User should Login successfully.
Actual Result	User logged in successfully.
Status	Success.

Table 7.3: Test Case of Admin Adding Huts and Users

Test Case ID	TC-3
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Add Huts, Users
Description	Check whether Huts and users added in the database or not.
Procedure	Enter all the required details and click the save button.
Expected Result	Huts and users should add in database successfully.
Actual Result	Huts and users added in database successfully.
Status	Success.

Table 7.4: Test Case of Huts Manager Adding menu

Test Case ID	TC-4
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Add menu
Description	Check whether the menu is added or not.
Procedure	Enter item name, price and image and click the save button.
Expected Result	Menu should Add successfully.
Actual Result	Menu Added successfully.
Status	Success.

Table 7.5: Test Case of Cart

Test Case ID	TC-5
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Cart
Description	Check whether the food item is added in the cart or not.
Procedure	Enter quantity and click on Add to Cart button.
Expected Result	Item should add in the cart.
Actual Result	Item Added successfully.
Status	Success.

Table 7.6: Test Case of checkout process

Test Case ID	TC-6
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Checkout Process
Description	When the customer clicks on order now button then check whether that order show on huts manager dashboard or not.
Procedure	Login to the system as a hut manager and click on the order module.
Expected Result	Ordered item should display on the huts manager dashboard.
Actual Result	Ordered item display successfully.
Status	Success.

Table 7.7: Test Case of Order Status

Test Case ID	TC-7
Tester	Nayab Aziz
Test Type	Black Box
Test Case Name	Order Status
Description	Check whether the order status is displayed correctly or not.
Procedure	Login to the system as a user and visit yours order page.
Expected Result	Order status should display correctly.
Actual Result	Order Status displayed correctly.
Status	Success.

7.5 Evaluation

Evaluation is a systematic determination of a subject's merit, worth and significance, using criteria governed by a set of standards. It can assist an organization, program, design, project or any other intervention or initiative to assess any aim, realizable concept/proposal, or any alternative, to help in decision-making; or to ascertain the degree of achievement or value in regard to the aim and objectives and results of any such action that has been completed.

Evaluation is the structured interpretation and giving of meaning to predicted or actual impacts of proposals or results. It looks at original objectives, and at what is either predicted or what was accomplished and how it was accomplished.

7.5.1 Purposes

Purposes Evaluation can be conducted for the purposes of decision making, judgements, conclusion, findings, new knowledge, organizational development and capacity building in response to the needs of identified stakeholders leading to improvement, decisions about future programming, and/or accountability ultimately informing social action ameliorating social problems and contributing to organizational or social value.

Chapter #08

Conclusion And Future Enhancement

7.1 Problems During Coding

When I implemented the Web Site, many difficulties were met. I tried to find the problems and solve them through the study.

Firstly, Installation was the very tedious step as I used this for this first time. I installed three to four time to get exact outcomes. This was awful to know about different modules and different versions of Visual Studio and SQL Server.

Secondly, I have to learn and revised all the basic concepts of PHP, HTML,CSS, JavaScript and SQL.

7.2 The Summary For Whole Thesis

The purpose of this project is to design a website for students and faculty members of Quaid-I-Azam university where they can easily order food from their university huts in their respective departments. For design process, to learn how to design a web product, to include the UML images graphic, functions analysis. Because I needed to learn some scripting languages and server-side languages which was an awesome experience.

For the final product, the functions of login and download. It has been tested on Windows PC which works finely. That means that the final product's cross platform design is successful. Although without the testing in real mobile devices, but if it could work on simulator, it will work fine on real devices too.

I will pay more attention on the beginning website analyzing and coding skill. Because analysis is the most important part in product development. And some problems come from the coding skill. That is the second point which I will work hard on.

7.3 Enhancement

Feature enhancement has its own importance, because with the passage of time system evolves and there is a need for more feature and functionalities, so system should allow future enhancement. Proposed system is developed by keeping this in mind.

7.3.1 Evaluation Of The System

After the completion of the project an important final step of evaluation is done. Evaluation is necessary step for the improvement in the future enhancement in the application The purpose of evaluation is to do the assessment of the whole system. In this section the evaluation of my system's design and implementation has been done and it showed that the system follows the requirement gathered at the start of the system.

7.3.2 Future Enhancements

In future, I have planned to include more facilities that are listed below.

- **Add Delivery Points:** In future I have planned to include delivery points like in each order the delivery point increase and after reaching to minimum limit the user will get amazing offers.

- **Add rating/review option:** In future I have decided to take review/ratings on the food or on huts services from which they ordered food.

- **Order Tracking System:** I have planned to include order tracking system so the registered users can easily track their orders without calling to huts manager to provide information to them about their orders.

- **Registration through University ID:** As in current system both registered and guest users can access website only if their device IP address matches with university Wi-Fi IP addresses. But sometimes you're far away from your department like sitting on the ground so in that case you won't be able to access website and order food so in future I have decided to restrict users to registers with their university ID so that they can access website and order food more easily from any part of the university.

References

Ian Somerville - Software Engineering_10thEd_2015

https://en.wikipedia.org/wiki/online_food_ordering

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-(diagram-name)/

https://www.computing.dcu.ie/~mward/mthesis/chapter8.pdf

https://ops.fhwa.dot.gov/publications/tptms/handbook/chapter_8. htm

https://en.wikipedia.org/wiki/Evaluation

https://en.wikipedia.org/wiki/PHP