

# **KITAB-GHAR**

**Final Year Project**

**Session 2015-2019**

A project submitted in partial fulfillment of the degree of

BS in Software Engineering



Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior College, Lahore

FALL 2019

Type (Nature of project)	[ <input checked="" type="checkbox"/> ] <b>Development</b> [ <input type="checkbox"/> ] <b>Research</b> [ <input type="checkbox"/> ] <b>R&amp;D</b>			
Area of specialization	Web			
FYP ID	FYP-BSSE-S19-017			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)		Ahsan Amanullah Khan	Ahsanamanullahkhan@gmail.com	
(ii)		Hafiz Muhammad Ahamd	<a href="mailto:Hafiz.m.ahmad57@gmail.com">Hafiz.m.ahmad57@gmail.com</a>	
(iii)		Azam Bilal sial	Azambilal.sial68@gmail.com	

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

### Plagiarism Free Certificate

This is to certify that, I \_\_\_\_\_ S/D of Muhammad Siddique, group leader of FYP under registration no \_\_\_\_\_ at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: \_\_\_\_\_ Name of Group Leader: \_\_\_\_\_ Signature: \_\_\_\_\_

Name of Supervisor: Dr. ABC

Co-Supervisor: Mr. XYZ

Designation: Lecturer

Designation: Associate Professor

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HoD: Dr. Arfan Jaffar

Signature: \_\_\_\_\_

# Project Report

## KITAB-GHAR

### Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

---

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

**HEAD OF THE DEPARTMENT**

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Dedication**

I want to dedicate this work to those students who wants to learn something even though they have no resources. I also want to dedicate this project to all of my friends who have crossed this stage without any resources and faced these types of difficulties.

## **Acknowledgements**

We would like to pay our gratitude to the almighty ALLAH who created us with not only the ability to design and program this system but also the power of patience.

We are obliged and thankful to our project supervisor Mr. **Muhammad Ahmad**, Senior Lecturer, Software Engineering Department, Superior University, for his continuous encouragement, motivation and professional guidance during the work of this project which has proven to be an integral part of it. Without his valuable support and guidance, this project could not elevate up this level of development from our point of view.

We would like to express our sincere and cordial gratitude to the people those who have supported us directly, evaluated and criticized our work in several phases during the development of this project and for preparing this dissertation indirectly.

## **Executive Summary**

We are developing a platform that works on Website. This source totally depends on facilitating students in Study and other Educational Material. In this project we provide students with a specific source so that he/she can gather the books for study from a single source and don't have to use other sources like Google Drive, Google Classroom and Drop box etc. It provides feature in which every student can sale/purchase the books of their desires and needs. It provides students with additional feature of providing the location of the bookshop which have the book they need. A Particular student can provide his/her own location so that they can sale their books on our website. Customer can compare different websites and prices given in these websites so its easy for them to buy books in reasonable price and can know which is giving good material and data. We also provide recommendation to customer so they will know what's new in their for them.

We full facilitate customer as well as sellers in this platform .As they get full information about books and the seller can sell books with full ease they only have to place their data of books what they have and customer can get what they want .In our website and also can compare this with other websites so they know which is providing authentic information and reasonable prices.

## Table of Contents

### Contents

Dedication .....	vi
Acknowledgements .....	vii
Executive Summary .....	viii
Table of Contents .....	ix
List of Figures .....	xv
List of Tables .....	xvi
Chapter 1 .....	17
Introduction.....	17
1.1. Background .....	2
1.2. Motivations and Challenges .....	2
1.3. Goals and Objectives.....	3
1.4. Literature Review/Existing Solutions .....	4
1.5. Proposed Solution .....	4
1.6. Project Plan .....	4
1.6.1. Work Breakdown Structure .....	5
Now describe further decomposition of work breakdown structure:.....	5
4.3 Database Design: .....	7
5.2 UI development:.....	7
5.3 Code development: .....	7

5.4 Hardware design and implementation: .....	7
6.1 Test plan:.....	7
8.1 Documentation:.....	8
1.6.2. Roles & Responsibility Matrix .....	9
1.6.3. Gantt Chart .....	10
Chapter 2.....	12
Software Requirement Specifications.....	12
2.1. Introduction .....	13
2.1.1. Purpose .....	13
Kitaab Ghar is a system in which we provide students the data related their books which they wants we also provide location from where they can get books and prices of different shops. And we also sell used books. People can buy and sell old books so it helps people Recycling the books and buying books in reasonable price. ....	13
2.1.2. Document Conventions .....	13
2.1.3. Intended Audience and Reading Suggestions .....	13
2.1.4. Product Scope .....	14
2.1.5. References .....	15
2.2. Overall Description .....	15
2.2.1. Product Perspective .....	15
2.2.2. Product Functions.....	15
2.2.3. User Classes and Characteristics .....	16
2.2.4. Operating Environment .....	17
2.2.5. Design and Implementation Constraints.....	17
2.2.6. User Documentation .....	18
2.2.7. Assumptions and Dependencies .....	18
2.3. External Interface Requirements .....	18

---

2.3.1.	User Interfaces .....	18
2.3.2.	Hardware Interfaces.....	19
2.3.3.	Software Interfaces .....	20
2.3.4.	Communications Interfaces .....	20
2.4.	System Features.....	20
2.4.1.	SF1- Create Account .....	22
2.4.1.1.	Description and Priority .....	22
2.4.1.2.	Stimulus/Response Sequences .....	22
2.4.1.3.	Functional Requirements .....	22
2.4.2.	SF2- Login.....	22
2.4.2.1.	Description and Priority .....	22
2.4.2.2.	Stimulus/Response Sequence.....	23
2.4.2.3.	Functional Requirements .....	23
2.4.3.	SF3- Place Order .....	23
2.4.3.1.	Description and Priority .....	23
2.4.3.2.	Stimulus/Response Sequences .....	23
2.4.3.3.	Functional Requirements .....	23
2.4.4.	SF4- Track Order.....	24
2.4.4.1.	Description and Priority .....	24
2.4.4.2.	Stimulus/Response Sequences .....	24
2.4.4.3.	Functional Requirements .....	24
2.4.5.	SF5-Add Customer .....	24
2.4.5.1.	Description and Priority .....	24
2.4.5.2.	Stimulus/Response Sequences .....	24
2.4.5.3.	Functional Requirements .....	24

2.4.6.	SF6-Search Customer .....	25
2.4.6.1.	Description and Priority .....	25
2.4.6.2.	Stimulus/Response Sequences .....	25
2.4.6.3.	Functional Requirements .....	25
2.4.7.	SF7- Manage Accounts .....	25
2.4.7.1.	Description and Priority .....	25
2.4.7.2.	Stimulus/Response Sequences .....	26
2.4.7.3.	Functional Requirements .....	26
2.5.	Other Nonfunctional Requirements .....	26
2.5.1.	Performance Requirements.....	26
2.5.2.	Safety Requirements.....	26
2.5.3.	Security Requirements.....	26
2.5.4.	Software Quality Attributes.....	26
2.5.5.	Business Rules .....	27
2.6.	Other Requirements.....	27
Chapter 3	.....	28
Use Case Analysis	.....	28
3.1.	Use Case Model .....	29
3.2.	Fully Dressed Use Cases .....	30
Chapter 4	.....	41
System Design	.....	41
4.1.	Architecture Diagram.....	42
4.2.	Domain Model.....	43
4.3.	Entity Relationship Diagram with data dictionary.....	46
4.4.	Class Diagram .....	49

4.5. `Sequence / Collaboration Diagram .....	50
4.6. Operation contracts .....	50
4.7. Operation contracts .....	54
4.8. Activity Diagram.....	59
4.9. State Transition Diagram .....	61
4.10. Component Diagram.....	67
4.11. Deployment Diagram .....	68
4.12. Data Flow diagram [ <i>only if structured approach is used - Level 0 and 1</i> ].....	69
Chapter 5.....	73
Implementation .....	73
5.1. Components, Libraries, Web Services and stubs .....	74
5.2. Tools and Techniques.....	74
5.3. Best Practices / Coding Standards.....	75
5.4. Version Control .....	75
Chapter 6.....	79
Testing and Evaluation .....	79
6.1. Use Case Testing:.....	80
6.2. Equivalence partitioning .....	84
6.3. Boundary value analysis: .....	88
6.4. Data flow testing: .....	88
6.5. Unit testing .....	90
6.6. Integration testing:.....	94
6.7. Performance testing:.....	94
6.8. Stress Testing: .....	94
Chapter 7.....	95
Faculty of CS&IT, The Superior College Lahore, Pakistan	<b>xiii</b>

Summary, Conclusion and Future Enhancements .....	95
7.1. Project Summary: .....	96
7.2. Achievements and Improvements: .....	96
7.3. Critical Review: .....	96
7.4. Lessons Learnt: .....	97
7.5. Future Enhancements/Recommendations: .....	97
Reference .....	98
and Bibliography.....	98

## List of Figures

Figure 1.6.1 work Breakdown Structure .....	5
Figure 1.6.2 Report Outline .....	10
Figure 3 Use Case Model.....	29
Figure 4.1 Architecture Diagram .....	42
Figure 4.2-5 Domain Model .....	43
Figure 4.3-6 ERD diagram.....	46
Figure 4.4-7 Class Diagram .....	49
Figure 5.5-8 Sequence Diagram .....	50
Figure 4.8-9 User Activity Diagram .....	59
Figure 4.8-10 Admin Activity Diagram.....	60
Figure 4.9.1-11 Login State Transition Diagram.....	61
Figure 4.9.2-12 Place order State Transition Diagram .....	62
Figure 4.9.3-13 Receipt State Transition Diagram .....	63
Figure 4.9.4-14 Registration State Transition Diagram.....	64
Figure 4.9.5-15 Admin State Transition Diagram .....	65
Figure 4.9.6-16 Update Customer State Transition Diagram .....	66
Figure 4.10-17 Component Diagram .....	67
Figure 4.11-18 Deployment Diagram .....	68
Figure 5.4-19 Main Screen .....	76
Figure 5.4-20 Registration Page .....	77
Figure 21 6.1. logi.....	89
Figure 22 6.2. Register.....	90
Figure 23 6.5.1. Test case for unot testing.....	92
Figure 24 6.5.2. test case for black case .....	93
Figure 25 6.5.3. test case for hyperlink.....	93
Figure 26 6.5.4. test case foe graphics .....	93

## List of Tables

Table 2.3.3 Software Interfaces.....	20
Table 4-3.1.1 Customer Table.....	47
Table 4-3.1.2 Admin Table.....	48
Table 4.3.1.3 Product Table.....	48
Table 4.3.1.1 Customer Table.....	50
Table 4.3.1.2 Admin Table.....	51
Table 4.3.1.3 Product Table.....	51

# Chapter 1

## Introduction

## Chapter 1: Introduction

Education is very important now a day. And to find material related field of your concern is a hectic thing to find .Student are unable to find books which are referred by teacher .Kitab Ghar is a system in which we provide books data from where students can get their desired books and location of the shop and we also provide sale of books.

We compare different websites and their prices so user can check what is best and from where they can get it.We also provide recommendation of new arrivals and books of their interest.

### 1.1. Background

Education plays a very important role in human life and it's been conducted in numerous ways in which counting on the culture and placement. Throughout the past century's men have traveled from country to country for business, migration, war and education. Want to avoid ancient education was invariably tremendous and invention of web brought this idea nearer to reality.

Online Study may be a reasonably electronic exchange of study connected documents or lectures that permits students to straight forwardly receive study material and learn employing an internet program. Generally, students get irritated by visiting the book stores and go too far away places in search of books and eventually they failed to get the book.

### 1.2. Motivations and Challenges

Computers and internet have also motivated the education system. The use of internet is common so it motivates the students. The things that motivates us the most is that everything is available on internet e.g. books, audio video lectures and many more things.

The challenges we face now a days in online study are these that during the searching of our study related materials we have to survey many web pages that consumes our so much time. Even though we find our relevant data it's not complete. Sometimes a student wants to find online courses, there are many free courses available but they aren't that good. If he is able to find good ones they are either paid or trial based.

### **1.3. Goals and Objectives**

#### **Goal:**

Our main goal is to provide each and every student with the help they need in their study.

#### **Time and energy efficiency:**

It is time saving and energy efficient. Students without any physical effort can login to their accounts and can have access to any type of study material they need.

#### **Accessibility:**

Students can have access to their books, tutorials and many other things while sitting anywhere.

#### **Scope:**

The scope of our project is that students can manage their study material easily and they can have chatrooms available for each course to communicate with other students. They can also find most accurate and relevant data. A Particular Student can upload their own lectures and will also be able to help other students and the benefit they get is that they can be hired by anyone as a tutor.

This project has well accurately and perfectly designed database so that no data can be misplaced or lost. It has well designed front end that will attract the viewers and looks cool. It is a friendly user interface.

## **1.4. Literature Review/Existing Solutions**

We have many platforms available online that provide us with the books for our study. Many platform only provide the facility of reading and viewing the books. Some other platform provide the facility of downloading the books but its paid.

Some platform shows the authors or printing of the books. But we are taking multiple points from these platforms and combining them into one so that we can provide student of all the country with a source that will helps them.

We compare different websites and their prices so user can check what is best and from where they can get it. We also provide recommendation of new arrivals and books of their interest.

## **1.5. Proposed Solution**

Kitaab Ghar is a system in which we provide students the data related their books which they wants we also provide location from where they can get books and prices of different shops. And we also sell used books. People can buy and sell old books so it helps people Recycling the books and buying books in reasonable price.

## **1.6. Project Plan**

Kitaab Ghar will be completed at the end of the December. The model will be used in this project is Agile. The purpose to use the agile is customer satisfaction by early and continuous delivery of valuable software. Welcome changing requirements, even in late development.

## 1.6.1. Work Breakdown Structure

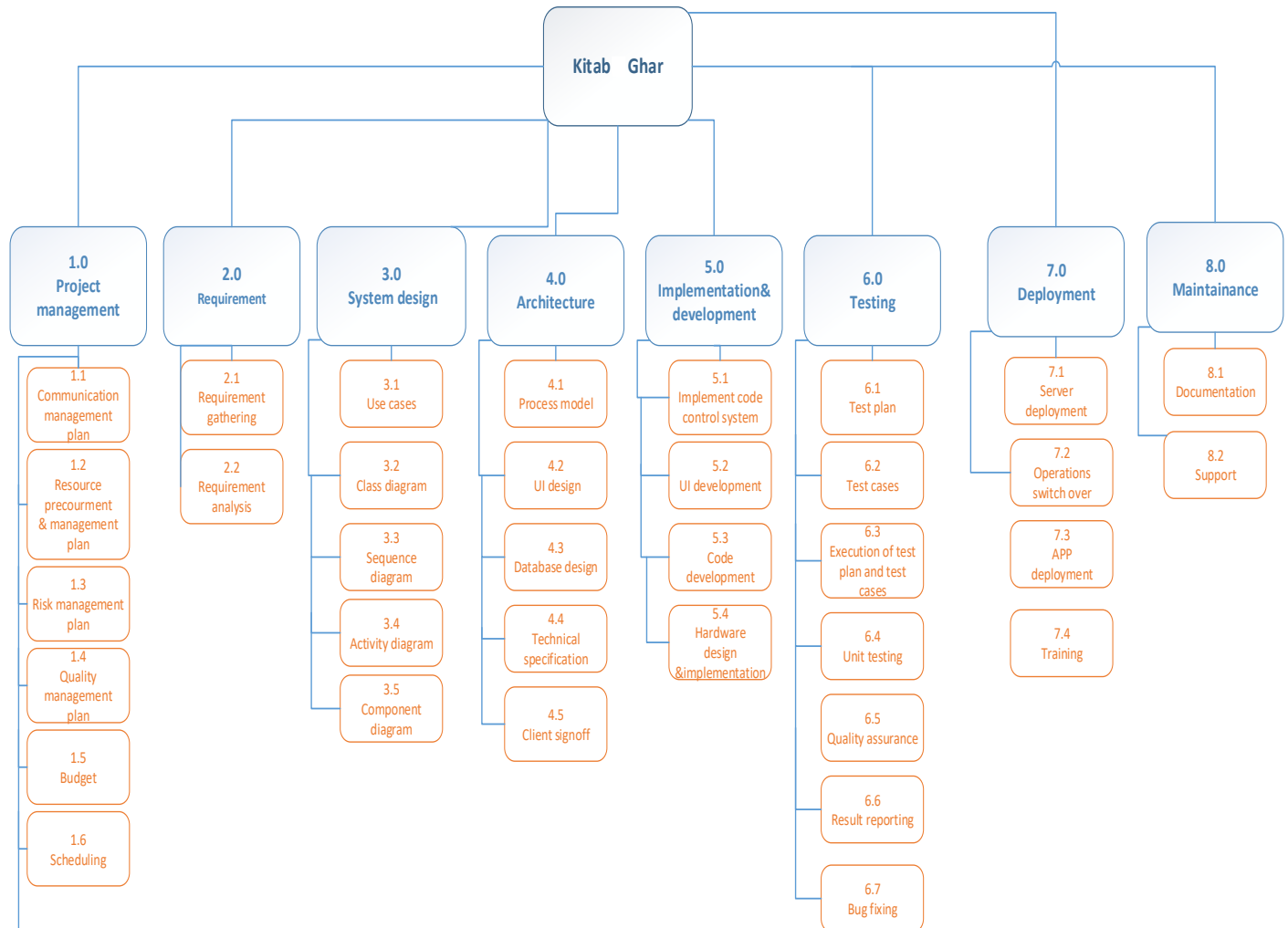


Figure 1.6.1 work Breakdown Structure

Now describe further decomposition of work breakdown structure:

### 1.5 Budget:

#### 1.5.1 Cost management

#### 1.5.2 Time management

#### 1.5.3 Scope management

### 2.1 Requirement Gathering:

**2.1.1 Brain storming**

**2.1.2 Survey on people facing problems**

**2.1.3 Communicating with people**

**2.1.4 Business requirement:**

**2.1.4.1 Evaluate Existing Processes**

**2.1.4.2 Document All Existing Users and User Roles**

**2.1.4.3 Audit Control**

**2.1.4.4 Reporting:**

**2.1.4.4.1 End User**

**2.1.4.4.2 Management**

**2.1.4.4.3 System Administrative/Technical**

**2.1.5 Third-Party interconnection**

**2.2 Requirement analysis:**

**2.2.1 Analyze the gathered requirement**

**2.2.2 Functional requirements**

**2.2.3 Define Specific UI Requirements**

**2.2.4 Define Specific Technology Requirements**

**2.2.5 Requirement document**

### **3.3 Database Design:**

#### **3.3.1** Schema design

#### **3.3.2** Normalization

#### **3.3.3** Sample data

### **4.4 Technical specification:**

#### **4.4.1** Platform specification

#### **4.4.2** Development Language specification

### **5.2 UI development:**

#### **5.2.1** Coding

### **5.3 Code development:**

#### **5.3.1** Coding

#### **5.3.2** Configuration

#### **5.3.3** Release for testing

### **5.4 Hardware design and implementation:**

#### **5.4.1** Servers preparations

#### **5.4.2** Backup policies and implementation

### **6.1 Test plan:**

#### **6.1.1** App Test plan

#### **6.1.2** Database Test plan

### **6.1.3 Backup Test plan**

## **6.2 Test cases:**

### **6.2.1 Writing test cases**

### **6.2.2 Sample Data**

## **6.7 Bug Fixing:**

### **6.6.1 Report test cases result**

### **6.6.2 Allocate team for bug deduction and fixing**

### **6.6.3 Re-testing**

## **7.4 Training:**

### **7.4.1 User training:**

#### **7.4.1.1 Make videos how to use**

#### **7.4.1.2 Give detail under the app**

### **7.4.2 System Administrator training:**

#### **7.4.2.1 Give sessions**

## **8.1 Documentation:**

### **8.1.1 End User Documentation**

### **8.1.2 Application Administration Documentation**

### **8.1.3 System Administrator Documentation**

### **8.1.4 Disaster Recovery**

## **8.2 Support:**

### 9.2.1 after delivery support plans

### 9.2.2 Resource allocation to support

### 9.2.3 Support procedures

## 1.6.2. Roles & Responsibility Matrix

### A. Introduction

- What we are going to do?
- Background
  - What currently happening in the market
- sMotivation & Challenges
  - Discussed in detail
- Goals and Objectives
  - Discussed how to achieve our goals.
- Existing System
  - Discussed what unique we are having, how could survive in a market
- Gap Analysis
  - Discussed Gap Analysis & Discussed what is our planned strategy?
- Proposed Solution
  - Discussed the solutions of the problems
- Project Plan
  - Made a project plan so we could complete our project in a time.
- Work Breakdown Structure
  - Discussed in detail
- Rules & Responsibility
  - Divided task in team members
- Gantt Chart

- Made a chart so could complete our project in a time.

### 1.6.3. Gantt Chart

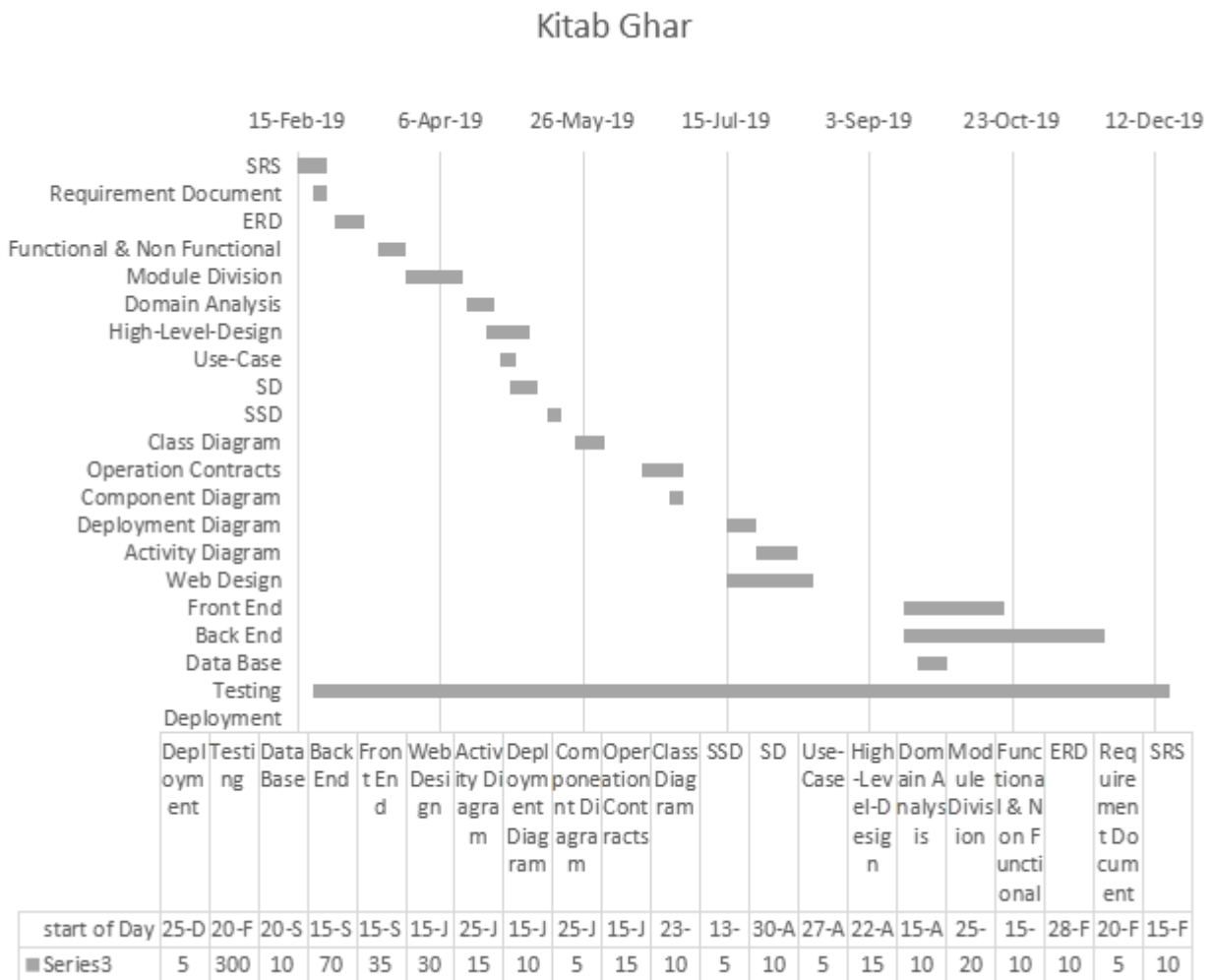


Figure 1.6.2 Report Outline

### B. Introduction

- What we are going to do?
  - Background
    - What currently happening in the market

- Motivation & Challenges
  - Discussed in detail
- Goals and Objectives
  - Discussed how to achieve our goals.
- Existing System
  - Discussed what unique we are having, how could survive in a market
- Gap Analysis
  - Discussed Gap Analysis & Discussed what is our planned strategy?
- Proposed Solution
  - Discussed the solutions of the problems
- Project Plan
  - Made a project plan so we could complete our project in a time.
- Work Breakdown Structure
  - Discussed in detail
- Rules & Responsibility
  - Divided task in team members
- Gantt Chart
  - Made a chart so could complete our project in a time.

# Chapter 2

## Software Requirement Specifications

## Chapter 2: Software Requirement Specifications

### 2.1. Introduction

#### 2.1.1. Purpose

Kitaab Ghar is a system in which we provide students the data related their books which they wants we also provide location from where they can get books and prices of different shops. And we also sell used books. People can buy and sell old books so it helps people Recycling the books and buying books in reasonable price.

#### 2.1.2. Document Conventions

This document follows calibri Format. Bold-faced text has been used to emphasize section and sub-section headings.

#### 2.1.3. Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, marketing staff, testers and documentation writers. Tools, techniques, languages and framework has described in this document which is helpful for developers.

The document includes the introduction that what is Kitaab Ghar. Problems of Customer and solutions. The scope of the project in which functionalities of system has described. The purpose of developing this Kitaab Ghar, which is helpful for the Students. The activity diagrams told about the processes involved in Kitaab Ghar.

**Overall Description** – Marketing staff must know the description of overall product to effectively advertise the product.

**System features** – Testers need an understanding of the system features to develop meaningful test cases.

## 2.1.4. Product Scope

Following are the scope of the developed systems:

### For Customer

**Online shopping:** Customer/Student can shop online by registering first.

**View Product:** Customer/Student can view product according to category and latest products/Books.

**Track Order:** Customer can track order after customer has finalized the order.

### For Admin

**Sales and purchases:** Admin can view sales and purchases from graph which will show the date-wise record.

**Category/Book Type:** Admin can add category/Book Type, view, edit and delete category/Book Type.

**Product:** Admin can add product/Book, view, edit manufacturer and delete product/Book.

**Product delivery status:** Admin can update sold and delivered product status.

### Benefits from admin view

Following are the benefit from the administrative view

- Helps to manage record of shop efficiently
- Easy to manage the whole system
- Time consuming and give better services
- Same application can be used for different online shopping purpose

### 2.1.5. References

List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

## 2.2. Overall Description

### 2.2.1. Product Perspective

Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

### 2.2.2. Product Functions

Following are the functions which our application/product will perform:

**Create Account:** The creation of new account needs some parameters like cnic, name, address and password.

**Login:** Customer and admin can login into their pages by entering valid username and password.

**View-Products:** Customer can view products on the page and all the description about that item.

**Place-Order:** Customer can place order by adding items to the cart and then can finalize the order.

**Track-Order:** Customer can track order by the given id.

**Add Book Price:** Admin will add Book Price to the Book manually.

**Manage Accounts:** Admin will manage accounts like Shop worker detail, daily sales and purchases and pending payments.

**Add Customer:** Admin can add customer through some details like cnic, name, contact no, date and address.

**Search Customer:** If an order of customer is ready and admin has to push notification. Admin will search that customer.

**Add-Items:** Admin can add items through their id's some description and weight.

**Push Notification:** When the product is ready, admin will send notification to customer.

**Make-Purchases:** Admin when purchase items/Book from vendors and customers.

**Update Customer:** admin can update customer when customer will return or buy another product.

**Logout:** Admin and customer will logout when they have done their tasks.

### **2.2.3. User Classes and Characteristics**

There are 3 users which will use this system:

- 1) Customer
- 2) Admin

**Customer:**

The customers are not expected to have a high educational and proficiency level or technical expertise. Hence, the user interface is user-friendly and available in English which is overall understandable.

**Admin:**

The admin will aware of all the system functions. The admin will take a tour before using this application

**2.2.4. Operating Environment**

Operating environment for the Kitaab Ghar system is as listed below:

- Centralized database
- Client/server system
- Operating system: Windows
- Database: SQL + database
- Software Platform: PHP / CodeIgniter

**2.2.5. Design and Implementation Constraints**

All the records will be saved into the database instead of writing on notebooks.

For database, mysql will be used.

The application will be available 24/7.

Customer must login before placing order.

### 2.2.6. User Documentation

The tutorial in which all the functions of Kitaab Ghar will be performed, given to admin. The tutorial will be in video format. From this video admin will understand their tasks and their limitations.

### 2.2.7. Assumptions and Dependencies

We are assuming that the hardware designed will work correctly with the third-party operating system. Some assumptions are as follows:

- Admin and customer must be connected to internet
- Before online shopping, customer will sign up
- Proper browsers should be installed
- Each customer must have an id and password for login

## 2.3. External Interface Requirements

### 2.3.1. User Interfaces

**View-Items:** Customer can view items on the home page and by going into specific category. Description of product will be available in this tab.

**Place-Order:**when customer will login, customer can place order from the given product, upload design and custom design.

**Track-Order:** When customer will login to his account, there will be a button of tracking. By entering the order id, status of the order will be displayed.

**Sales:**Admin can view total sales and purchases on daily basis. Only admin can access the database. Admin can view the sales and purchases on the graph where numbers are also displayed. Admin can view the total orders placed and orders that have been delivered.

**Send-Notification:**When the product will be ready, the admin will press the push notification button, which will send notification to that customer.

**Inbox:**The admin will receive the uploaded images in the inbox tab, and then customer and admin can chat about design, shape and weight.

- Front-end software: HTML + CSS + JS
- Back-end software: phpmyadmin + mysql + PHP

### **2.3.2. Hardware Interfaces**

The hardware interfaces involves:

- Windows.
- A browser which supports HTML & Javascript.
- For invoice, printer is used with the computer.

### 2.3.3. Software Interfaces

Following are the software used for the Kitaab Ghar application:

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the record of sales and purchases we used the phpmyadmin and mysql database.
PHP / Codeignitor	To implement the project we have PHP and its framework codeignitor.
Bootstrap	The framework and classes of HTML and CSS are involved in its framework called as bootstrap.

*Table 2.3.3 Software Interfaces*

### 2.3.4. Communications Interfaces

First the customer when they asked a question for anything related, admin can send reply message to customer in chat box.

After receiving some query, admin will reply that customer by an e-mail. Chat can be continued on e-mails but not on our system. The application will only use an internet connection and hence require HTTP for transmission of data for communication.

## 2.4. System Features

Following are the functions which our application/product will perform:

**Create Account:** The creation of new account needs some parameters like cnic, name, address and password.

**Login:** Customer and admin can login into their pages by entering valid username and password.

**View-Products:** Customer can view products on the page and all the description about that item.

**Place-Order:** Customer can place order by adding items to the cart and then can finalize the order.

**Track-Order:** Customer can track order by the given id.

**Manage Accounts:** Admin will manage accounts like Shop worker details, daily sales and purchases and pending payments.

**Add Customer:** Admin can add customer through some details like cnic, name, contact no, date and address.

**Search Customer:** If an order of customer is ready and admin has to push notification. Admin will search that customer.

**Add-Items:** Admin can add items through their id's some description.

**Push Notification:** When the product is ready, admin will send notification to customer.

**Make-Purchases:** Admin when purchase items/Books from vendors and customers.

**Update Customer:** admin can update customer when customer will return or buy another product.

**Logout:** Admin and customer will logout when they have done their tasks.

## **2.4.1. SF1- Create Account**

### **2.4.1.1. Description and Priority**

The customer will create an account before purchase book. The fields like cnic, name, address and some password must be entered.

The priority is high for this system function.

### **2.4.1.2. Stimulus/Response Sequences**

When create account button is triggered, it will take you on login page. On login page customer will provide cnic and password and can more proceed with shopping.

### **2.4.1.3. Functional Requirements**

REQ-SF1-1: Customer must be connected to internet before online shopping.

REQ-SF1-2: For account opening, customer must need valid cnic, contact no and address. All the fields must be entered.

REQ-SF1-3: When the create account button will be pressed, the confirmation message will send to customer.

## **2.4.2. SF2- Login**

### **2.4.2.1. Description and Priority**

In login function, customer will provide cnic and password. After create an account page, this page will be displayed. After login page the customer will see the categories and can add items to the cart.

The priority of login is high.

#### **2.4.2.2. Stimulus/Response Sequence**

When a customer will successfully login. The customer can shop from categories like upload design, custom design and available designs.

#### **2.4.2.3. Functional Requirements**

REQ-SF2-1: customer must create an account.

REQ-SF2-2: Customer must remember cnic and password.

REQ-SF2-3: Customer must verify an account by message.

### **2.4.3. SF3- Place Order**

#### **2.4.3.1. Description and Priority**

In place order customer can buy things. When the customer has added items to the cart, customer can order the products. After order, the customer will be given an order\_id.

The priority of this system function is medium.

#### **2.4.3.2. Stimulus/Response Sequences**

After placing order, customer will receive confirmation message that your order has been placed. Now, the customer can use the track order facility in login session.

#### **2.4.3.3. Functional Requirements**

REQ-SF3-1: When items are added into the cart, the customer can checkout.

REQ-SF3-2: Customer has to choose at least one option from categories.

REQ-SF3-3: Customer will receive a confirmation message after order.

## 2.4.4. SF4- Track Order

### 2.4.4.1. Description and Priority

Customer can track order by the given order\_id. The status of the order will be displayed to the customer. The priority of track order is low in system functions.

### 2.4.4.2. Stimulus/Response Sequences

When a customer will track order. The system will ask to enter order\_id. If the id is valid, the status will be displayed. Otherwise error will be displayed.

### 2.4.4.3. Functional Requirements

REQ-SF4-1: Customer must place an order before using this track order facility.

REQ-SF4-2: order\_id is very important. Customer must remember that order\_id.

REQ-SF4-3: After login page, track order will be displayed. So, customer must login.

## 2.4.5. SF5-Add Customer

### 2.4.5.1. Description and Priority

If a customer visit physical shop instead of online. The customer will be added by the admin. All the fields in create account then filled by the admin.

The priority of this system function is high.

### 2.4.5.2. Stimulus/Response Sequences

Admin can search the customer only if they are registered into the database. After adding customer, admin can give username and password to the customer.

### 2.4.5.3. Functional Requirements

REQ-SF5-1: Customer must provide cnic and valid information.

REQ-SF5-2: Admin will ask customer which product/Book customer wants to buy.

REQ-SF5-3: Admin will add the items into the cart.

## **2.4.6. SF6-Search Customer**

### **2.4.6.1. Description and Priority**

When the customer has finalized the order. Now admin will search for the specific customer to push notification. Admin will search customer by the `cus_id`. For update customer, admin will also search the customer.

The priority of this system feature is medium.

### **2.4.6.2. Stimulus/Response Sequences**

The information of particular customer will be displayed. The information will contain the `order_id`, name, contact no, address and the items in his car.

### **2.4.6.3. Functional Requirements**

REQ-SF6-1: Customer must be present in the list.

REQ-SF6-2: Admin must know the `cus_id` for which he will search.

REQ-SF6-3: The product of that customer must be ready if admin wants to push notification.

## **2.4.7. SF7- Manage Accounts**

### **2.4.7.1. Description and Priority**

Admin will login to his account. Admin can view total sales and purchases on the graph. Admin can view the pending payments to vendors. Admin can view the salaries of the Shop worker.

The priority of Accounts management is high.

### **2.4.7.2. Stimulus/Response Sequences**

The webpage will open after login. Different modules like Shop Worker, total sales and purchases, Books buy from vendor and its payments and amount pending. All these are included in accounts management.

### **2.4.7.3. Functional Requirements**

REQ-SF7-1: Admin has logged into an account.

## **2.5. Other Nonfunctional Requirements**

### **2.5.1. Performance Requirements**

As the system is the service providing application. An application will provide its services (shopping) all the time. The system Kitaab Ghar will be running on server 24/7. Customer can shop anytime therefore the system will remain on.

### **2.5.2. Safety Requirements**

Unauthorized person cannot create account as a customer has to verify an email before logging in into the page.

### **2.5.3. Security Requirements**

The overall system is operating by the admin. No other person has access to the system. Security increased when one person has full power.

### **2.5.4. Software Quality Attributes**

#### **Reliability**

The system can be used by different customers at the same time. Increase in customers will not affect the system speed.

#### **Availability**

The system is available every time 24/7.

### **Portability**

Same customer can login from different computer.

### **Robustness**

The system will not crash with the burden of user and more traffic.

## **2.5.5. Business Rules**

Following are the users which will use the system:

- Admin
- Customer

**Admin** has the full authorization of controlling the system and access the database.

**Customer** can only place order and has no direct relation with the system's database.

## **2.6. Other Requirements**

At the point undertaking the principal matter to build up is to survey the credibility of a task or item. It is essentially utilized as a measure how down to earth or gainful the advancement of a product system will be to you (or association). This movement repeats for the duration of the life cycle.

There are numerous kinds of possibilities:

- Technical
- Economic
- Specification
- Information
- Motivational
- Legal and Ethical

# Chapter 3

## Use Case Analysis

# Chapter 3: System Analysis

The system boundary is described in this chapter. The actors and the functionalities they are performing are represented in system boundary. The attributes are fully explained in fully dressed use cases. Fully dressed use cases include pre-condition and post-condition.

## 3.1. Use Case Model

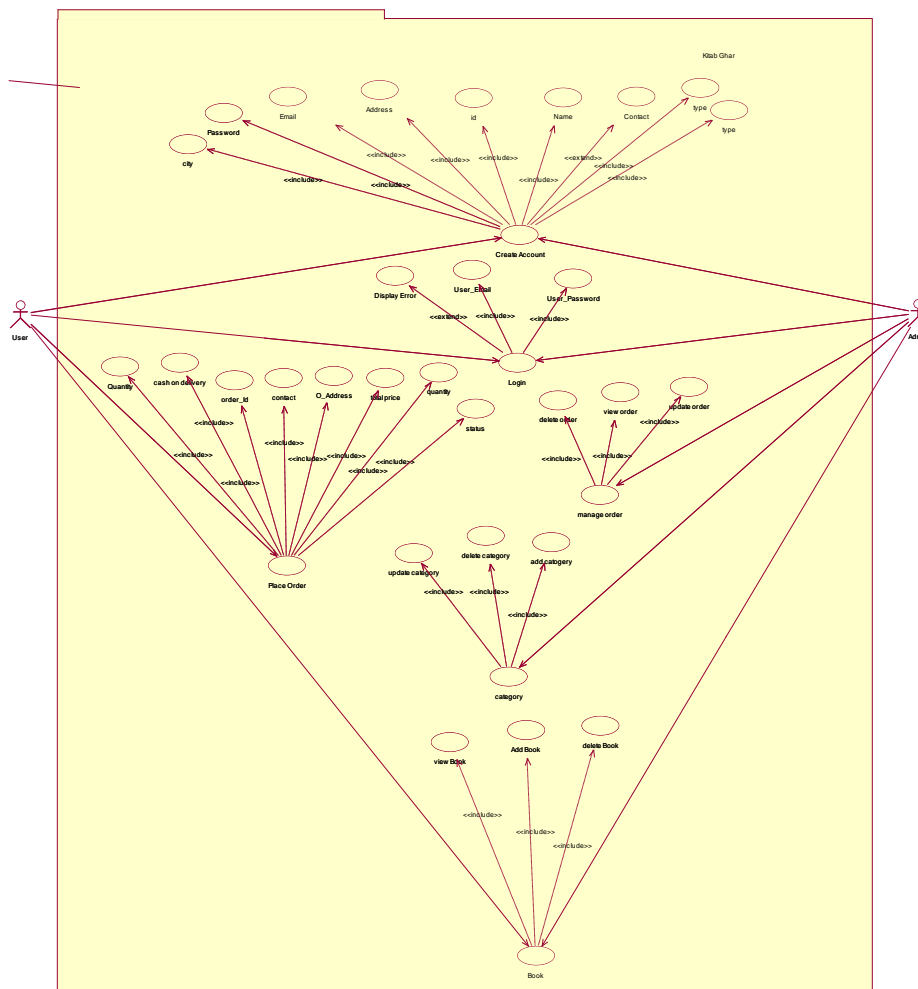


Figure 3 Use Case Model

## 3.2. Fully Dressed Use Cases

**Use Case 1.** Create Account

**Primary Actor:**unregistered customer

**Scope:**New customer able to sign up

**Level:** customer register himself

**Stake Holder:** Admin

**Pre-conditions:**None

**Post-condition:**

Successfully Create account for new customer

Customer give the online order

**Main Success Scenario:**

After create account customer give order online.

Customer search the product

Customer track their order

**Extension:**

Customer passes a wrong command and after realizing he can go back.

Admin not receive the request for create account of new user

**Special Requirement:**

Security

## **Use Case 2. Add Customer**

**Scope:** Admin able to add customer

**Level:** Admin add customer

**Primary Actor:** Admin

**Stake holder:** Admin

**Pre-conditions:** Customer information must be authorized.

### **Main Success Scenario:**

Function is start when admin want add customer information

Admin add customer by its id

Admin add customer by its address

Admin add customer by its phone number

Admin add customer by its name

### **Extension:**

If the admin adds the customer information with same details then the system show message  
“Record already exist”

### **Special Requirement:**

Efficiency

## **Use Case 3. Place Order**

**Scope:** customer able to give order online

**Level:** customer place order

**Primary Actor:** Customer

**Stake holder:** Customer

**Pre-conditions:**

Customer must be login their account.

System is connected to power source, display is turned on and system have to collect inputs.

**Post-condition:**

Giving order successfully

Receipt generated with total amount.

**Main success scenario:**

Function is start when user want give order online

User selects language

User confirm order

**Extension:**

Customer passes a wrong command and after realizing he can go back.

#### **Use Case 4.** Tracking Order

**Scope:** customer able to track order

**Level:** customer tracking their order

**Primary Actor:** customer

**Description:** After place order customer also can tracking order of processing and after complete the order customer received the message or call by admin

**Pre-conditions:** order must be place successfully and make payment paid

**Main Success scenario:**

Function is start when user give order successfully and after that they want track their order information

Customer check order processing time

Customer received message after complete order

Message sends by admin or auto generated

**Extension:** If the customer can not want tracking of own order than he/she visit our shop

**Post-condition:**

Show message order is completed successfully

**Use Case 5.** Manage Accounts

**Scope:** Admin manage their all accounts

**Level:** admin manage accounts

**Primary Actor:** Admin

**Description:** Admin manage the all accounts information. Admin also check the liabilities and daily income.

**Pre-conditions:** Admin must have an authorized access for use these systems.

**Main Success Scenario:**

Function is start when admin wants manage accounts and check the liabilities

Admin check the liabilities

Admin check the daily income

Admin check the pending payments

Admin pay the Shop worker pay

Admin pay to wholesale dealer

**Post-condition:**

Admin know the all information about their Shop worker detail

Admin also know the daily income

Admin also know the pending payments

**Use Case 6. Make Purchase**

**Scope:** admin make purchase

**Level:** make purchase

**Primary Actor:** Admin

**Description:** Admin manage the purchase record. Admin buy a Books from customer and any vendor. And give a payment to customer or vendor.

**Pre-conditions:**

Admin must know about Book Price.

**Main Success Scenario:**

Function is start when any vendor come to shop for sale the Books than admin make a purchase from this vendor and buy a Books.

Admin manage purchase record

Admin purchase Books from any customer

Admin give payment to customer after buying Books

**Extension:**

If admin don't know about the Books price than they not make a purchase of Books

**Post-condition:**

Admin give digital receipt to customer after buying Books or stationary

Admin save customer information for security

**Use Case 7. Login**

**Scope:** customer login their account

**Level:** customer login account

**Primary Actor:** Customer

**Description:** After creating successful account customer can login their account with authorized information.

**Pre-conditions:** Customer must be entering a correct information for login their account.

**Main Success Scenario:**

Faculty of CS&IT, The Superior College Lahore, Pakistan

Function is start when customer wants login their account

Customer enter their name or mail

Customer enter their password for login their account

**Extension:** Customer passes a wrong command and after realizing he can go back.

**Post-condition:**

Customer account opened

Customer used the system

Customer place the order online

**Use Case 8.** Update Customer record

**Scope:** admin update the customer record

**Level:** admin update record

**Primary Actor:** Admin

**Description:** Admin update their customer record if anyone customer changing personal information than admin update it and save in our database

**Pre-conditions:** To update customer record customer must be have already account or detailed in our system if they have than admin update it

**Main Success Scenario:**

Function is start when admin want update the customer record

Admin update their customer record

Record update according to situation

**Post-condition:**

After update customer information admin use this correct information

**Use Case 9.** Push Notification

**Scope:** Admin push the notification to customer

**Level:** Admin push notification

**Primary Actor:** Admin

**Description:** Admin push the notification for customer when their order complete.

**Pre-conditions:**

Admin enter their information for login their account

Must be complete the order of customer

**Main Success Scenario:**

Function is start when customer order is complete and Admin want send message to customer

Admin login their account

Complete the order of customer

**Extension:**

If Admin send message to wrong contact

If Admin not send the notification for completion order to customer.

**Post-condition:**

After complete the order send the notification by message or mail to customer for reminder their completion order.

**Use Case 10.** Login

**Scope:** Admin login their account

**Level:** admin login account

**Primary Actor:** Admin

**Description:** After creating successful account admin can login their account with authorized information.

**Pre-conditions:**

Admin must be registered their information.

Admin entering a correct information for login their account.

**Main Success Scenario:**

Function is start when admin wants login their account

Customer enter their name or mail

Customer enter their password for login their account

**Extension:** Admin passes a wrong command and after realizing he can go back.

**Post-condition:**

Admin account opened

Admin used the system

Admin now add and update customer information

Admin can manage accounts

**Use Case 11.** Search Customer

**Scope:** admin search customer record

**Level:** admin search customer

**Primary Actor:** Admin

**Description:** Admin search customer record by their ID.

**Pre-conditions:**

Admin must be login their account

Customer information added already

**Main Success Scenario:**

Function is start when admin want search the customer record for check their account is cleared

Admin enter their information for login their account

Admin search the customer record by using their ID

**Extension:** Admin passes a wrong ID and after realizing he can go back.

**Post-condition:**

Admin account opened

Admin used the system

Admin search the customer record

## **Use Case 12. Change Password**

**Scope:** customer want change password

**Level:** customer change the password

**Primary Actor:** Customer

**Description:** customer request for change the password.

### **Pre-conditions:**

Send confirmation message to customer for change password

### **Main Success Scenario:**

Function is start when the customer wants change their account password for security

When customer wants to change password, they send request for change

Receive notification for change password

### **Extension:**

If customer don't know own old password than they not change password

This use case used for generate the request for changing password

### **Post-condition:**

Customer send request for change password

Change password successfully.

# Chapter 4

## System Design

## Chapter 4: System Design

This chapter is all about the flow of operations. There is a graphical representation of our system. It consists of all the activities which are happening in system. The relationships with entities are enlisted. The sequence of functionalities and processes are explained. The state of admin and customer has described. The deployment of overall system has shown in the chapter.

### 4.1. Architecture Diagram

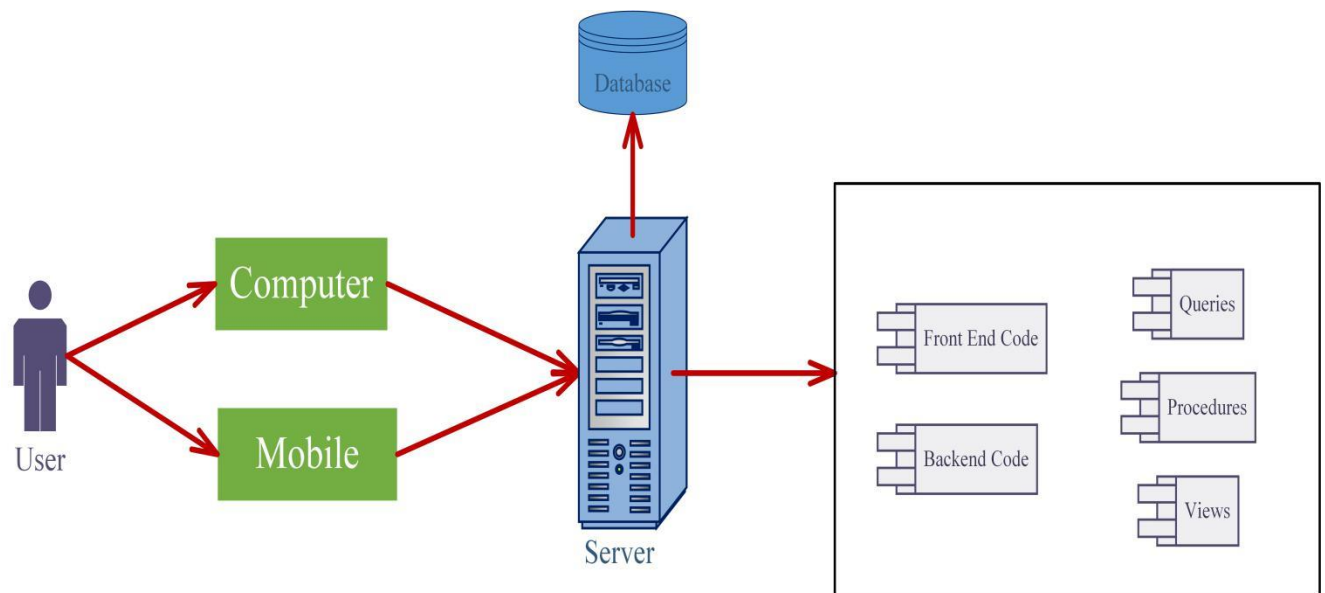


Figure 4.1 Architecture Diagram

## 4.2. Domain Model

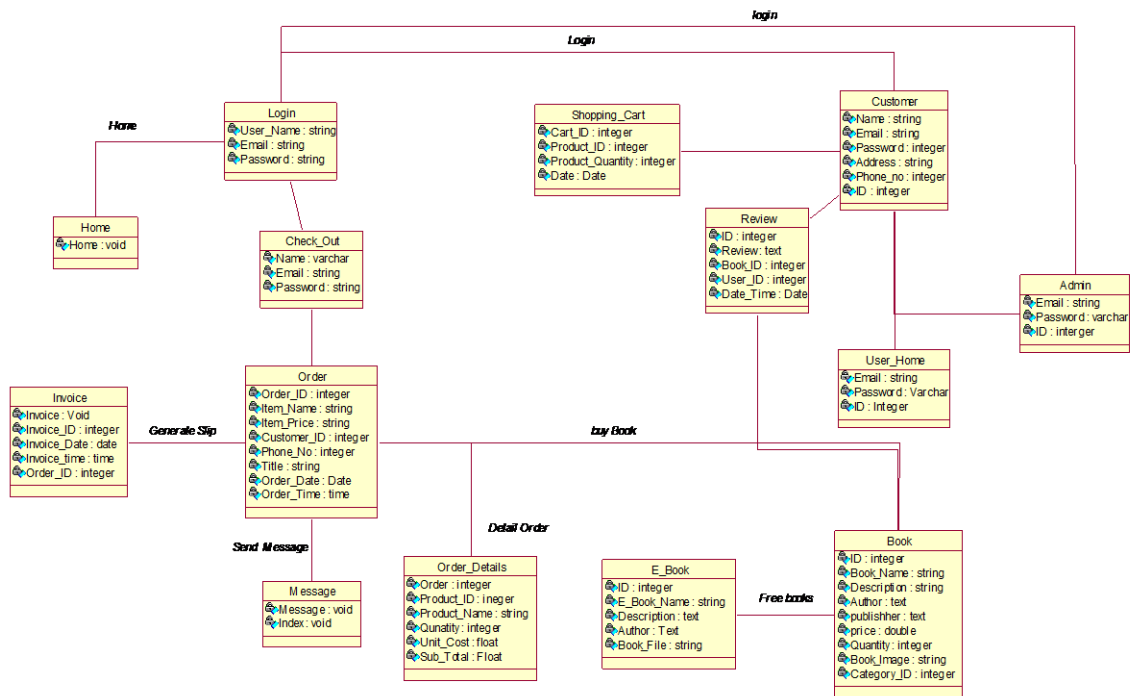


Figure 4.2-5 Domain Model

### 4.3.1. Data Dictionary

#### 4.3.1.1. Customer table:

<b>Entity Name</b>	Customer			
<b>Entity Description</b>	It stores the information of customer			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
Name	Stores name of the customer	Varchar	Not null	
Id/cnic	Stores cnic of customer	Number	Primary key, Not null	
Address	Stores address of customer	VARCHAR	Not null	
contactNo	Stores contact detail of customer	Number	Not null	
Password	Stores password of customer	Password	Not null	
Email	Stores Email of customer	Email	Not null	

Table 4-3.1.1 Customer Table

**4.3.1.2. Admin table:**

<b>Entity Name</b>	Admin			
<b>Entity Description</b>	Stores all the information about admin and login details			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
username	Username of admin	INT	Not Null	
Password	Stores password of admin	VARCHAR	Not Null	

*Table 4-3.1.2 Admin Table***4.3.1.3. Product table:**

<b>Entity Name</b>	Product			
<b>Entity Description</b>	It stores the description of product			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
Id	Every product has specific id	Number	Primary key, Not null	
Name	Name of product	VARCHAR	Not null	
Price	The selling price for product	Number	Not null	
Description	Stores value points of product	VARCHAR	Not null	

*Table 4.3.1.3 Product table*

### 4.3. Entity Relationship Diagram with data dictionary

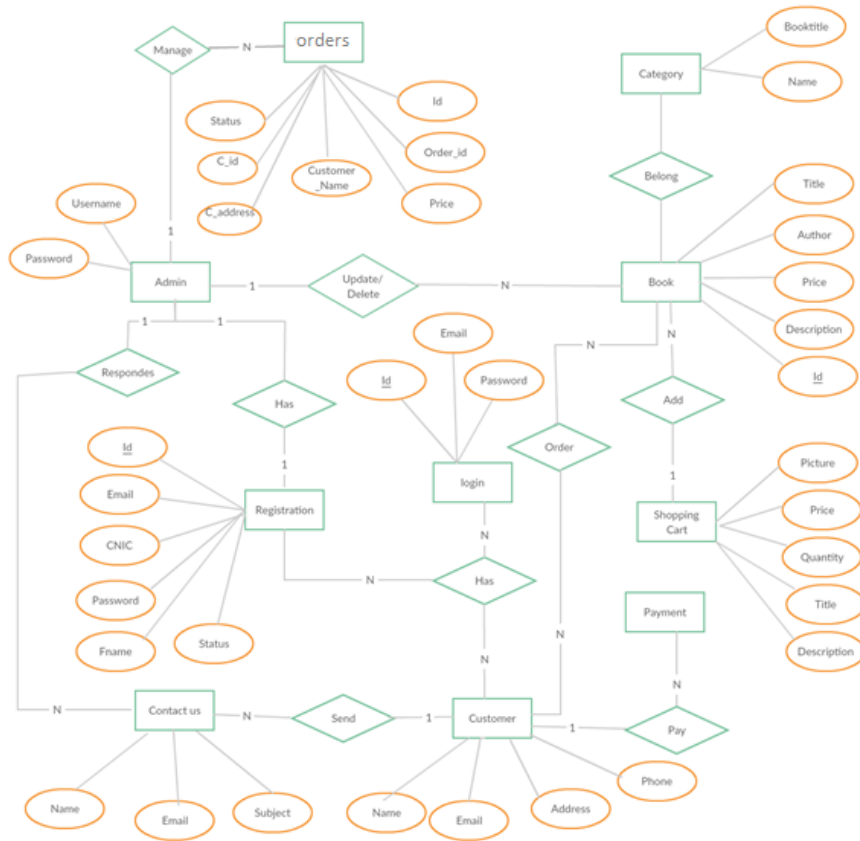


Figure 4.3-6 ERD diagram

### 4.3.1. Data Dictionary

#### 4.3.1.1. Customer table:

<b>Entity Name</b>	Customer			
<b>Entity Description</b>	It stores the information of customer			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
Name	Stores name of the customer	Varchar	Not null	
Id/cnic	Stores cnic of customer	Number	Primary key, Not null	
Address	Stores address of customer	VARCHAR	Not null	
contactNo	Stores contact detail of customer	Number	Not null	
Password	Stores password of customer	Password	Not null	
Email	Stores Email of customer	Email	Not null	

*Table 4.3.1.1 Customer Table*

**4.3.1.2. Admin table:**

<b>Entity Name</b>	Admin			
<b>Entity Description</b>	Stores all the information about admin and login details			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
username	Username of admin	INT	Not Null	
Password	Stores password of admin	VARCHAR	Not Null	

*Table 4.3.1.2 Admin table***4.3.1.3. Product table:**

<b>Entity Name</b>	Product			
<b>Entity Description</b>	It stores the description of product			
<b>Column Name</b>	<b>Column Description</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Reference</b>
Id	Every product has specific id	Number	Primary key,Not null	
Name	Name of product	VARCHAR	Not null	
Price	The selling price for product	Number	Not null	
description	Stores value points of product	VARCHAR	Not null	

*Table 4.3.1.3 Product Table*

## 4.4. Class Diagram

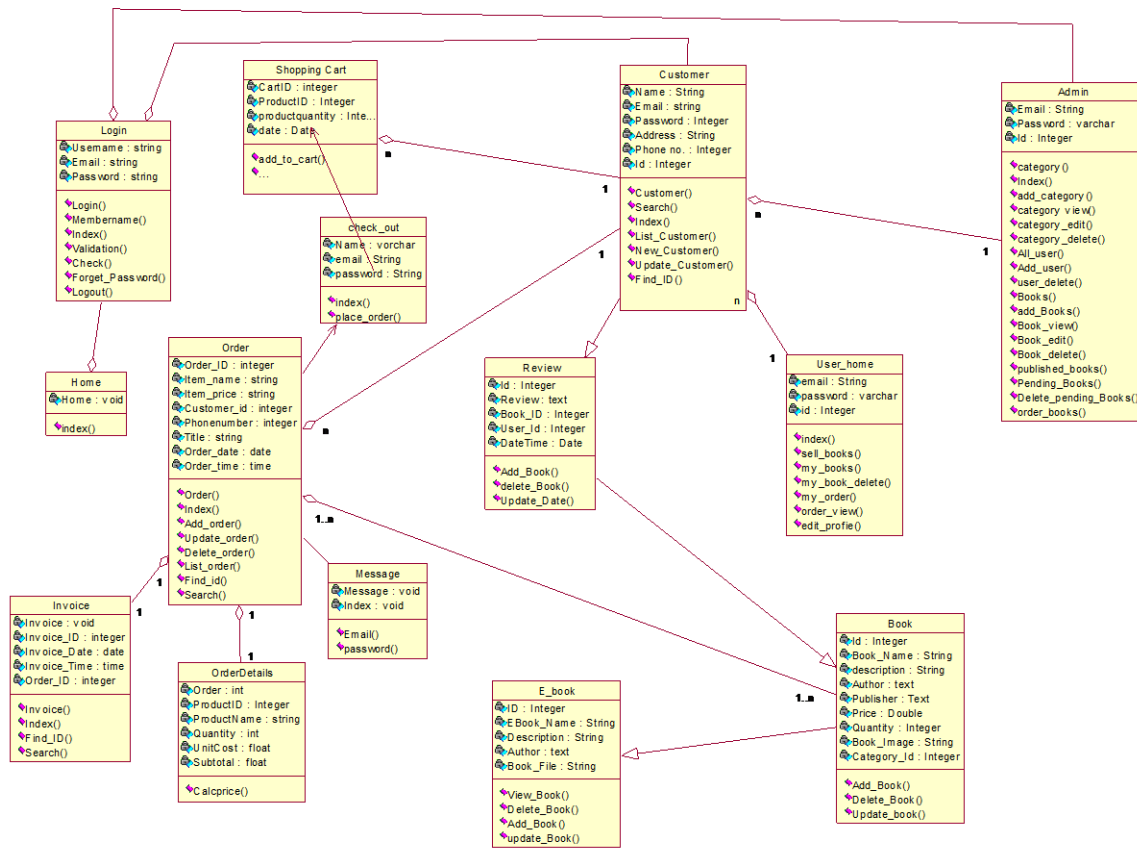


Figure 4.4-7 Class Diagram

## 4.5. `Sequence / Collaboration Diagram

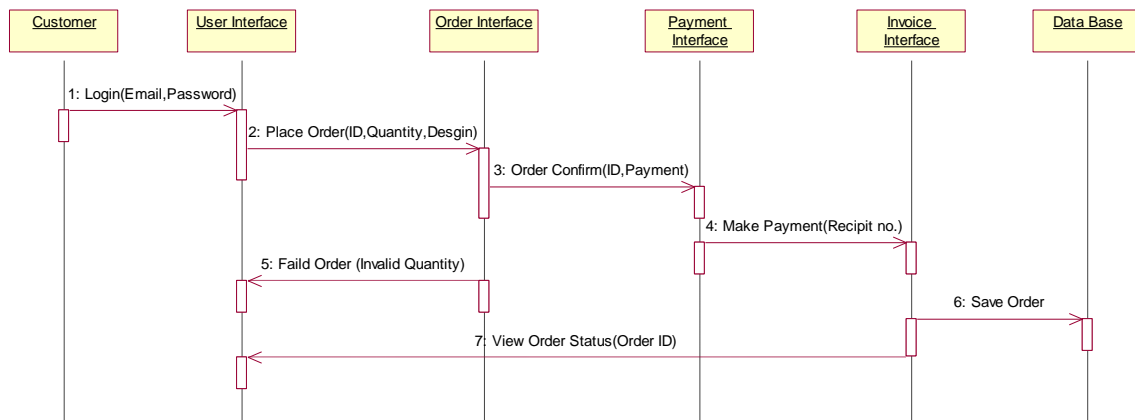


Figure 5.5-8 Sequence Diagram

## 4.6. Operation contracts

### Operation 1:

Registration request (name, phone number, email, address)

### Cross References:

Use case: registration

Scenario: registration

### Precondition:

None

### Post condition:

- A new registration Request instance “regreq” is created (instance creation)
- Regreq associated with that user who send request for registration

**Operation 2:**

Registration (email, Password)

**Cross References:**

Use case: registration

Scenario: registration

**Precondition:**

Must be registration request instance regreq is generated

**Post condition:**

- A registration instance “registration” is created (instance creation)
- New account instance “a” is created

**Operation 3:**

Admin Login

**Cross References:**

Use case: add customer

Scenario: add customer

**Precondition:**

The admin must have username and password before login.

**Post condition:**

- A login instance “login” is created(instance creation)
- Login. Username became username, login. password became password (attribute modification)

- login was associated with get add customer (association formed)

**Operation 4:**

Update customer info

**Cross References:**

Use case: add customer

Scenario: add customer

**Precondition:**

The admin must have username and password before login.

An account instance “a” is created

Instance “a” association with admin

**Post condition:**

- All attributes of instance “a” is show

**Operation 5:**

Customer login

**Cross References:**

Use case: Place order

Scenario: Place Order

**Precondition:**

The customer must have enter the valid mail and password for login.

**Post condition:**

- An order instance “o” is created(instance creation)
- o. id became id , o. name became name, o. quantity became quantity (attribute modification)
- “o” was associated with internal order system (association formed)

#### **Operation 6:**

Admin Login

#### **Cross References:**

Use case: Manage Accounts

Scenario: Manage Accounts

#### **Precondition:**

The Admin must have entered the valid mail and password for login

#### **Post condition:**

- A Manage Accounts instance “m” is created (instance creation)
- m. daily incomes became daily income. pending payments became pending payments, check balance became check balance (attribute modification)
- “m” was associated with search customer (association formed)

#### **Operation 8:**

Admin Login

#### **Cross References:**

Use case: Make Purchase

Scenario: Make purchase

**Precondition:**

The admin must have entered the valid mail and password for login

**Post condition:**

- A make purchase instance “p” is created (instance creation)
- P. customer became customer. p. Vendor became vendor (attribute modification)
- “p” was associated with make purchase (association formed)

**Operation 9:**

Admin Logout (adminID)

**Cross References:**

None

**Precondition:**

An account instance “a” is created.

Admin association with “a”

**Post condition:**

- Account instance “a” with broken login instance “i”
- Login instance “l” destroyed

## 4.7. Operation contracts

**Operation 1:**

Registration request (name, phone number, email, address)

**Cross References:**

Use case: registration

Scenario: registration

**Precondition:**

None

**Post condition:**

- A new registration Request instance “regreq” is created (instance creation)
- Regreq associated with that user who send request for registration

**Operation 2:**

Registration (email, Password)

**Cross References:**

Use case: registration

Scenario: registration

**Precondition:**

Must be registration request instance regreq is generated

**Post condition:**

- A registration instance “registration” is created (instance creation)
- New account instance “a” is created

**Operation 3:**

Admin Login

**Cross References:**

Use case: add customer

Scenario: add customer

**Precondition:**

The admin must have username and password before login.

**Post condition:**

- A login instance “login” is created(instance creation)
- Login. username became username, login. password became password (attribute modification)
- login was associated with get add customer (association formed)

**Operation 4:**

Update customer info

**Cross References:**

Use case: add customer

Scenario: add customer

**Precondition:**

The admin must have username and password before login.

An account instance “a” is created

Instance “a” association with admin

**Post condition:**

- All attributes of instance “a” is show

**Operation 5:**

Customer login

**Cross References:**

Use case: Place order

Scenario: Place Order

**Precondition:**

The customer must have enter the valid mail and password for login.

**Post condition:**

- An order instance “o” is created(instance creation)
- o. id became id , o. name became name, o. quantity became quantity (attribute modification)
- “o” was associated with internal order system (association formed)

**Operation 6:**

Admin Login

**Cross References:**

Use case: Manage Accounts

Scenario: Manage Accounts

**Precondition:**

The Admin must have entered the valid mail and password for login

**Post condition:**

- A Manage Accounts instance “m” is created (instance creation)
- m. daily incomes became daily income. pending payments became pending payments, check balance became check balance (attribute modification)

- “m” was associated with search customer (association formed)

**Operation 8:**

Admin Login

**Cross References:**

Use case: Make Purchase

Scenario: Make purchase

**Precondition:**

The admin must have entered the valid mail and password for login

**Post condition:**

- A make purchase instance “p” is created (instance creation)
- P. customer became customer. p. Vendor became vendor (attribute modification)
- “p” was associated with make purchase (association formed)

**Operation 9:**

Admin Logout (adminID)

**Cross References:**

None

**Precondition:**

An account instance “a” is created.

Admin association with “a”

**Post condition:**

- Account instance “a” with broken login instance “i”

- Login instance "I" destroyed

## 4.8. Activity Diagram

User

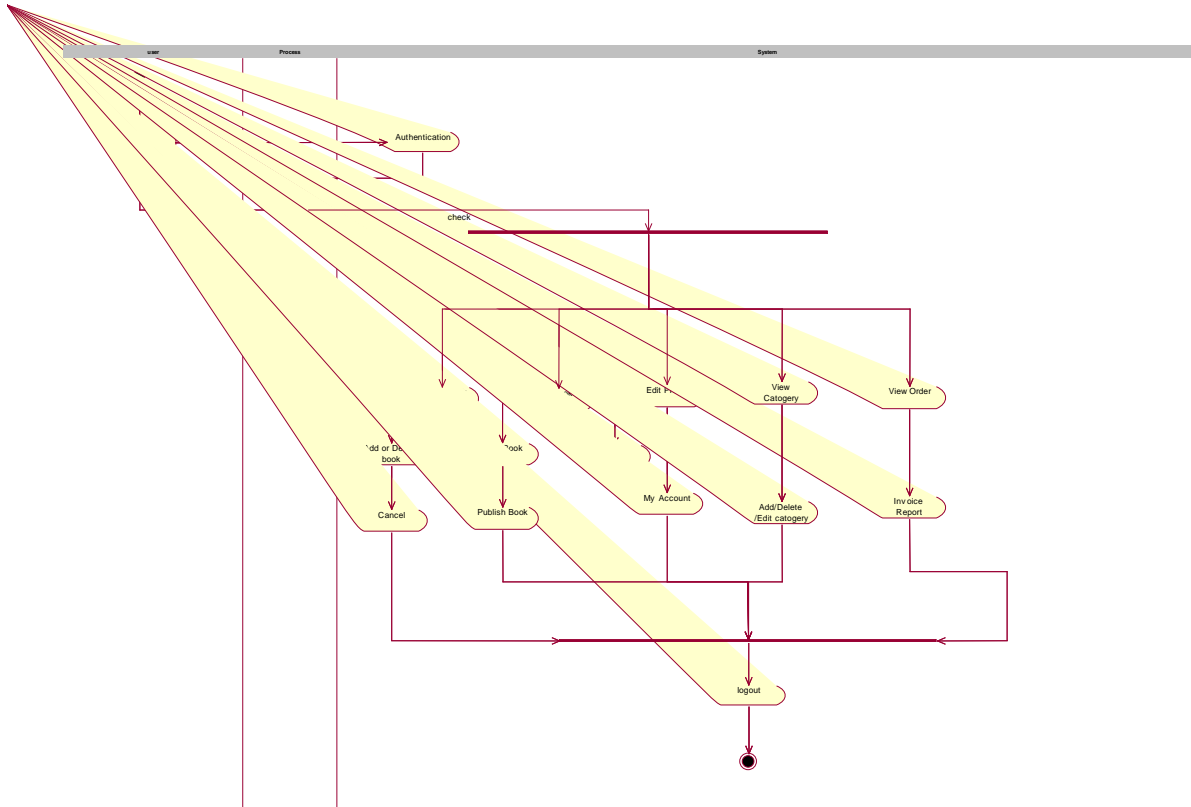


Figure 4.8-9 User Activity Diagram

# Admin

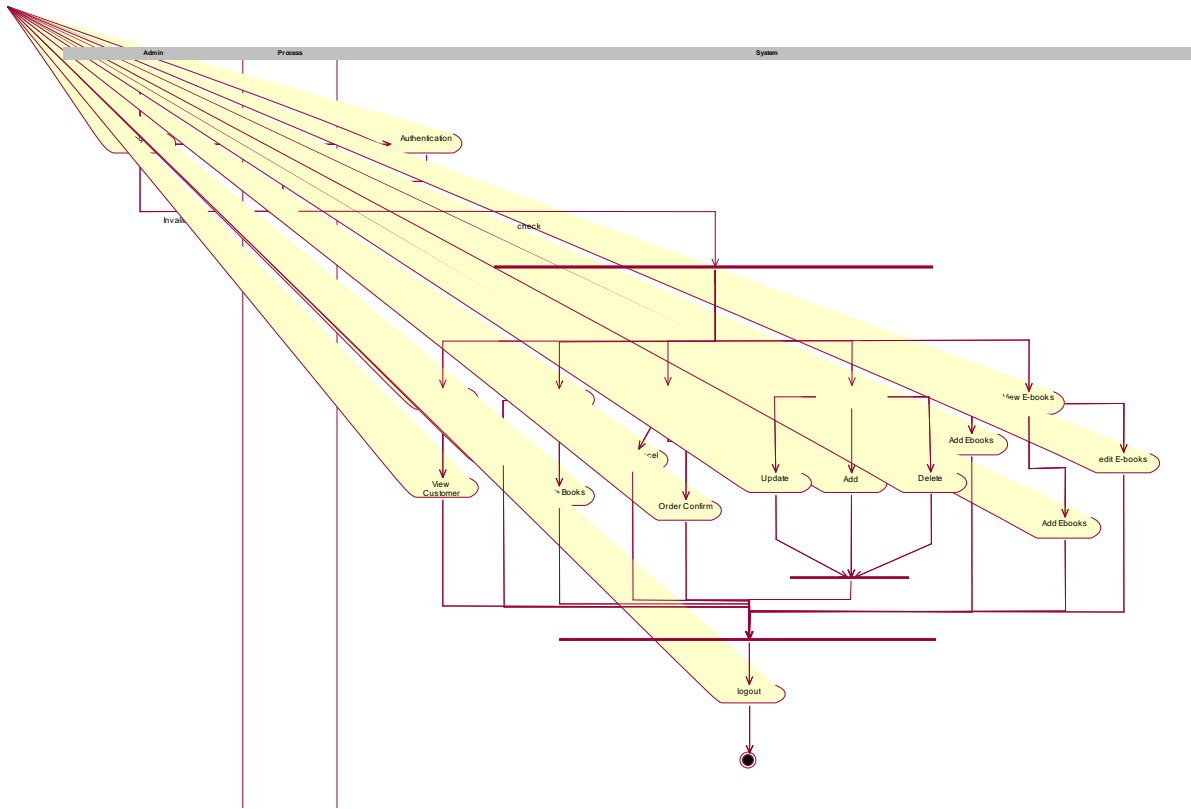


Figure 4.8-10 Admin Activity Diagram

## 4.9. State Transition Diagram

### 4.9.1. For Login

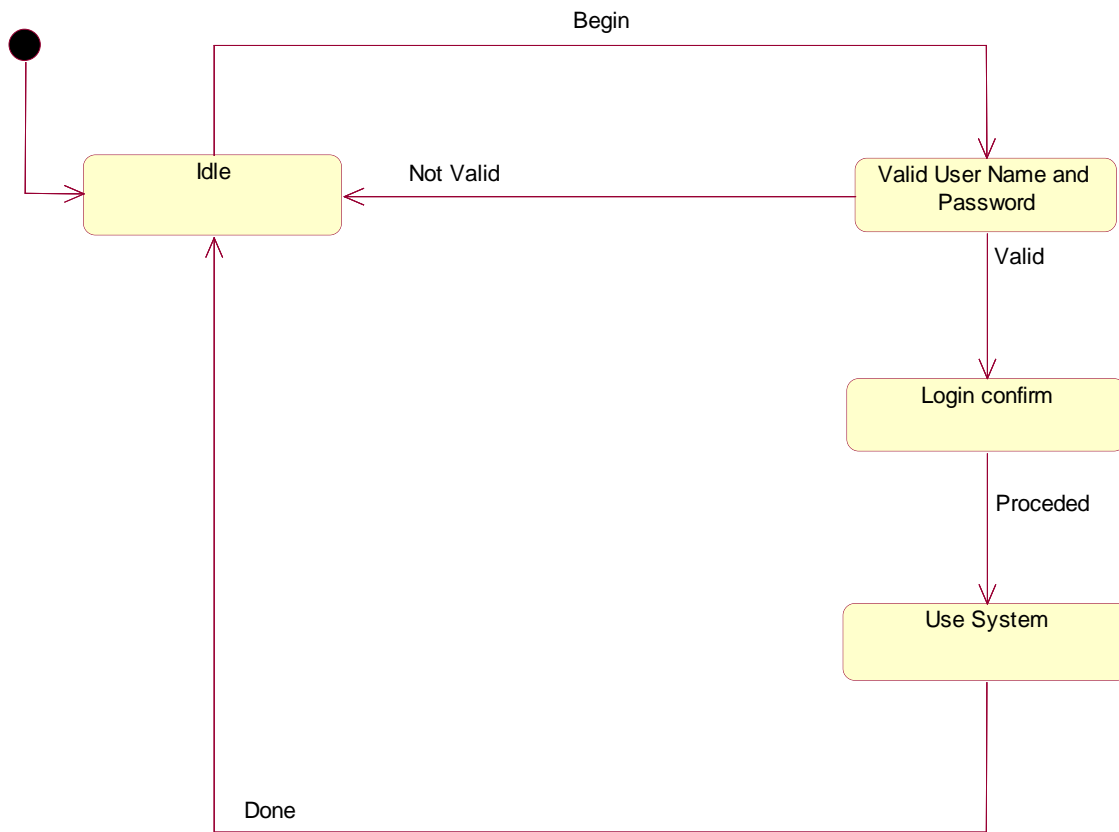


Figure 4.9.1-11 Login State Transition Diagram

#### 4.9.2. For Place Order

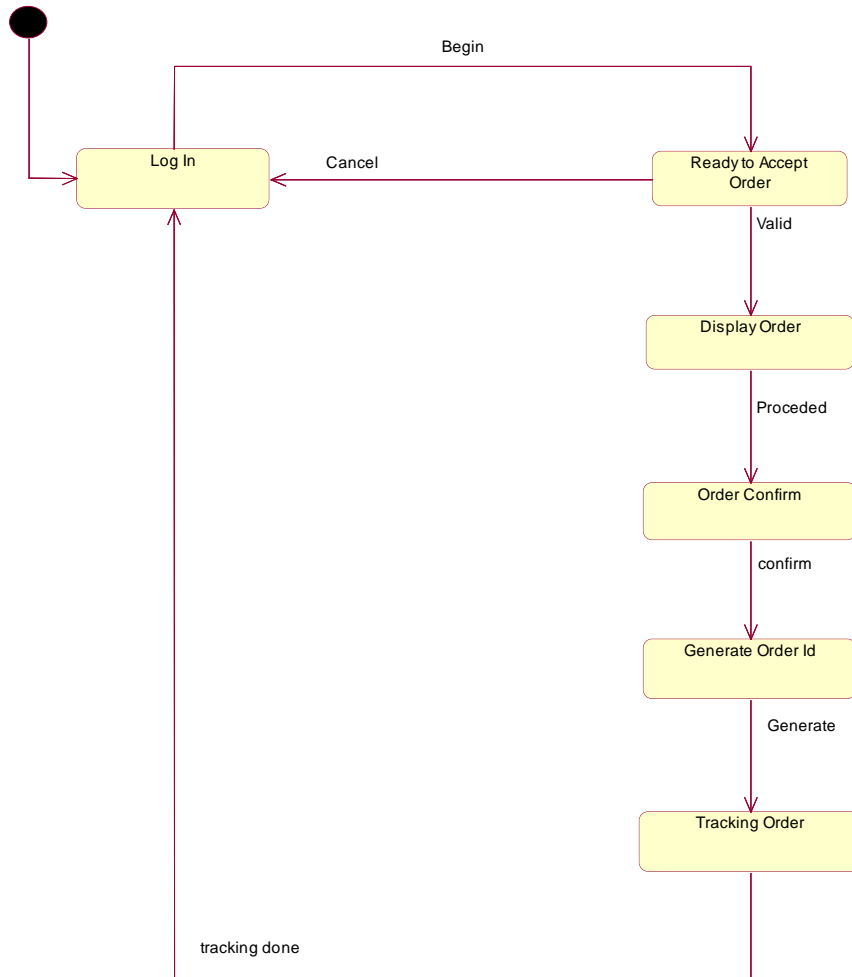


Figure 4.9.2-12 Place order State Transition Diagram



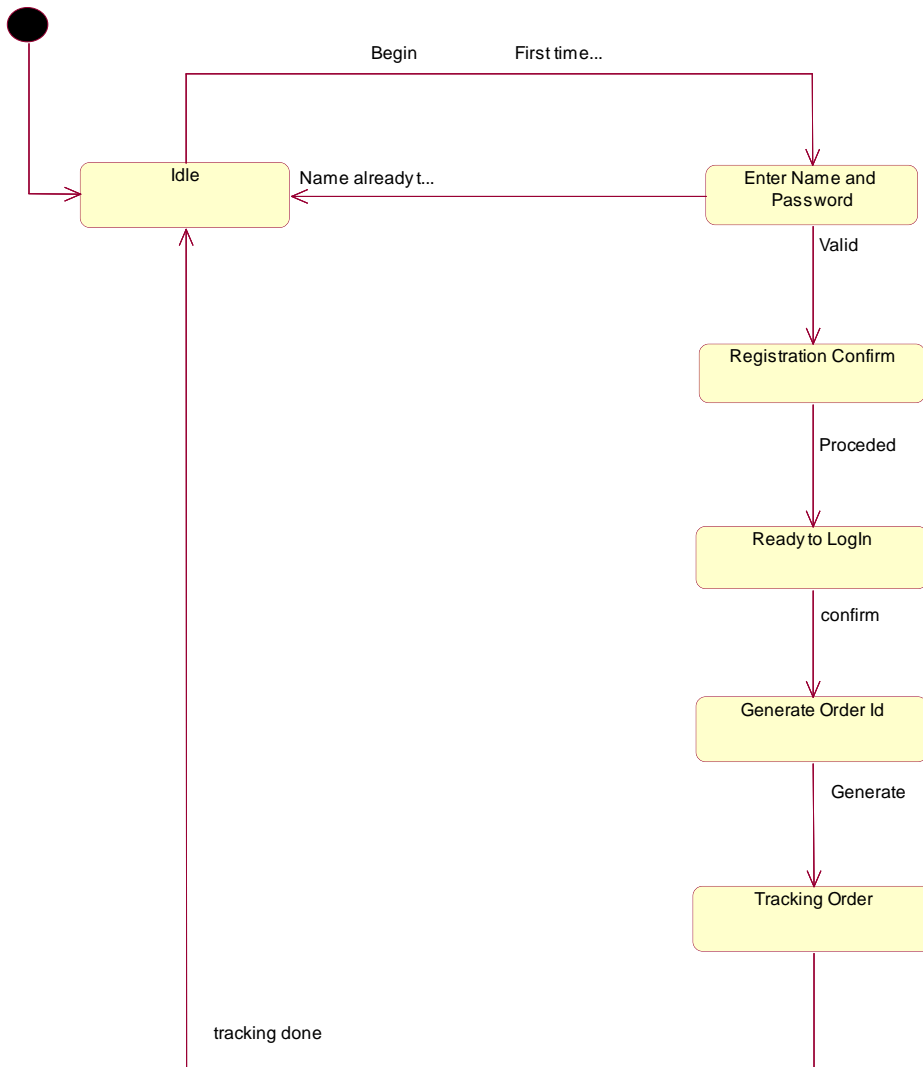


Figure 4.9.4-14 Registration State Transition Diagram

#### 4.9.5. For Admin

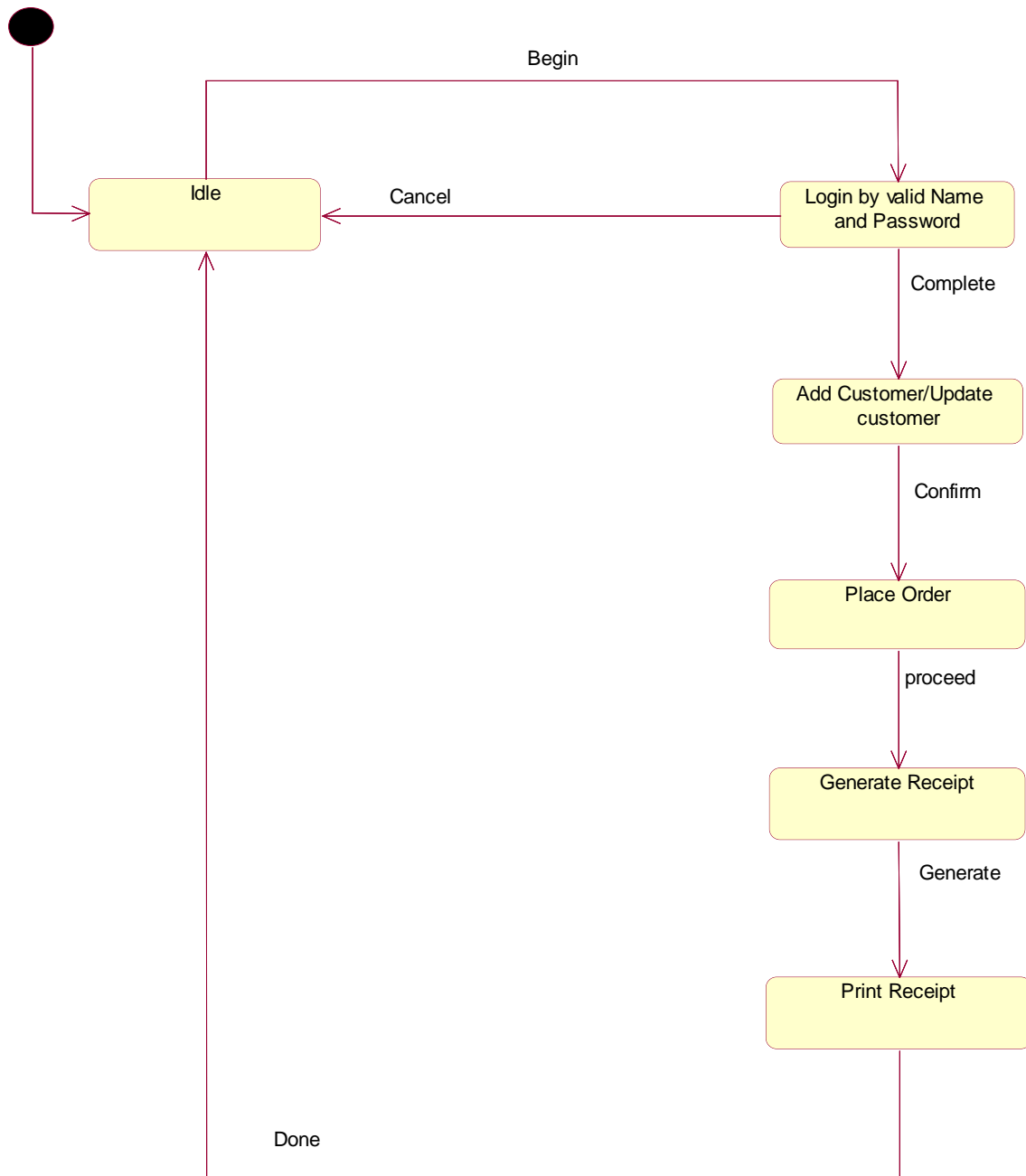


Figure 4.9.5-15 Admin State Transition Diagram



## 4.10. Component Diagram

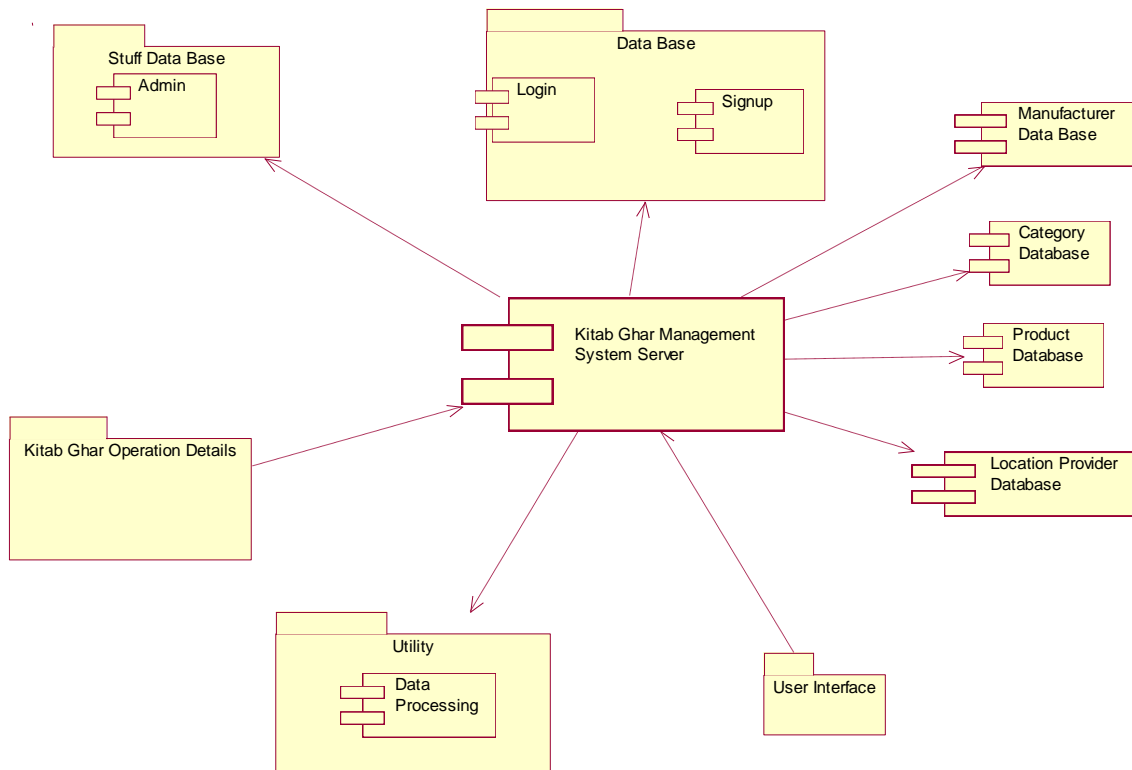


Figure 4.10-17 Component Diagram

### 4.11. Deployment Diagram

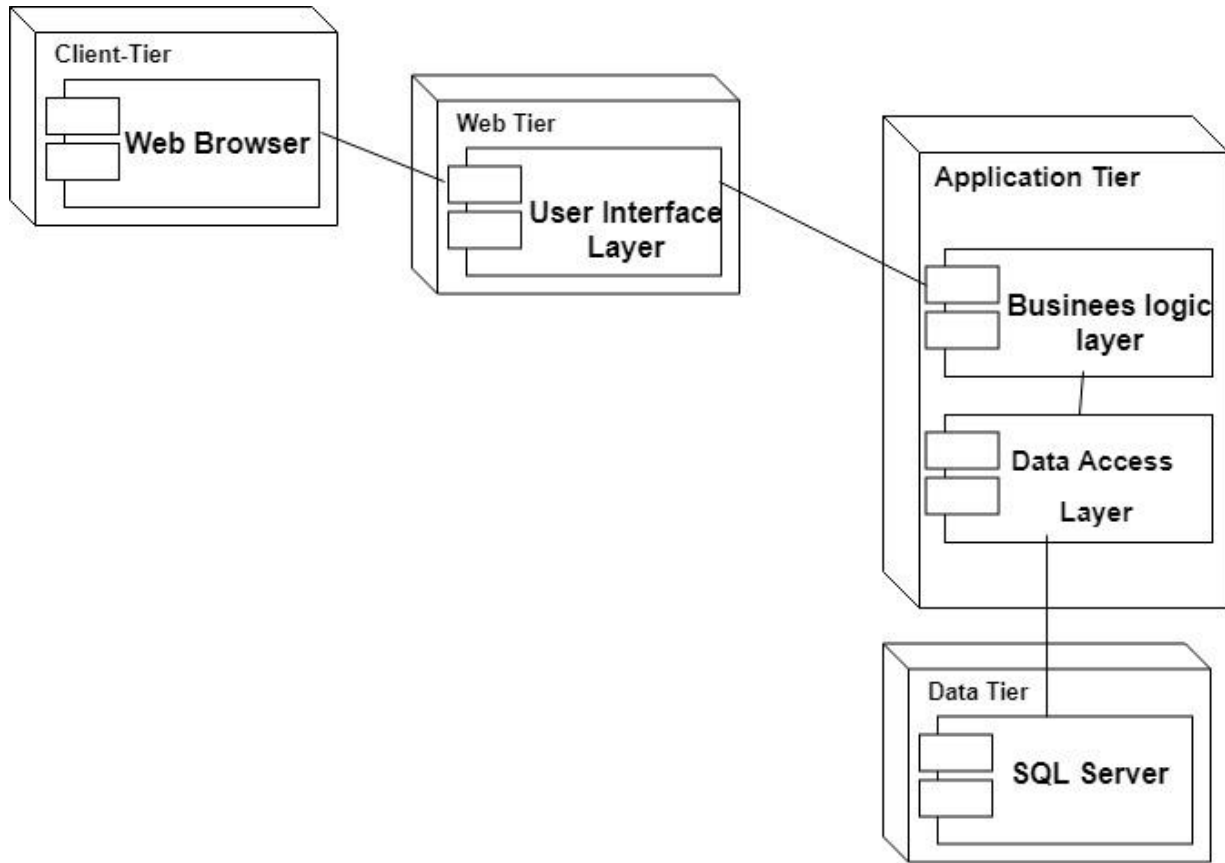


Figure 4.11-18 Deployment Diagram

### 4.12. Data Flow diagram [only if structured approach is used - Level 0 and 1]

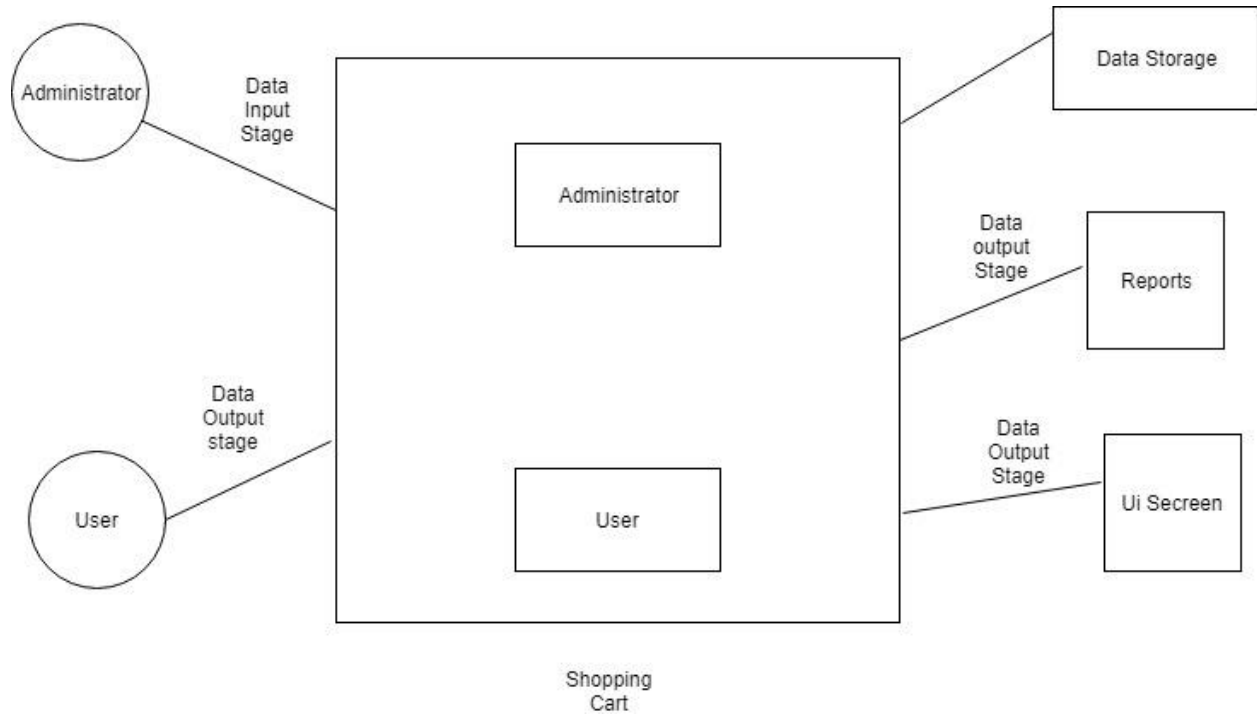


Figure 4.12-Data Flow Diagram

#### For Login

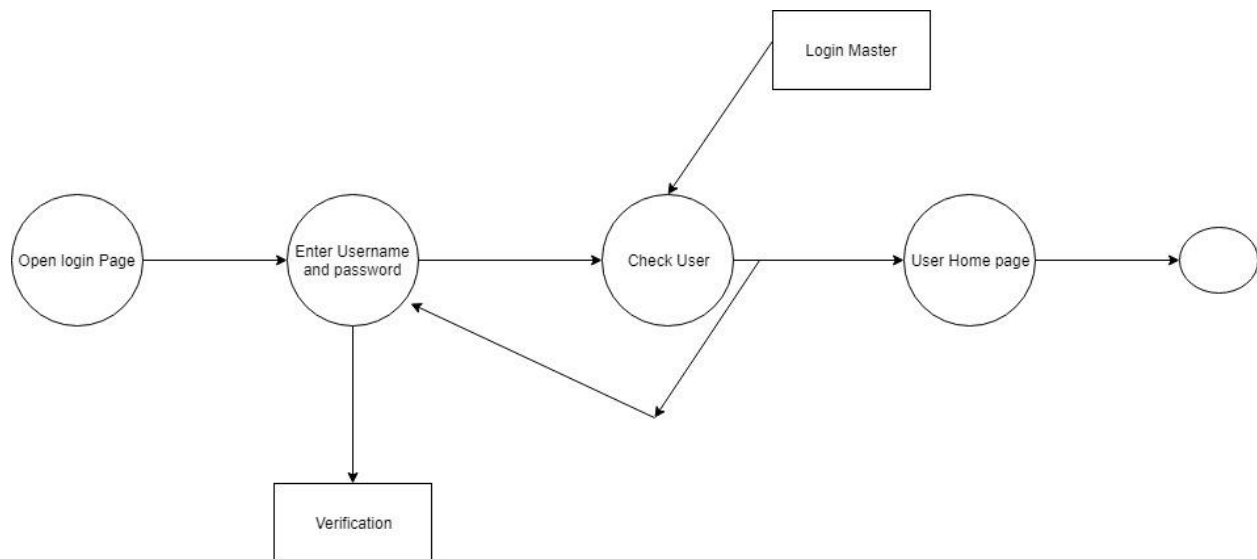


Figure 4.12 .1Admin Data Flow Diagram

**For Registration**

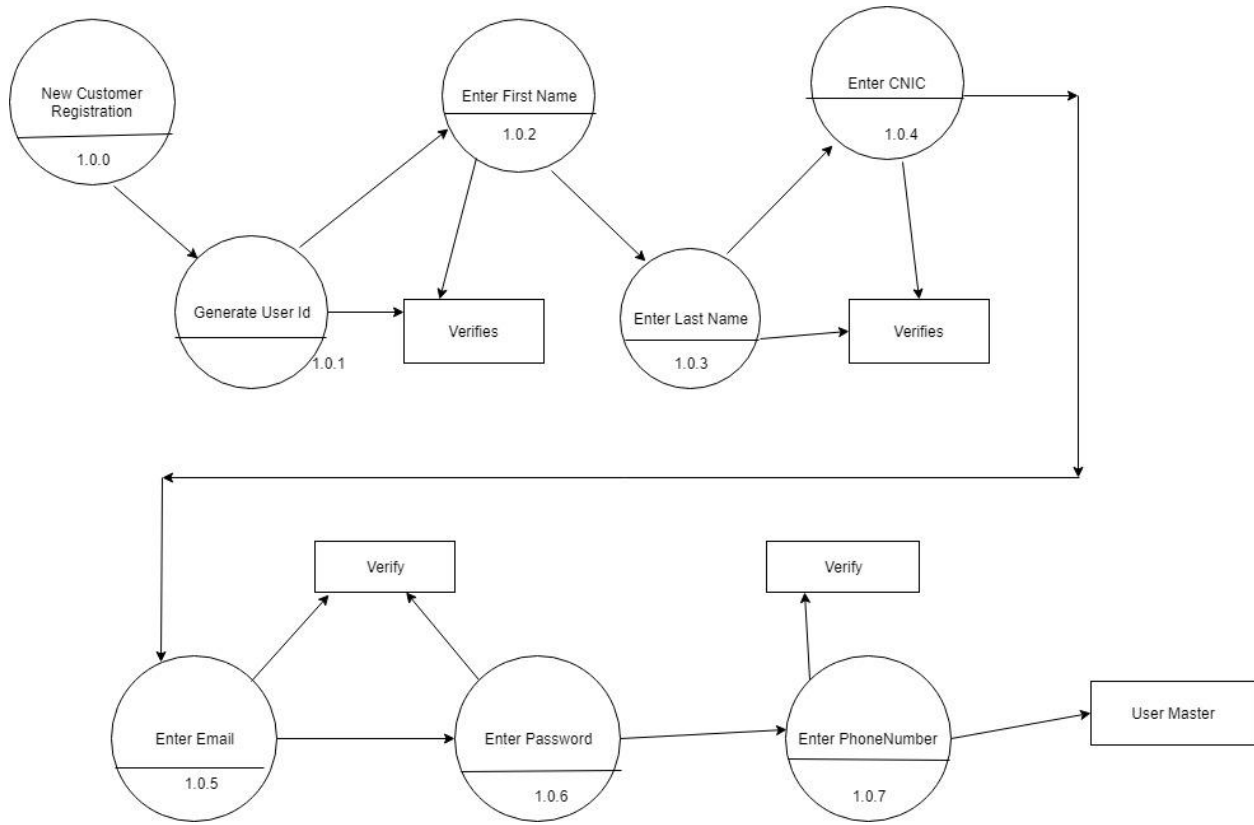


Figure 4.12.2- Registration Data Flow Diagram

## Manage Product

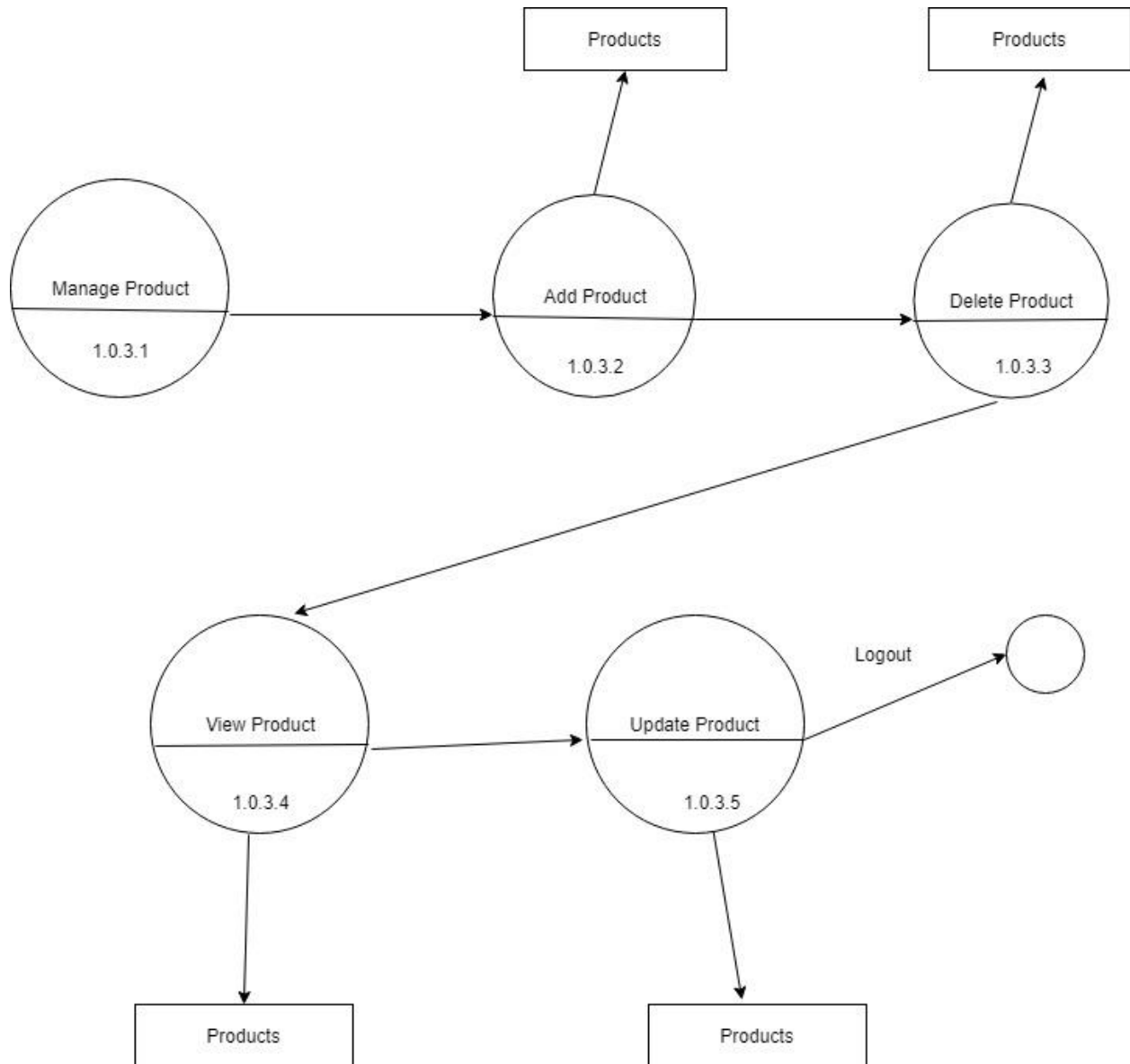


Figure 4.12.3 Manage Product



# Chapter 5

## Implementation

## Chapter 5: Implementation

This chapter is all about the deployment of the project. It involves algorithms and some prototype of the project. Libraries, version and environment in which project is deployed are stated in this chapter.

### 5.1. Components, Libraries, Web Services and stubs

We are using HTML, CSS, JS, PHP and Codeignitor. The libraries are as follows:

- Encryption
- form validation
- session
- pagination
- form
- schema

### 5.2. Tools and Techniques

Codeignitor

Sublime Text

Notepad ++

XAMP

### **5.3. Best Practices / Coding Standards**

The coding standard we used to make this project same as w3c. The libraries of codeignitor are used. Model, views and controllers are used to design the overall project. The classes in bootstrap framework are used.

### **5.4. Version Control**

PHP version: 7

HTML: 5

CSS: 3

Bootstrap: 4

Codeignitor: 3.1.9

JavaScript: 5

## **UI Screenshots**

### **1. Main Screen**

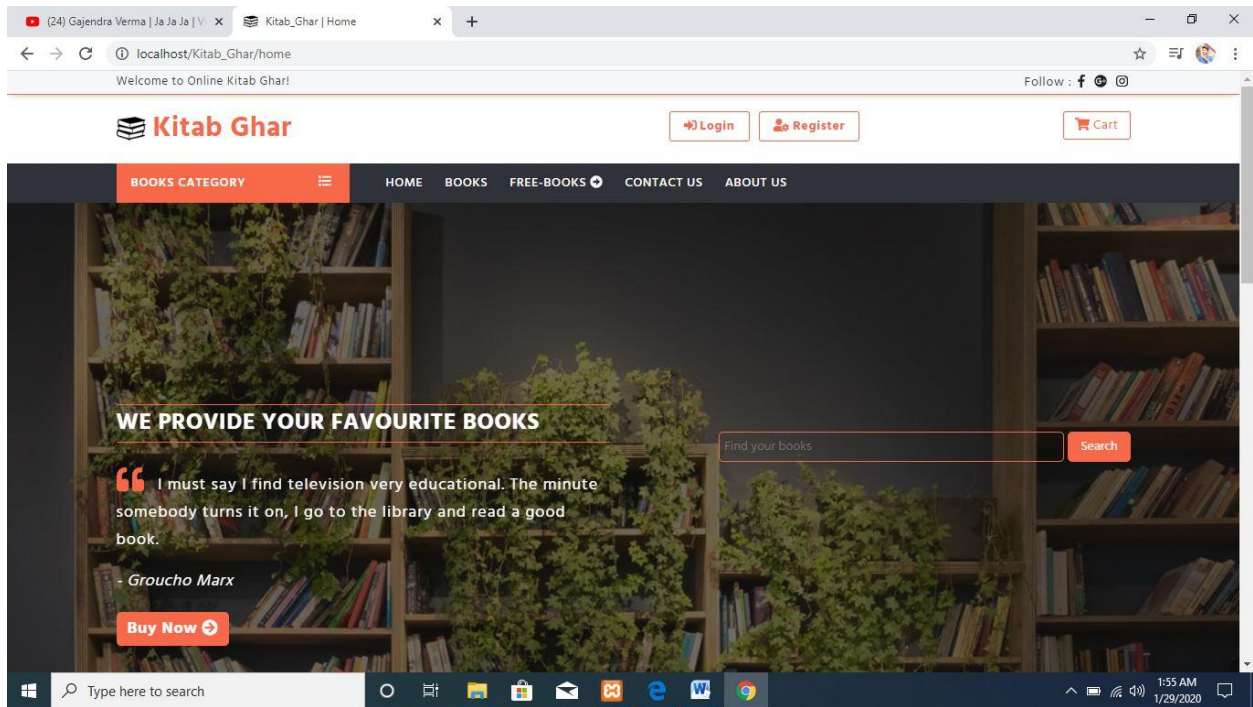
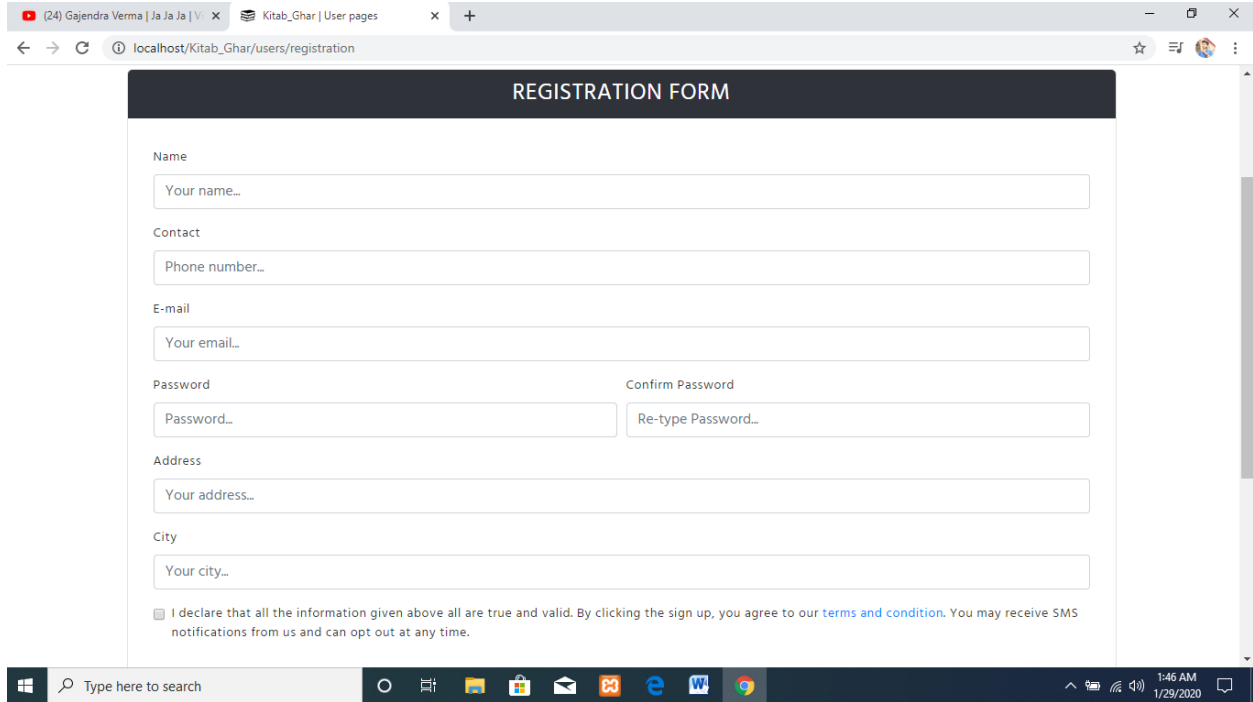


Figure 5.4-19 Main Screen

## 2. Registration Page



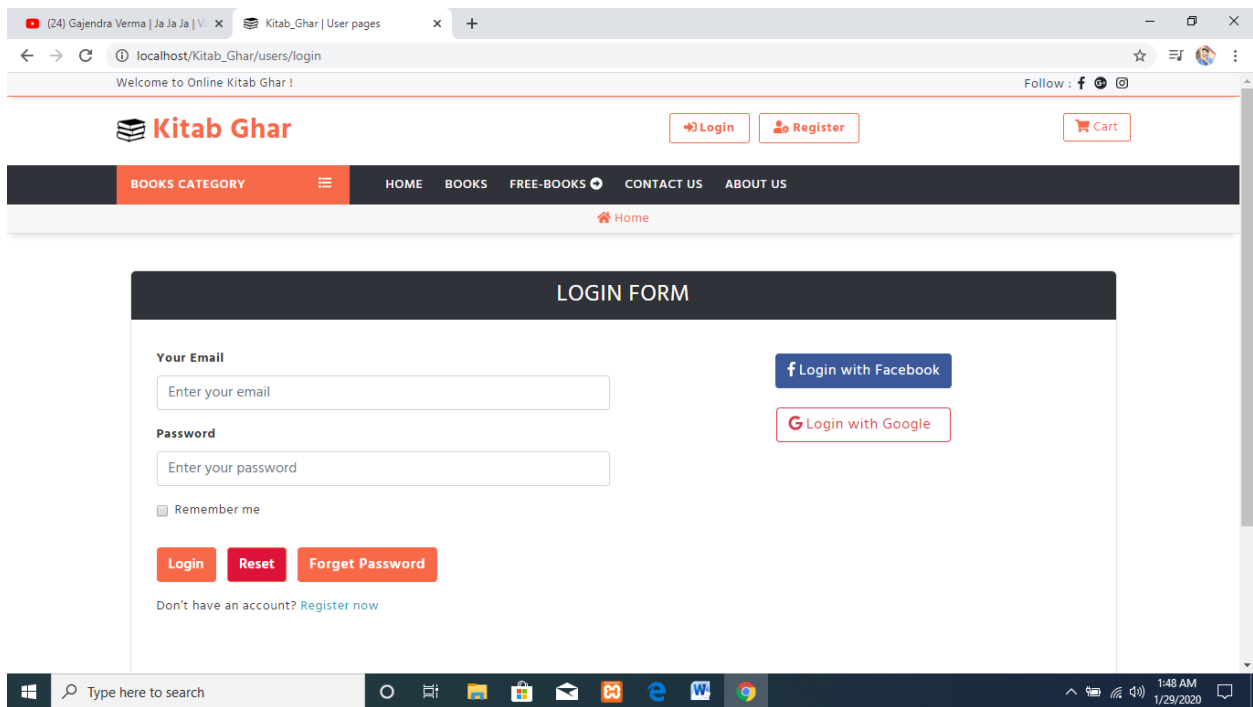
The screenshot shows a web browser window with the URL `localhost/Kitab_Ghar/users/registration`. The page title is "REGISTRATION FORM". The form contains the following fields:

- Name:
- Contact:
- E-mail:
- Password:
- Confirm Password:
- Address:
- City:

Below the fields, there is a checkbox and text:  I declare that all the information given above all are true and valid. By clicking the sign up, you agree to our [terms and condition](#). You may receive SMS notifications from us and can opt out at any time.

Figure 5.4-20 Registration Page

### 3.Login



The screenshot shows a web browser window with the URL `localhost/Kitab_Ghar/users/login`. The page title is "LOGIN FORM". The page content includes:

- Header: "Welcome to Online Kitab Ghar!" and "Follow : f t i".
- Navigation: "Kitab Ghar" logo, "Login", "Register", and "Cart" buttons.
- Menu: "BOOKS CATEGORY", "HOME", "BOOKS", "FREE-BOOKS", "CONTACT US", "ABOUT US".
- Form Fields:
  - Your Email:
  - Password:
  - Remember me:
- Buttons: "Login", "Reset", "Forget Password", "Login with Facebook", "Login with Google".
- Footer: "Don't have an account? [Register now](#)".

## 4.User

The screenshot shows a web browser window with the URL `localhost/Kitab_Ghar/user-home`. The page title is "Welcome to Online Kitab Ghar!". The main content area is titled "Welcome, Khalid Book Depo" and includes a sub-header: "You can sell your book with a proper price. You can see your book informations. You can track your order from here, and you can see your order details. You can also edit your profile information from here, like as your name, password and other informations."

**PROFILE DETAILS**

- Khalid Book Depo**
- user@user.com
- 034509664
- DHA SECTOR BB house 456 Ghazi Road Lahore lahore
- Joined from: 21-Jan, 20

**YOUR ACTIVITIES**

- Total Books:** You have uploaded total 0 Books.
- Published Books:** 0
- Orders:** You have placed 2 orders till now.
- Reviews:** You have written 1 reviews of books.

## 4.Admin

The screenshot shows a web browser window with the URL `localhost/Kitab_Ghar/admin/index`. The page title is "Home / Admin Dashboard". A message at the top states: "Logged in successfully. You have logged in as an admin."

**Admin Dashboard**

Welcome, Admin. You have logged in as an admin. Now you can monitor the full application

Category	Count
Total Category	7
Total Books	28
Total Users	6
Total e-books	3
Total orders	4
Pending Books	0

© 2020 Azam Bilal, Inc.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

Test & Evaluation is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess progress of design, performance, supportability, etc. Developmental test and evaluation are engineering tool used to reduce risk throughout the acquisition cycle.

Testing is a mechanism to assure quality of a product, system, or capability (e.g., right product, built right). To be effective, testing cannot occur only at the end of a development. It must be addressed continuously throughout the entire life cycle.

### 6.1. Use Case Testing:

**Use case testing** is a technique that helps us identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.

#### Use Case Test 1:

<b>Use Case Name</b>	<b>Register</b>
<b>Use Case Description</b>	User registration for using the system.
<b>Actors</b>	User, Shop keeper, Admin.
<b>Pre-Condition</b>	User must open the website.
<b>Post-Condition</b>	User register with system after registration successfully to the system.
<b>Actor</b> <b>System</b>	<ul style="list-style-type: none"> <li>• User should be register himself via First name, last name, password, Address, Phone, Title, Address.</li> <li>• System Validate all the input fields.</li> <li>• Registered with system.</li> </ul>
<b>Extension</b>	If any input field was invalid then system generates error.

**Use Case Test 2:**

<b>Use Case Name</b>	<b>Login</b>
<b>Use Case Description</b>	A user login to access the system features.
<b>Actors</b>	User, Shop keeper, Admin.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. User must have to register in system.</li> <li>2. User must have to open the system.</li> </ol>
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. User have login successful.</li> <li>2. User have shown the all details about system.</li> </ol>
<b>Actor</b> <b>System</b>	<ul style="list-style-type: none"> <li>• Enter Username and password.</li> <li>• System validate User name and password.</li> <li>• Allow access to system dashboard.</li> </ul>
<b>Extension</b>	If invalid password and username,  Then system should show error messages.

**Use Case Test 3:**

<b>Use Case Name</b>	<b>Courses</b>
<b>Use Case Description</b>	Admin can send notifications.
<b>Actors</b>	Shop Keeper, Admin.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. User must have to register in system.</li> </ol>

	<ol style="list-style-type: none"> <li>2. User must have to open the system.</li> <li>3. The user must fulfill the specific goals to clear the specific activity.</li> </ol>
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. User can upload the course.</li> <li>2. User can delete the course.</li> <li>3. User can update the course.</li> </ol>
<b>Actor</b> <b>System</b>	<ul style="list-style-type: none"> <li>• Shop Keeper &amp; Admin must login successfully can view the course.</li> <li>• Valid input fields.</li> <li>• System can access the admin to send notification.</li> </ul>
<b>Extension</b>	<p>Invalid input fields</p> <p>System show error message.</p>

#### Use Case Test 4:

<b>Use Case Name</b>	<b>Search .</b>
<b>Use Case Description</b>	user can search books.
<b>Actors</b>	User, Shop keeper, Admin..
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. User must be register and login in system.</li> <li>2. The user must fulfill the specific goals to clear the specific activity.</li> </ol>
<b>Post-Condition</b>	User can get information about books and their relevant.
<b>Actor</b> <b>System</b>	<ul style="list-style-type: none"> <li>• User must login successfully can view the books.</li> <li>• Valid input fields.</li> <li>• System can access the admin to send notification.</li> </ul>

<b>Extension</b>	Invalid input fields System show error message.
------------------	--

**Use Case Test 5:**

<b>Use Case Name</b>	<b>Manage information</b>
<b>Use Case Description</b>	Admin manage the information about books and their relevant data.
<b>Actors</b>	Admin.
<b>Pre-Condition</b>	Admin must be register and login in system.
<b>Post-Condition</b>	1. Admin can get information about books and their relevant.
<b>Actor</b> <b>System</b>	<ul style="list-style-type: none"> <li>• Admin must login successfully can view the information.</li> <li>• Valid input fields.</li> <li>• Admin manage books data.</li> </ul>
<b>Extension</b>	<ul style="list-style-type: none"> <li>• Invalid input fields System show error message.</li> <li>• If admin not login then he is not able to manage the information</li> </ul>

## 6.2. Equivalence partitioning

<b>Test Case Suite</b>	TS001
<b>Test Case id</b>	TC001
<b>Test Case summary</b>	To verify the registration.
<b>Related Requirement</b>	User should be register himself/herself via Username, First name, Last Name, Password, Address, Phone, Title.
<b>Test Procedure</b>	<ol style="list-style-type: none"> <li>1. Enter username in username field</li> <li>2. Enter first name in first name field</li> <li>3. Enter last name in last name field</li> <li>4. Enter password in password field</li> <li>5. Enter address in address field</li> <li>6. Enter Title in title field.</li> </ol>
<b>Test Data</b>	<ul style="list-style-type: none"> <li>• User Name</li> <li>• First Name</li> <li>• Last Name</li> <li>• Password</li> <li>• Address</li> <li>• Title</li> </ul>
<b>Expected Result</b>	<p>System should register user if user enter valid fields.</p> <p>System should generate error if all the inputs are not valid.</p>
<b>Actual Result</b>	<p>System entered the user if user enter valid fields.</p> <p>System generate error if all the inputs are not valid.</p>
<b>Status</b>	Valid
<b>Remarks</b>	This is a registration process test case
<b>Created By</b>	Ahsan Amanullah khan

<b>Date of Creation</b>	22/01/2020
<b>Test Environment</b>	OS: Windows 8 Google Chrome

## Test Case 2:

<b>Test Case Suite</b>	TS002
<b>Test Case id</b>	TC002
<b>Test Case summary</b>	To verify the login.
<b>Related Requirement</b>	User should be able to login to user name and password.
<b>Test Procedure</b>	<ol style="list-style-type: none"> <li>1. Enter user name in user name field.</li> <li>2. Enter password in password field.</li> </ol>
<b>Test Data</b>	User Name: Ahsan Amanullah khan Password: User Identified
<b>Expected Result</b>	<ol style="list-style-type: none"> <li>1. System should login user if user entered valid fields</li> <li>2. System should generate error in case of entering invalid field.</li> </ol>
<b>Actual Result</b>	System entered the user if user enter valid fields. System generate error if all the inputs are not valid.
<b>Status</b>	Valid.
<b>Remarks</b>	This is a login procedure test case.
<b>Created By</b>	Ahsan Amanullah khan
<b>Date of Creation</b>	22/01/2020
<b>Test Environment</b>	OS: Windows 8

	Google Chrome
--	---------------

### Test Case 3:

<b>Test Case Suite</b>	<b>TS003</b>
<b>Test Case id</b>	<b>TC003</b>
<b>Test Case summary</b>	To verify user
<b>Related Requirement</b>	User must be login to system.
<b>Test Procedure</b>	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Update details of books.</li> <li>3. Guide users about books.</li> <li>4. Logout</li> </ol>
<b>Test Data</b>	<p>Update in books.</p> <p>Give reviews to shopkeepers.</p>
<b>Expected Result</b>	<p>System should store data.</p> <p>System should retrieve data.</p> <p>System should generate error Pop-up message against invalid data enter.</p>
<b>Actual Result</b>	<ol style="list-style-type: none"> <li>1.System should store data.</li> <li>2.System should retrieve data.</li> <li>3.System should generate error Pop-Up message against invalid data enter.</li> </ol>
<b>Status</b>	Valid.

<b>Remarks</b>	Teacher detect responsibilities test case
<b>Created By</b>	Ahsan Amanullah khan
<b>Date of Creation</b>	22/01/2020
<b>Test Environment</b>	OS: Windows 8 Google Chrome

### Test Case 4:

<b>Test Case Suite</b>	<b>TS004</b>
<b>Test Case id</b>	<b>TC004</b>
<b>Test Case summary</b>	To verify admin responsibilities
<b>Related Requirement</b>	Admin dashboard.
<b>Test Procedure</b>	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Defect Information</li> <li>3. Add information/Remove Information</li> <li>4. Modify information</li> <li>5. Report generate od diseases</li> <li>6. Logout</li> </ol>
<b>Test Data</b>	<ol style="list-style-type: none"> <li>1. Search Defects.</li> <li>2. Show results against defect information.</li> <li>3. Generate Defect Log.</li> </ol>
<b>Expected Result</b>	1. System should update the data if valid input fields given.

	System should generate error Pop-Up message against invalid data.
<b>Actual Result</b>	If we entered valid data against valid fields then system update data, otherwise generate error Pop-Up message.
<b>Status</b>	Valid.
<b>Remarks</b>	Update and store defect Information test case.
<b>Created By</b>	Ahsan Amanullah khan
<b>Date of Creation</b>	22/01/2020
<b>Test Environment</b>	OS: Windows 8 Google Chrome

### 6.3. Boundary value analysis:

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.

- So, these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside-Just Outside values are called boundary values and the testing is called "boundary testing".
- The basic idea in boundary value testing is to select input variable values at their:
  1. Minimum
  2. Just above the minimum
  3. A nominal value
  4. Just below the maximum
  5. Maximum

### 6.4. Data flow testing:

This complexity makes a program full of risk. So testing these variable data flow makes sense and Dataflow testing concept can be used to test all the variables from the program.

## Levels of Dataflow Testing:

### Static data flow testing

- Identifying Potential Defects within the program.
- Analyzing code for defects
- The code will not be executed

### Dynamic data flow testing

- We will execute the program
- Looks like a control flow testing
- List out the path need test.

### Important flow control of system:

### Login:

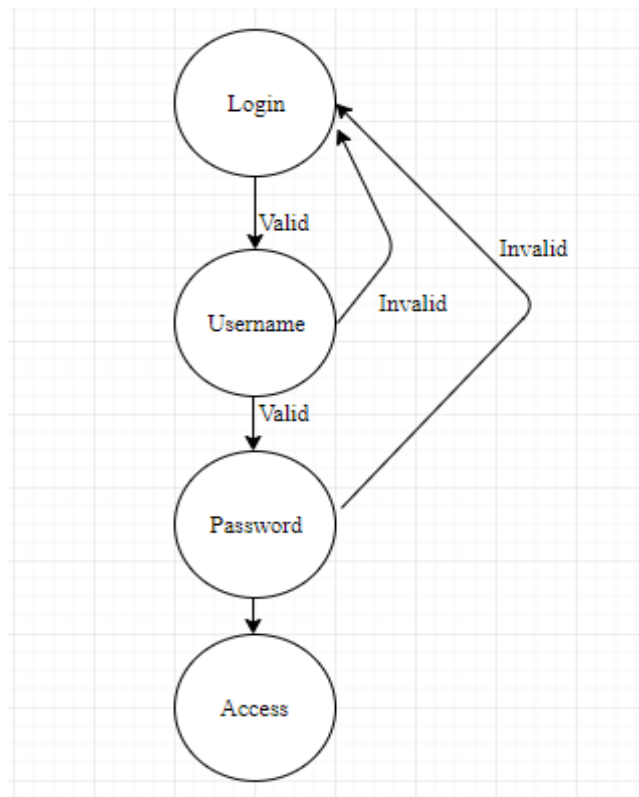


Figure 21 6.1. logi

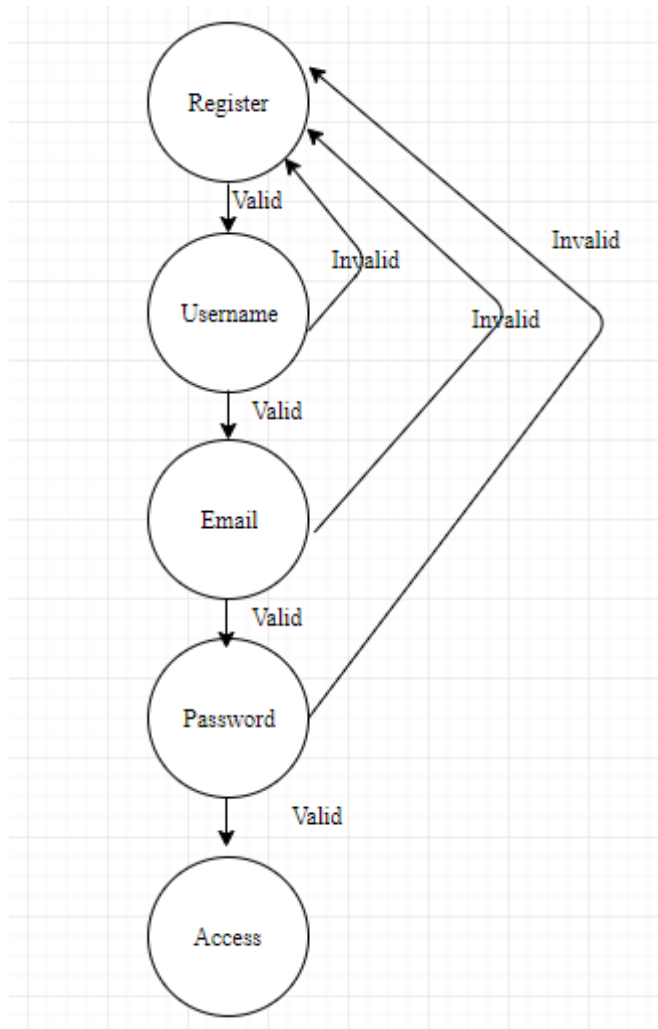
**Register:**

Figure 22 6.2. Register

## 6.5. Unit testing

Unit testing is a white-box testing technique. The main consideration in this test is verification of all modules of the software system. Each module is unit tested, as follows:

- Unit testing for the module User Management is tested by valid user or invalid user.

- Unit testing for the module User Account management is provided by testing Login, Logout and creating new or deleting updating existing user.
- Unit testing for the module of upload books.
- Unit testing for the module of reviews of users on uploaded books by shopkeepers.
- Unit testing for the module Detail of information of crops and their relative data.
- Unit testing for the module Add/update/delete information.

### Test cases for unit testing of modules:

Test Case No	Test item	Input Specifications	Output Specifications	Pass / Fail	Description
1	Login	Email, Password	Error message, Welcome Message	Pass	To verify login mechanism and user authentication
2	Registration	Name, Password, Email, Address Phone No	Validation Error, Successful Registration Message	Pass	To Register a New User
3	Add information	Information Category, information Name	Validation Error, Successful Add information	Pass	To Add a new information
4	View reviews	Books must be uploaded	Validation Error,	Pass	To View reviews

5	Logout	Must login	Validation Person	Pass	Logout
---	--------	------------	----------------------	------	--------

Figure 23 6.5.1. Test case for unot testing

### Test cases for black box testing of documentation / text:

Test Case No	Test item	Input Specifications	Output Specifications	Pass / Fail	Description
1	Syntax Grammatical Correctness Of the Content	Text written on each page of the system	Grammatically correct content or error	Pass	To check the content for grammatical & syntactical correctness
2	Semantics	Text written on each page of the system	Semantics correctness or Bugs / errors in text	Pass	To check the text for the use of improper phrases terminologies and semantic correctness
3	Validity of contact information	All contact information provided in Contact Us or other pages of the system	Correct / incorrect information	Pass	To check all the contact information for validity
4	Text in title	Text in browser's	Bugs / errors	Pass	To check the text in title bar of each page

	bar	title bar in each page	in text		for relevance & correctness
--	-----	------------------------	---------	--	-----------------------------

Figure 24 6.5.2. test case for black case

### Test cases for black box testing of hyperlinks:

Test Case No	Test item	Input Specifications	Output Specifications	Pass / Fail	Description
1	Hyperlinks connectivity	All text and image hyperlinks and their sub-links on each page	Working Broken Incorrect Links	Pass	To check all the links

Figure 25 6.5.3. test case for hyperlink

### Test cases for black box testing of graphics:

Test Case No	Test item	Input Specifications	Output Specifications	Pass / Fail	Description
1	Load & Display	All graphics on each page	Displayed properly/bugs , errors or delay in loading	Pass	To check that all the graphics display and load properly
2	Balance	All graphics on each page	All images placed properly/ bugs errors	Pass	To check that all the graphics for their size and placement

Figure 26 6.5.4. test case for graphics

## **6.6. Integration testing:**

Integration testing is the technique for testing the interfaces of software components. Each software components in unit tested, and all the components are integrated to perform together. The tests are conducted to ensure that the components are working properly after interfacing.

All modules are integrated by an incremental approach, and integration testing of the system is performed as follows:

- Integrate login module and test that the software is properly connected to the database.
- Integrate all modules and test by inserting updating, and deleting records.

## **6.7. Performance testing:**

Performance testing uses criteria to check that the system functions according to the specifications. All unit modules have been tested successfully. The integration of these unit modules produces reports in the required format.

## **6.8. Stress Testing:**

Performance testing should be carried out after the implementation of the product. We shall test the speed, accuracy, and effectiveness of functions as well as quantitative test e.g. response time of any function. We have tested the performance of our system by checking the working of every complex operation, repeatedly, multiple times. We have followed the following steps.

- List of all complex functions especially database related function.
- Assess response time and accuracy of core functions.
- Risk assessment of error or exception code (if any).

# Chapter 7

## Summary, Conclusion and Future Enhancements

## **Chapter 7: Summary, Conclusion & Future Enhancements**

### **7.1. Project Summary:**

We are going to develop a web application for online books purchasing system. We are developing this web application according to the students need. We are changing the learning environment for students. They can access to books of their desire whenever they want. Users can register here and get connected to our web application.

In old days students have too much problems for searching books. They have to go to market in search of books. So, they have to pay high amount for those books. That's seems difficult for them. Some students have to travel 2 hours to 4 hours a day for going to get their books.

### **7.2. Achievements and Improvements:**

### **7.3. Critical Review:**

We will use the functions of the system, and then give some information about how it works.

We are satisfied with our system and also satisfied the usability of the system it should be more attractive. The system is meeting our expectations. It is secure system and our data protected in the system. In business value our project or system helps us to grow it all over Pakistan and we will keep our customers online and manage them online.

#### **7.4. Lessons Learnt:**

In this project learn many new things learn language selection is the main subject for us for this project. We want to work on different platform and we want to use something different so then we decided to work on PHP MVC. It was so difficult for us to done the project on php. We learn how to use the framework and MySQL. We have created database on SQL server, the relationship between the different modules is confusing. We also generate a prototype to understand this system. We also understand how modules are connected this reduce the errors because all data has been collected in one place.

#### **7.5. Future Enhancements/Recommendations:**

In future we will take this system on big level so every person can facilitate from this system.

Every person can enroll and learn from all over the world have in our system.

# Reference and Bibliography

Reference and Bibliography

## Index

### [A]

Acknowledgment vi

Assumption and Dependencies 18,19

Architecture Diagram 43

Activity Diagram 60,61,62,63,64,65,67

## **[B]**

Background 2

Business Rules 36

## **[C]**

Communication Interface 20

Class Diagram 55

Component Diagram 68

Components 83

## **[D]**

Dedication v

Document Conventions 13

Design and Implementation Constraints 17

Domain Modal 51

Data Dictionary 53, 54