Price Comparison Web-based Application

By

Zeeshan Afzal Satti

Supervised By

Dr. Akmal Saeed Khattak

Department of Computer Science

Quaid-i-Azam University

Islamabad

2013-2017

Table of Contents

List of Figures

List of Tables

Chapter 1 Software Project Management Plan

Chapter 1

Software Project Management Plan (SPMP)

1.1 Introduction

This introduction provides information about software project management plan (SPMP) document for the Price Comparison Web-based Application (PCWBA). The document briefly describes the project overview, project deliverables, project organization that include software process model, roles and responsibilities, tools and techniques, and project management plan about the Price Comparison Web-based Application (PCWBA).

1.1.1 Project Overview

This project is meant to provide ease to those buyers that buys products online and manually visit different websites to view the prices of same product from different sellers but this application shows prices of same product at one place that is get from different websites. In this era of internet when more and more buyers are going online to make purchases, price comparison sites have really proved their worth[1]. The basic aim of this web-based application is to compare the prices of products that is gathered from different websites through web crawling techniques, web scraping (web harvesting or web data extraction) and vice versa. For someone who wants to save money, and receive the best value for money, they may meditate about price comparison websites. The main emphasis or focus of this project is to look different airlines that offer their services and to compare the prices of flights to find a cheap ticket and also providing useful travel options on a particular search route is very helpful to passengers. This whole idea of price comparison can then be extended to all other category products which include books, tickets, computers, laptops, mobiles, household items etc. Basically, this web-based application is a price comparison portal for online comparison of prices. This application also provide analysis on the price history of products. Customer can also provide feedback and rate the product about their experience.

1.1.2 Project Deliverables

Table 1.1 Project Deliverables

Deliverable Name	**Description**	**Delivery Date**
Software Project Management Plan (SPMP)	Description of the software approach and associated milestones	September 30th, 2016
Software requirement Specification (SRS)	Description of expected software features, constraints, interfaces and other attributes.	November 8th, 2016

[1] https://www.awksolutions.com/price-comparison-portfolio/

Software Design Description (SDD)	Description of how the software will meet the requirements. Also describes the rationale of design decisions taken.	December 30th, 2016
Final Product With Source Code	Working product with source code is given in this deliverable.	February 9th , 2017
Software Test Documentation (STD)	Description of the plan and specifications to verify and validate the software and the results.	February 22th, 2017

1.2 Project Organization

Project organization provides information about which software process model is followed, what are the roles of different team members and which tools and techniques are used in the project.

1.2.1 Software Process Model

The software process model used for this project will be a Waterfall Process Model. The reason is because we have two semesters for this project. In one semester, we only gather requirement and refined those requirements and next semester we start coding of the project. So, when the requirements are well understood, it is reasonable to use waterfall approach. The implementation is going in systematic sequential approach to development. Systematic Sequential approach means that first there is a **communication** phase that include project description and requirement gathering and second there is a **planning** phase in which we estimate and schedule a project and then there is third step that deals with **analysis and design phase** and then **construction** phase which include coding and testing. Then in last, there is a **deployment** phase which follows delivery, support and feedback. So, this is done in linear fashion and not include any haphazard approach. In this project, requirements are well-specified by the supervisor and unchanging. These all points leads toward the Waterfall Process Model.

1.2.2 Roles and Responsibilities

There is no team member involved in this project because Price Comparison Web-based Application (PCWBA) is an individual project. So, I have different roles and responsibilities which is stated as follows:

- Develop a project plan.
- Establish a project schedule and determine each phase.
- Manage deliverables according to the plan.
- Determine the methodology used in the project.
- Provide regular updates to supervisor about project progress (status).
- Ensure availability of essential resources such as development cost etc.
- Carry ultimate responsibility for the project.

- Approve project deliverables.
- Gather requirements from domain users.
- Document technical and business requirements.
- Establishes standards, tools and procedures that is use in the project, including Issue, Risk, Change and Information Management.
- Develop product according to user requirements.
- Resolving technical issues that arise during development.
- Verify that project deliverables meet the requirements.
- Test solutions to validate objectives (verify and validate final product).
- Deploy the project in the real environment.

1.2.3 Tools and Techniques

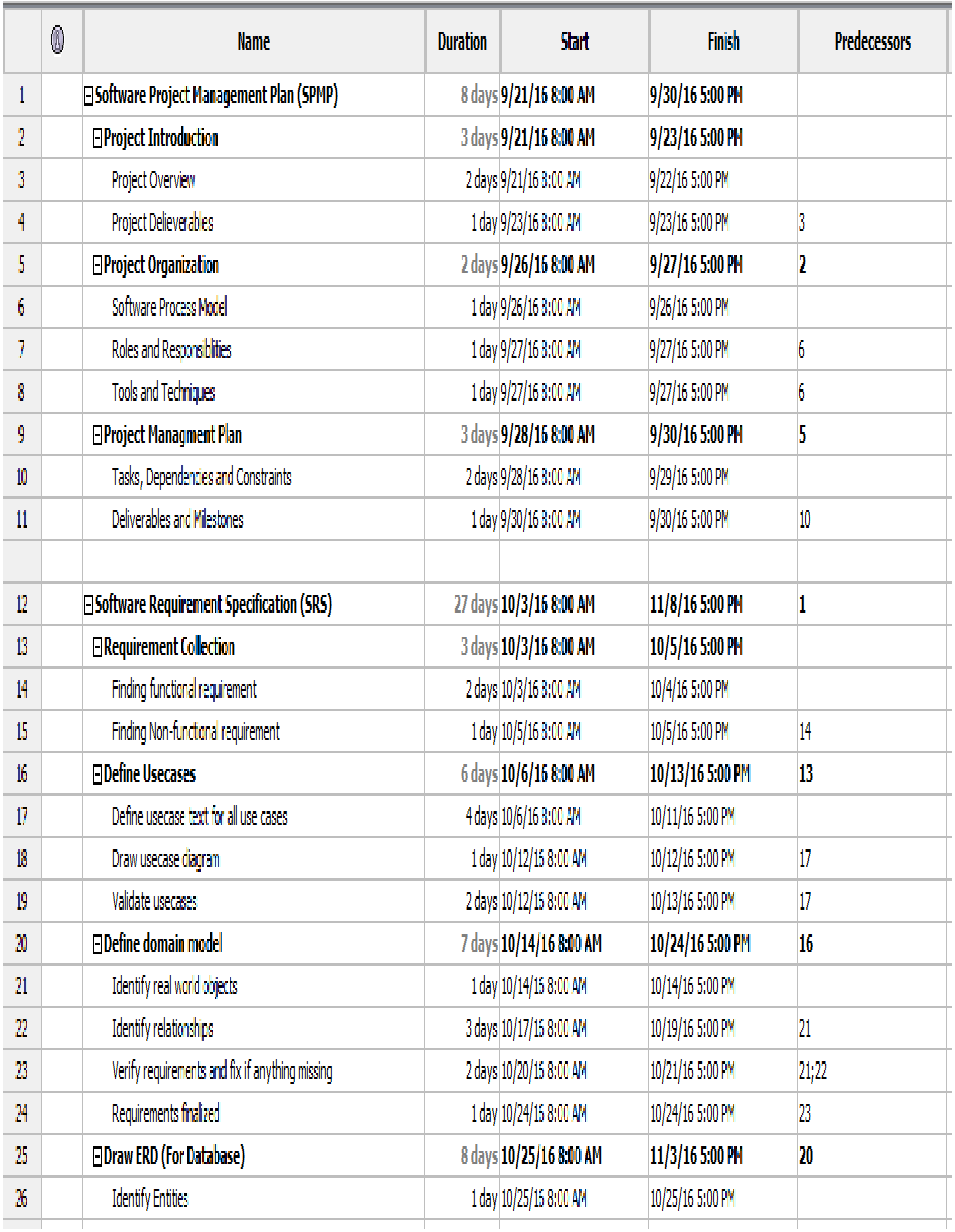
A tool is an item or implement used for a specific purpose. A tool can be a physical object or a technical object such as a web authoring tool or software program. A technique is a systematic procedure, or routine used to accomplish a specific activity or task. Tools that are used in this project are follows:

- Visual Studio 2012 for ASP.Net MVC 5
- Microsoft SQL Server for Database
- Microsoft Visio for use case diagram, domain model, ERD and class diagram.
- Project libre for software project management plan.
- Dreamweaver CS6 for Web designing.
- Adobe Photoshop for image manipulations (animations and provide special effect to images).

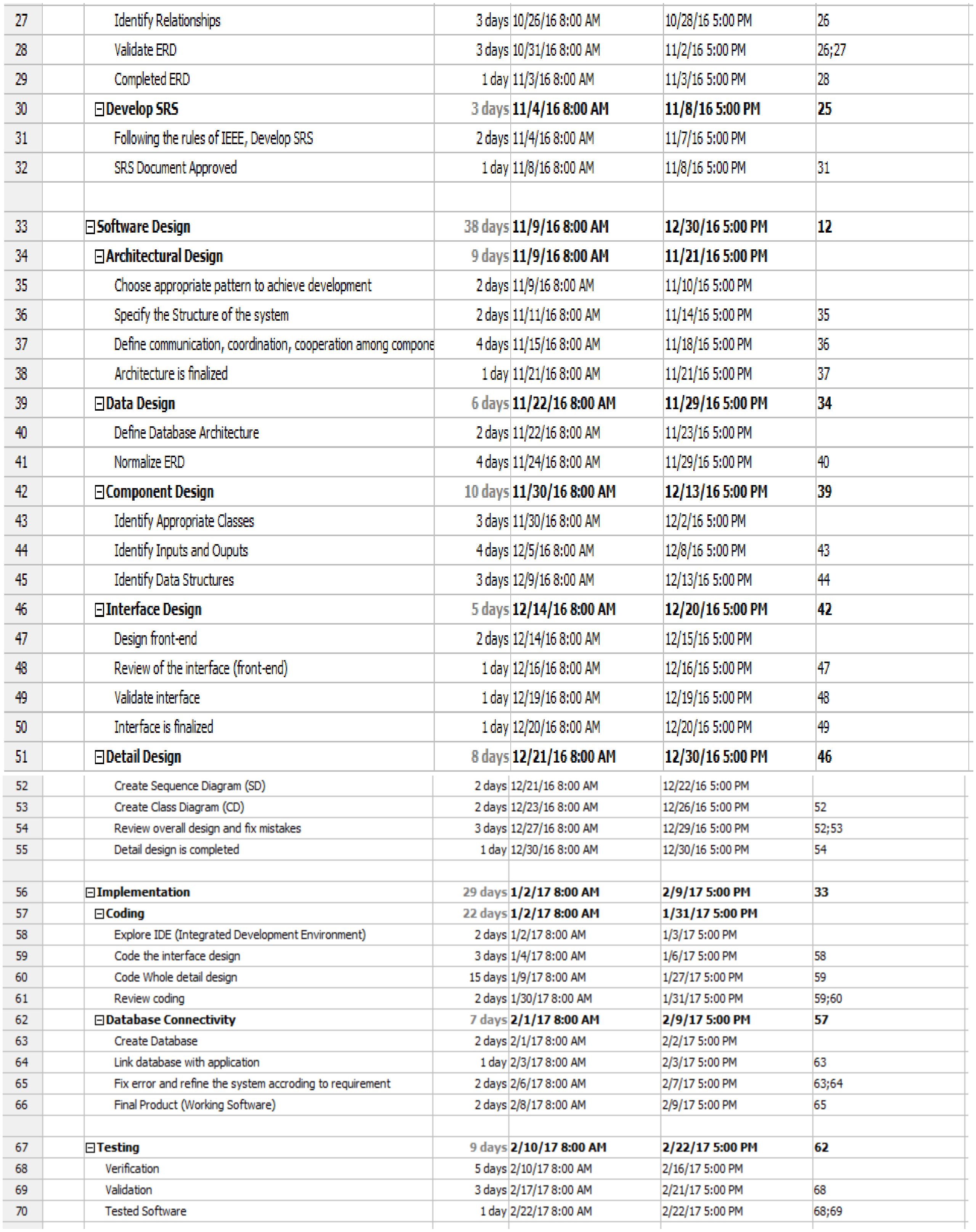
The object oriented analysis technique will be used to successfully complete the project.

1.3 Project Management Plan

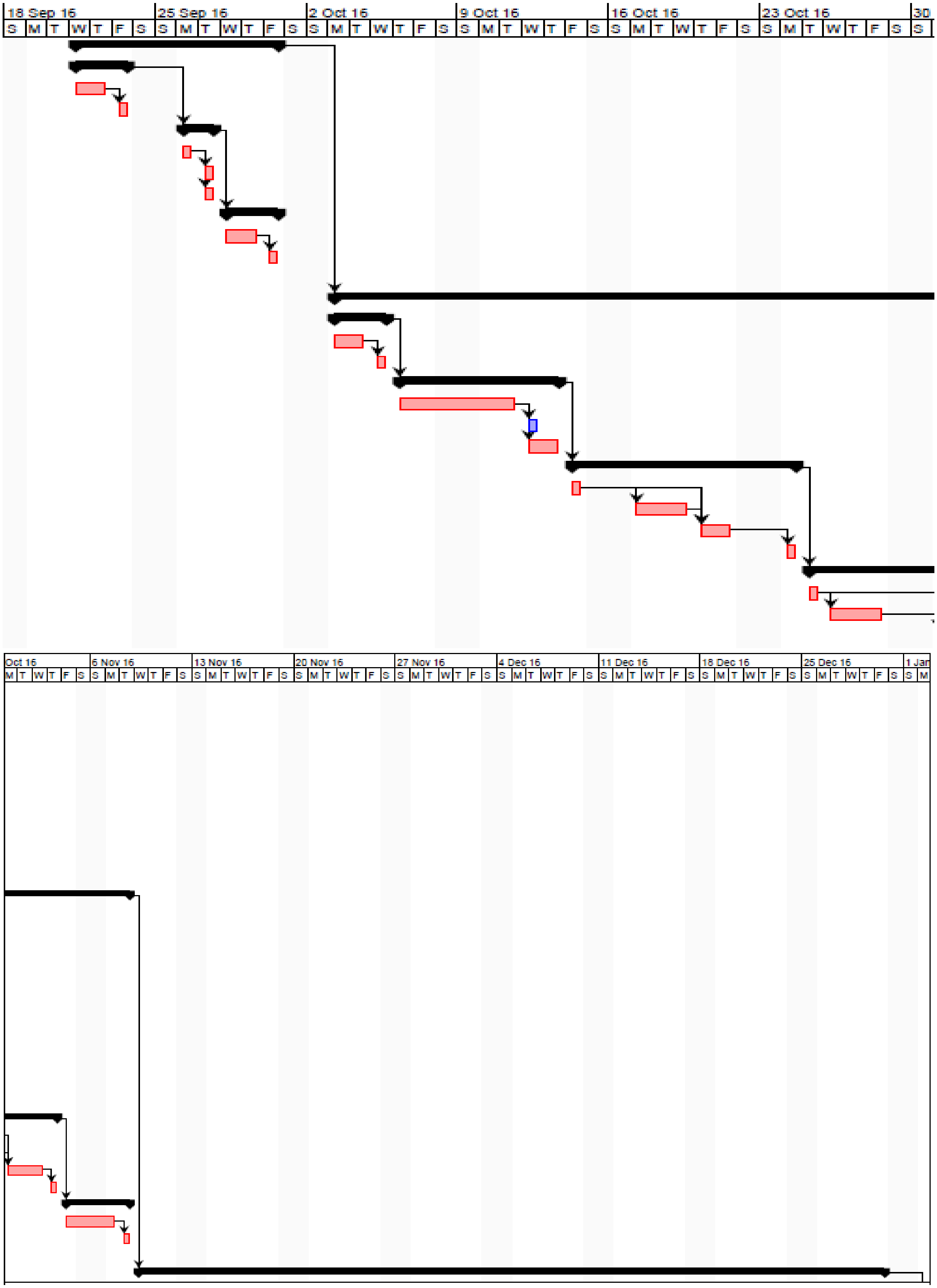
Software project management plan is the art and science of planning and leading software projects. In this project management, software projects are planned, implemented, monitored and controlled. The project management plan for this project is given on below:

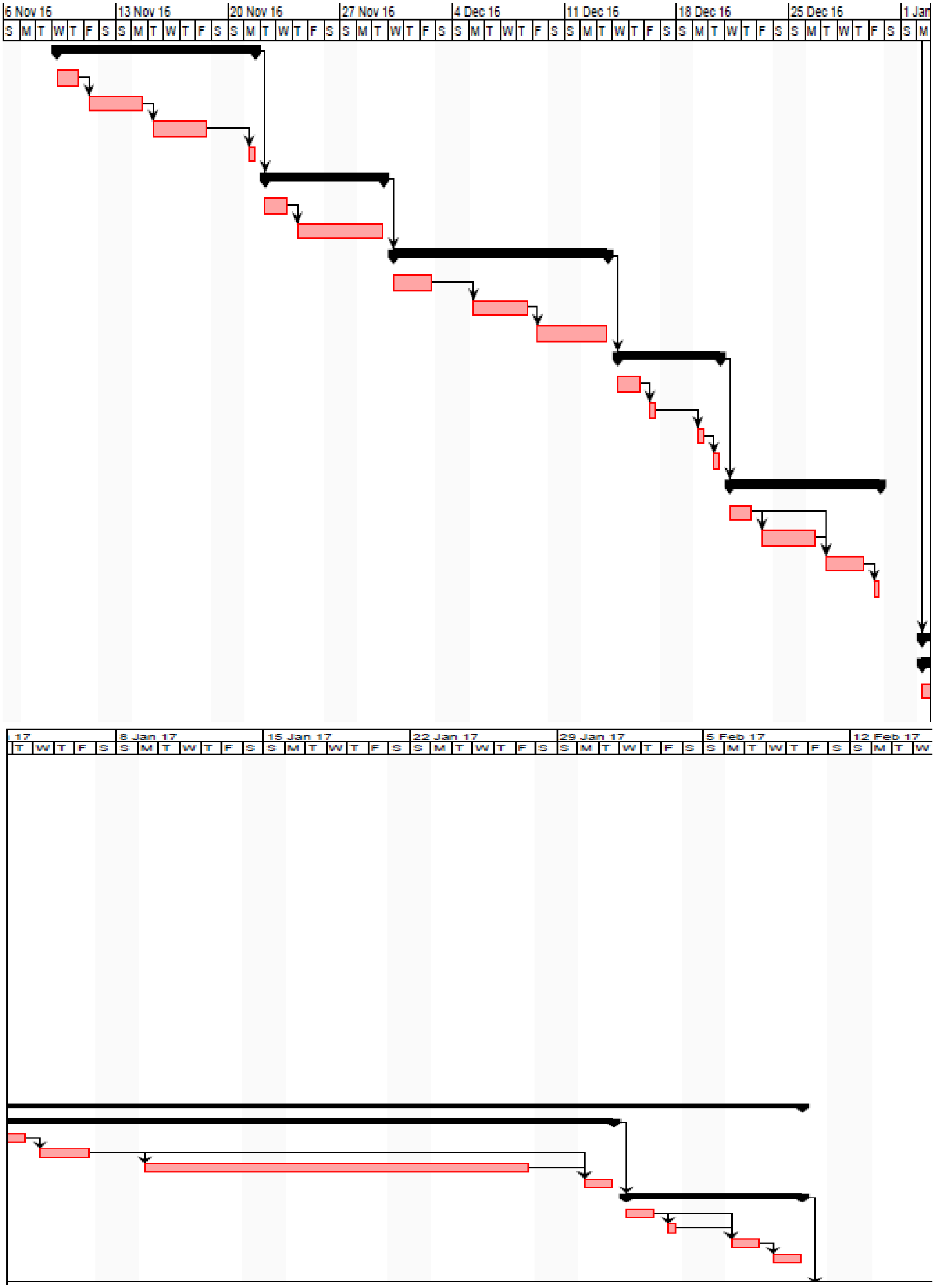


	ⓘ	Name	Duration	Start	Finish	Predecessors
1		⊟Software Project Management Plan (SPMP)	8 days	9/21/16 8:00 AM	9/30/16 5:00 PM	
2		⊟Project Introduction	3 days	9/21/16 8:00 AM	9/23/16 5:00 PM	
3		Project Overview	2 days	9/21/16 8:00 AM	9/22/16 5:00 PM	
4		Project Delieverables	1 day	9/23/16 8:00 AM	9/23/16 5:00 PM	3
5		⊟Project Organization	2 days	9/26/16 8:00 AM	9/27/16 5:00 PM	2
6		Software Process Model	1 day	9/26/16 8:00 AM	9/26/16 5:00 PM	
7		Roles and Responsiblities	1 day	9/27/16 8:00 AM	9/27/16 5:00 PM	6
8		Tools and Techniques	1 day	9/27/16 8:00 AM	9/27/16 5:00 PM	6
9		⊟Project Managment Plan	3 days	9/28/16 8:00 AM	9/30/16 5:00 PM	5
10		Tasks, Dependencies and Constraints	2 days	9/28/16 8:00 AM	9/29/16 5:00 PM	
11		Deliverables and Milestones	1 day	9/30/16 8:00 AM	9/30/16 5:00 PM	10
12		⊟Software Requirement Specification (SRS)	27 days	10/3/16 8:00 AM	11/8/16 5:00 PM	1
13		⊟Requirement Collection	3 days	10/3/16 8:00 AM	10/5/16 5:00 PM	
14		Finding functional requirement	2 days	10/3/16 8:00 AM	10/4/16 5:00 PM	
15		Finding Non-functional requirement	1 day	10/5/16 8:00 AM	10/5/16 5:00 PM	14
16		⊟Define Usecases	6 days	10/6/16 8:00 AM	10/13/16 5:00 PM	13
17		Define usecase text for all use cases	4 days	10/6/16 8:00 AM	10/11/16 5:00 PM	
18		Draw usecase diagram	1 day	10/12/16 8:00 AM	10/12/16 5:00 PM	17
19		Validate usecases	2 days	10/12/16 8:00 AM	10/13/16 5:00 PM	17
20		⊟Define domain model	7 days	10/14/16 8:00 AM	10/24/16 5:00 PM	16
21		Identify real world objects	1 day	10/14/16 8:00 AM	10/14/16 5:00 PM	
22		Identify relationships	3 days	10/17/16 8:00 AM	10/19/16 5:00 PM	21
23		Verify requirements and fix if anything missing	2 days	10/20/16 8:00 AM	10/21/16 5:00 PM	21;22
24		Requirements finalized	1 day	10/24/16 8:00 AM	10/24/16 5:00 PM	23
25		⊟Draw ERD (For Database)	8 days	10/25/16 8:00 AM	11/3/16 5:00 PM	20
26		Identify Entities	1 day	10/25/16 8:00 AM	10/25/16 5:00 PM	



27		Identify Relationships	3 days	10/26/16 8:00 AM	10/28/16 5:00 PM	26
28		Validate ERD	3 days	10/31/16 8:00 AM	11/2/16 5:00 PM	26;27
29		Completed ERD	1 day	11/3/16 8:00 AM	11/3/16 5:00 PM	28
30		**⊟Develop SRS**	**3 days**	**11/4/16 8:00 AM**	**11/8/16 5:00 PM**	**25**
31		Following the rules of IEEE, Develop SRS	2 days	11/4/16 8:00 AM	11/7/16 5:00 PM	
32		SRS Document Approved	1 day	11/8/16 8:00 AM	11/8/16 5:00 PM	31
33		**⊟Software Design**	**38 days**	**11/9/16 8:00 AM**	**12/30/16 5:00 PM**	**12**
34		**⊟Architectural Design**	**9 days**	**11/9/16 8:00 AM**	**11/21/16 5:00 PM**	
35		Choose appropriate pattern to achieve development	2 days	11/9/16 8:00 AM	11/10/16 5:00 PM	
36		Specify the Structure of the system	2 days	11/11/16 8:00 AM	11/14/16 5:00 PM	35
37		Define communication, coordination, cooperation among compone	4 days	11/15/16 8:00 AM	11/18/16 5:00 PM	36
38		Architecture is finalized	1 day	11/21/16 8:00 AM	11/21/16 5:00 PM	37
39		**⊟Data Design**	**6 days**	**11/22/16 8:00 AM**	**11/29/16 5:00 PM**	**34**
40		Define Database Architecture	2 days	11/22/16 8:00 AM	11/23/16 5:00 PM	
41		Normalize ERD	4 days	11/24/16 8:00 AM	11/29/16 5:00 PM	40
42		**⊟Component Design**	**10 days**	**11/30/16 8:00 AM**	**12/13/16 5:00 PM**	**39**
43		Identify Appropriate Classes	3 days	11/30/16 8:00 AM	12/2/16 5:00 PM	
44		Identify Inputs and Ouputs	4 days	12/5/16 8:00 AM	12/8/16 5:00 PM	43
45		Identify Data Structures	3 days	12/9/16 8:00 AM	12/13/16 5:00 PM	44
46		**⊟Interface Design**	**5 days**	**12/14/16 8:00 AM**	**12/20/16 5:00 PM**	**42**
47		Design front-end	2 days	12/14/16 8:00 AM	12/15/16 5:00 PM	
48		Review of the interface (front-end)	1 day	12/16/16 8:00 AM	12/16/16 5:00 PM	47
49		Validate interface	1 day	12/19/16 8:00 AM	12/19/16 5:00 PM	48
50		Interface is finalized	1 day	12/20/16 8:00 AM	12/20/16 5:00 PM	49
51		**⊟Detail Design**	**8 days**	**12/21/16 8:00 AM**	**12/30/16 5:00 PM**	**46**
52		Create Sequence Diagram (SD)	2 days	12/21/16 8:00 AM	12/22/16 5:00 PM	
53		Create Class Diagram (CD)	2 days	12/23/16 8:00 AM	12/26/16 5:00 PM	52
54		Review overall design and fix mistakes	3 days	12/27/16 8:00 AM	12/29/16 5:00 PM	52;53
55		Detail design is completed	1 day	12/30/16 8:00 AM	12/30/16 5:00 PM	54
56		**⊟Implementation**	**29 days**	**1/2/17 8:00 AM**	**2/9/17 5:00 PM**	**33**
57		**⊟Coding**	**22 days**	**1/2/17 8:00 AM**	**1/31/17 5:00 PM**	
58		Explore IDE (Integrated Development Environment)	2 days	1/2/17 8:00 AM	1/3/17 5:00 PM	
59		Code the interface design	3 days	1/4/17 8:00 AM	1/6/17 5:00 PM	58
60		Code Whole detail design	15 days	1/9/17 8:00 AM	1/27/17 5:00 PM	59
61		Review coding	2 days	1/30/17 8:00 AM	1/31/17 5:00 PM	59;60
62		**⊟Database Connectivity**	**7 days**	**2/1/17 8:00 AM**	**2/9/17 5:00 PM**	**57**
63		Create Database	2 days	2/1/17 8:00 AM	2/2/17 5:00 PM	
64		Link database with application	1 day	2/3/17 8:00 AM	2/3/17 5:00 PM	63
65		Fix error and refine the system accroding to requirement	2 days	2/6/17 8:00 AM	2/7/17 5:00 PM	63;64
66		Final Product (Working Software)	2 days	2/8/17 8:00 AM	2/9/17 5:00 PM	65
67		**⊟Testing**	**9 days**	**2/10/17 8:00 AM**	**2/22/17 5:00 PM**	**62**
68		Verification	5 days	2/10/17 8:00 AM	2/16/17 5:00 PM	
69		Validation	3 days	2/17/17 8:00 AM	2/21/17 5:00 PM	68
70		Tested Software	1 day	2/22/17 8:00 AM	2/22/17 5:00 PM	68;69


18 Sep 16
25 Sep 16
2 Oct 16
9 Oct 16
16 Oct 16
23 Oct 16
30
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Oct 16
6 Nov 16
13 Nov 16
20 Nov 16
27 Nov 16
4 Dec 16
11 Dec 16
18 Dec 16
25 Dec 16
1 Jan
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M


6 Nov 16
13 Nov 16
20 Nov 16
27 Nov 16
4 Dec 16
11 Dec 16
18 Dec 16
25 Dec 16
1 Jan
S M T W T F S
17
8 Jan 17
15 Jan 17
22 Jan 17
29 Jan 17
5 Feb 17
12 Feb 17
T W T F S

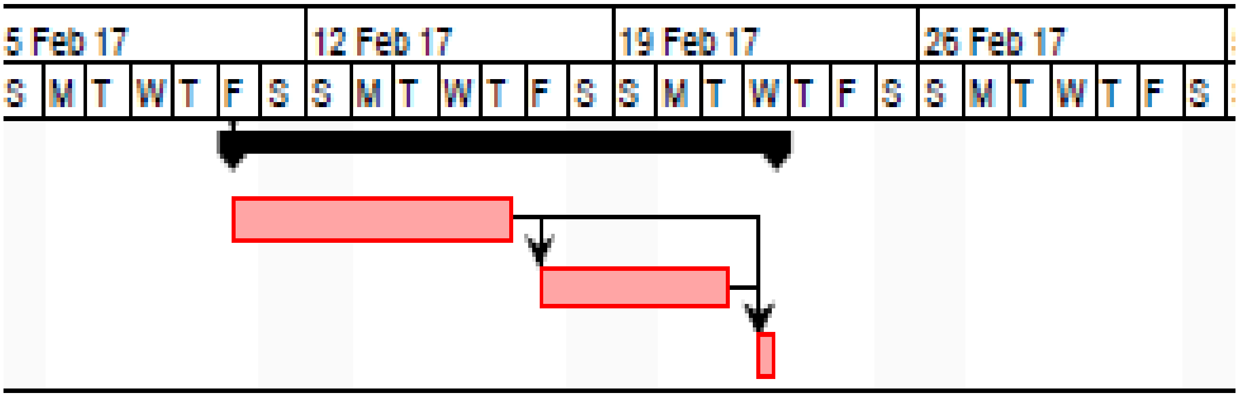


Figure 1.1 Project Plan (Gantt chart)

Chapter 2

Software Requirements Specification (SRS)

2.1 Introduction

This Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Price Comparison Web-based Application with the release number 1.0. This SRS will facilitate for an end user to completely understand of what is to be expected from the Price Comparison Web-based Application. This SRS will provide the foundation for the project. From this SRS, the application can be designed, constructed, and finally tested.

The end users will be able to use this SRS as a test to see that the software engineer will be constructing the system to their expectations. If it is not according to their expectations the end users can specify how it is not to their liking and the software engineer will change the SRS to fit the end users' needs. The SRS is organized into two main sections. The first is introduction which describes product overview and product scope, product objective and major functions of project and the second is the Specific Requirements which includes interfaces (hardware interface, user interface and software interface), system product features (functional requirements) and non-functional requirements.

2.2 Product Overview

This Price Comparison Web-based Application (PCWBA) is basically designed to replace the traditional approach. In traditional approach, one can individually visiting hundreds of different websites to look the prices of the same product of different suppliers but this can be done on single click by this application. It saves time of the user and give better user experience. It not only compare the prices but also showing interesting analysis about changing in price. It provide facility to get feedback of different customers about the product. The main focus of project is look different airlines that offer their services and to compare the prices of flights to find a low-priced ticket and also providing suitable travel options on a specific search route is very useful to travelers. Further this idea of price comparison of flights can be prolonged to all other category products that include laptops, books, computers, mobiles, etc.

2.3 Product Scope

Price Comparison Web-based Application (PCWBA) main function is to get product price from different websites. PCWBA compare these product prices with each other. This application will display interesting analysis of prices showing past history of product prices. This application will consider customer opinions on products. Customers also rate the product according to their user experience about product. Further it includes price development for your favorite product in order to have a price alert through email in case one wants to wait for an even better price. User can add product in their carts and see those products later. User can view his/her history. User can place order online if the product provide facility of online transactions. Stakeholders include buyers that buys products online and visit different websites to view the prices of same product and sellers to view the prices of their competitive sellers at same place.

2.4 Product Objectives

The primary objective of this application is to make price comparison portal for online comparison of prices. This application keep up to date offers from different merchants with regularly updated prices and comprehensive researched product information. This application are valuable to help companies to analyze prices of products of their competitive companies. Users can check out fluctuations in prices. This application are designed for consumers to find the right products at the right price. This application increase the competitiveness of product markets and enhance market efficiency. User can view the prices and can order products at one place.

2.5 Product Functions

The PCWBA major function is to get up to date offers from different websites with frequently updated prices and complete information about product. Another focus is to compare flights of different airlines to find an inexpensive tickets and also providing useful travel options on a particular search route. This idea of flight price comparison can then be stretched to all other category products like laptops, mobiles prices etc. This application will show interesting analysis of prices showing past history through graphs and charts. This application will consider customer opinions on products and will rate a product based on opinion of different customers. Another function include the registration of a user to use the features that include includes price development for user favorite product in order to have a price alert through email or message in case one wants to wait for an even better price. If a product allow online booking then the registered user can also ordered the product.

2.6 Definitions, Acronyms and Abbreviations

Table 2.1 Abbreviations used in SRS

PCWBA	Price Comparison Web-based Application
SRS	Software Requirement Specification
HCI	Human Computer Interaction

GUI	Graphical User Interface
IIS	Internet Information Server
OS	Operating System
SQL	Structured Query Language
HTTP	Hyper Text Transfer Protocol
SMTP	Simple Mail Transfer Protocol
XSS	Cross-site Scripting
ERD	Entity Relationship Diagram

2.7 User Classes and Characteristics

Users of this web application are non-registered user, registered user and administrator. It is assumed that all type of user have basic knowledge of computers and Internet browsing. Administrator is a technical person that has experience about the software development and well-known about the development of the Price Comparison Web-based Application (PCWBA). The following are the description of users:

1. Non-Registered User:
    - It can only compare the product prices and features.
    - View the analysis of price of product on the behalf of previous history.
2. Registered User:
    - It can do all work that a non-registered user can do.
    - It used the price alert feature that is to develop price alert through email in case for better price that is set by that user.
    - It give feedback about the product and rate the product with stars.
    - It also ordered the product if the product allowed online ordered.
    - It can add product in cart to view later.
3. Administrator:
    - Administrators of the PCWBA should have more knowledge of internal modules of the system and are able to rectify small problems that may arise due to database crashes and power failures. They have privileges to directly access or modify the database.

2.8 Design and Implementation Constraints

- There is at least 3GB hard disk and 1GB RAM to run this website (PCWBA).
- The information of all users, prices history and price alert must be stored in a database that is accessible by the website.
- Microsoft SQL Server 2014 will be used as SQL engine and database.

- Microsoft Visual Studio 2012 is used to build this web based application.
- ASP.Net MVC 5 is used as a server side language to develop PCWBA.
- Users may access PCWBA from any computer that has browser and an Internet connection.
- GUI is provided only in English language.

2.9 User Documentation

User manual is provided at the end of project which provide all information's such as how to use this system easily and efficiently. The user manual will include product overview, complete configuration of the used software (such as SQL server), technical details, backup procedure and contact information which will include email address.

2.10 Assumptions and Dependencies

Following are the assumptions in PCWBA:

1. PCWBA users should have some basic knowledge about data entry and must know about the basics of computer.
2. PCWBA is web based application therefore it can be used from all operating system.
3. Microsoft SQL server 2012 is used to store the database.

Following are the Dependencies in PCWBA:

1. Internet should be available 24 hours.
2. Database should be connected to the system 24 hours.

2.11 Specific Requirements

Specific requirements covering interface, functional and non-functional requirements. This is obviously the most significant part of SRS. The requirements may document external interfaces, describe system functionality and performance, and specify logical database requirements, design constraints, emergent system properties and quality characteristics.

2.11.1 External Interface Requirements

According to Richard Thayer (2002), "External interface requirements specify hardware, software, or database elements with which a system or component must interface...." This section provides information to ensure that the system will communicate properly with external components. If different portions of the product have different external interfaces, incorporate an instance of this section within the detailed requirements for each such portion.

2.11.1.1 User Interface

Price Comparison Web-based Application (PCWBA) has Friendly user interface, online help and user guide (user documentation) that is sufficient to educate the users how to use this application without facing any problems or difficulties. This application should be designed for easy to use, providing help instructions, and appropriate error messages for invalid user inputs. The color scheme is friendly and one can feel pleased to use this website. The user interface is built on the principal of "Human Computer Interaction (HCI)". GUI (Graphical User interface) is provided only in English. Shortcut keys are provided through which user can easily use PCWBA. There is help option in PCWBA that guide user to use application.

2.11.1.2 Hardware Interface

1. Client Side

Table 2.2 Client side hardware interfaces

Processor	Intel core processor
Primary Memory	512 MB RAM or more
Hard Disk Space	Minimum 100 MB

2. Server Side

Table 2.3 Server side hardware interfaces

Processor	Intel core processor
Primary Memory	1 GB RAM or more
Hard Disk Space	Minimum 3 GB

2.11.1.3 Software Interfaces

- Client:

  Web Browser (any), Operating System (any)

- Web Server:

  IIS (Internet Information Server)

- Database Server:

  Microsoft SQL Server 2014

- Development End:

  ASP.Net MVC 5, HTML, JSON, AJAX, OS (Windows)

2.11.1.4 Communications Protocols

User can interact with the web server by using HTTP/HTTPS. Website also using SMTP (Simple Mail Transfer Protocol) to avail the mail service to their clients. E-mail service is used for the verification of account registration and for price alert feature. Price Comparison Web-based Application (PCWBA) also interact with different websites using HTTP/HTTPS protocol to get prices of different products for comparison.

2.11.2 Software Product Features

The software product features illustrates the functional requirements and the major services provided by the product. In this software product feature, use case diagram and description is added to explain functional requirements and major services provided by PCWBA.

2.11.2.1 Use Case Diagram

Use Case diagram provides a succinct visual context diagram for the system, illustrating the external actors and how they use the system. A use case diagram is an excellent picture of the system context. Use-case diagram represents that how different users interact with the components of system.

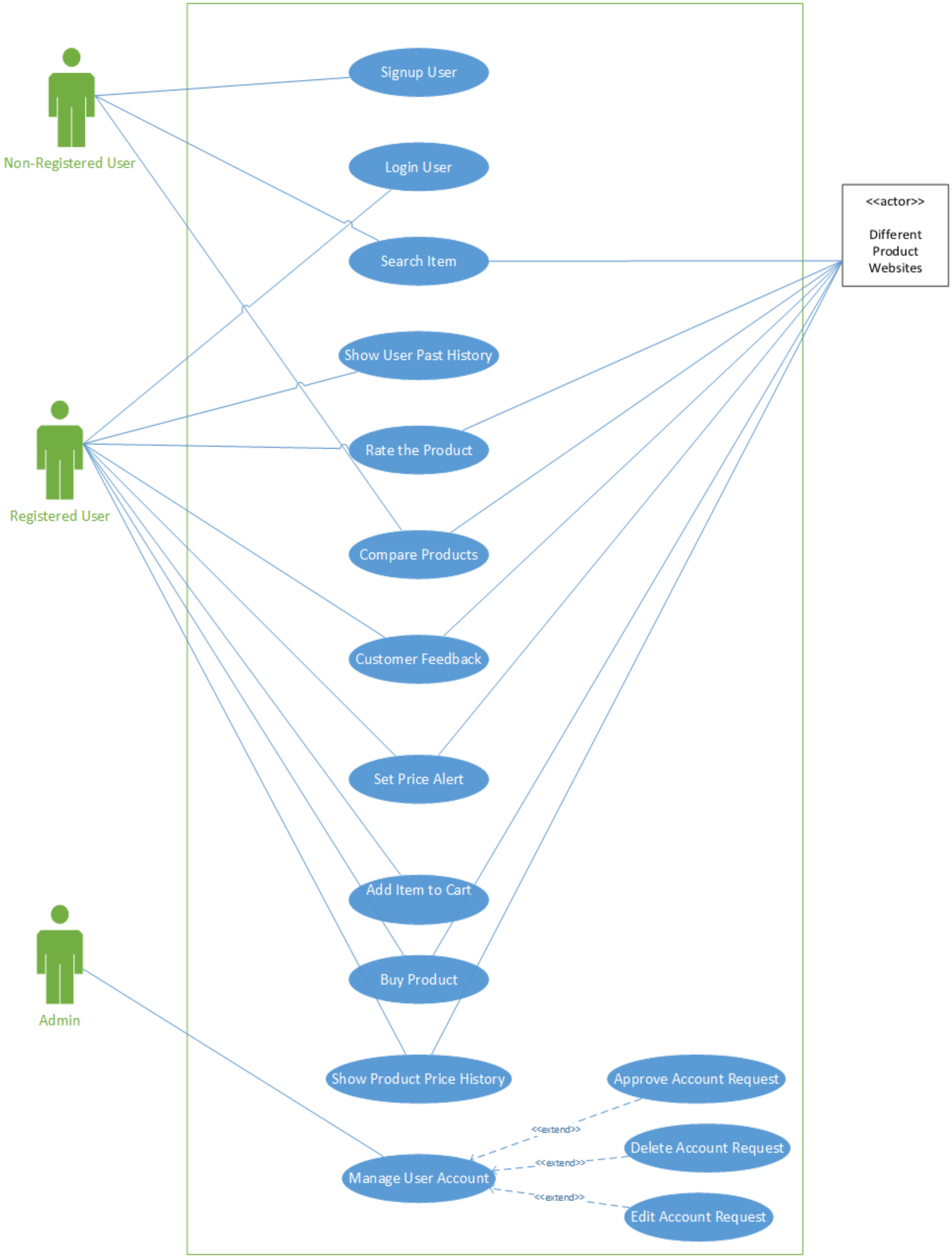


Figure 2.1 Use Case Diagram

2.11.2.2 Use Case Description

Writing use cases description is an excellent technique to understand and describe requirements. Use cases are a mechanism to help keep it simple and understandable for all stakeholders. Informally, they are stories of using a system to meet goals.

UC1: Login User

Primary actor: Registered User

Stakeholder and interest: The user logs in to authenticate his or her role in the system and to perform a task in the system.

Pre-condition: User has already account in system.

Post-condition: User logged into the system successfully and perform actions appropriate for his or her role.

Basic flow: (Success Scenario)

1) The system requests that the actor enter his/her name and password.
2) The actor enters his/her name and password.
3) The system validates the entered name and password and logs the actor into the system.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
3a) Empty fields are entered by user. Empty or half-filled form is submitted by user.
3b) Invalid Username/Password which means system determines that the username/password does not match a username/password for any account.
3c) Deactivated username and password.

Special Requirements: (None)

Technology and data variation list: (Keyboard is used to enter the information about log in. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week.

UC2: Sign Up User

Primary actor: Non-registered User

Stakeholder and interest: User wanted to signs up in system.

Pre-condition: User does not already exist account in system.

Post-condition: User sign up into the system successfully.

Basic flow: (Success Scenario)

1) The system requests that enter data in relevant fields which includes name, choose user name, password, confirm password, birthday, gender, mobile phone, email, mobile phone, and location.
2) The actor enters data in relevant fields.
3) System prompts captcha for verification anonymous user.
4) The actor check captcha for verification.
5) The system validates the fields and create the account into a system.
6) System sends mail to Unregistered User's email.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
3a) Given Information is invalid like username is incorrect or username already exist.
3b) Empty field is entered by user then system displays an error message.

Special Requirements: (None)

Technology and data variation list: (Keyboard is used to enter the information about sign up. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC3: <u>Search Item</u>

Primary actor: Non-registered User/ registered User

Stakeholder and interest: User wanted to search Item.

Pre-condition: User must be on search screen.

Post-condition: User search item successfully.

Basic flow: (Success Scenario)

1) The system requests that the actor entered the item name in filed.
2) The actor enters name of item.
3) The system get data from different website on the basis of entered item name and show prices and descriptions of products.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**

2a) Empty field is entered by user then system prompts that fill the file for successful search.
2b) Invalid character like special characters are entered by user.

Special Requirements: (None)

Technology and data variation list: (Keyboard is used to enter the information about item. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC4: <u>Show User Past History</u>

Primary actor: Registered User/Admin

Stakeholder and interest: User wanted to view his/her past history.

Pre-condition: User already logged in.

Post-condition: User view his/her past history successfully.

Basic flow: (Success Scenario)

1) User select the past history view option.
2) User select the history time like one month, two month or from beginning.
3) The user can view all history about him/her in which include price alert details, rate product details etc.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
2a) User not select history time then system prompts that select history time for successful view his/her history.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the history information option. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC5: <u>Rate the Product</u>

Primary actor: Registered User

Stakeholder and interest: User wanted to rate the product.

Pre-condition: User already logged in.

Post-condition: User rate the product successfully.

Basic flow: (Success Scenario)

1) User select the rate product.
2) User rate the product by checked the stars that ranges from zero to five.
3) User click on submit to rate the product.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) User select the rate product option without logged into the system
2a) User not checked any stars to rate the product.
3a) User not submit the ratting of the product.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to rate the product. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC6: <u>Compare Products</u>

Primary actor: Non-registered User/Registered User

Stakeholder and interest: User wanted to compare the products (price and description).

Pre-condition: User search the item.

Post-condition: User compare the product successfully.

Basic flow: (Success Scenario)

1) User select the products to compare like flights.
2) User click on compare to view the comparison of product.
3) User view the products comparison.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
2a) User select no product for comparison. System prompts that minimum two products required for comparison.

2b) User only select one product for comparison. System prompts that minimum two products required for comparison.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the products for comparison. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC7: Customer Feedback (Comments)

Primary actor: Registered User

Stakeholder and interest: User wanted to give comments on the products.

Pre-condition: User logged into the system.

Post-condition: User give comment on the product successfully.

Basic flow: (Success Scenario)

1) User select the products to give comments like PIA flight KCH to ISB.
2) User write the comment and submit it.
3) System validate the comment like no abusive language is used.
4) User successfully placed the comment.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) User wanted to give feedback without logged into the system.
3a) User submit comment box empty. System prompt to fill the comment.
3b) User use abusive language in comments. System warns the user.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the products for feedback and keyboard is used to entered comment in comment box. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC8: Set Price Alert

Primary actor: Registered User

Stakeholder and interest: User wanted to set price alert for the favorite product

Pre-condition: User logged into the system.

Post-condition: User set price alert for the favorite product.

Basic flow: (Success Scenario)

1) User select the products to set price alert.
2) User set price alert for that product.
3) User select the method through which message is informed that his/her entered price is reached like through text message or email.
4) User submit price alert and system validates.
5) User successfully set price alert.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) User wanted to set price alert without logged into the system.
3b) user does not select any method for price alert message.
4a) User submit price alert text box empty. System prompt to fill the comment.
4b) User entered invalid data in price alert text box.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the products for price alert and keyboard is used to entered data in the fields. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC9: <u>Add Item to Cart</u>

Primary actor: Registered User

Stakeholder and interest: User wanted to add item in the cart.

Pre-condition: User has a registered account in the system and has logged on to the website.

Post-condition: User add item in the cart successfully.

Basic flow: (Success Scenario)

1) User chooses a product they wish to buy and clicks the ‘Add to cart’ button to send their product to the cart.
2) User is brought to the cart page and displayed the items they have chosen to buy.

3) User successfully add item in the cart.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) User wanted to add item in the cart without logged into the system.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the products to add in the cart. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC10: <u>Logout User</u>

Primary actor: Registered User

Stakeholder and interest: User wanted to logout to the system.

Pre-condition: User has a registered account in the system and has logged on to the website.

Post-condition: The user must be logged in to the system.

Basic flow: (Success Scenario)

1) The user clicks logout to close his current session.
2) The user can no longer access nonpublic information. User successfully logout.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the logout option in the application. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC11: <u>Buy Product</u>

Primary actor: Registered User

Stakeholder and interest: User wanted to buy the product.

Pre-condition: User has a registered account in the system and has logged on to the website.

Post-condition: user buy the product successfully.

Basic flow: (Success Scenario)

1) The user selects the product to buy.
2) User entered relevant details in the form to buy
3) The user can buy the product successfully.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) Product does not provide online order or online transactions.
2a) Entered data is invalid or empty fields are entered by user.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the buy option in the application and keyboard is used to fill information in the fields. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC12: <u>Show Product Price Past History</u>

Primary actor: Registered User

Stakeholder and interest: User wanted to see the product price history.

Pre-condition: User has a registered account in the system and has logged on to the website.

Post-condition: user view the product past history successfully.

Basic flow: (Success Scenario)

1) The user selects the product to view history.
2) The user can view the product price history successfully in the form of graph or chart.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
1b) User select product without logged on to the system.

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the product price history option in the application. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC13: <u>Manage User Account</u>

Primary actor: Admin

Stakeholder and interest: Add wanted to manage accounts of the users.

Pre-condition: Admin has an account in the system and given all the root level privileges to add user in the system and has logged on to the website.

Post-condition: Admin manages account successfully.

Extension points: Approved Account Request, step2. Delete Account Request, step 2. Edit Account Request, step 2.

Basic flow: (Success Scenario)

1) Admin selects the manage account option.
2) Admin complete the form for the manage account.
3) Admin submit the form for the manage account.
4) Admin manage account successfully.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**
3a) Admin submit form without filling full.

Special Requirements: (None)

Technology and data variation list: (Keyboard is used to enter the related information of faculty member. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC14: <u>Approved Account Request</u>

Primary actor: Admin

Stakeholder and interest: Admin wanted to approve user account request.

Pre-condition: Admin has an account in the system and given all the root level privileges to delete users of the system and logged on to the website.

Post-condition: Admin approved account successfully.

Trigger: Admin wanted to approve the registration of account.

Basic flow: (Success Scenario)

1) Admin selects the approve account option in the manage account option.
2) Admin view the list of request to approve account.
3) Admin select the user and click approve to register user in system.
4) Admin approved account successfully.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the user to approve account request of the user. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC15: <u>Delete Account Request</u>

Primary actor: Admin

Stakeholder and interest: Admin wanted to delete user accounts.

Pre-condition: Admin has an account in the system and given all the root level privileges to modify the system and has logged on to the website.

Post-condition: Admin deleted account successfully.

Trigger: Admin wanted to approve the delete account request.

Basic flow: (Success Scenario)

1) Admin selects the delete account option in the manage account option.
2) Admin view the list of request to delete account.
3) Admin select the user and approve delete request to delete the user record from the system.
4) Admin approved delete account request successfully.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the user to delete request of the user. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

UC16: **Edit Account Request**

Primary actor: Admin

Stakeholder and interest: Admin wanted to edit user account information's.

Pre-condition: Admin has an account in the system and given all the root level privileges to modify the system and has logged on to the website.

Post-condition: Admin edit user account request successfully.

Trigger: Admin wanted to approve edit account request.

Basic flow: (Success Scenario)

1) Admin selects the edit account option in the manage account option.
2) Admin view the list of request to edit account information.
3) Admin select the user and approve edit request to edit the user record in the system.
4) Admin approved edit account request successfully.

Alternate flows: (Failure Scenario)

1a) Server fails at any time. In this case, data is shifted to backup server**.**

Special Requirements: (None)

Technology and data variation list: (Mouse is used to select the user and keyboard to edit the user account information. Monitor is to display output or touch screen for enter or display).

Frequency of occurrence: Several times in a day or week**.**

2.11.3 Software System Attributes

Software system attributes are non-functional requirements that are used to evaluate the performance of a system. It is imperative for operators to look at the products they are purchasing from all the quality attributes that may affect them in future.

2.11.3.1 Reliability

PCWBA (Price Comparison Web-based Application) is 90% reliable. There is no occurrence of failure in case of submitting empty fields or any invalid data that is entered by user then application handles this user action by giving appropriate message. The database may get crashed at any certain time due to virus, operating system failure or server problem. Therefore, there is database backup that used in case of database crash. Fault tolerance will also use in case of server failure or crash.

2.11.3.2 Availability

High Availability is the measure of the quality of a software to keep functioning in spite of problems. The system shall be available 24 hours. All the users are able to access the system at any time.

2.11.3.3 Security

This application is a web-based so, it will be hosted on server and all the user data will be kept on Microsoft SQL Server database. PCWBA used secured database that required information like server type, server name and password to access database. Application should be able to protect user password in database after applying cryptography algorithms. Access to the various subsystems will be protected by a user authentication (log in screen) that requires a user name and password. All input need to be encoded and validated to prevent SQL injection and XSS (Cross Site Scripting). User can only perform operation under the permission. System will have different types of users (described in user class and characteristics) and every user has access constraints.

2.11.3.4 Maintainability

There should be aspect of maintainability for the system. This application should be easy to extend because it follows the object oriented approach. The code should be written in a way that it favors implementation of new functions. PCWBA is built under the principles of software engineering so it is easy to change in case of changing in requirements.

2.11.3.5 Portability

This is web based application so main purpose of developing web-based application is to improve the portability of system. To improve portability, system should run on variety of platforms and variety of connection speeds. System should be lightweight so that it can run on a machine with slow internet connection. This is web application so it is accessed from all platforms like Windows, Linux, Ubuntu and vice versa.

2.11.3.6 Performance

Performance requirements define acceptable response times for system functionality:

1. The load time for user interface screens shall take no longer than two seconds.
2. The log in information shall be verified within ten seconds.
3. Queries shall return results within five to ten seconds.
4. Product details (Price information) get from different website within five to ten seconds.
5. Expected number of simultaneous user should be at least 600. System should be able to deal with at least 600 users at the same time.

2.11.4 Database Requirements

Microsoft SQL Server will be used as SQL engine and database.

Entity Relationship Diagram

ERD is a top-down approach to database design that begins by identifying the important data called entities and relationships between the data that must be represented in the model. We then add more details such as the information we want to hold about the entities and relationships called attributes and any constraints on the entities, relationships, and attributes. ERD is a Logical Data Diagram that does not define actual data storage. It defines the Entities at a conceptual level. The ER model supported with additional semantic concepts is called the Enhanced Entity–Relationship (EER) model. In concepts of the EER model, namely specialization/generalization, aggregation, and composition. In my database requirement there is a concept of generalization so, EER diagram is draw to illustrate the tables and their relations.

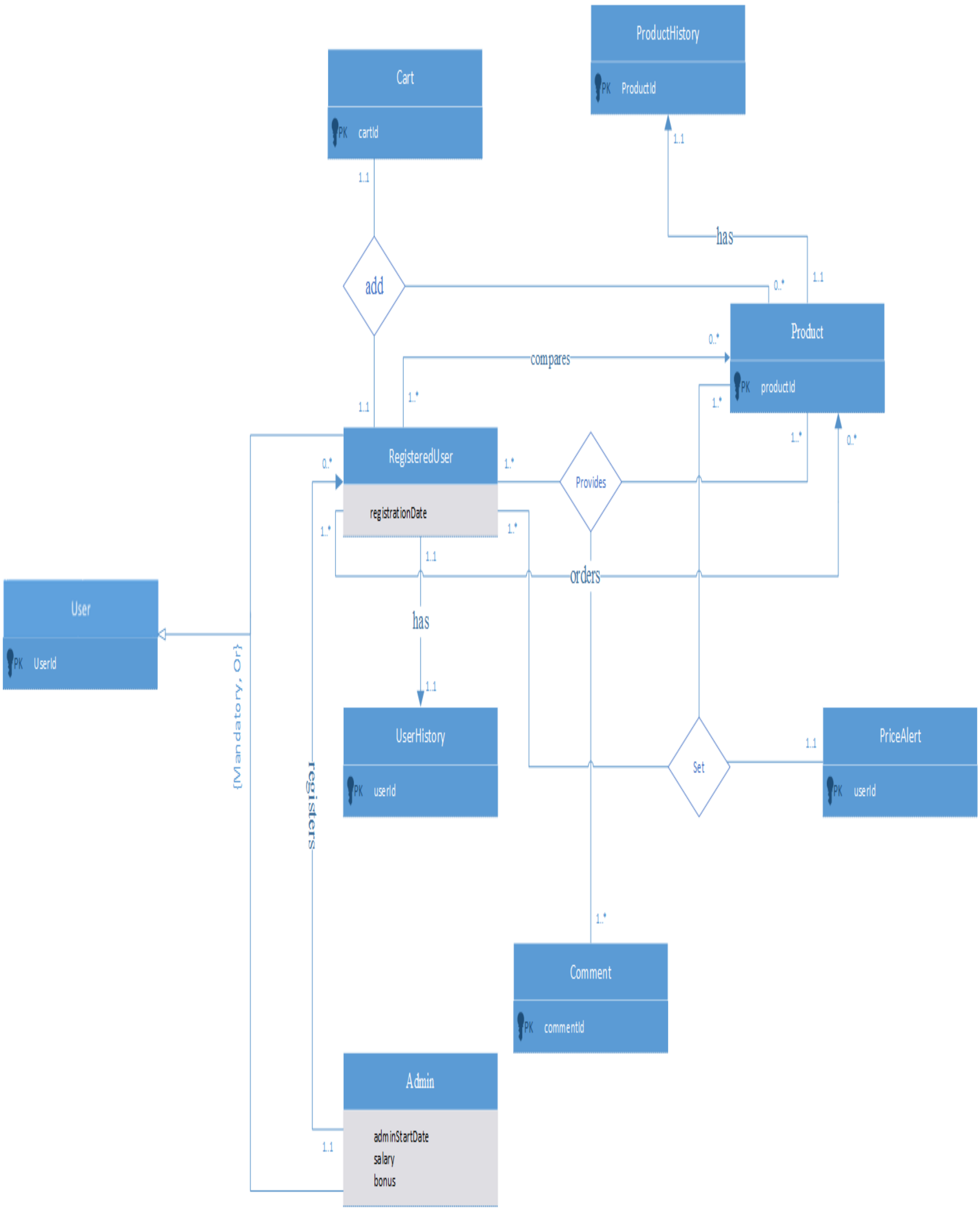


Figure 2.2 Enhanced Entity-Relationship Diagram (EERD)

2.12 Domain Model

A domain model is a visual representation of conceptual classes or real-world objects in a domain of interest. They have also been called conceptual models, domain object models, and analysis object mode. Using UML notation, a domain model is illustrated with a set of class diagrams in which no operations are defined.

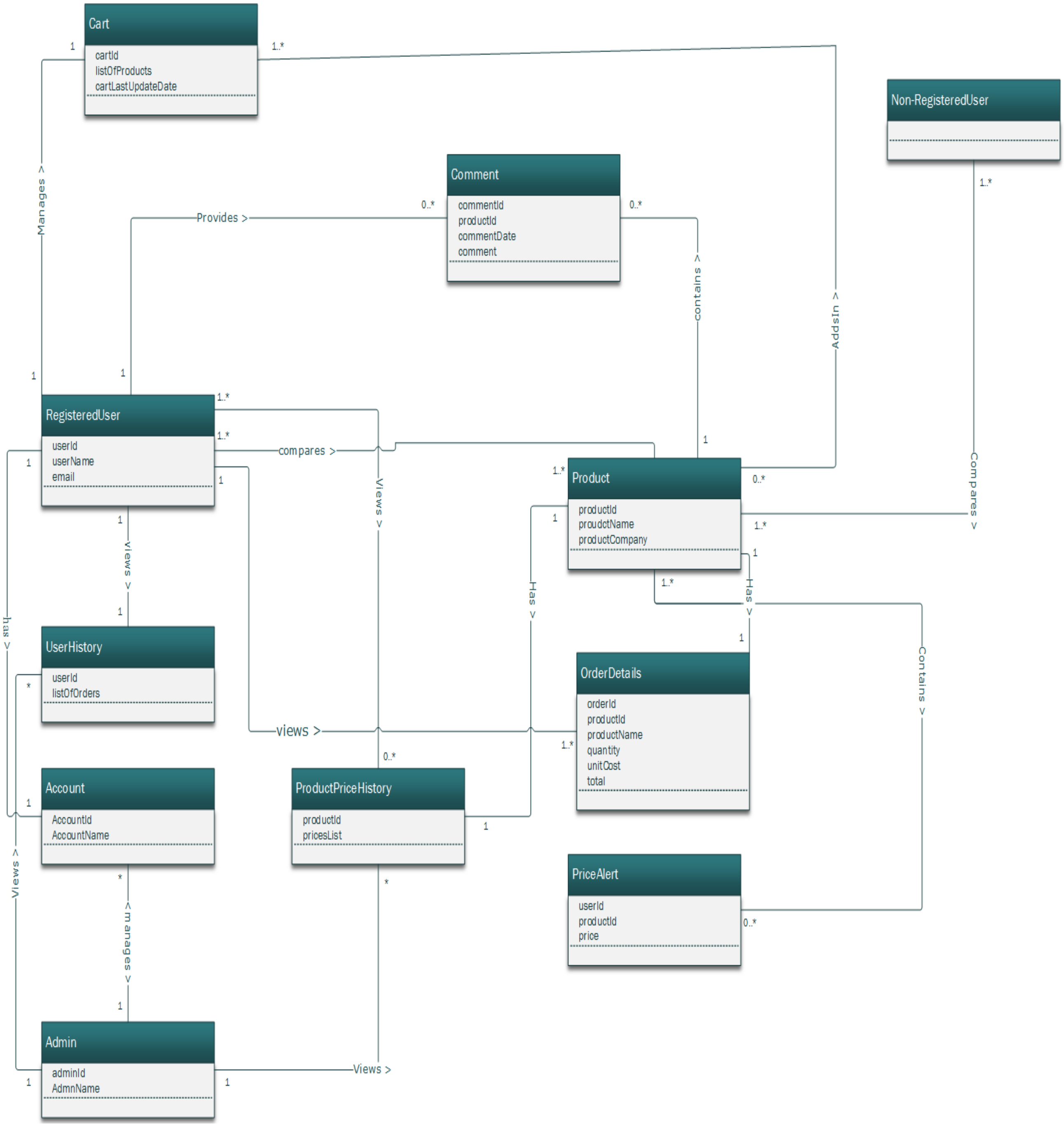


Figure 2.3 Domain Model

Chapter 3

Software Design Description

3.1 Introduction

A software design description (SDD) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. This SDD contains system architecture design, detailed description of components, user interaction design which contains screen images (interfaces) and sequence diagram (SD). This SDD also contain design class diagram (DCD) that 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/modules), and their relationships.

3.1.1 Requirements Traceability Matrix

Requirement Traceability Matrix or RTM captures all requirements proposed by the client or development team.

Table 3.1 Requirements Traceability Matrix

Req. Id	Requirement Name	Interface Screen	Test Case	Sequence Diagram	Domain Model	Class Diagram
UC1	Login	IF-2	TC-001	SD-2	Account	Registered User
UC2	Signup	IF-1	TC-002	SD-1	Account	RegisteredUser
UC3	Search Item	IF-3	TC-003	SD-3	Product	Product
UC4	Show User Past History	IF-17	TC-0010	SD-5	User History	User History
UC5	Rate The Product	IF-5	TC-007	SD-12	Product	Product
UC6	Compare Product	IF-4	TC-003	SD-4	Product	Product
UC7	Customer Feedback	IF-5	TC-009	SD-12	Product History	Product History
UC8	Set Price Alert	IF-6	TC-005	SD-8	Price Alert	Price Alert
UC9	Add Item To Cart	IF-10	TC-0012	SD-7	Cart	Cart
UC10	Logout User	IF-10	TC-0012	SD-11	Account	Account
UC11	Order Product	IF-11	TC-0013	SD-10	Order	Order
UC12	Show Product Past History	IF-7	TC-006	SD-6	Product History	Product History

UC13	Manage User Account	IF-12	-	-	Admin	Admin
UC14	Approved Account Request	IF-13	TC-004	SD-1	Admin	Admin
UC15	Delete Account Request	IF-15	TC-008	SD-7	Admin	Admin
UC16	Edit Account Request	IF-14	TC-0011	-	Admin	Admin

3.2 System Architecture Design

An architecture is the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style. Architectural design is the resolution of the requirements in the design of the software, the hardware and networking, operations, policies, and so forth.

3.2.1 Chosen System Architecture

3-Tier architecture is used to make the Price Comparison Web-based Application (PCWBA). Further, in 3-Tier architecture I used Model View Controller (MVC) to build PCWBA. Basically, Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage and data access are developed and maintained as independent modules or most often on separate platforms. Basically, there are 3 layers, tier 1 (presentation tier, GUI tier), tier 2 (business objects, business logic tier) and tier 3 (data access tier). These tiers can be developed and tested separately. The Three-tier Architecture contains:

Presentation tier

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing and shopping cart contents. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI).

Application tier (business logic, logic tier, or middle tier)

The logical tier is pulled out from the presentation tier and, as its own layer, it controls an application’s functionality by performing detailed processing.

Data tier

The data tier includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data. The data access layer should provide an API to the application tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms. As with the separation of any tier, there are costs for implementation and often costs to performance in exchange for improved scalability and maintainability.

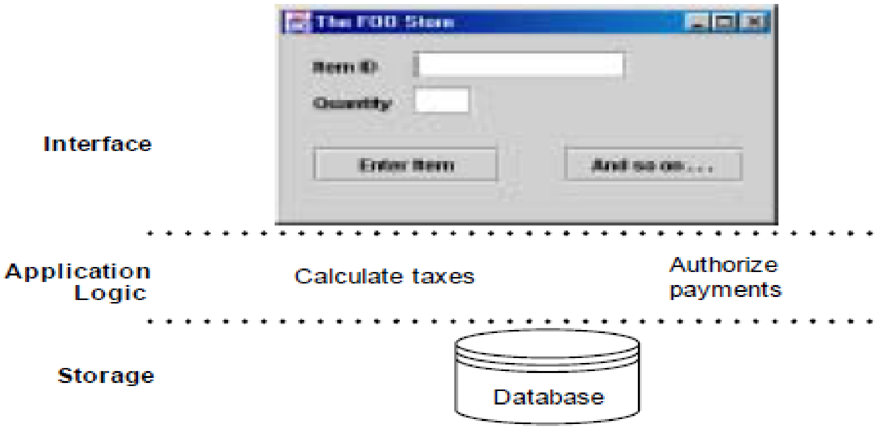


Figure 3.1 Classical 3-tier Architecture (Thinner Client)

Model **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. It is the one of the way to provide Three-tier architecture to your application. MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows. A Model View Controller pattern is made up of the following three parts:

Model

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

The view

A presentation of data in a particular format, triggered by a controller's decision to present the data. They are script based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

Controller

The controller is responsible for responding to user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

The following figure illustrates the interaction between Model, View and Controller.

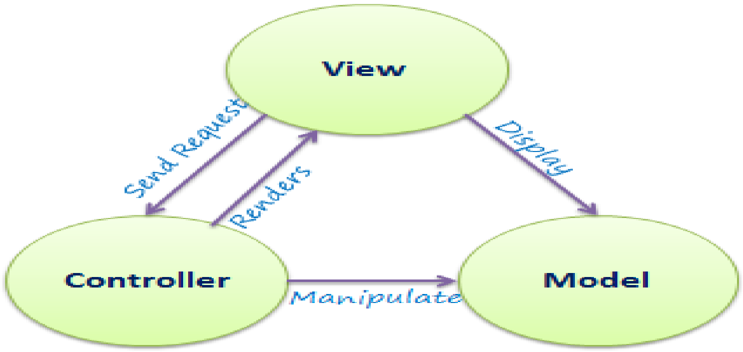


Figure 3.2 MVC Architecture

The following figure illustrates the flow of the user's request in ASP.NET MVC.

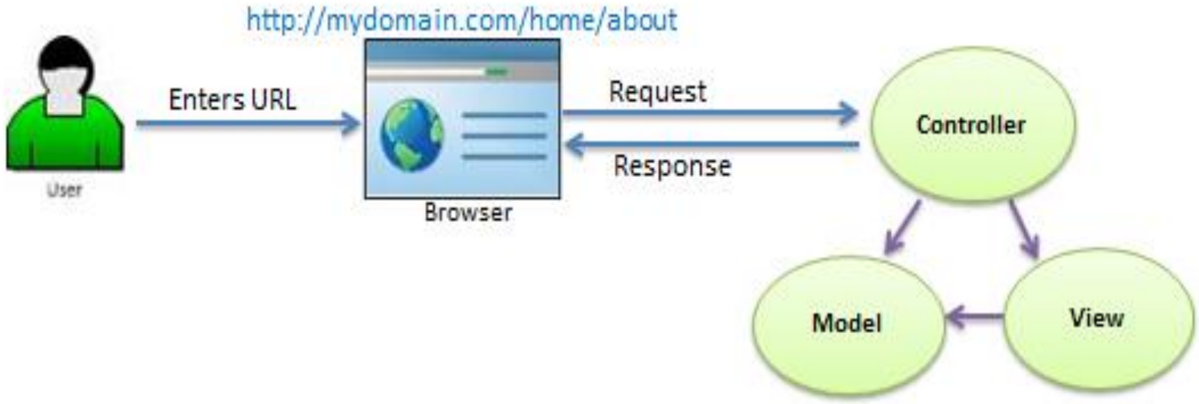


Figure 3.3 Request/Response in MVC Architecture

As per the above figure, when the user enters a URL in the browser, it goes to the server and calls appropriate controller. Then, the Controller uses the appropriate View and Model and creates the response and sends it back to the user. If we express Model View Controller (MVC) in 3-tier Architecture style. We can relate as follows:

Presentation Tier: "Controllers and Views" from MVC Pattern.

Business Tier: "Model (Data)" from MVC Pattern.

Data Access Tier: Original Data Access Tier.

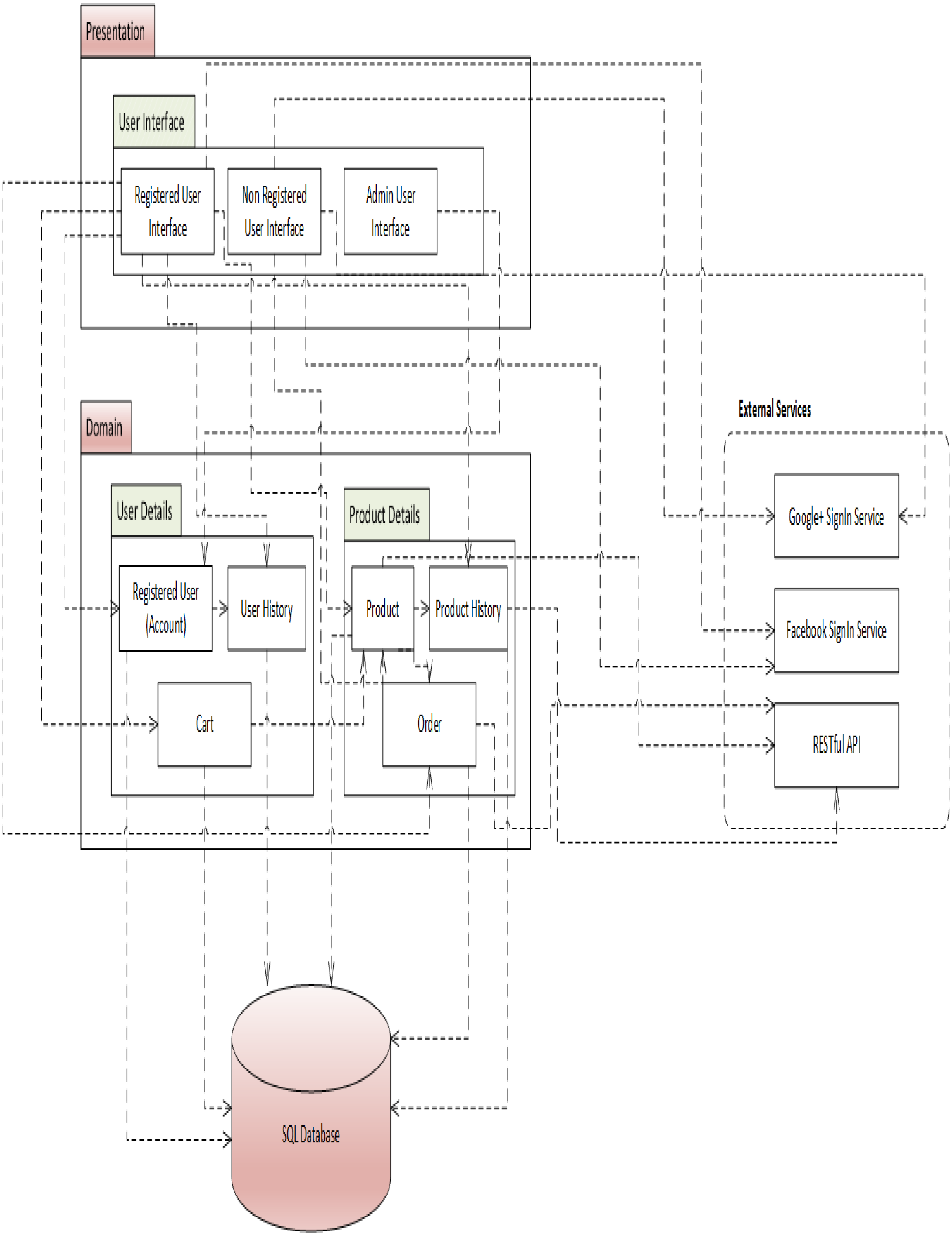


Figure 3.4 Logical View of layers in PCWBA

3.2.2 Discussion of Alternative Design

We can also choose 2-tier architecture design but it has some issues like data integrity issues. As we know that 2-tier architecture is an application architecture that separates the code base from the DBMS (database management system). The application code resides on a "fat client" or "Think client" which is used to process the data. The problem with 2-tier applications is that they become complex and hard to support as the user base increases in size. So, I will go to 3-tier architecture that better re-use, improved security that client is not direct access to database and better performance because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.

3.3 Detailed Description of Components

Component Diagram

In the Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Table 3.2 Components Overview

Component Or Module	Description
Product	Encapsulate the product details
Order	Encapsulate the order details of product
Web Service	Service that provide facility to get the details of products from different websites.
Registered User	User to view products, maintain cart and has history.
Admin	Approved Registered User to perform actions
Product History	Encapsulate history of product
User History	Track record of user actions in PCWBA
Cart	Used to keep record of user favorite products for future. Cart contains Products that is added by registered user.

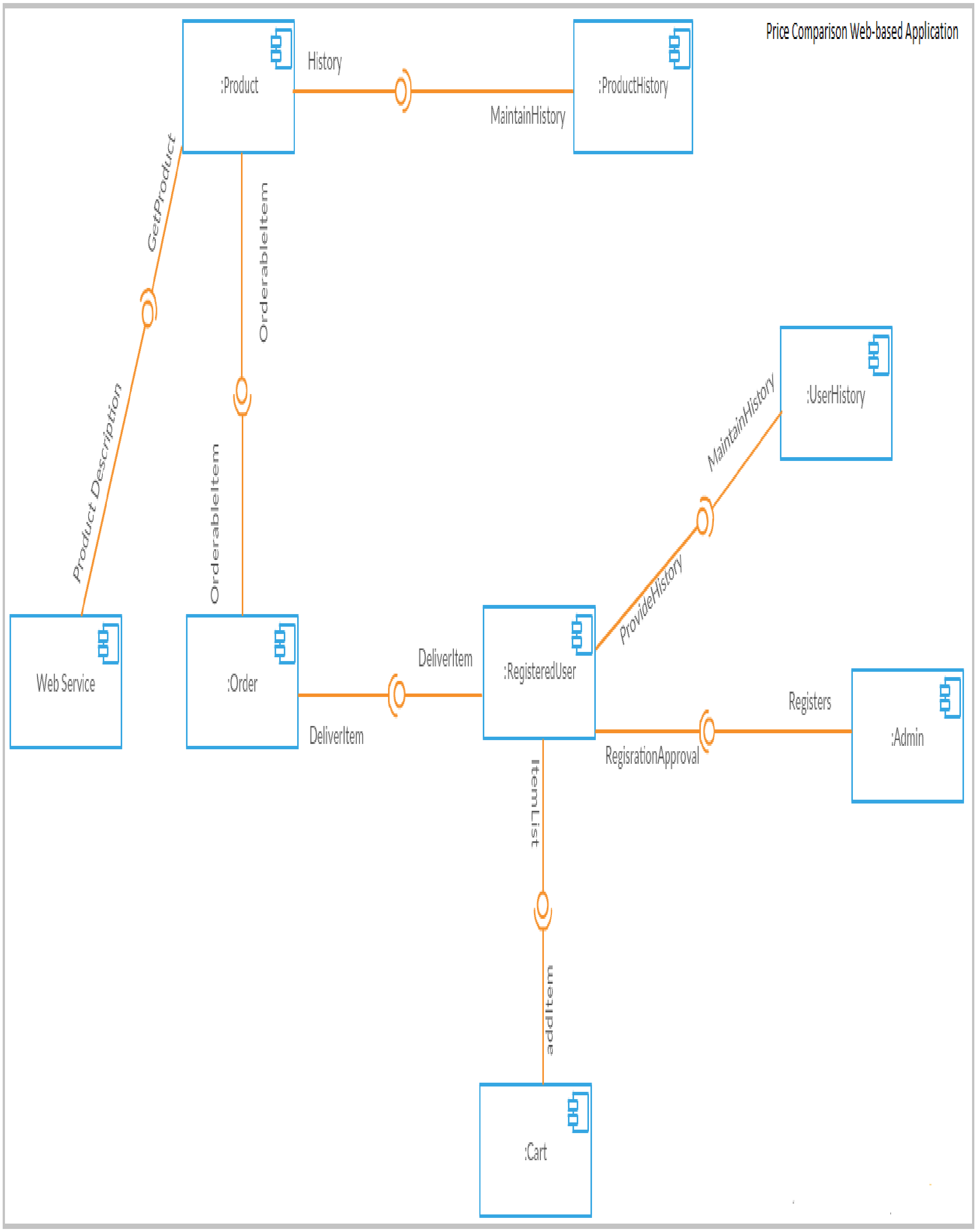


Figure 3.5 Component Diagram of PCWBA

3.4 User Interaction Design

3.4.1 Screen Images

Screen images is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. It is an activity that can occur in software development. Screen images (interfaces) has several benefits such as software designer and implementer can get valuable feedback from the users early in the project. Screen images shows that shows what the screens may look like after the completion of project. Screen images are basically prototyping of system so it can improve the quality of requirements and specifications provided to developers. Following are the screens in PCWBA (Price Comparison Web-based Application).

3.4.1.1 IF-1: Sign Up Screen

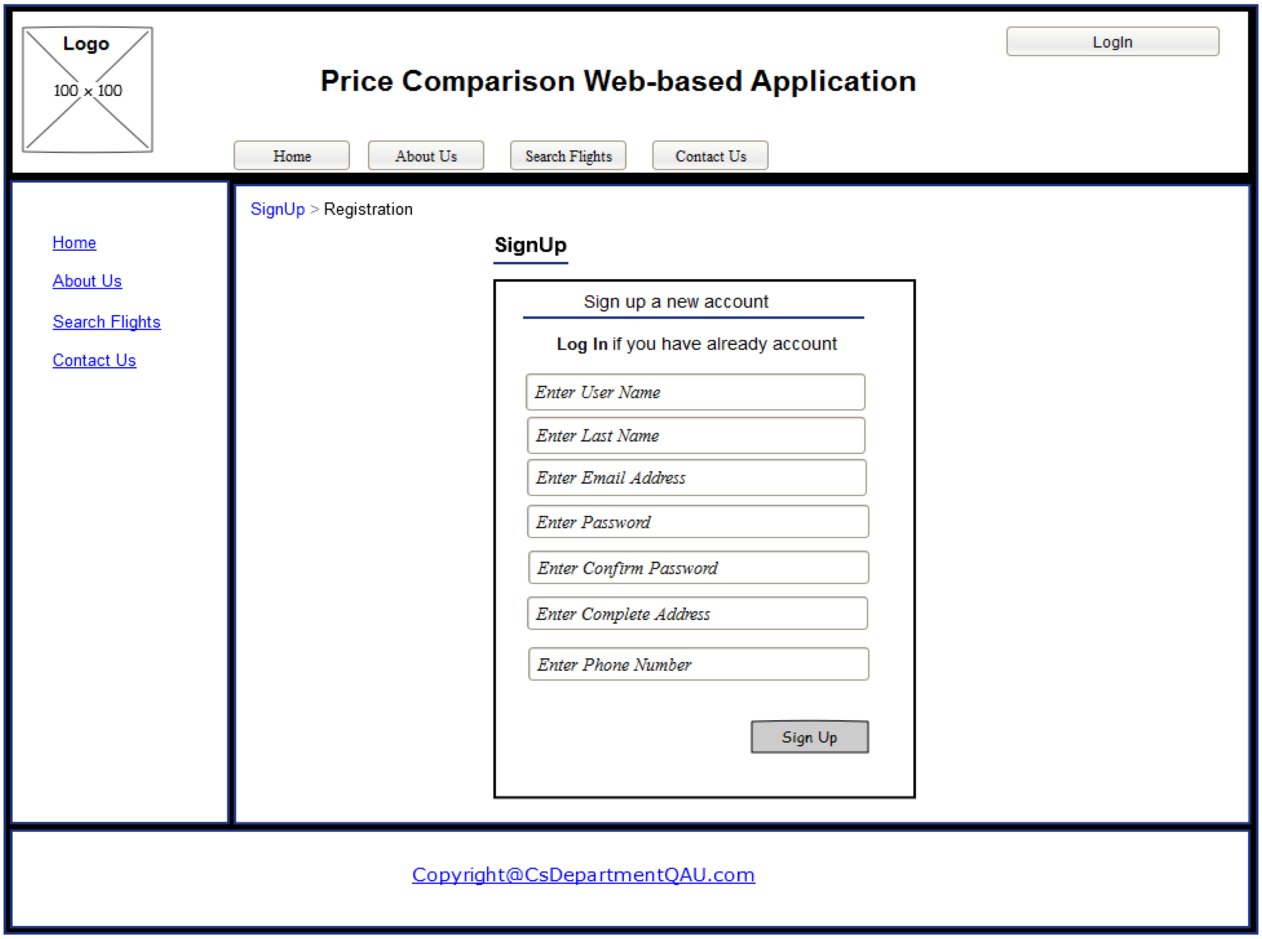


Figure 3.6 Sign up Interface

3.4.1.2 IF-2: Login Screen

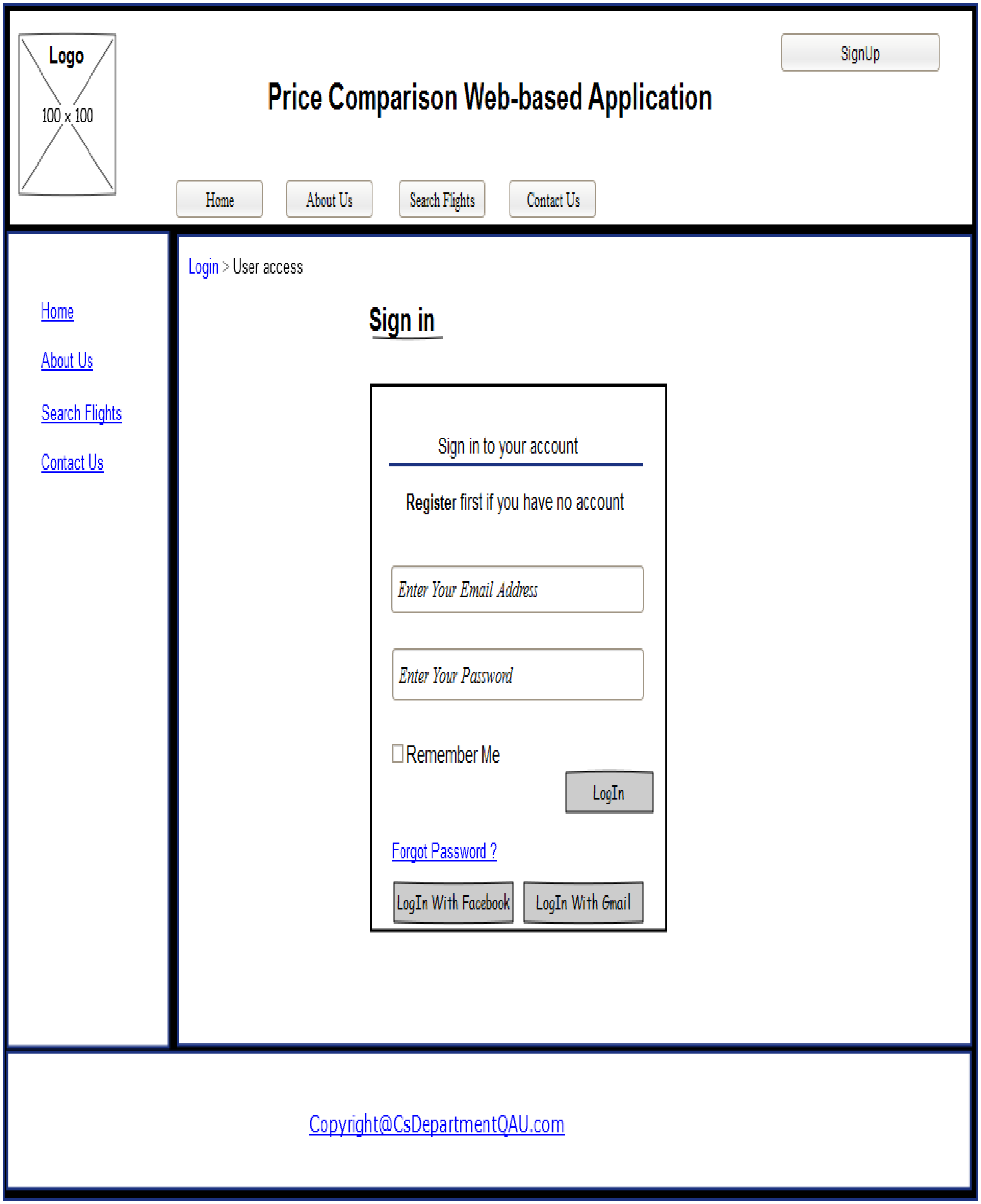
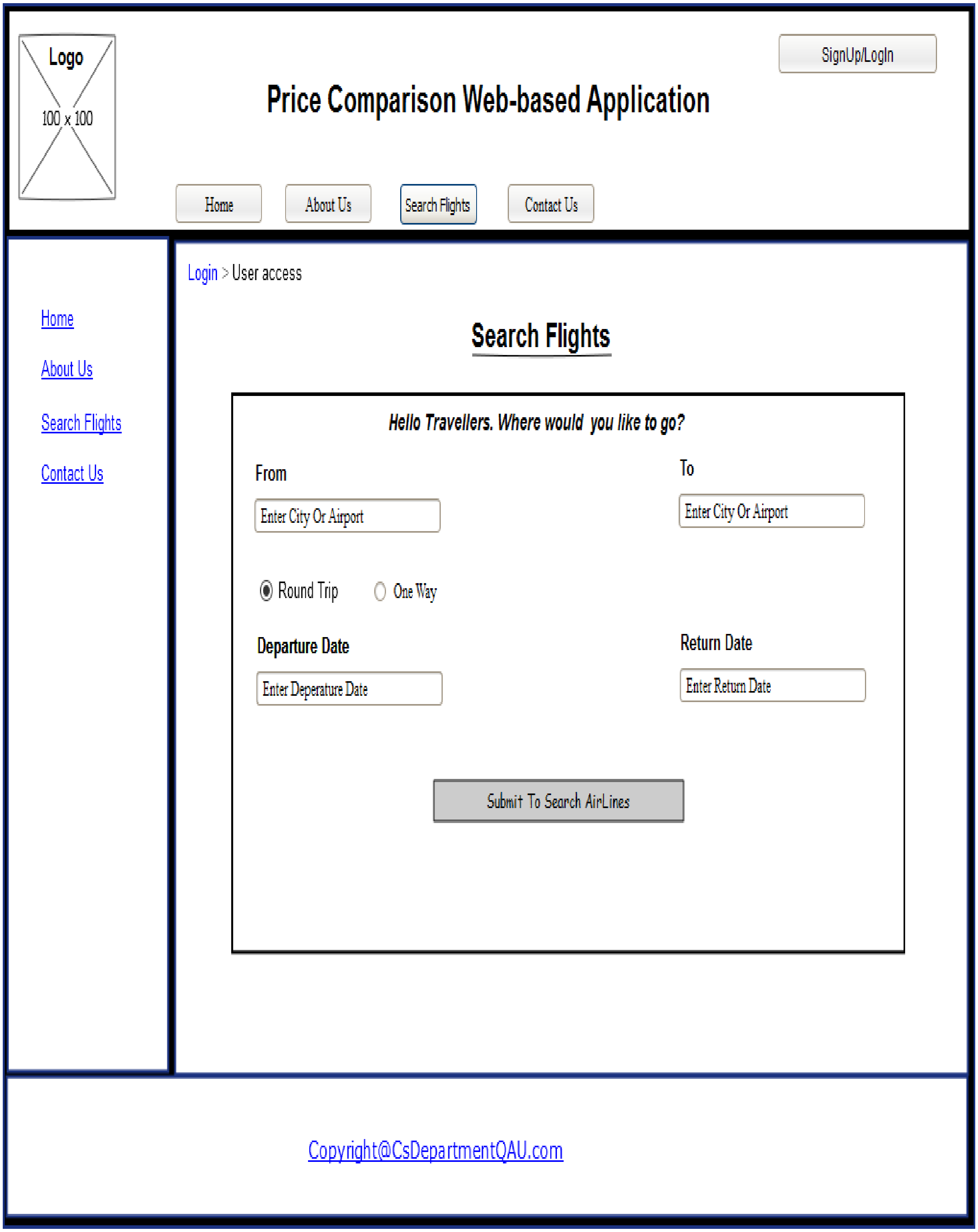


Figure 3.7 Login Interface

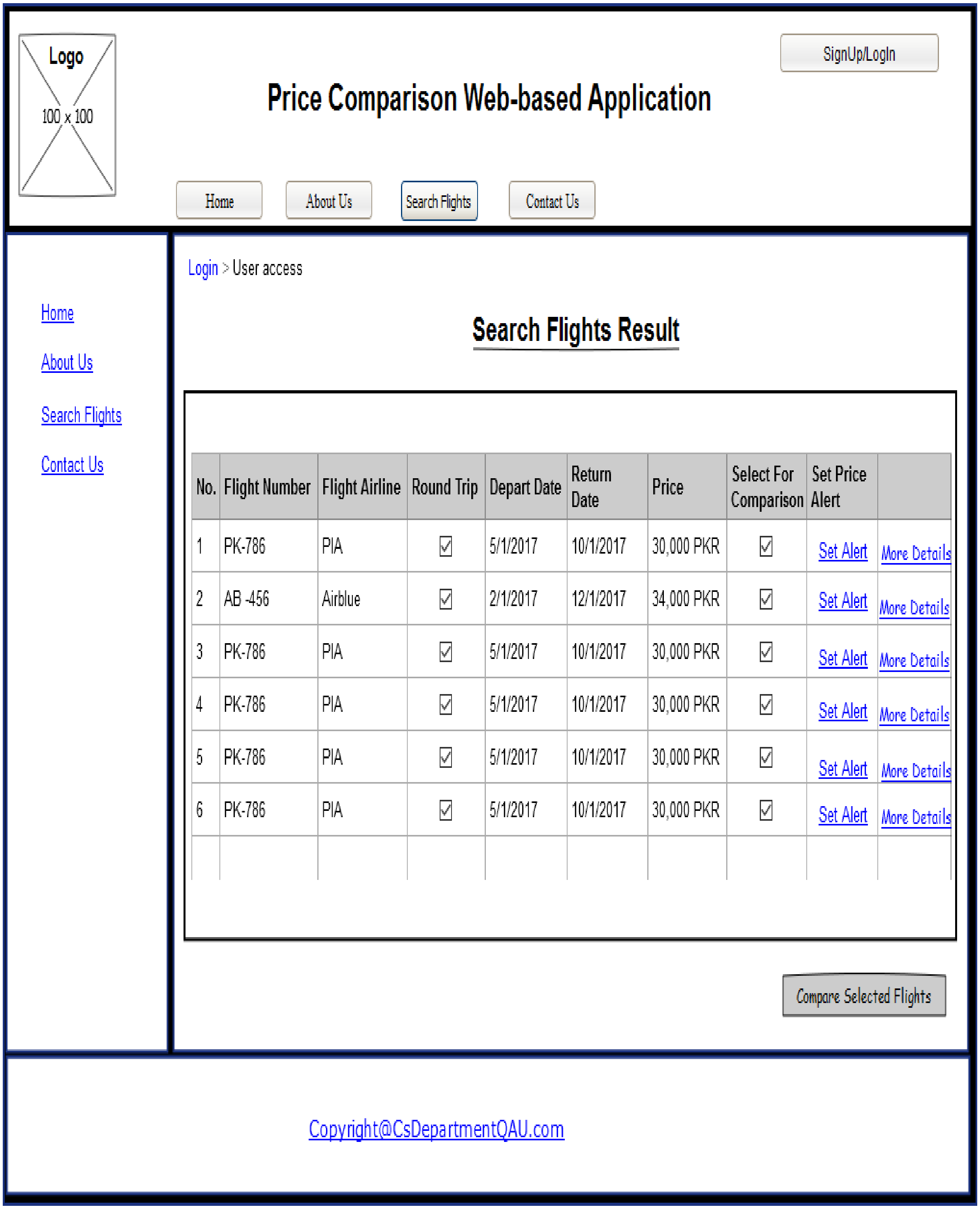
3.4.1.3 IF-3: Search Flight Screen



Hello Travellers. Where would you like to go?

Figure 3.8 Search Flights Interface

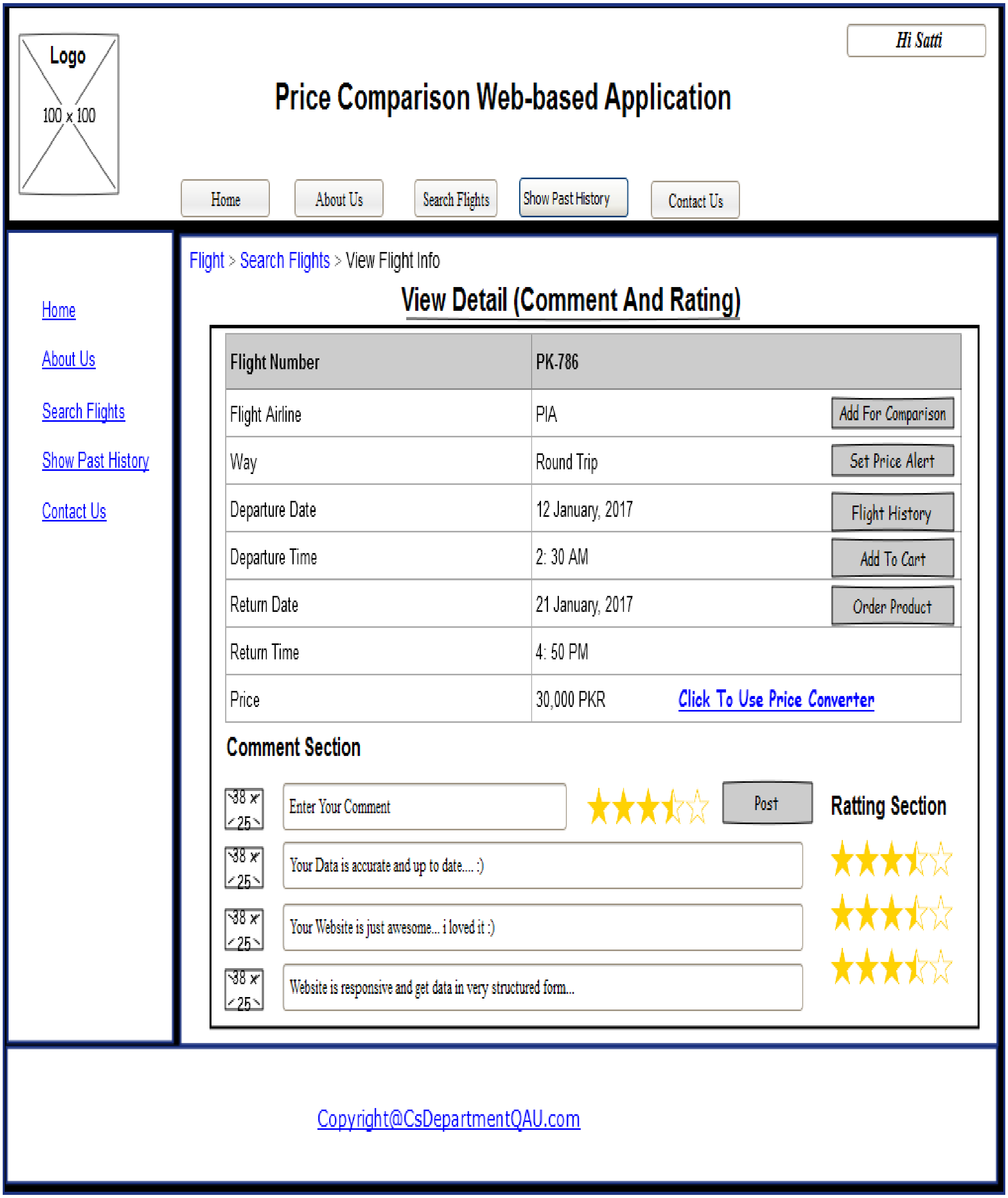
3.4.1.4 IF-4: Search Flight Response Screen



No.	Flight Number	Flight Airline	Round Trip	Depart Date	Return Date	Price	Select For Comparison	Set Price Alert	
1	PK-786	PIA	☑	5/1/2017	10/1/2017	30,000 PKR	☑	Set Alert	More Details
2	AB -456	Airblue	☑	2/1/2017	12/1/2017	34,000 PKR	☑	Set Alert	More Details
3	PK-786	PIA	☑	5/1/2017	10/1/2017	30,000 PKR	☑	Set Alert	More Details
4	PK-786	PIA	☑	5/1/2017	10/1/2017	30,000 PKR	☑	Set Alert	More Details
5	PK-786	PIA	☑	5/1/2017	10/1/2017	30,000 PKR	☑	Set Alert	More Details
6	PK-786	PIA	☑	5/1/2017	10/1/2017	30,000 PKR	☑	Set Alert	More Details

Figure 3.9 Search Flights Response Interface

3.4.1.5 IF-5: View Details Flight Screen



Price Comparison Web-based Application

Flight > Search Flights > View Flight Info

View Detail (Comment And Rating)

Flight Number	PK-786
Flight Airline	PIA
Way	Round Trip
Departure Date	12 January, 2017
Departure Time	2: 30 AM
Return Date	21 January, 2017
Return Time	4: 50 PM
Price	30,000 PKR Click To Use Price Converter

Comment Section

Enter Your Comment ★★★⯪☆

Rating Section

Your Data is accurate and up to date.... :) ★★★⯪☆

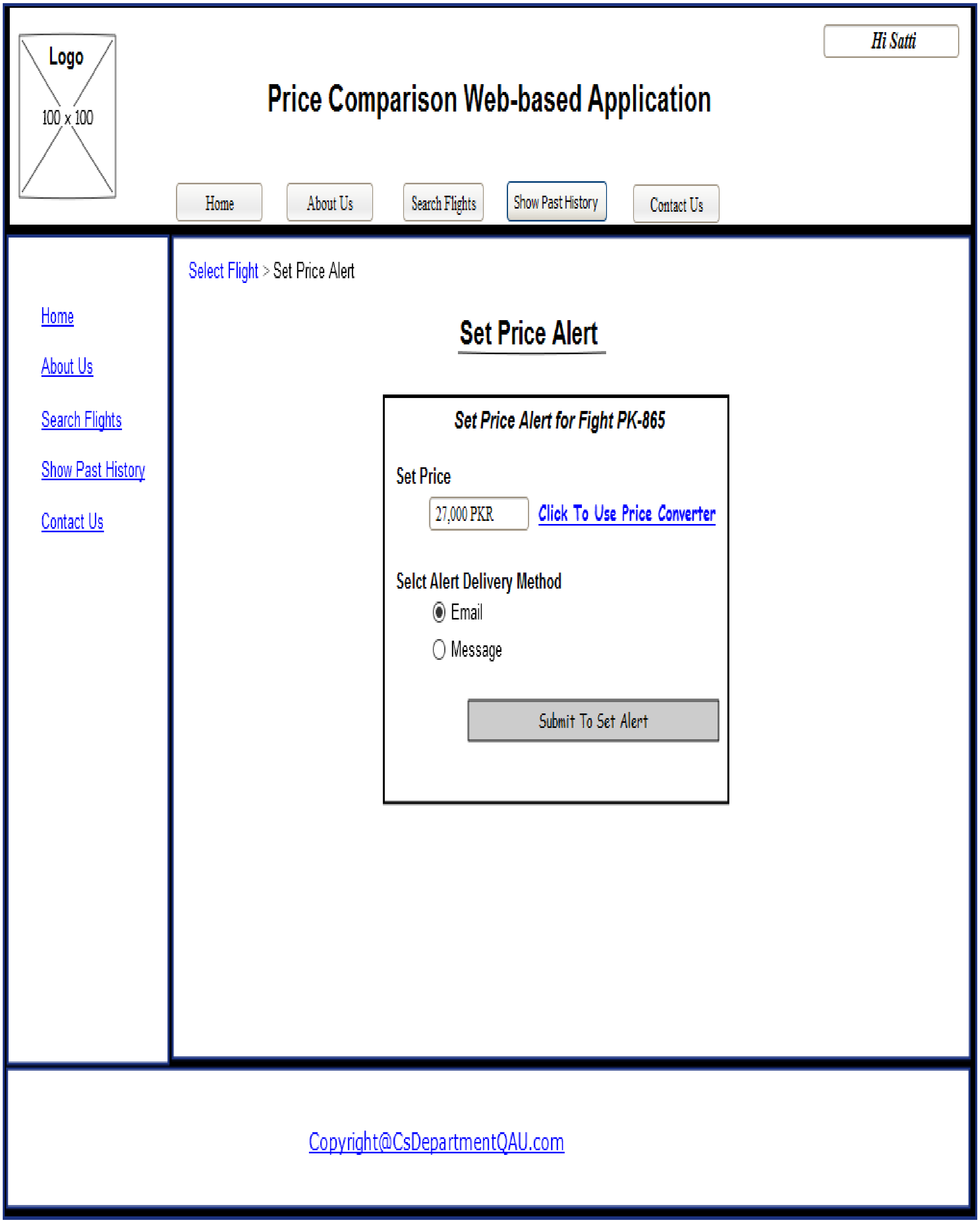
Your Website is just awesome... i loved it :) ★★★⯪☆

Website is responsive and get data in very structured form... ★★★⯪☆

Copyright@CsDepartmentQAU.com

Figure 3.10 View Detail Flight (Rate and Comment) Interface

3.4.1.6 IF-6: Set Price Alert for Flight



Set Price Alert for Fight PK-865

Set Price

27,000 PKR

Selct Alert Delivery Method

● Email

○ Message

Figure 3.11 Set Price Alert Interface

3.4.1.7 IF-7: Search Flight History

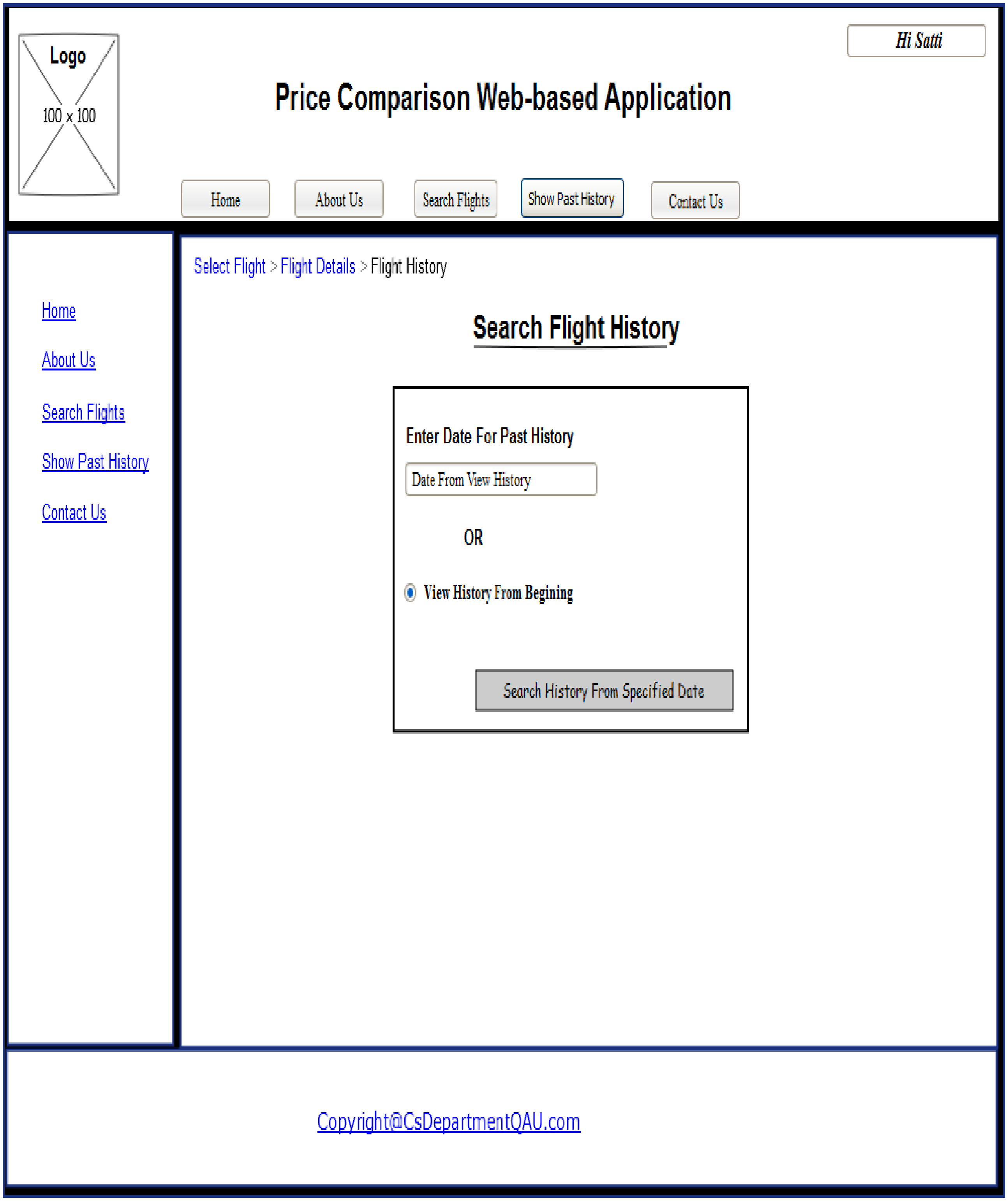
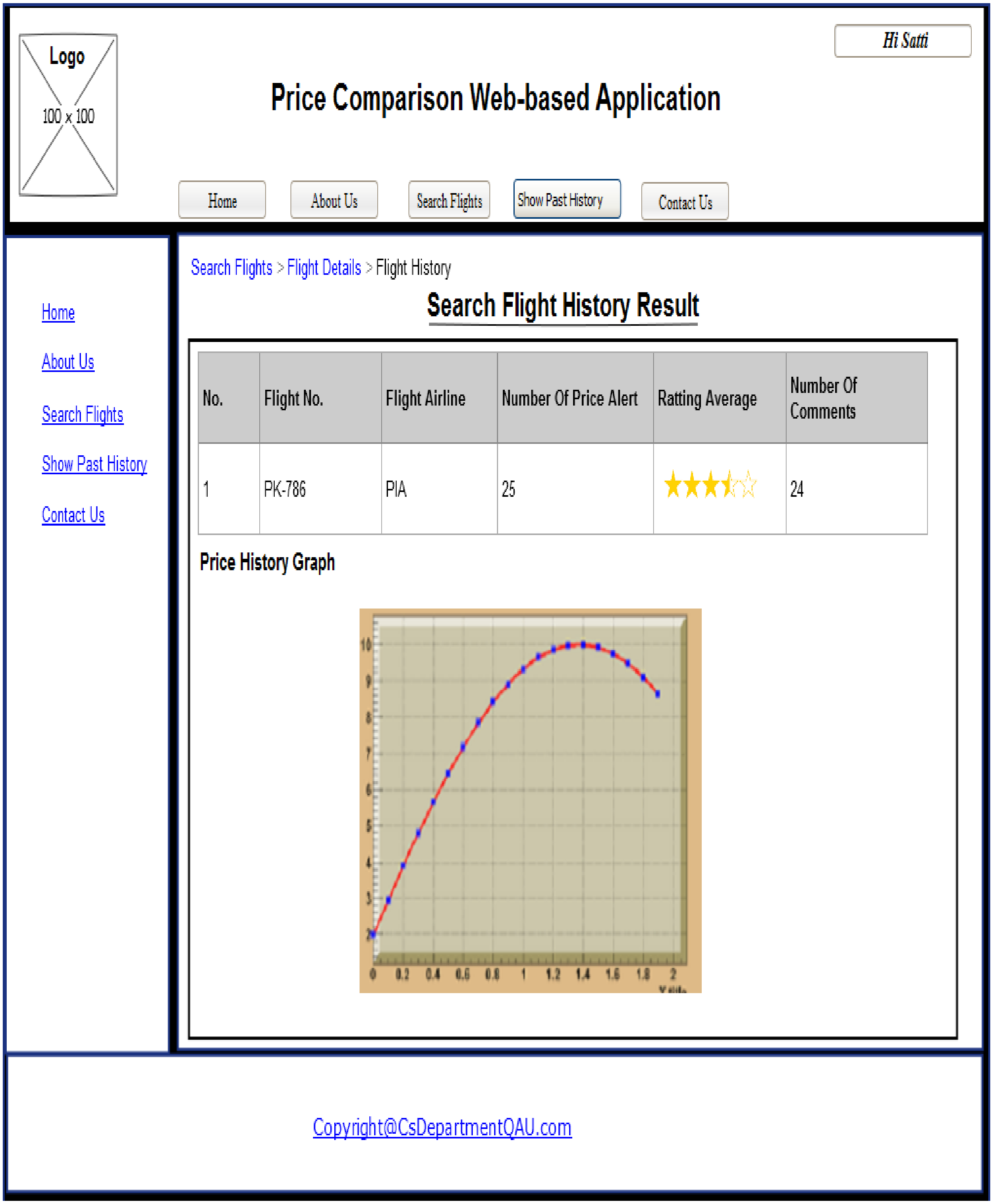

Logo
100 x 100
Hi Satti
Price Comparison Web-based Application
Home
About Us
Search Flights
Show Past History
Contact Us
Home
About Us
Search Flights
Show Past History
Contact Us
Select Flight > Flight Details > Flight History
Search Flight History
Enter Date For Past History
Date From View History
OR
● View History From Begining
Search History From Specified Date
Copyright@CsDepartmentQAU.com

Figure 3.12 Search Flight History Interface

3.4.1.8 IF-8: Search Flight History Response



Price Comparison Web-based Application

Home

About Us

Search Flights

Show Past History

Contact Us

Search Flights > Flight Details > Flight History

Search Flight History Result

No.	Flight No.	Flight Airline	Number Of Price Alert	Ratting Average	Number Of Comments
1	PK-786	PIA	25	★★★⯪☆	24

Price History Graph

Copyright@CsDepartmentQAU.com

Figure 3.13 Search Flight History Response Interface

3.4.1.9 IF-9: Contact Us

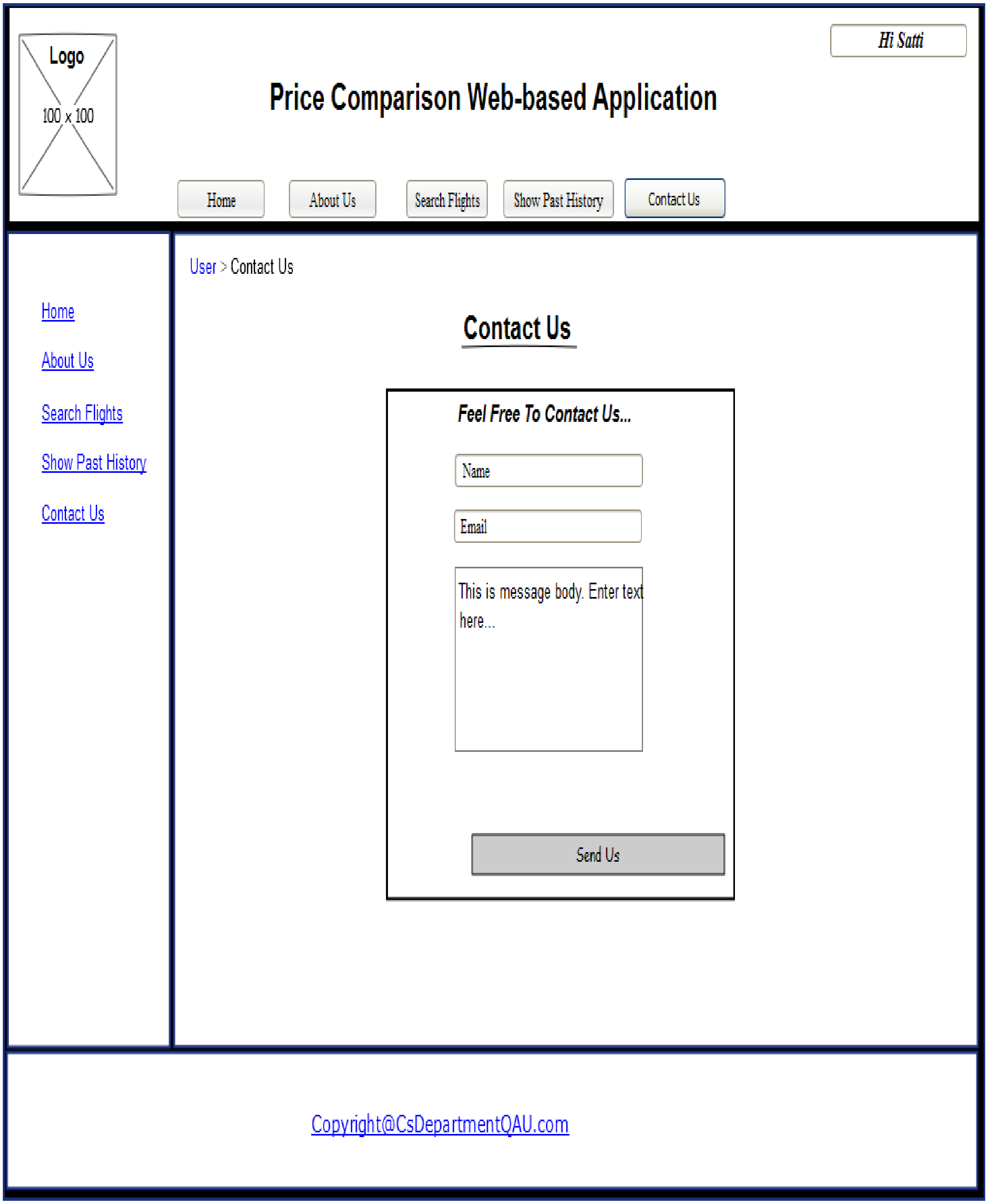
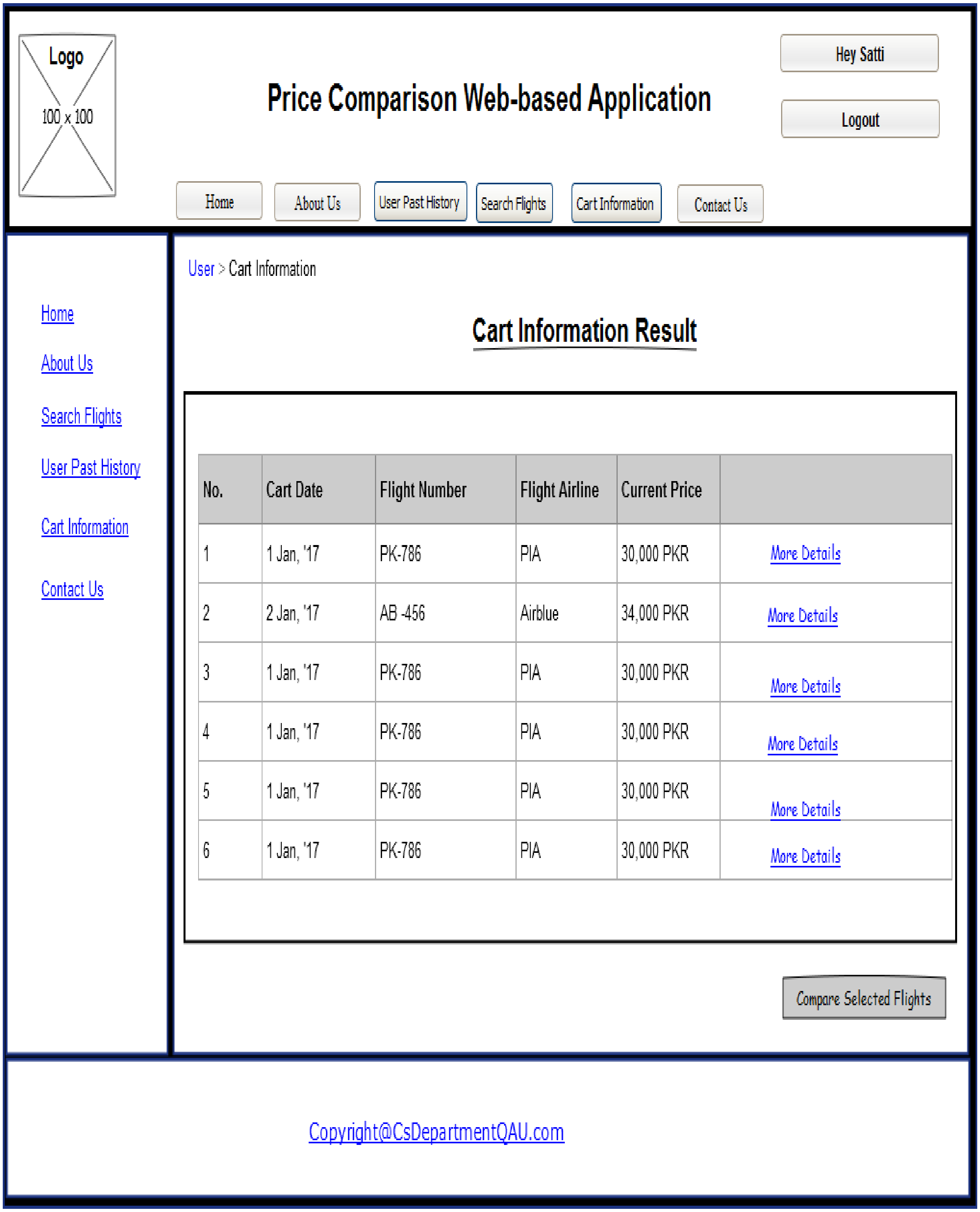


Figure 3.14 Contact Us Interface

3.4.1.10 IF- 10: Cart Information



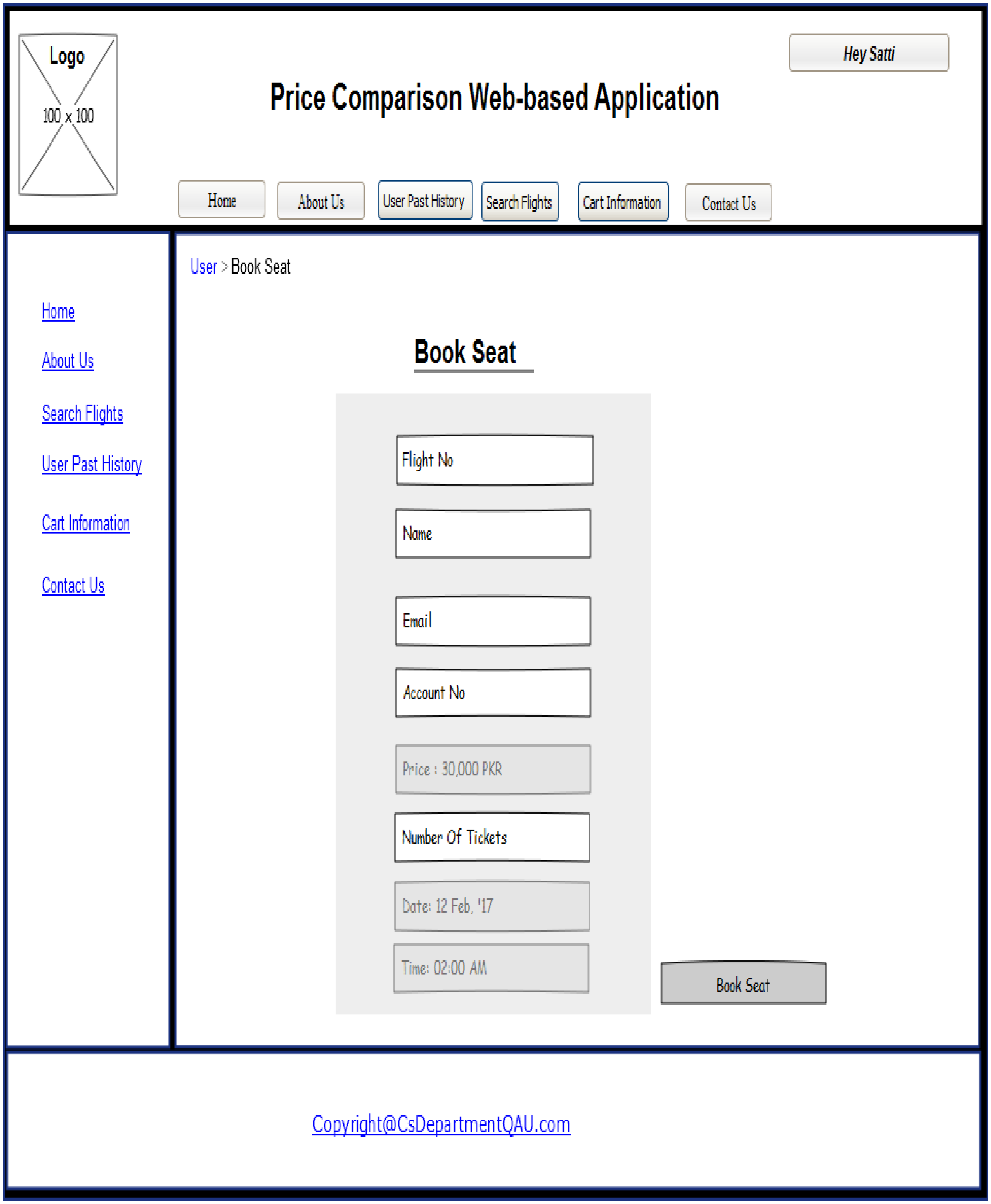
Price Comparison Web-based Application

Cart Information Result

No.	Cart Date	Flight Number	Flight Airline	Current Price	
1	1 Jan, '17	PK-786	PIA	30,000 PKR	More Details
2	2 Jan, '17	AB -456	Airblue	34,000 PKR	More Details
3	1 Jan, '17	PK-786	PIA	30,000 PKR	More Details
4	1 Jan, '17	PK-786	PIA	30,000 PKR	More Details
5	1 Jan, '17	PK-786	PIA	30,000 PKR	More Details
6	1 Jan, '17	PK-786	PIA	30,000 PKR	More Details

Figure 3.15 Cart Information Interface

3.4.1.11 IF- 11: Book Seat



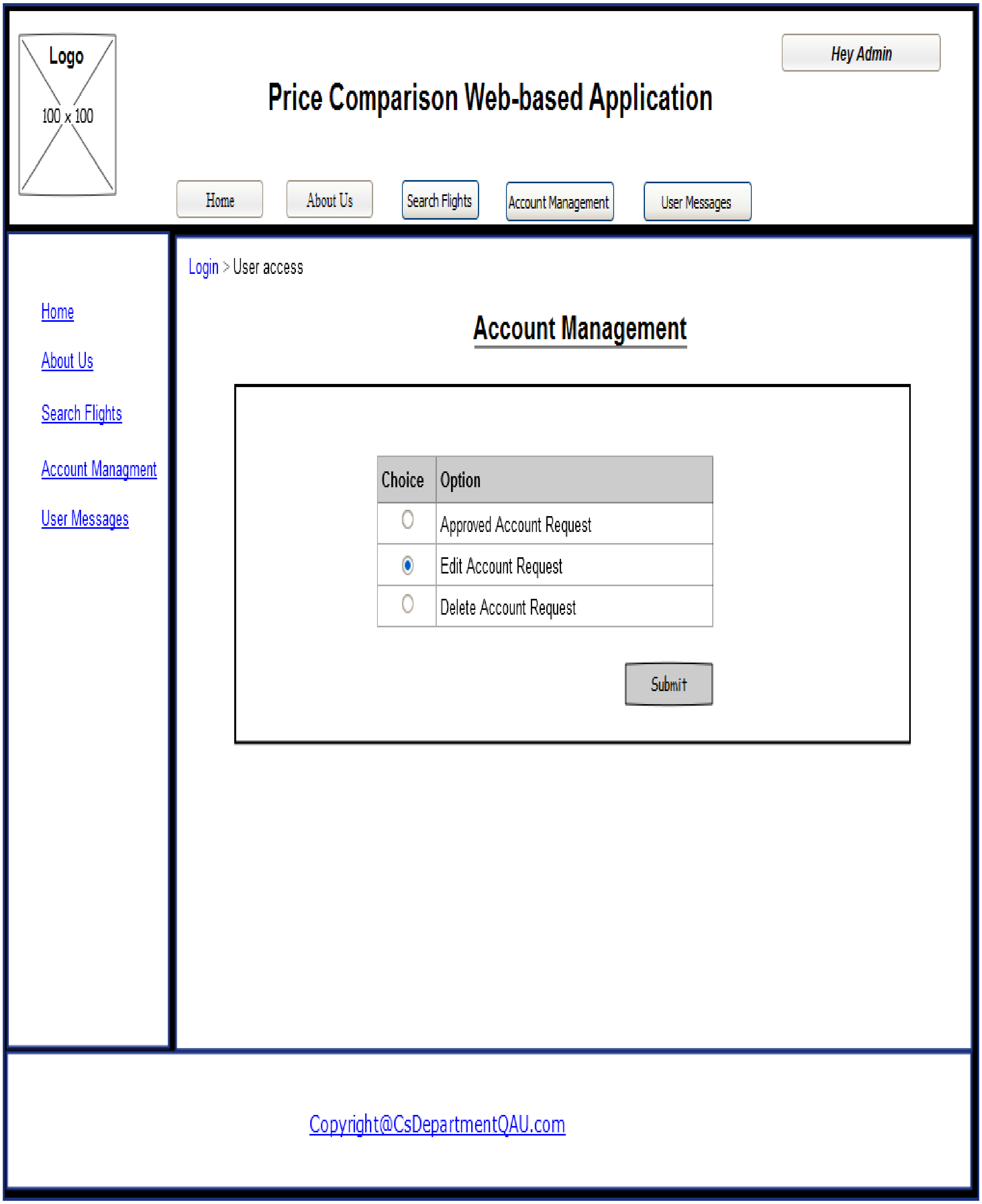
Price : 30,000 PKR

Date: 12 Feb, '17

Time: 02:00 AM

Figure 3.16 Book Seat Interface

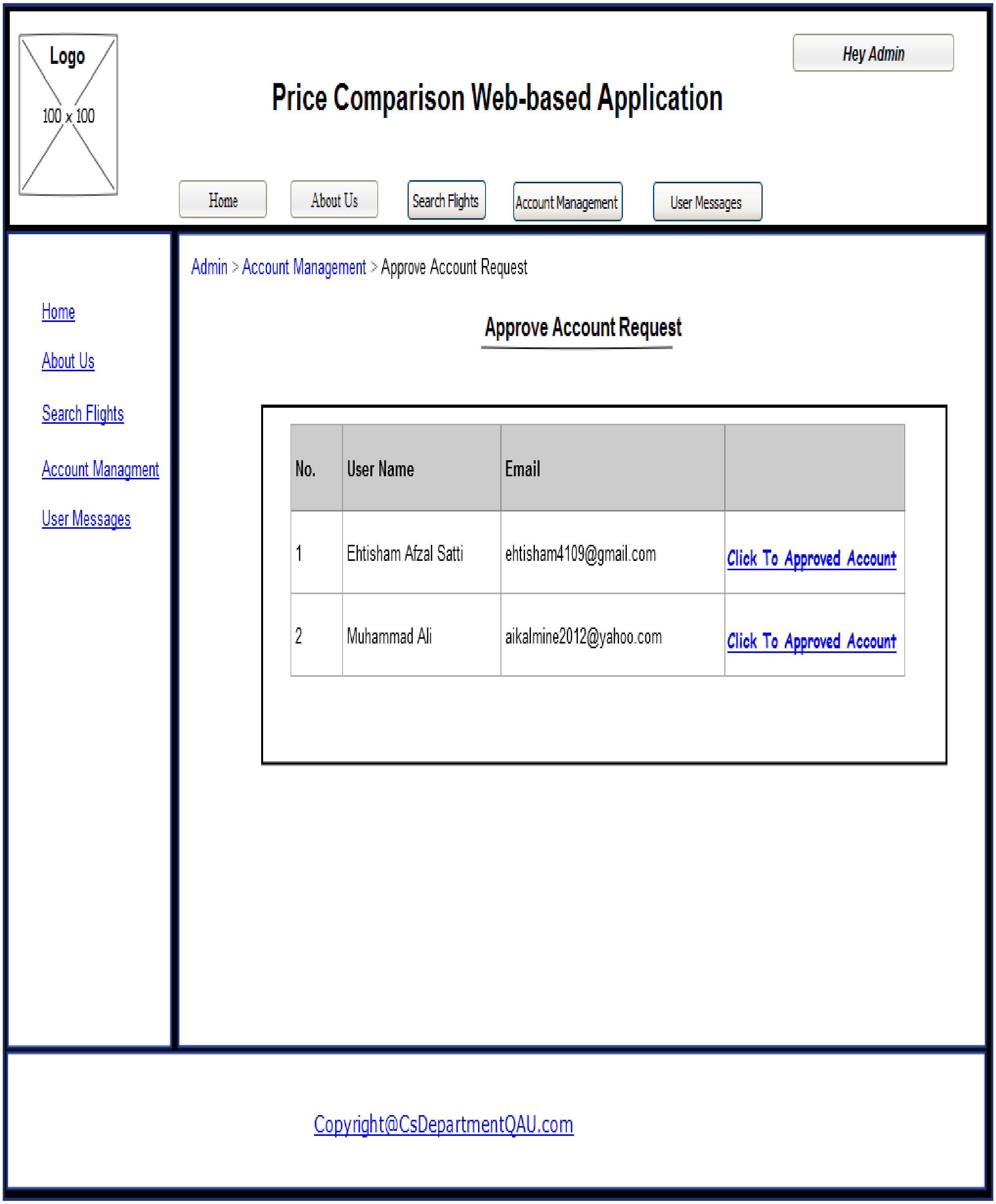
3.4.1.12 IF-12: Admin Account Option Screen



Choice	Option
○	Approved Account Request
●	Edit Account Request
○	Delete Account Request

Figure 3.17 Admin Account Option Interface

3.4.1.13 IF-13: Admin Approved Account Screen



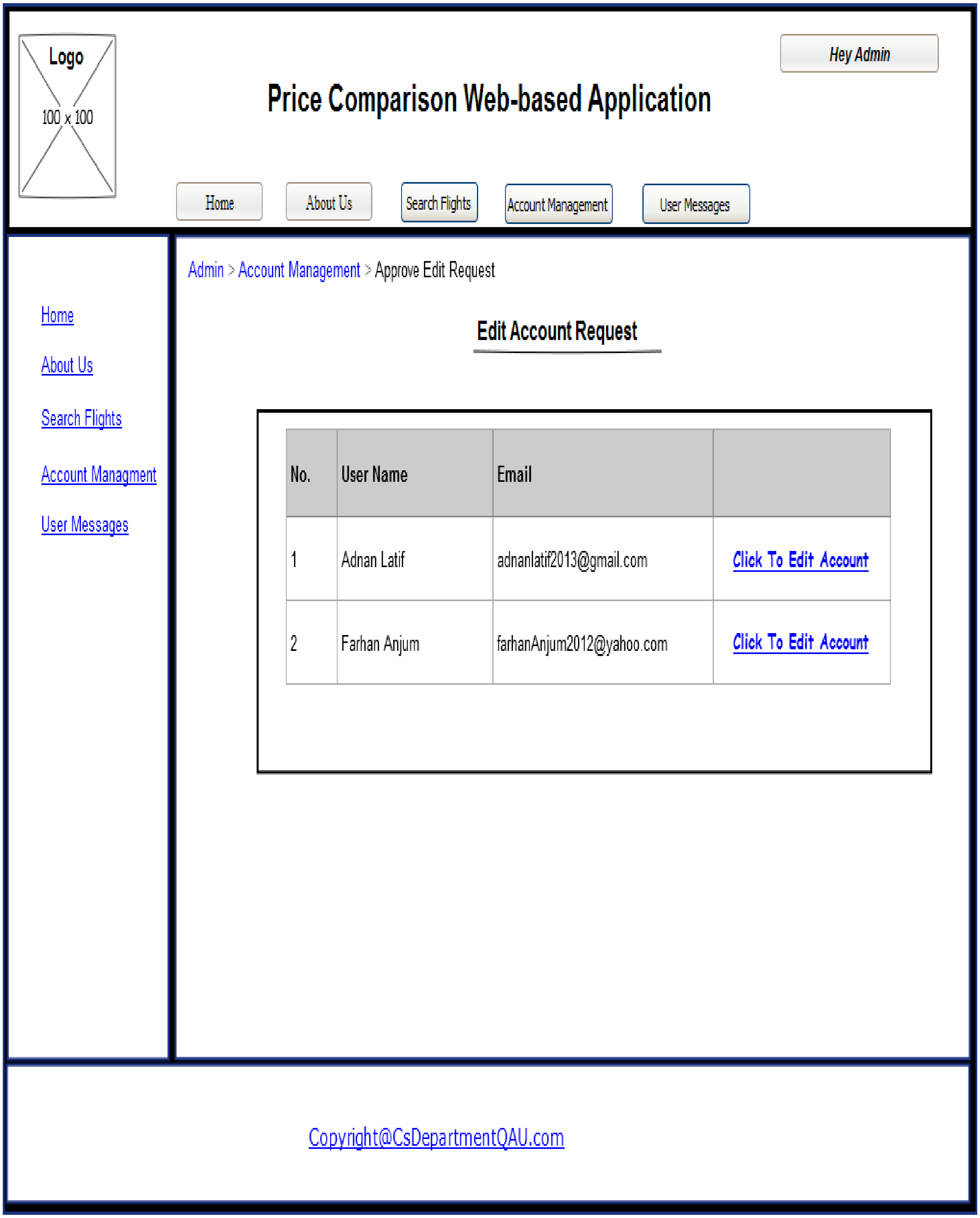
Admin > Account Management > Approve Account Request

Approve Account Request

No.	User Name	Email	
1	Ehtisham Afzal Satti	ehtisham4109@gmail.com	Click To Approved Account
2	Muhammad Ali	aikalmine2012@yahoo.com	Click To Approved Account

Figure 3.18 Admin Approved Account Interface

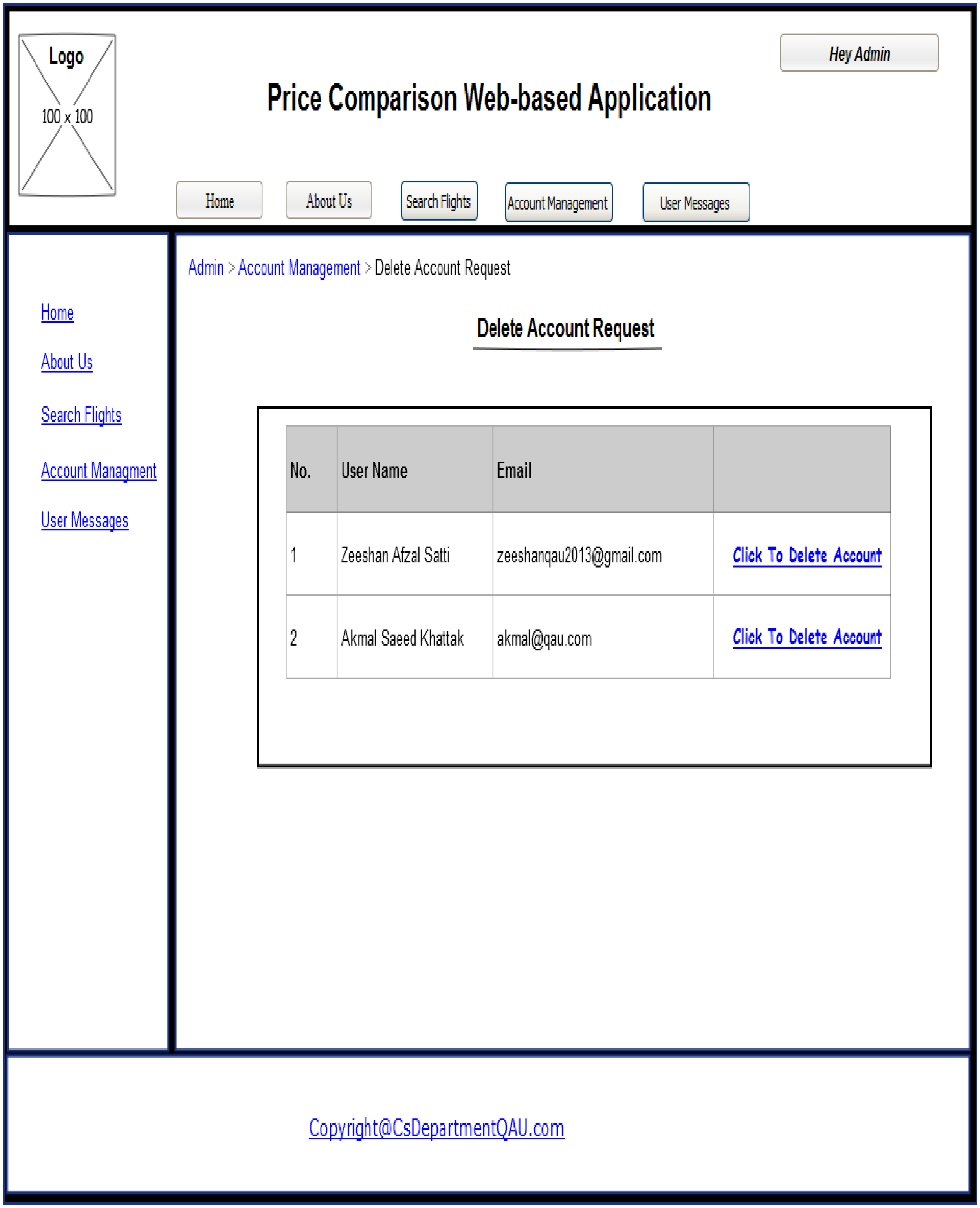
3.4.1.14 IF-14: Admin Edit Account Screen



No.	User Name	Email	
1	Adnan Latif	adnanlatif2013@gmail.com	Click To Edit Account
2	Farhan Anjum	farhanAnjum2012@yahoo.com	Click To Edit Account

Figure 3.19 Admin Edit Account Interface

3.4.1.15 IF-15: Admin Delete Account Screen

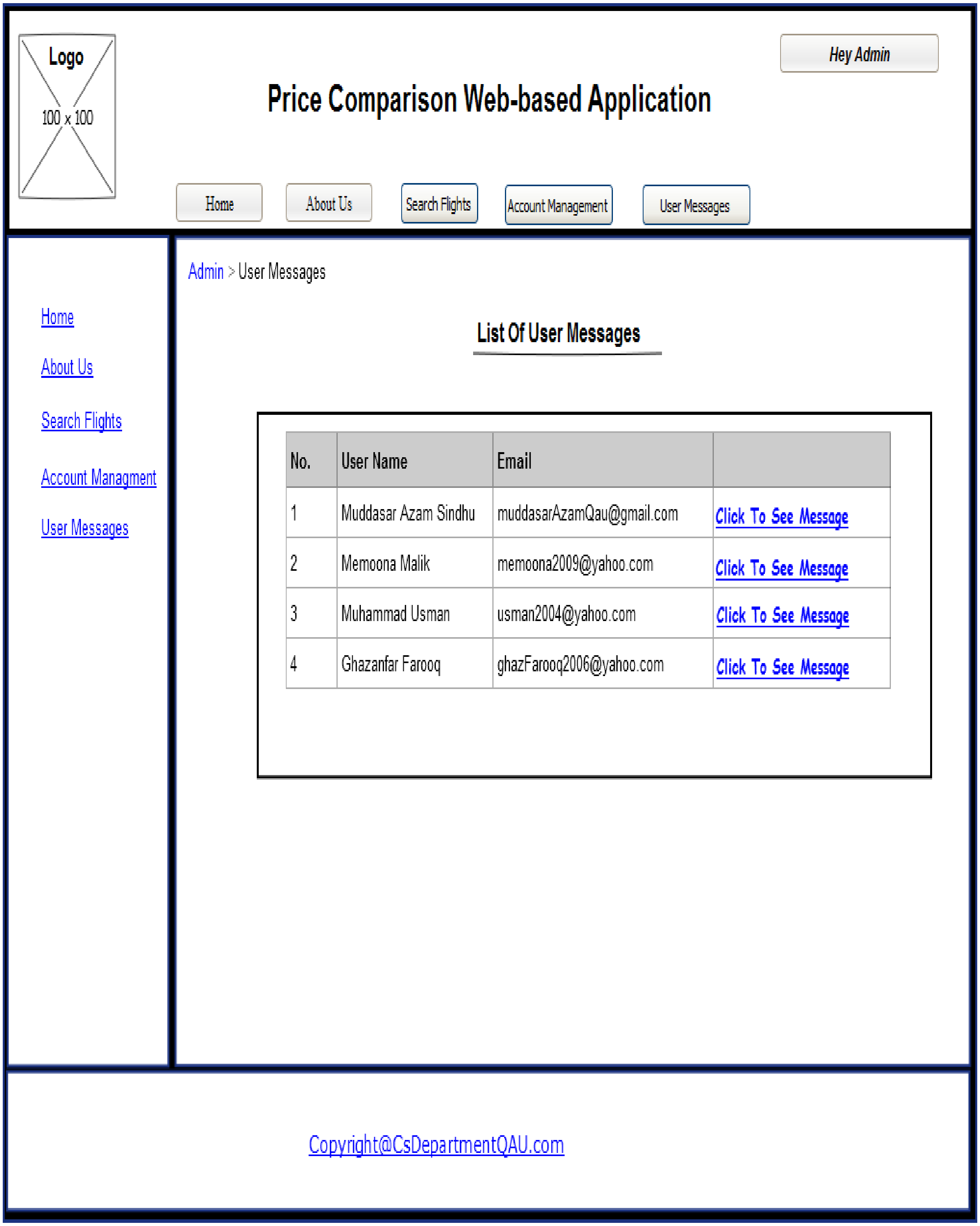


Delete Account Request

No.	User Name	Email	
1	Zeeshan Afzal Satti	zeeshanqau2013@gmail.com	Click To Delete Account
2	Akmal Saeed Khattak	akmal@qau.com	Click To Delete Account

Figure 3.20 Admin Delete Account Interface

3.4.1.16 IF-16: Admin Views List of Messages



No.	User Name	Email	
1	Muddasar Azam Sindhu	muddasarAzamQau@gmail.com	Click To See Message
2	Memoona Malik	memoona2009@yahoo.com	Click To See Message
3	Muhammad Usman	usman2004@yahoo.com	Click To See Message
4	Ghazanfar Farooq	ghazFarooq2006@yahoo.com	Click To See Message

Figure 3.21 Admin Views List of Messages Interface

3.4.1.17 IF-17: Search User History

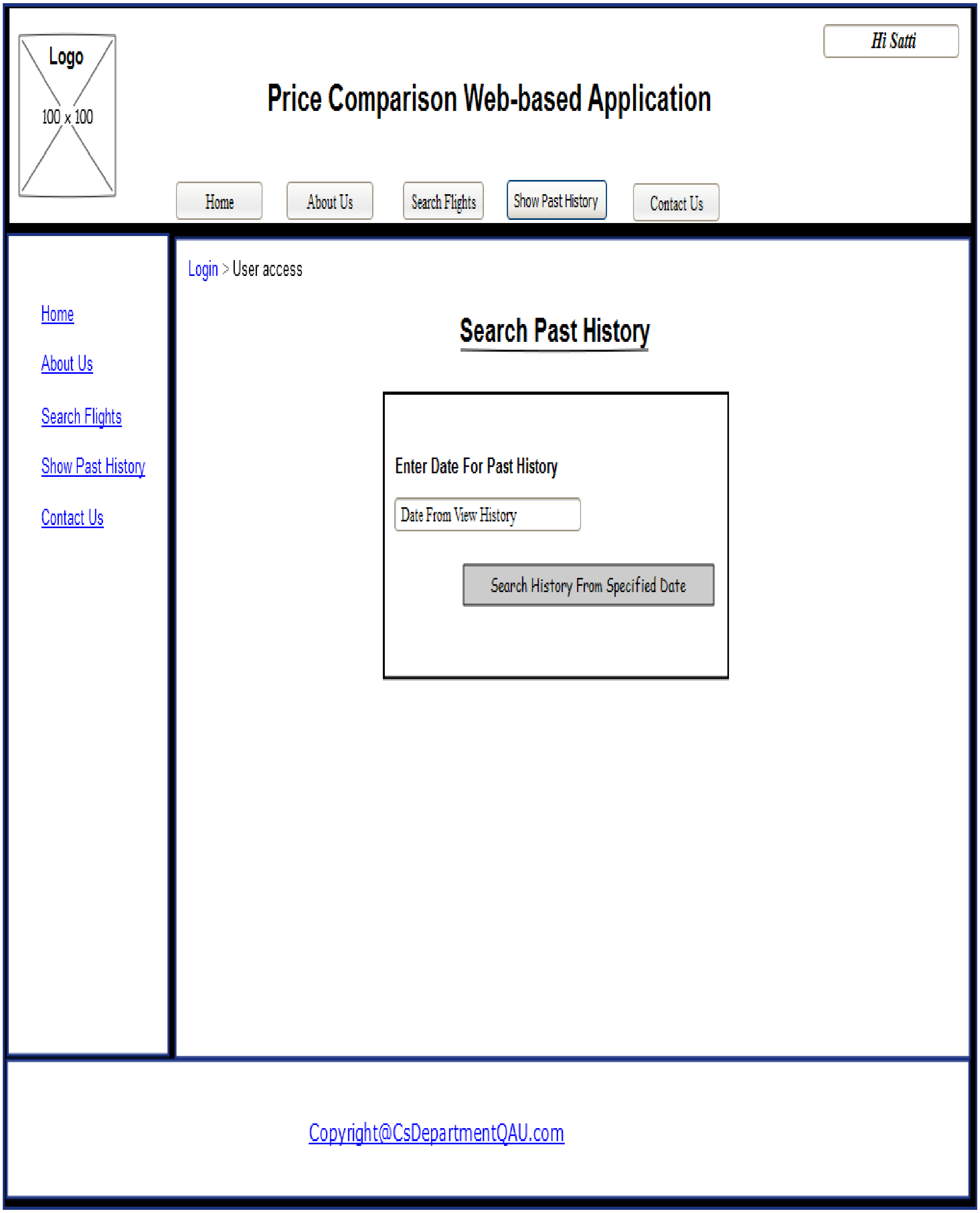
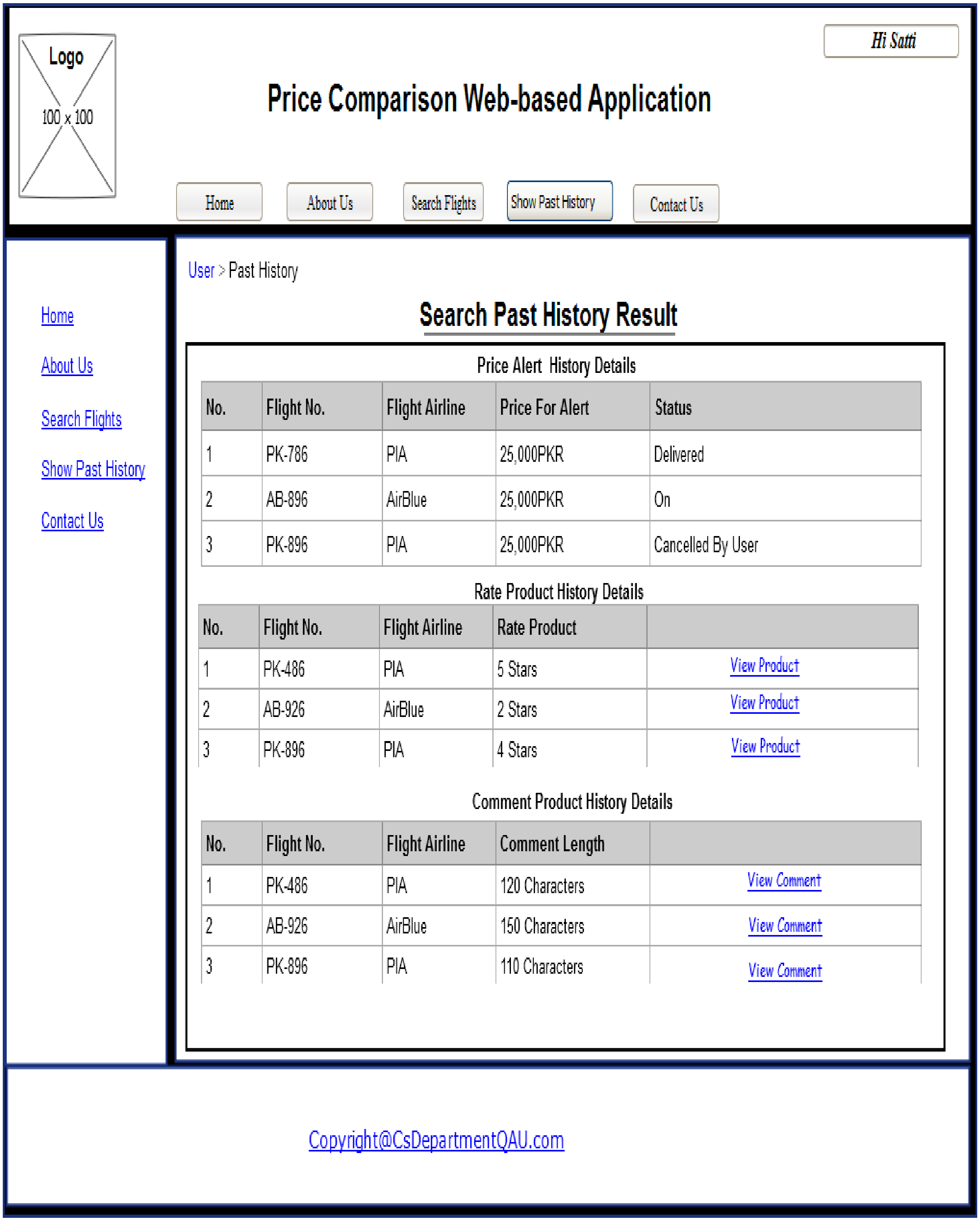


Figure 3.22 Search User History Interface

3.4.1.18 IF-18: Search User History Response



Search Past History Result

Price Alert History Details

No.	Flight No.	Flight Airline	Price For Alert	Status
1	PK-786	PIA	25,000PKR	Delivered
2	AB-896	AirBlue	25,000PKR	On
3	PK-896	PIA	25,000PKR	Cancelled By User

Rate Product History Details

No.	Flight No.	Flight Airline	Rate Product	
1	PK-486	PIA	5 Stars	View Product
2	AB-926	AirBlue	2 Stars	View Product
3	PK-896	PIA	4 Stars	View Product

Comment Product History Details

No.	Flight No.	Flight Airline	Comment Length	
1	PK-486	PIA	120 Characters	View Comment
2	AB-926	AirBlue	150 Characters	View Comment
3	PK-896	PIA	110 Characters	View Comment

Copyright@CsDepartmentQAU.com

Figure 3.23 Search User History Interface

3.4.2 Objects and Actions

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. Following are the sequence diagrams of PCWBA:

3.4.2.1 SD-1: Sign Up User

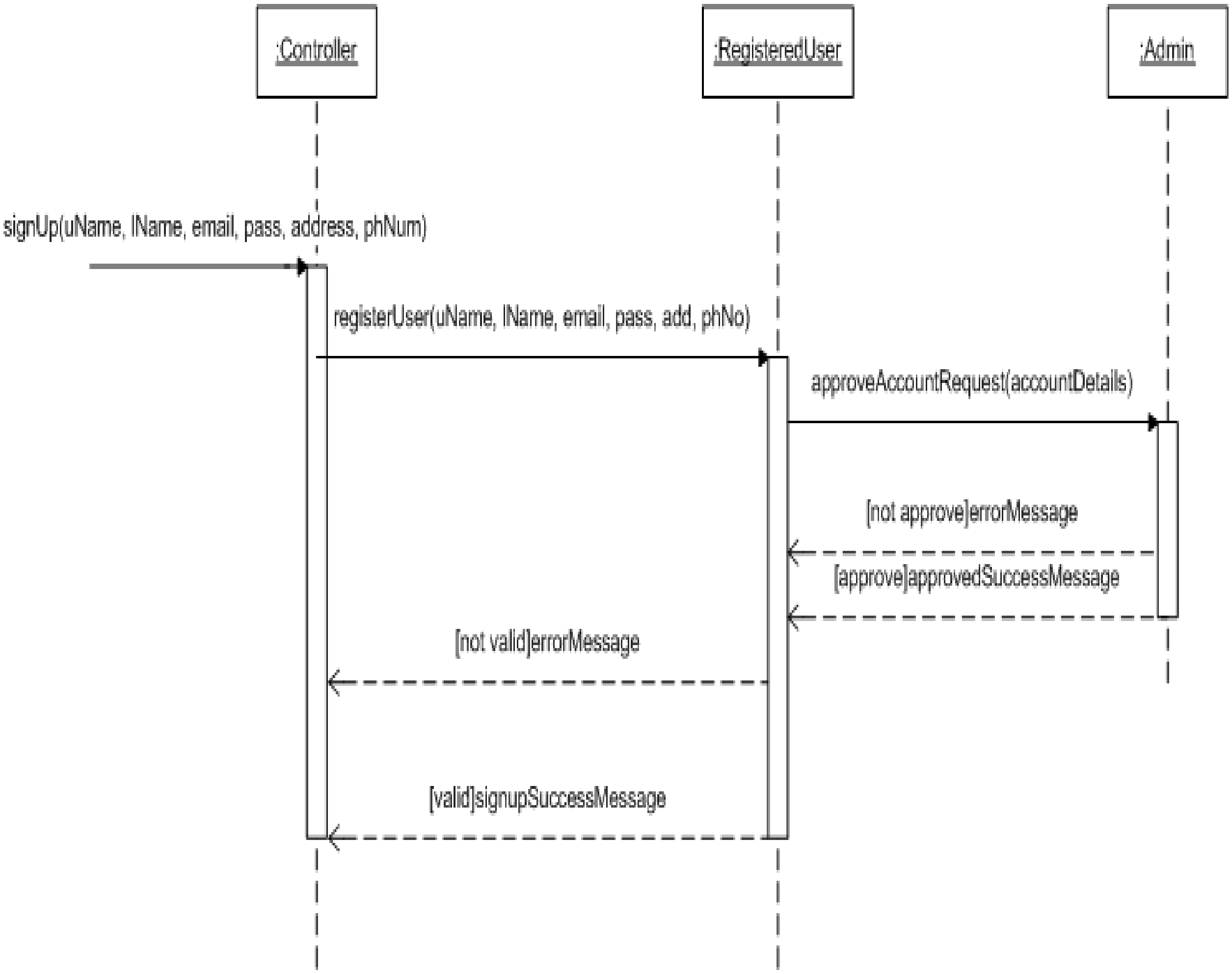

:Controller
:RegisteredUser
:Admin
signUp(uName, lName, email, pass, address, phNum)
registerUser(uName, lName, email, pass, add, phNo)
approveAccountRequest(accountDetails)
[not approve]errorMessage
[approve]approvedSuccessMessage
[not valid]errorMessage
[valid]signupSuccessMessage

Figure 3.24 Sign up Sequence Diagram

3.4.2.2 SD-2: Log in User

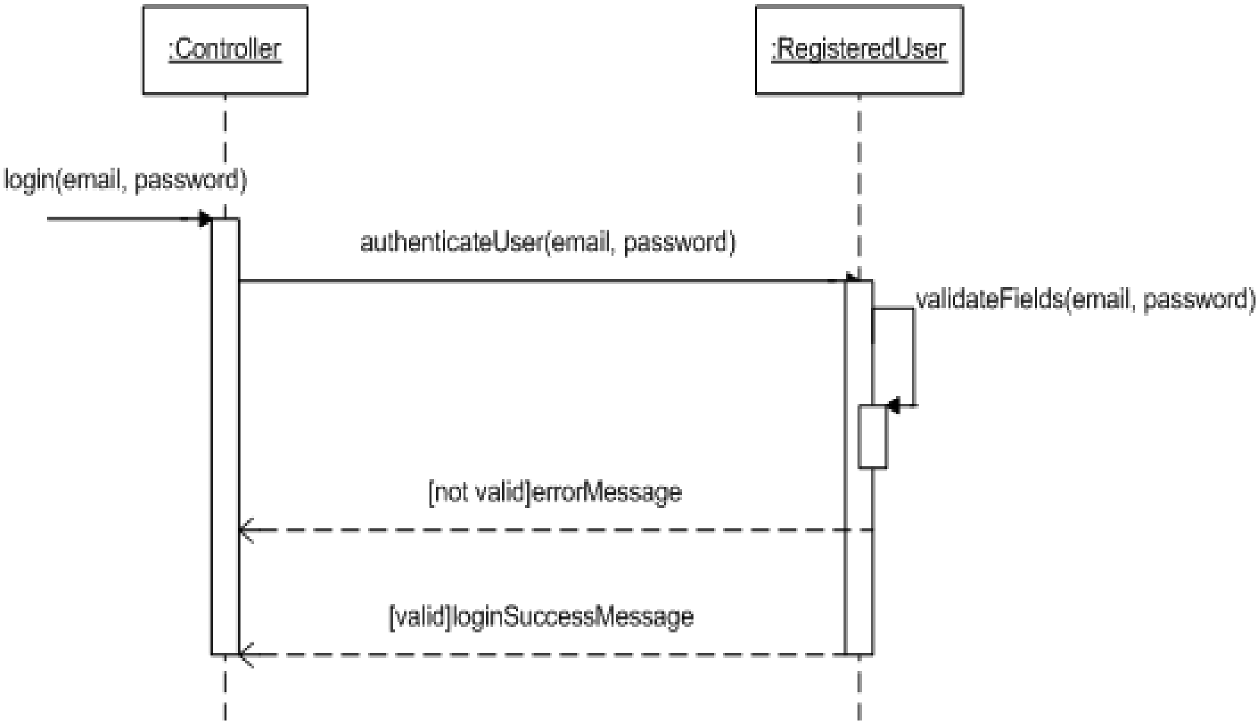

:Controller
:RegisteredUser
login(email, password)
authenticateUser(email, password)
validateFields(email, password)
[not valid]errorMessage
[valid]loginSuccessMessage

Figure 3.25 Login Sequence Diagram

3.4.2.3 SD-3: Search Product

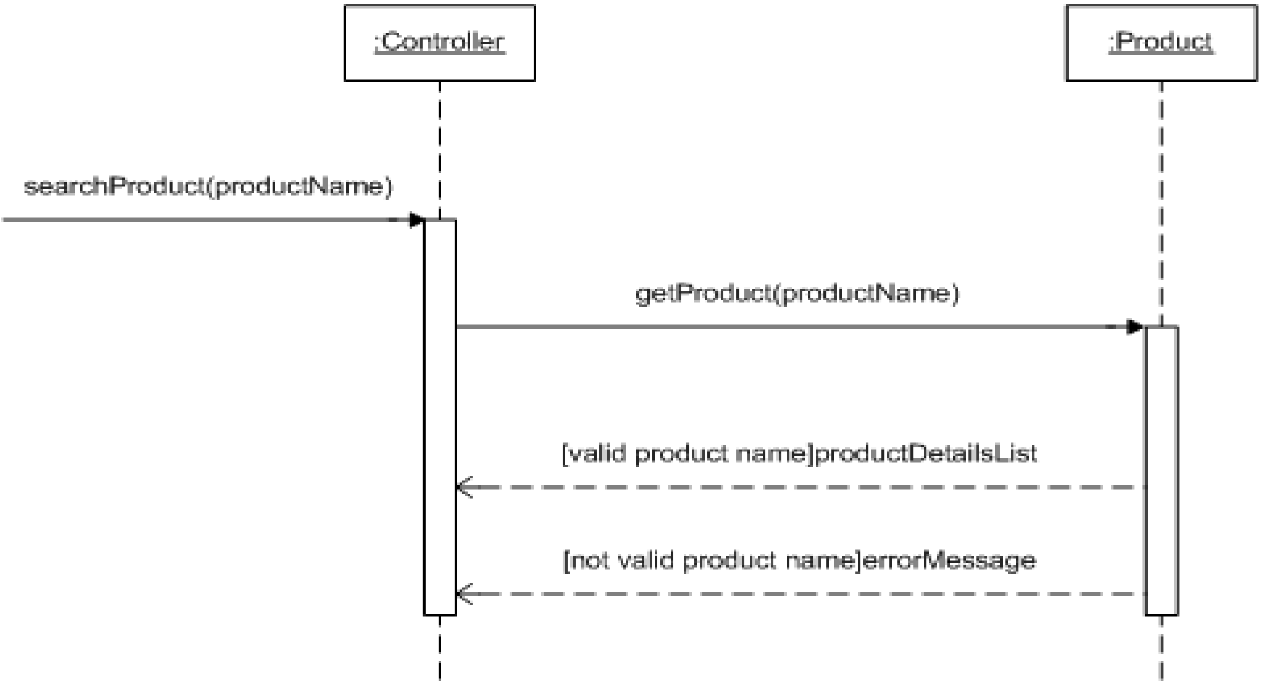

:Controller
:Product
searchProduct(productName)
getProduct(productName)
[valid product name]productDetailsList
[not valid product name]errorMessage

Figure 3.26 Search Product Sequence Diagram

3.4.2.4 SD-4: Compare Product

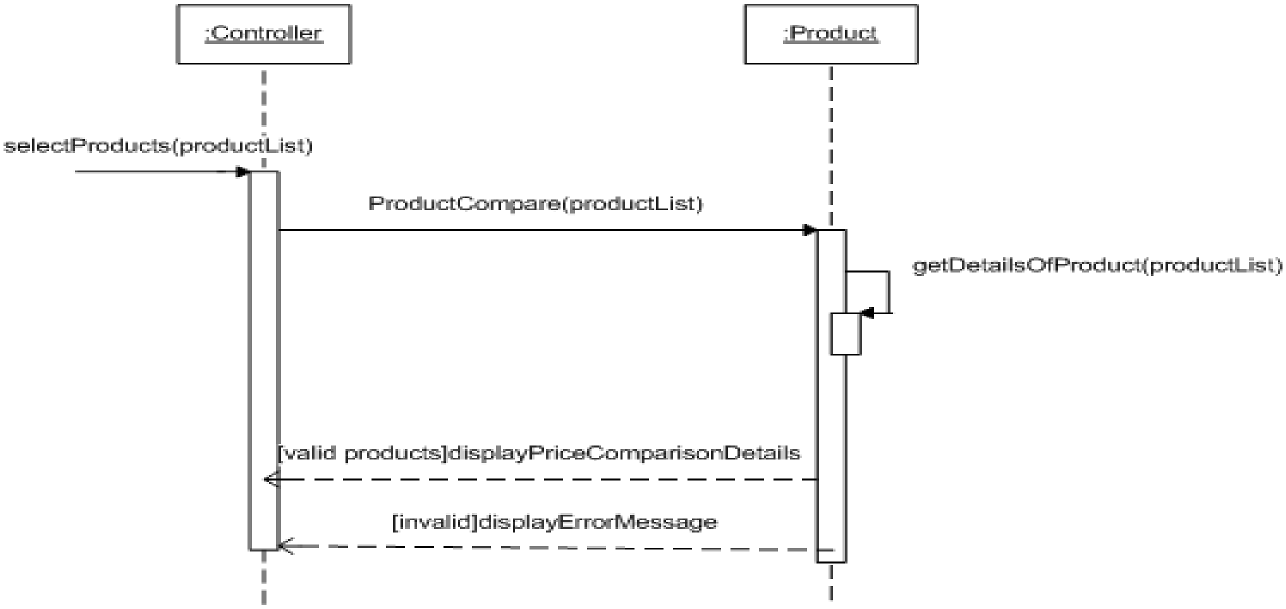

:Controller
:Product
selectProducts(productList)
ProductCompare(productList)
getDetailsOfProduct(productList)
[valid products]displayPriceComparisonDetails
[invalid]displayErrorMessage

Figure 3.27 Compare Product Sequence Diagram

3.4.2.5 SD-5: User History

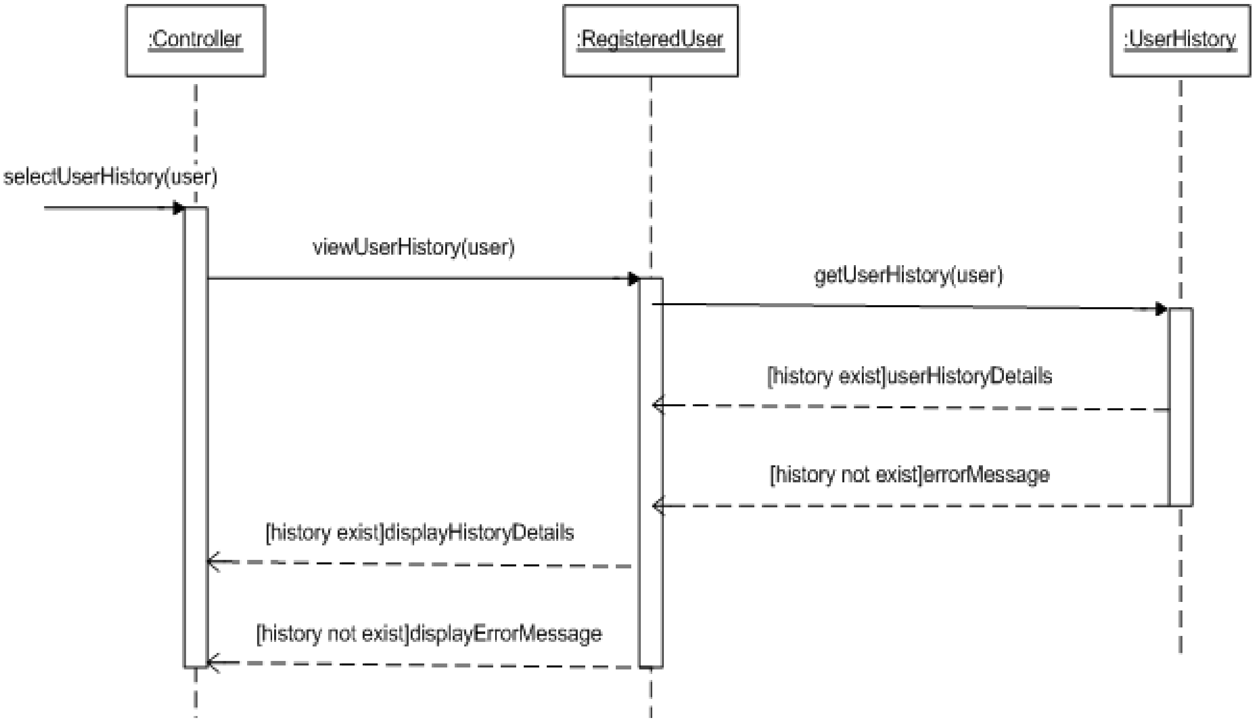

:Controller
:RegisteredUser
:UserHistory
selectUserHistory(user)
viewUserHistory(user)
getUserHistory(user)
[history exist]userHistoryDetails
[history not exist]errorMessage
[history exist]displayHistoryDetails
[history not exist]displayErrorMessage

Figure 3.28 User History Sequence Diagram

3.4.2.6 SD-6: Product History

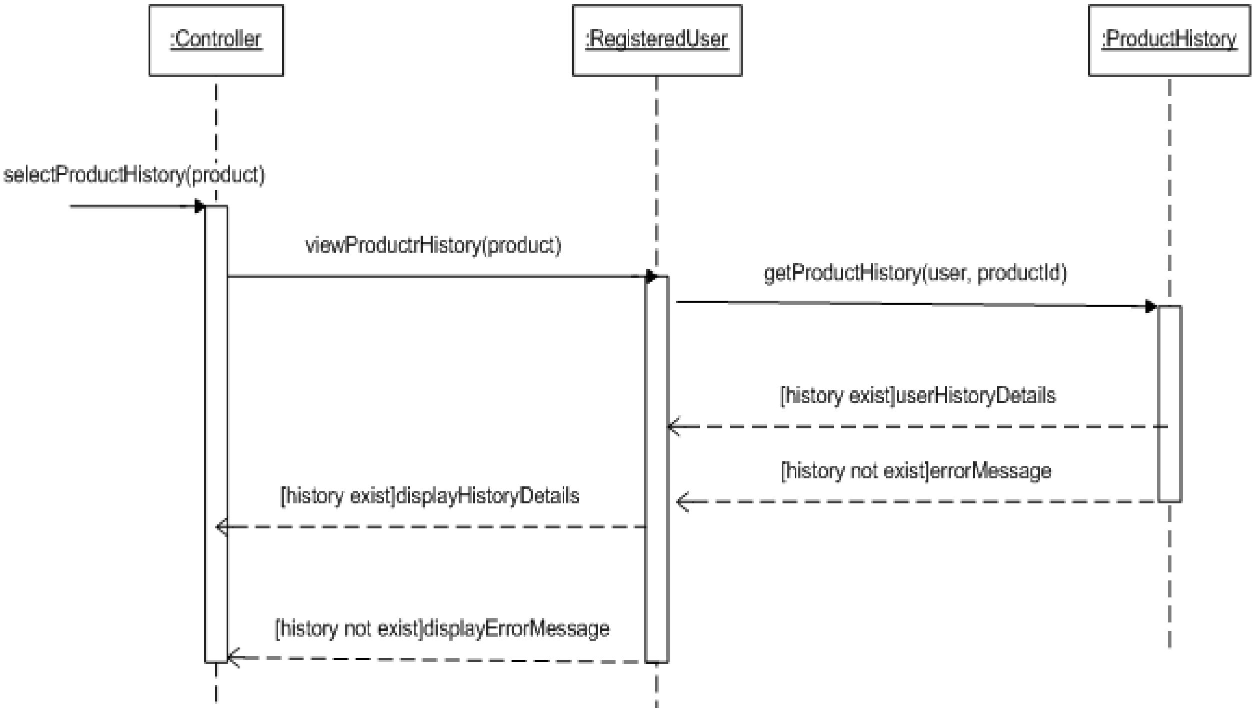

:Controller
:RegisteredUser
:ProductHistory
selectProductHistory(product)
viewProductrHistory(product)
getProductHistory(user, productId)
[history exist]userHistoryDetails
[history not exist]errorMessage
[history exist]displayHistoryDetails
[history not exist]displayErrorMessage

Figure 3.29 Product History Sequence Diagram

3.4.2.7 SD-7: Delete Account

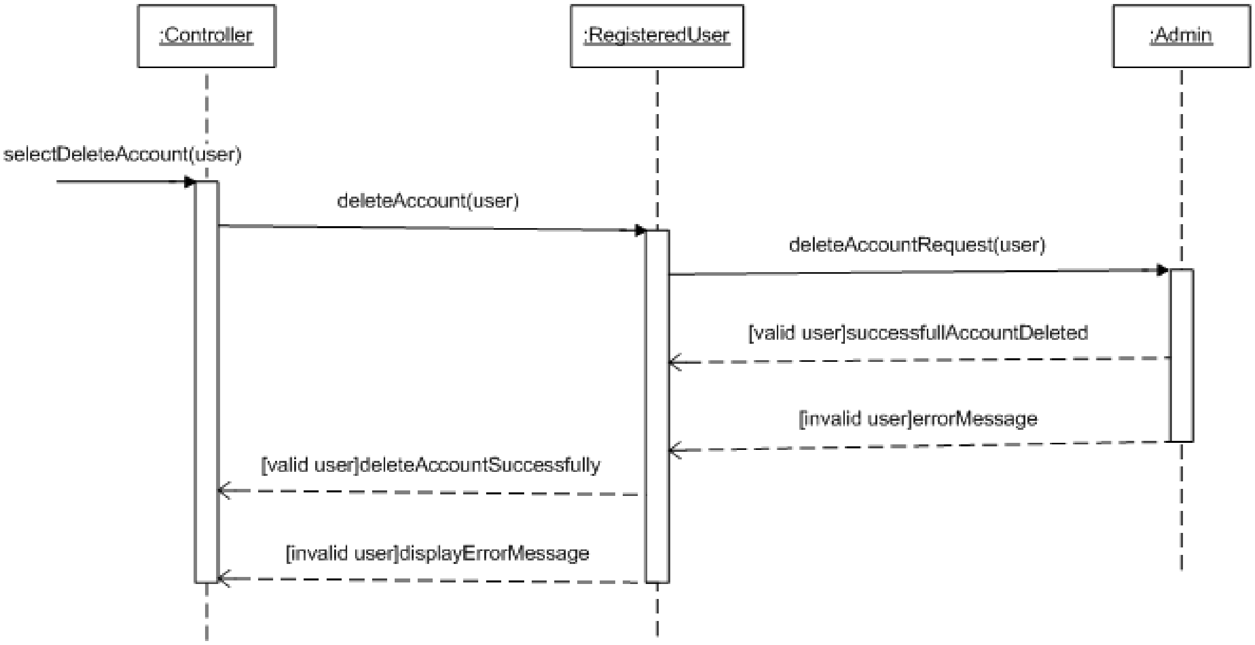

:Controller
:RegisteredUser
:Admin
selectDeleteAccount(user)
deleteAccount(user)
deleteAccountRequest(user)
[valid user]successfullAccountDeleted
[invalid user]errorMessage
[valid user]deleteAccountSuccessfully
[invalid user]displayErrorMessage

Figure 3.30 Delete Account Sequence Diagram

3.4.2.8 SD-8: Set Price Alert

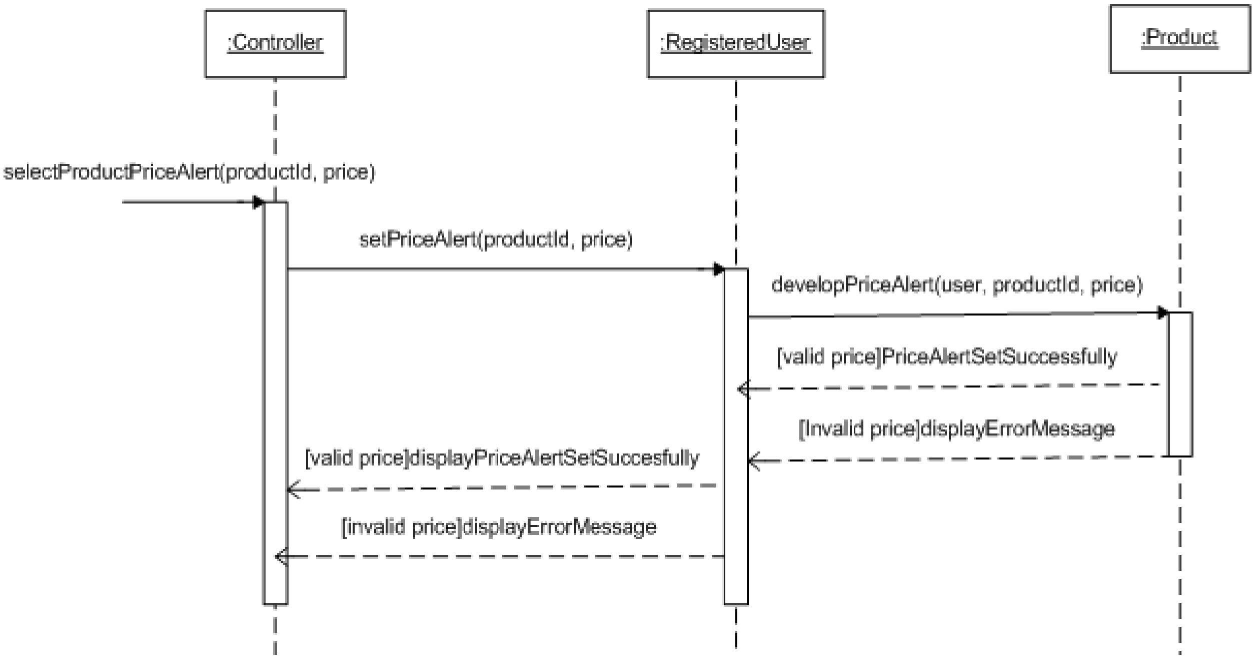

:Controller
:RegisteredUser
:Product
selectProductPriceAlert(productId, price)
setPriceAlert(productId, price)
developPriceAlert(user, productId, price)
[valid price]PriceAlertSetSuccessfully
[Invalid price]displayErrorMessage
[valid price]displayPriceAlertSetSuccesfully
[invalid price]displayErrorMessage

Figure 3.31 Set Price Alert Sequence Diagram

3.4.2.9 SD-9: Add Product in Cart

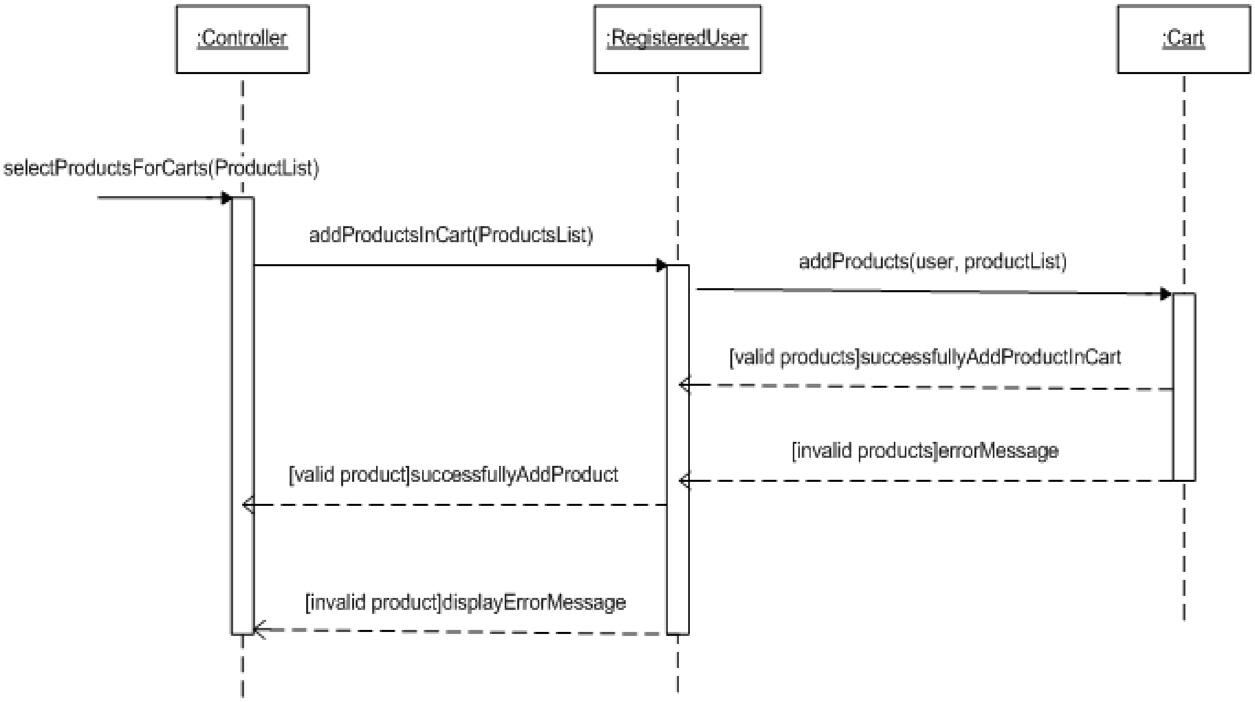

:Controller
:RegisteredUser
:Cart
selectProductsForCarts(ProductList)
addProductsInCart(ProductsList)
addProducts(user, productList)
[valid products]successfullyAddProductInCart
[invalid products]errorMessage
[valid product]successfullyAddProduct
[invalid product]displayErrorMessage

Figure 3.32 Add Product in Cart Sequence Diagram

3.4.2.10 SD-10: Order Product

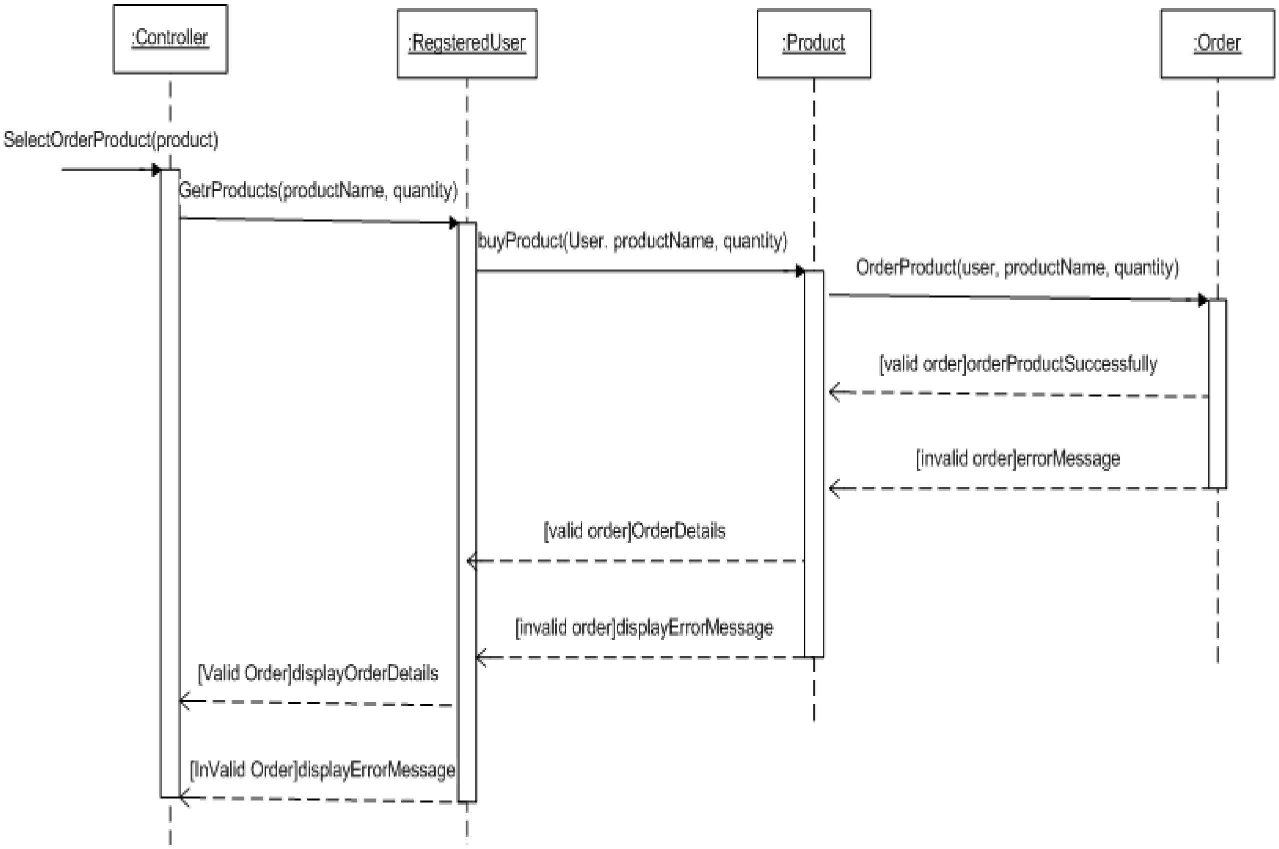

:Controller
:RegsteredUser
:Product
:Order
SelectOrderProduct(product)
GetrProducts(productName, quantity)
buyProduct(User. productName, quantity)
OrderProduct(user, productName, quantity)
[valid order]orderProductSuccessfully
[invalid order]errorMessage
[valid order]OrderDetails
[invalid order]displayErrorMessage
[Valid Order]displayOrderDetails
[InValid Order]displayErrorMessage

Figure 3.33 Order Product Sequence Diagram

3.4.2.11 SD-11: Logout User

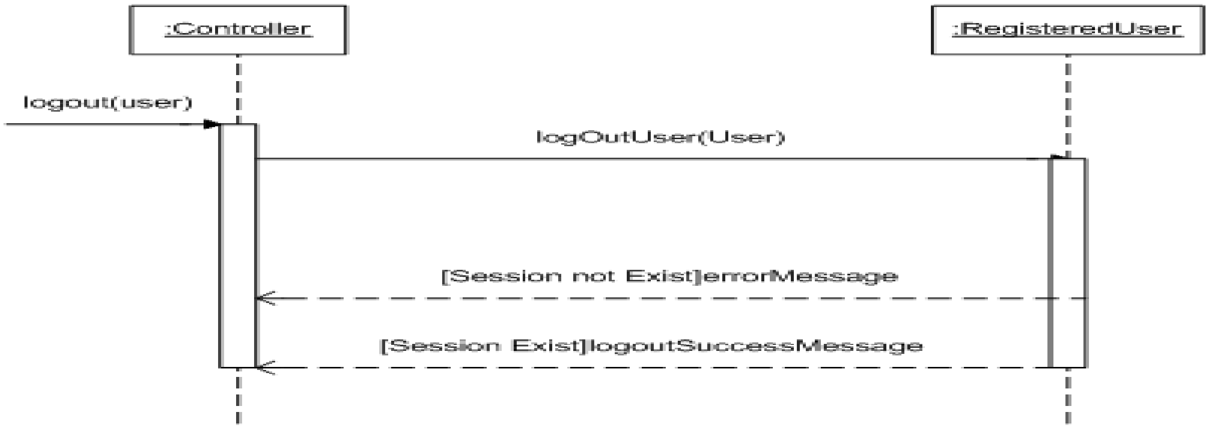

:Controller
:RegisteredUser
logout(user)
logOutUser(User)
[Session not Exist]errorMessage
[Session Exist]logoutSuccessMessage

Figure 3.34 Logout User Sequence Diagram

3.4.2.12 SD-12: Rate and Comment Product

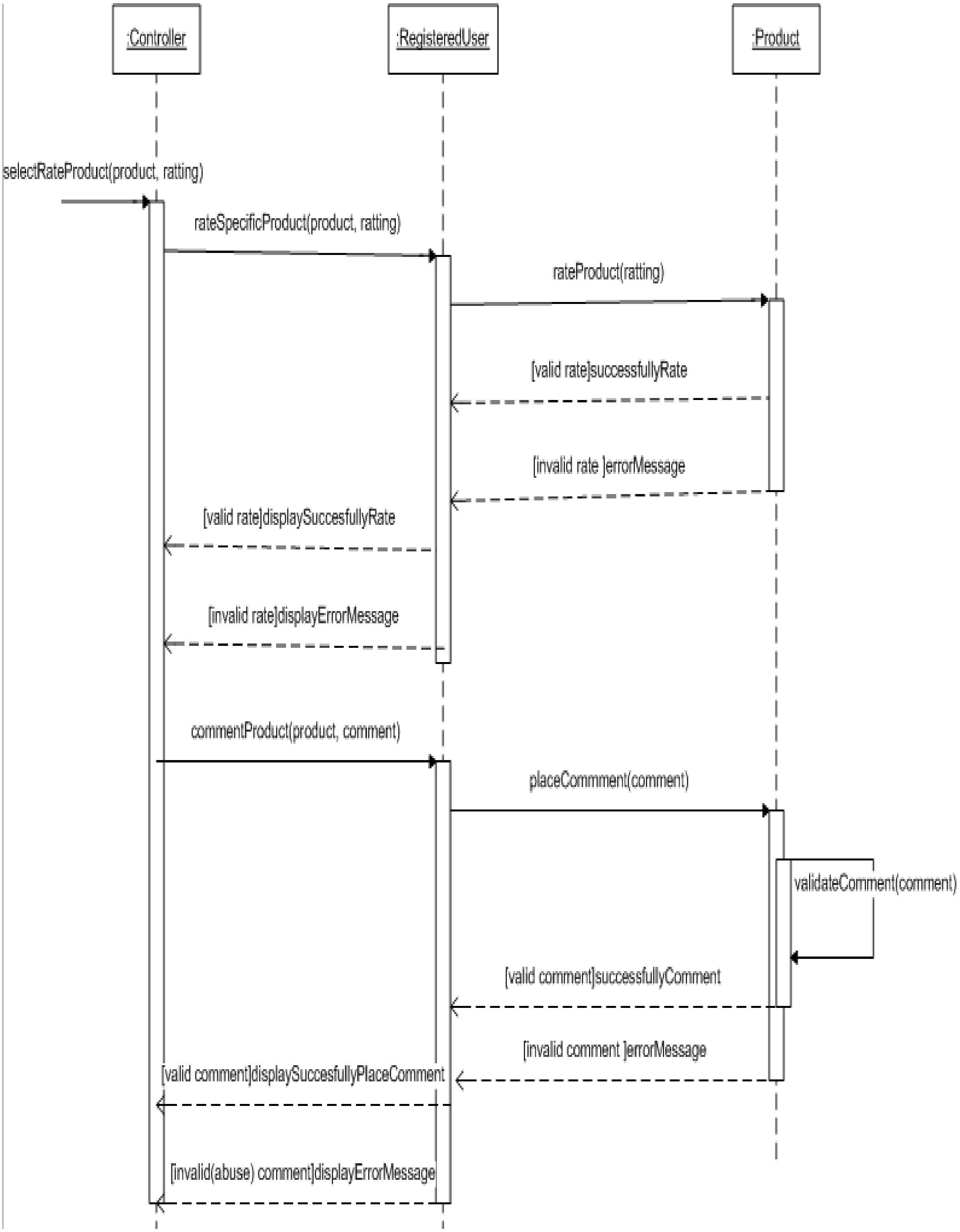

:Controller
:RegisteredUser
:Product
selectRateProduct(product, ratting)
rateSpecificProduct(product, ratting)
rateProduct(ratting)
[valid rate]successfullyRate
[invalid rate]errorMessage
[valid rate]displaySuccesfullyRate
[invalid rate]displayErrorMessage
commentProduct(product, comment)
placeCommment(comment)
validateComment(comment)
[valid comment]successfullyComment
[invalid comment]errorMessage
[valid comment]displaySuccesfullyPlaceComment
[invalid(abuse) comment]displayErrorMessage

Figure 3.35 Rate and Comment Product Sequence Diagram

3.5 Class Diagram

Class diagrams depict the software classes and their relationships. This diagram defines individual classes along with their attributes, types of the attributes, and operations, associations between classes and navigability (direction of association) that define attribute visibility, and dependency relationships that define non-attribute visibility. Conceptual models (Domain Model) and collaboration diagrams are very useful to identify the software classes (classes in class diagram). PCWBA (Price Compare Web-based Application) contain following classes. Some of them are model classes that interact with SQL database to perform CRUD (Create, Retrieve, Update and Delete) operations.

1. Controller

2. Admin

3. Registered User

4. Non-Registered User

5. Admin

6. Product

7. Order

8. Product History

9. User History

These all are the software classes which means that these classes should be implemented while in PCWBA (Price Comparison Web-based Application) implementation. Following class diagram also describes types of relationships that exist between classes.

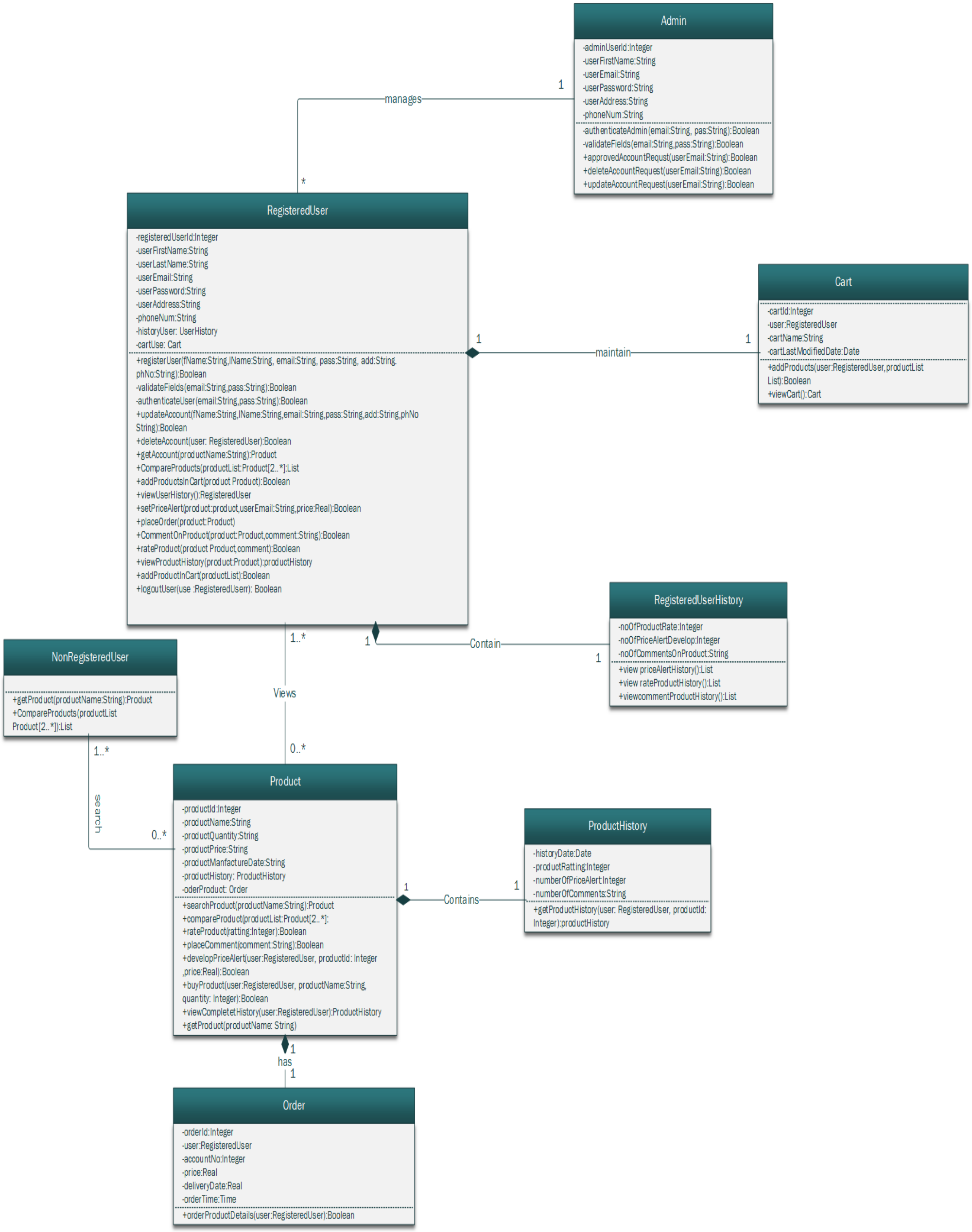

Admin
-adminUserId:Integer
-userFirstName:String
-userEmail:String
-userPassword:String
-userAddress:String
-phoneNum:String
-authenticateAdmin(email:String, pas:String):Boolean
-validateFields(email:String,pass:String):Boolean
+approvedAccountRequst(userEmail:String):Boolean
+deleteAccountRequest(userEmail:String):Boolean
+updateAccountRequest(userEmail:String):Boolean
manages
1
*
RegisteredUser
-registeredUserId:Integer
-userFirstName:String
-userLastName:String
-userEmail:String
-userPassword:String
-userAddress:String
-phoneNum:String
-historyUser: UserHistory
-cartUse: Cart
+registerUser(fName:String,lName:String, email:String, pass:String, add:String.
phNo:String):Boolean
-validateFields(email:String,pass:String):Boolean
-authenticateUser(email:String,pass:String):Boolean
+updateAccount(fName:String,lName:String,email:String,pass:String,add:String,phNo
String):Boolean
+deleteAccount(user: RegisteredUser):Boolean
+getAccount(productName:String):Product
+CompareProducts(productList:Product[2..*]:List
+addProductsInCart(product Product):Boolean
+viewUserHistory():RegisteredUser
+setPriceAlert(product:product,userEmail:String,price:Real):Boolean
+placeOrder(product:Product)
+CommentOnProduct(product:Product,comment:String):Boolean
+rateProduct(product Product,comment):Boolean
+viewProductHistory(product:Product):productHistory
+addProductInCart(productList):Boolean
+logoutUser(use :RegisteredUserr): Boolean
1
maintain
1
Cart
-cartId:Integer
-user:RegisteredUser
-cartName:String
-cartLastModifiedDate:Date
+addProducts(user:RegisteredUser,productList
List):Boolean
+viewCart():Cart
1..*
1
Contain
1
RegisteredUserHistory
-noOfProductRate:Integer
-noOfPriceAlertDevelop:Integer
-noOfCommentsOnProduct:String
+view priceAlertHistory():List
+view rateProductHistory():List
+viewcommentProductHistory():List
NonRegisteredUser
+getProduct(productName:String):Product
+CompareProducts(productList
Product[2..*]):List
Views
1..*
0..*
search
0..*
Product
-productId:Integer
-productName:String
-productQuantity:String
-productPrice:String
-productManfactureDate:String
-productHistory: ProductHistory
-oderProduct: Order
+searchProduct(productName:String):Product
+compareProduct(productList:Product[2..*]:
+rateProduct(ratting:Integer):Boolean
+placeComment(comment:String):Boolean
+developPriceAlert(user:RegisteredUser, productId: Integer
,price:Real):Boolean
+buyProduct(user:RegisteredUser, productName:String,
quantity: Integer):Boolean
+viewCompletetHistory(user:RegisteredUser):ProductHistory
+getProduct(productName: String)
1
Contains
1
ProductHistory
-historyDate:Date
-productRatting:Integer
-numberOfPriceAlert:Integer
-numberOfComments:String
+getProductHistory(user: RegisteredUser, productId:
Integer):productHistory
1
has
1
Order
-orderId:Integer
-user:RegisteredUser
-accountNo:Integer
-price:Real
-deliveryDate:Real
-orderTime:Time
+orderProductDetails(user:RegisteredUser):Boolean

Figure 3.36 Designed Class Diagram (DCD)

Chapter 4

System Implementation

4.1 Introduction:

This chapter is related to system implementation. The chapter mentions the tools, framework and API's used to develop 'Price Comparison Web-based Application (PCWBA)'. Framework selection is about the framework which is used to build system. Similarly, Language selection is about the programming language and database selection is about the database that is used for storing and retrieving data. After the design phase, the implementation phase comes. In this phase we decide how to implement our design and which techniques to use.

4.2 Framework Selection:

The '.NET Framework' is used to develop this system. The .NET Framework provides a comprehensive programming model for building all kinds of applications on Windows, from mobile to web to desktop. The .NET Framework is an application development platform that provides services for building, deploying, and running desktop, Web, and phone applications and Web services. It helps in designing portable scalable and robust applications. The .NET's class library consists of classes, interfaces and value types the help in speeding up the development process and provides access to system functionality. The .Net supplications Components and controls are built on the foundation of .Net Framework types.

4.3 Language Selection:

In .NET Framework, C# is used to successfully develop Price Comparison Web-based Application (PCWBA). C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft. The most recent version is C# 7.0 which was released in 2017 along with Visual Studio 2017.

4.4 Database Selection:

Microsoft SQL Server 2014 is selected for data storage. Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

4.5 Software Used:

Following software is used in developing Price Comparison Web-based Application:

1. Visual Studio 2012 for ASP.Net MVC 5.
2. Microsoft SQL Server for Database.
3. Sublime Text 3 for writing services in PHP Co
4. Microsoft Visio for use case diagram, domain model, ERD and class diagram.
5. Project libre for software project management plan.
6. Dreamweaver CS6 for Web designing.
7. Adobe Photoshop for image manipulations (animations and provide special effect to images).

4.6 Product Comparison:

Comparison of following products is successfully done in PCWBA (Price Comparison Web-based Application (PCWBA). Product list is following:

1. Flights Price Comparison
2. Books Price Comparison
3. Mobiles Price Comparison
4. Other Products.
   i) Clothing
      (a) Kurtas & Shalwar Kameez
      (b) Unstitched Fabric
      (c) Formal Wear
      (d) Dupattas, Scarves & Stoles
      (e) Sport Wear
      (f) Winter Wear
   ii) Shoes
      (a) Casual Shoes
      (b) Formal Shoes
      (c) Boots
      (d) Sandals & Slippers
      (e) Sneakers
      (f) Sports Shoes
   iii) Accessories
      (a) Bags
      (b) Brooches
      (c) Eyewear
      (d) Hair Accessories
      (e) Jewellery
      (f) Watches

iv) Fragrances
  (a) Ambiance
  (b) Attar
  (c) Colognes & Perfumes
  (d) Deodorants
  (e) Gift Set
v) Make up
  (a) Eyes
  (b) Face
  (c) Lips
  (d) Tools & Accessories
vi) Hair Care
  (a) Hair Accessories
  (b) Hair Extensions
  (c) Hair Products
  (d) Hair Styling Tools
vii) Bath & Body
  (a) Bath & Shower
  (b) Cleanses & Lotions
  (c) Sun Care
viii) Personal Care
  (a) Face Care
  (b) Hair Removal
  (c) Oral Care
  (d) Other
ix) Health Care
  (a) Dietary Supplements
  (b) Fitness & Medical Equipment
x) Hands, Feet & Nail Care
  (a) Manicure & Pedicure
  (b) Nail Polish & Accessories
xi) Mobile Phones
  (a) Featured Phones
  (b) Smartphones
xii) Accessories
  (a) Memory Cards
  (b) Phone Batteries
  (c) Phone Chargers
  (d) Phone Charms
  (e) Power Banks
  (f) Surface & Screen Protectors
xiii) Land Line
  (a) Corded Phones

(b) Cordless Phones

xiv) Laptops

(a) MacBook’s
(b) Mini & Netbooks
(c) Notebooks
(d) Tablet PCs

xv) Storage

(a) External Hard Drives
(b) Internal Hard Drives
(c) Memory Cards
(d) Micro SD Cards
(e) Solid State Drives
(f) USB/Flash Drives

xvi) Peripherals & Accessories

(a) Keyboard
(b) Laptop Sleeves
(c) Mouse
(d) Networking
(e) Webcams
(f) Other Computer Accessories

xvii) Components & Spare Parts

(a) Power Supplies
(b) Graphic Cards
(c) Adapters
(d) Batteries & Chargers

xviii) Printers & Scanners

(a) Ink Cartridges & Toners
(b) Laser Printers
(c) Scanners
(d) Deskjet, Inkjet & Office jet Printers

xix) Tablets & Accessories

(a) Bluetooth Accessories
(b) Other Tablet Accessories
(c) Stylus
(d) Tablet Cases & Covers
(e) Tablet Chargers
(f) Tablets and eBook Readers

xx) Desktops & Monitors

(a) All-in-Ones
(b) Monitors

xxi) Memory

(a) Memory Sticks

xxii) iPod & MP3 Players

(a) Audio Recorders
(b) Headphones & Headsets
(c) MP3 Players
(d) Portable Speakers

xxiii) Small Appliances
(a) Beverage Preparation
(b) Cooking
(c) Ironing & Laundry
(d) Mixing & Blending
(e) Other Small Appliances
(f) Toasting

xxiv) Televisions
(a) 3D TVs
(b) LCD TVs
(c) LED TVs
(d) Smart TVs

xxv) Washers & Dryers
(a) Dishwashers
(b) Top Load Dryers
(c) Washer & Dryer Combo
(d) Washing Machines

xxvi) Generators & Power Suppliers
(a) Generators
(b) Inverters
(c) Other Power Accessories
(d) Travel Accessories
(e) UPS
(f) Voltage Converters

xxvii) Cooling & Heating
(a) Air Conditioners
(b) Air Treatment
(c) Fans
(d) Heaters

xxviii) Refrigerators & Freezers
(a) Freezers
(b) Refrigerators

xxix) Home Theater
(a) Hi-Fi Systems
(b) Speaker Systems

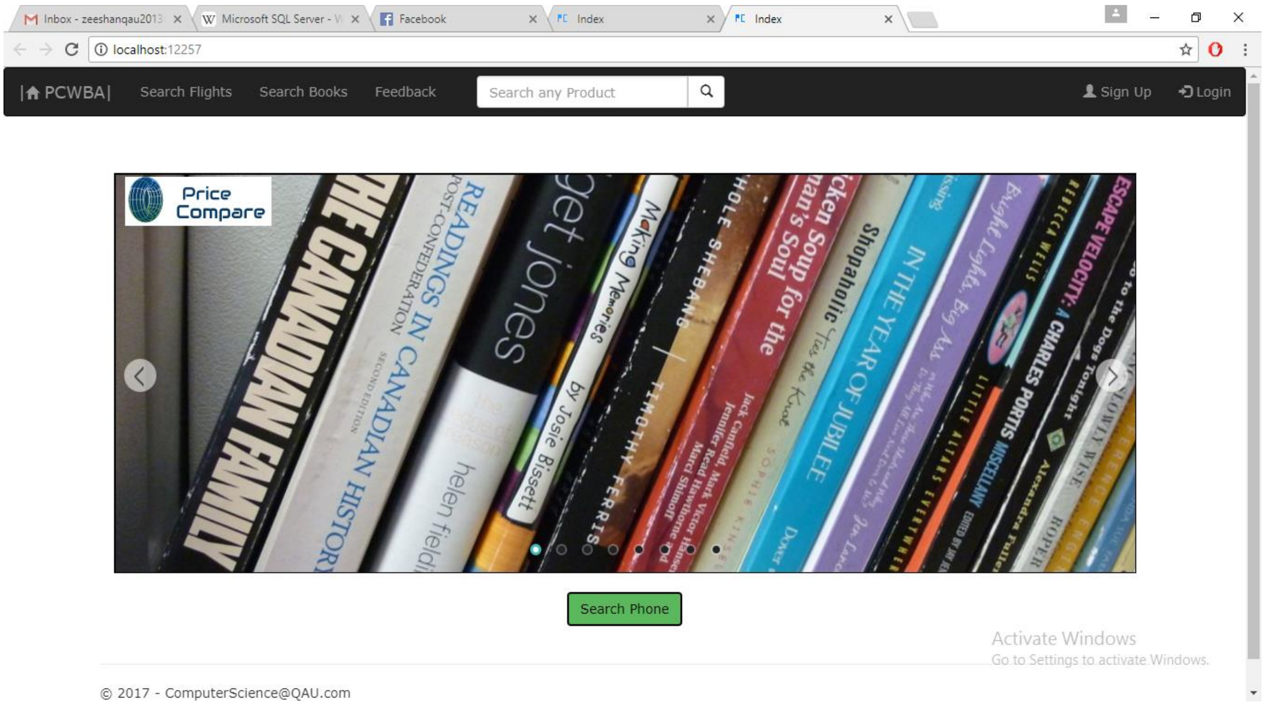
xxx) DVD & Blu-ray Players
(a) Blu-ray Players
(b) DVD Players

xxxi) Hi-Fi & Stereo

(a) Accessories
(b) Speakers
xxxii) TV & Video Accessories
(a) Cables

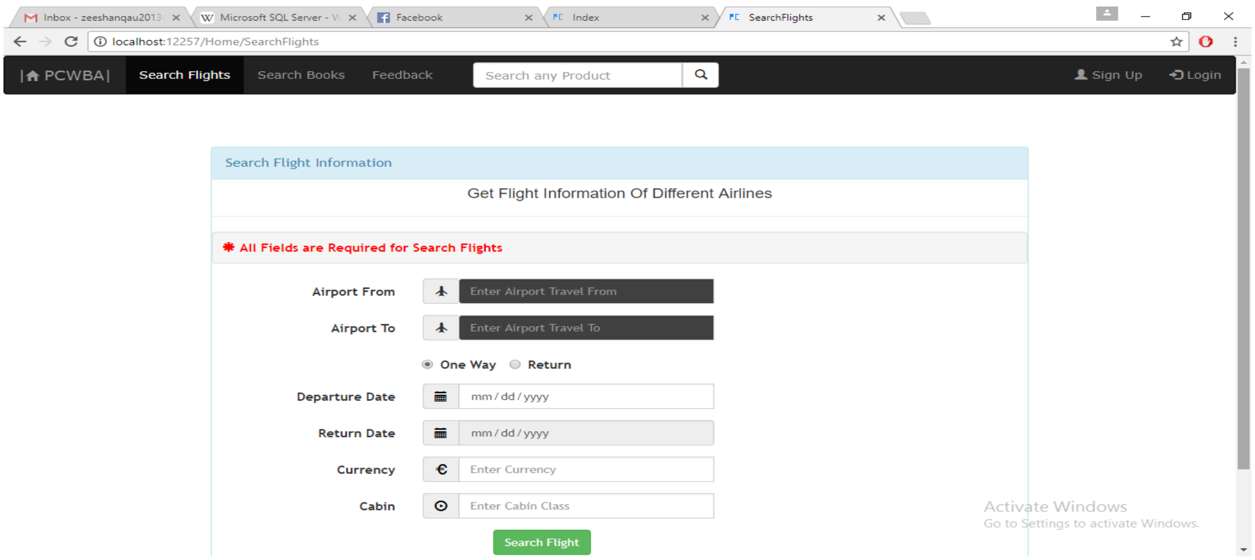
4.4 System Screen Images:

4.4.1 SC- 1: Main Page



© 2017 - ComputerScience@QAU.com

4.4.2 SC- 2: Search Flight



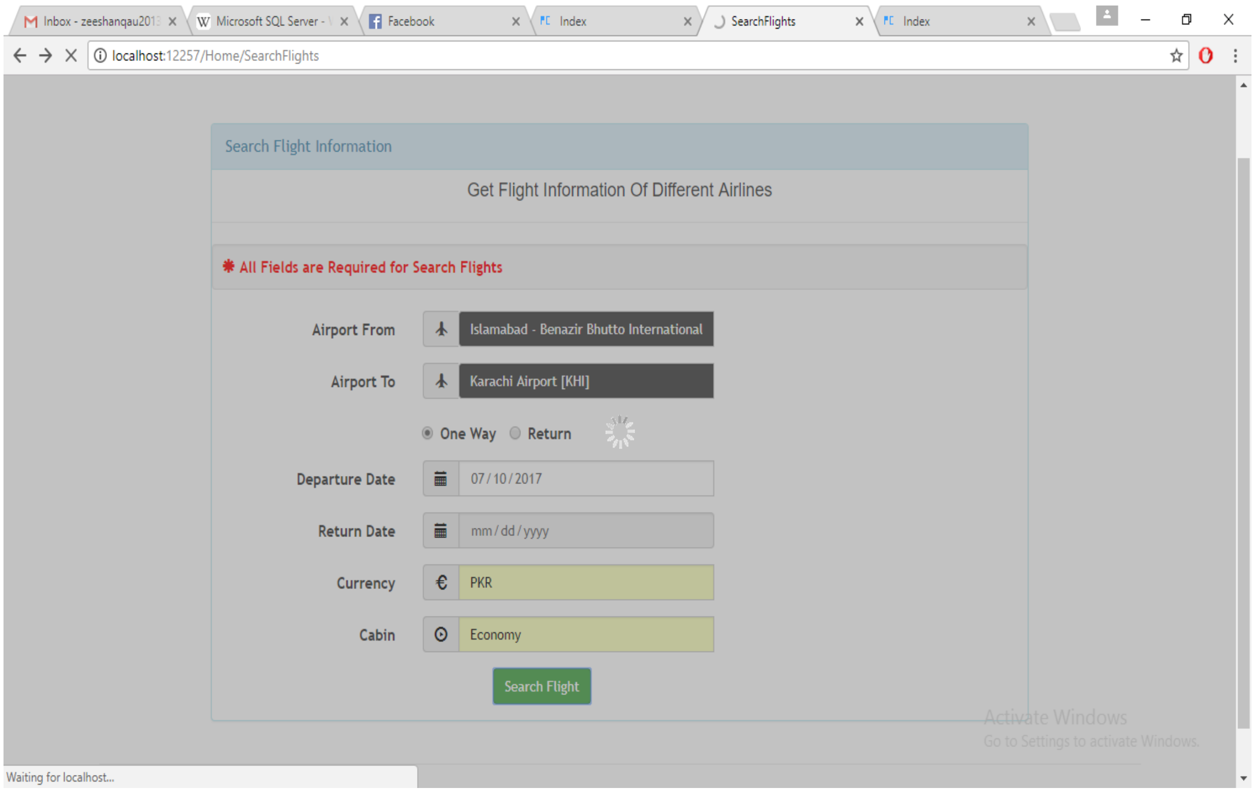
Search Flight Information

Get Flight Information Of Different Airlines

✱ All Fields are Required for Search Flights

Airport From	Enter Airport Travel From
Airport To	Enter Airport Travel To
	● One Way ○ Return
Departure Date	mm / dd / yyyy
Return Date	mm / dd / yyyy
Currency	Enter Currency
Cabin	Enter Cabin Class

4.4.3 SC- 3: Search Flight-II



Search Flight Information

Get Flight Information Of Different Airlines

✱ All Fields are Required for Search Flights

Airport From: Islamabad - Benazir Bhutto International

Airport To: Karachi Airport [KHI]

● One Way ○ Return

Departure Date: 07/10/2017

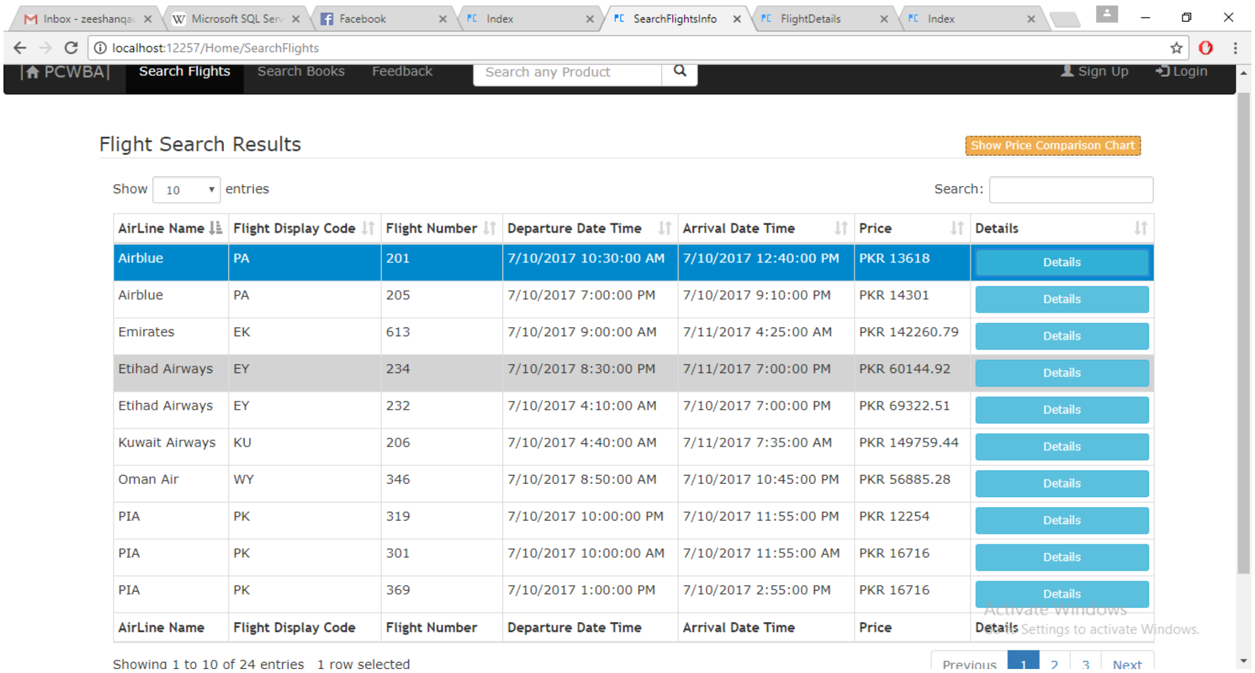
Return Date: mm/dd/yyyy

Currency: PKR

Cabin: Economy

Search Flight

4.4.4 SC-4: Flight Results



Flight Search Results

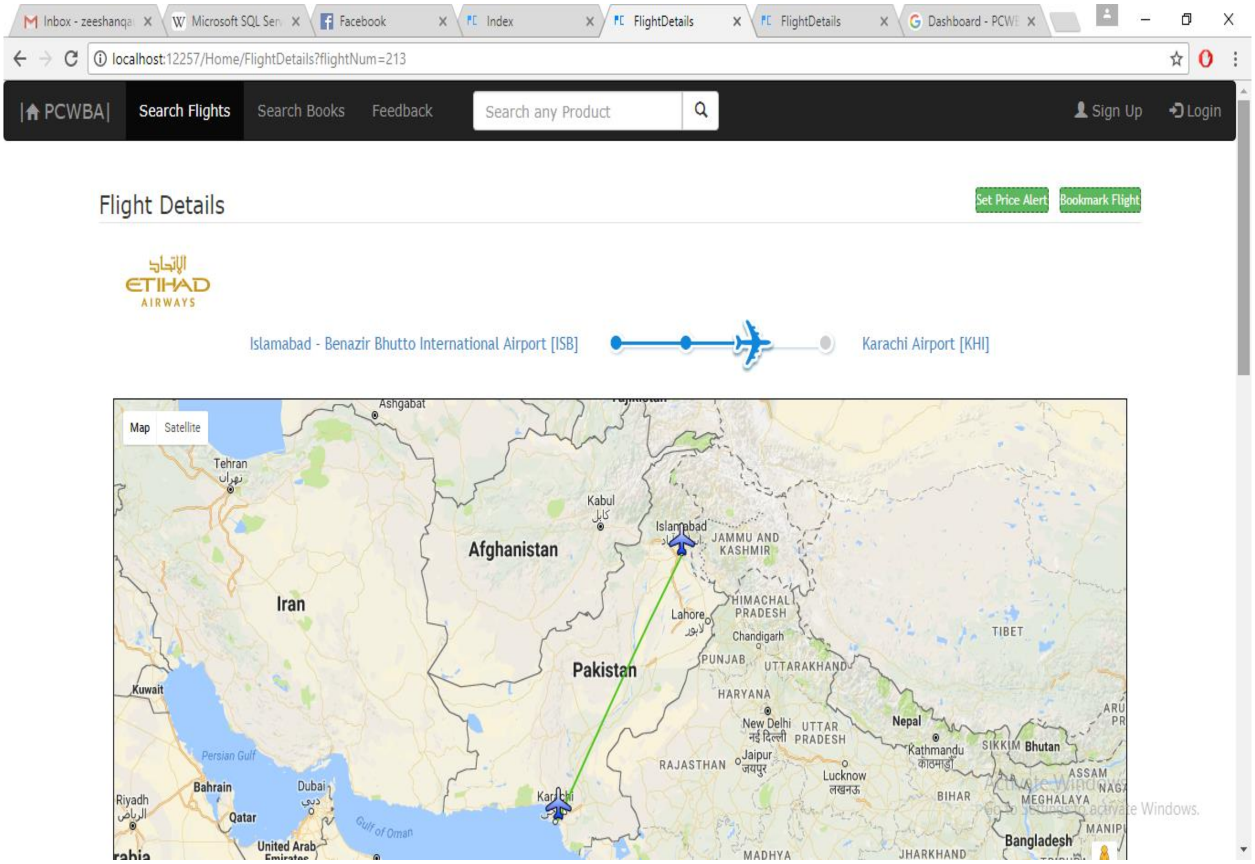
Show 10 entries Search:

AirLine Name	Flight Display Code	Flight Number	Departure Date Time	Arrival Date Time	Price	Details
Airblue	PA	201	7/10/2017 10:30:00 AM	7/10/2017 12:40:00 PM	PKR 13618	Details
Airblue	PA	205	7/10/2017 7:00:00 PM	7/10/2017 9:10:00 PM	PKR 14301	Details
Emirates	EK	613	7/10/2017 9:00:00 AM	7/11/2017 4:25:00 AM	PKR 142260.79	Details
Etihad Airways	EY	234	7/10/2017 8:30:00 PM	7/11/2017 7:00:00 PM	PKR 60144.92	Details
Etihad Airways	EY	232	7/10/2017 4:10:00 AM	7/10/2017 7:00:00 PM	PKR 69322.51	Details
Kuwait Airways	KU	206	7/10/2017 4:40:00 AM	7/11/2017 7:35:00 AM	PKR 149759.44	Details
Oman Air	WY	346	7/10/2017 8:50:00 AM	7/10/2017 10:45:00 PM	PKR 56885.28	Details
PIA	PK	319	7/10/2017 10:00:00 PM	7/10/2017 11:55:00 PM	PKR 12254	Details
PIA	PK	301	7/10/2017 10:00:00 AM	7/10/2017 11:55:00 AM	PKR 16716	Details
PIA	PK	369	7/10/2017 1:00:00 PM	7/10/2017 2:55:00 PM	PKR 16716	Details
AirLine Name	Flight Display Code	Flight Number	Departure Date Time	Arrival Date Time	Price	Details

Showing 1 to 10 of 24 entries 1 row selected

Previous 1 2 3 Next

4.4.5 SC-5: Flight Detail

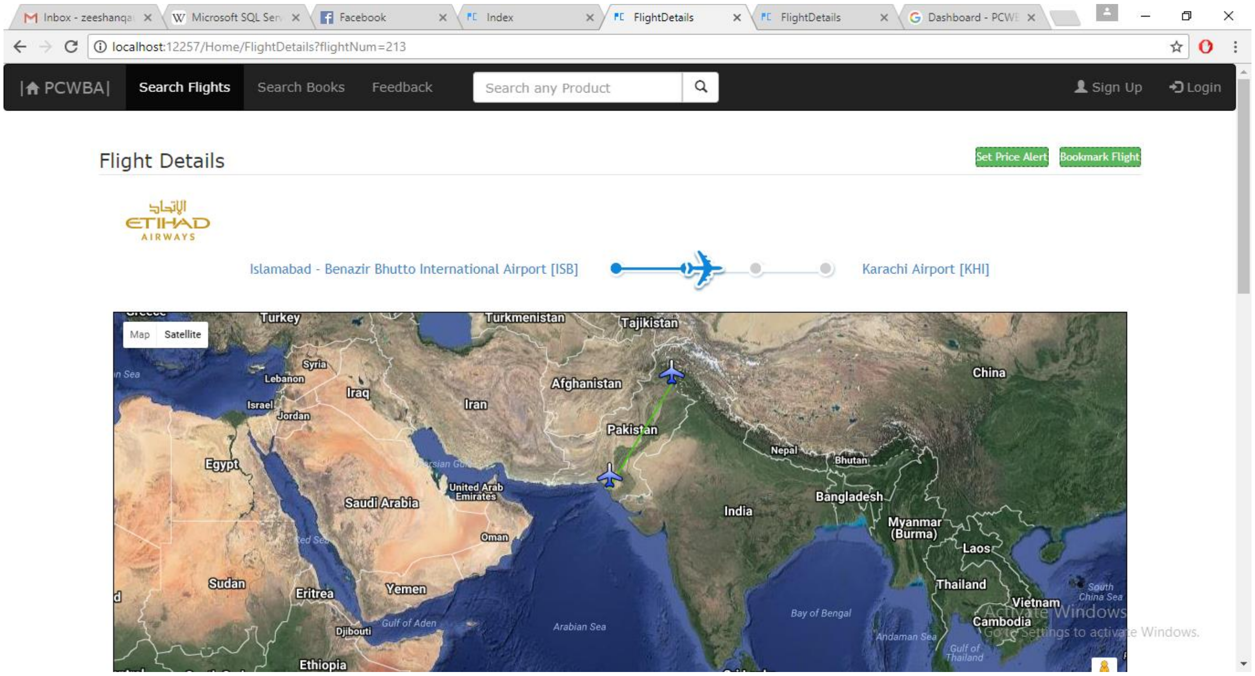


Flight Details

Islamabad - Benazir Bhutto International Airport [ISB]

Karachi Airport [KHI]

4.4.6 SC-6: Flight Detail-II

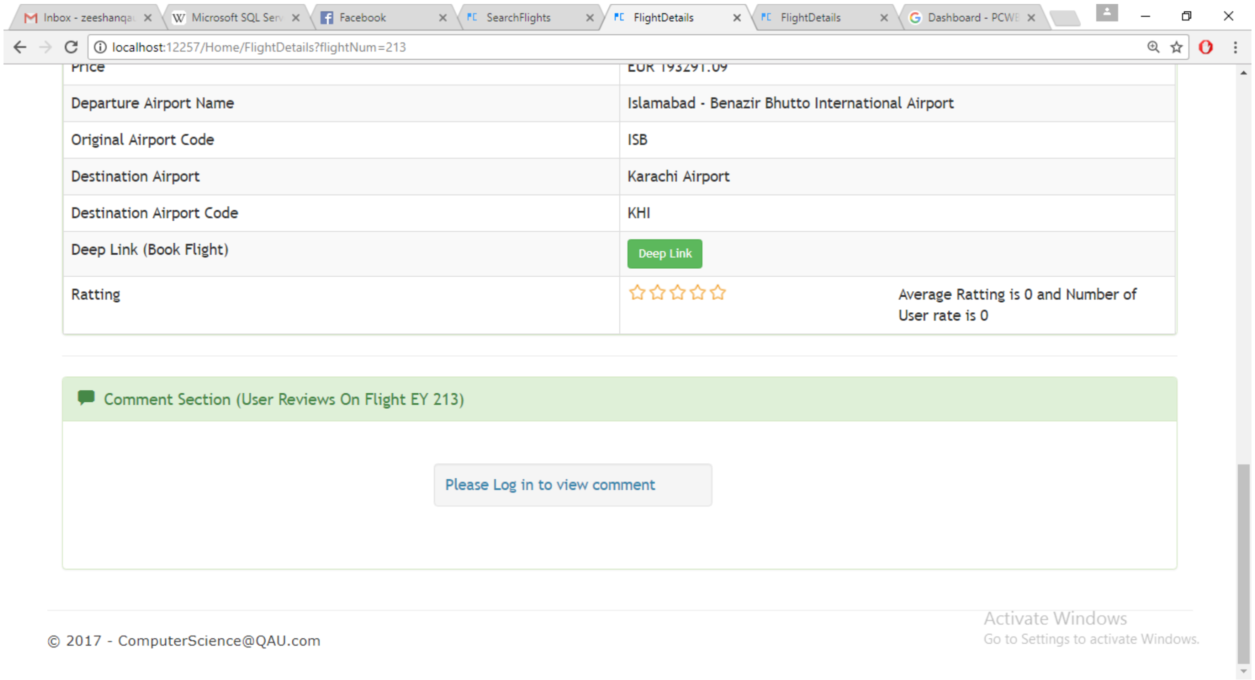


Flight Details

Islamabad - Benazir Bhutto International Airport [ISB]

Karachi Airport [KHI]

4.4.7 SC-7: Flight Detail Comment and Rate



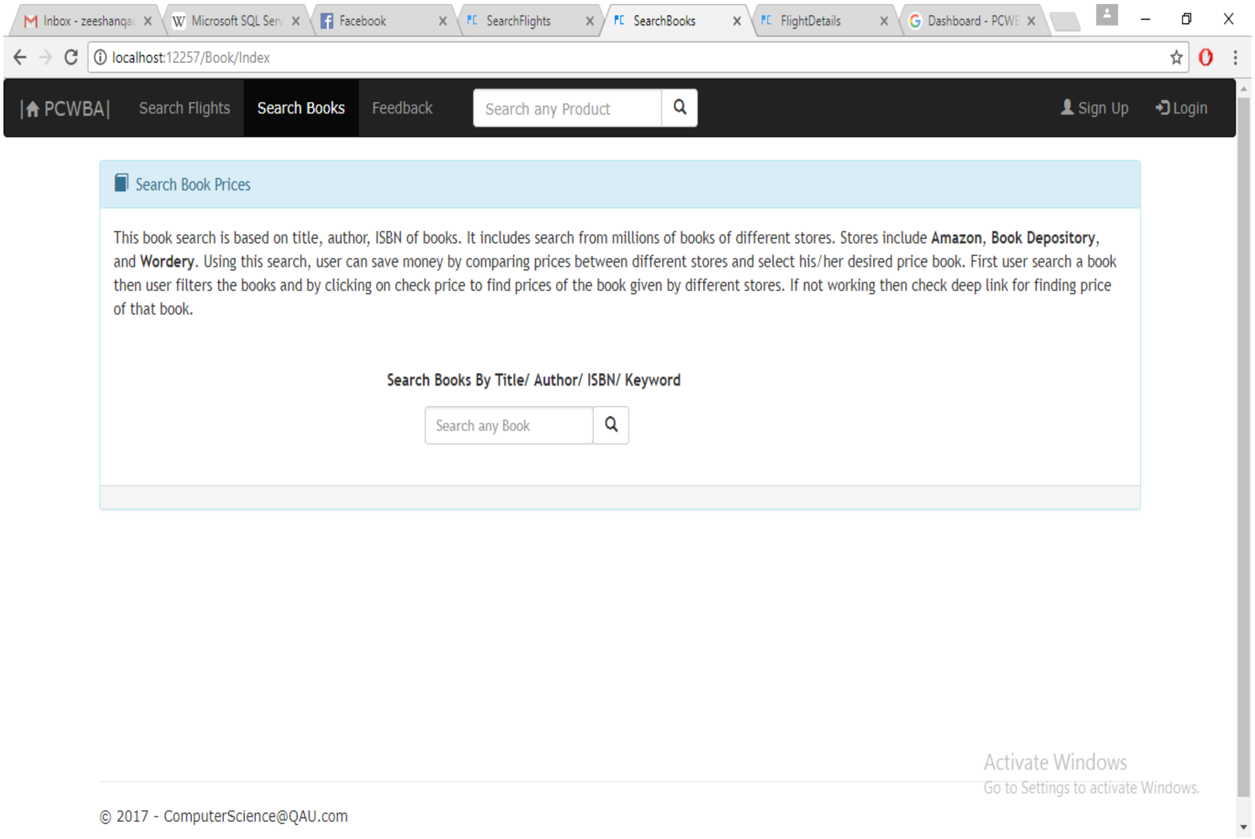
Price	EUR 193291.09
Departure Airport Name	Islamabad - Benazir Bhutto International Airport
Original Airport Code	ISB
Destination Airport	Karachi Airport
Destination Airport Code	KHI
Deep Link (Book Flight)	Deep Link
Ratting	☆☆☆☆☆ Average Ratting is 0 and Number of User rate is 0

Comment Section (User Reviews On Flight EY 213)

Please Log in to view comment

© 2017 - ComputerScience@QAU.com

4.4.8 SC-8: Search Books



Search Book Prices

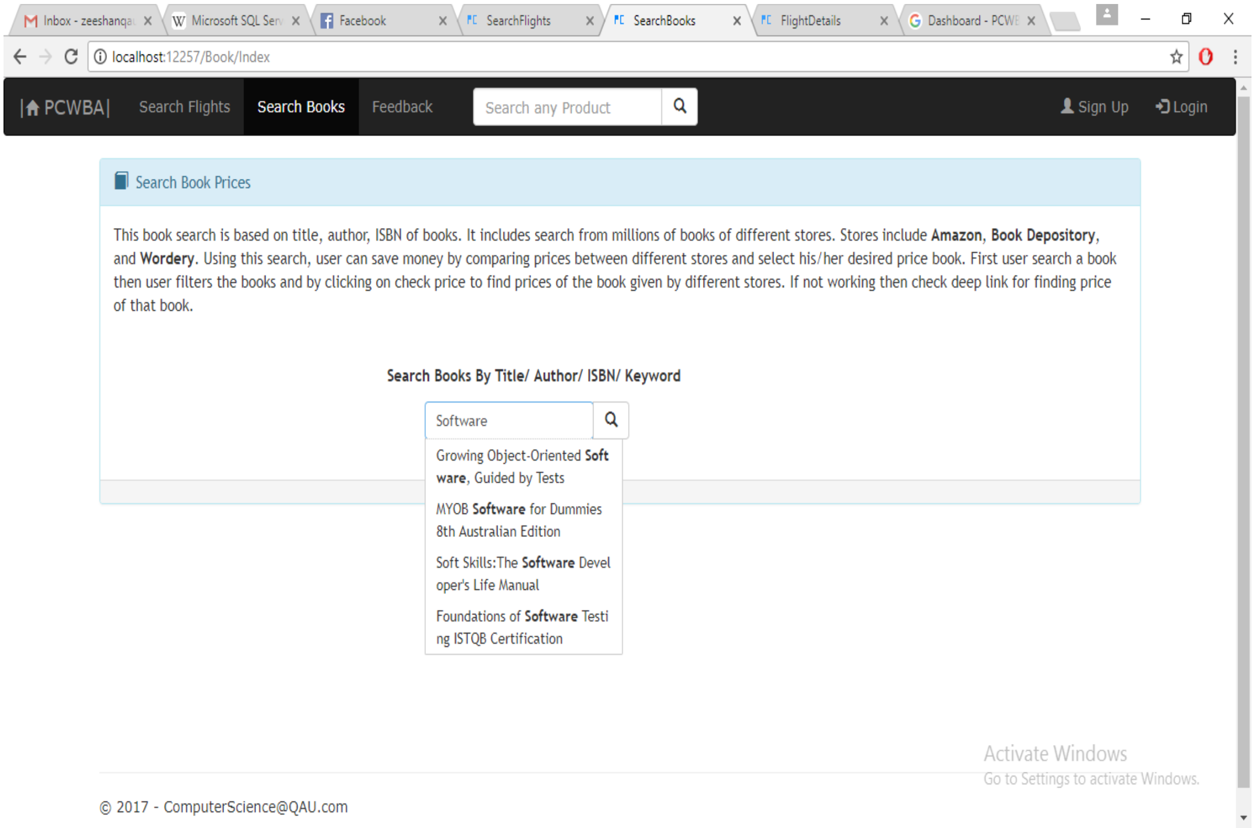
This book search is based on title, author, ISBN of books. It includes search from millions of books of different stores. Stores include **Amazon**, **Book Depository**, and **Wordery**. Using this search, user can save money by comparing prices between different stores and select his/her desired price book. First user search a book then user filters the books and by clicking on check price to find prices of the book given by different stores. If not working then check deep link for finding price of that book.

Search Books By Title/ Author/ ISBN/ Keyword

Search any Book

© 2017 - ComputerScience@QAU.com

4.4.9 SC-9: Search Book-II



Search Book Prices

This book search is based on title, author, ISBN of books. It includes search from millions of books of different stores. Stores include **Amazon**, **Book Depository**, and **Wordery**. Using this search, user can save money by comparing prices between different stores and select his/her desired price book. First user search a book then user filters the books and by clicking on check price to find prices of the book given by different stores. If not working then check deep link for finding price of that book.

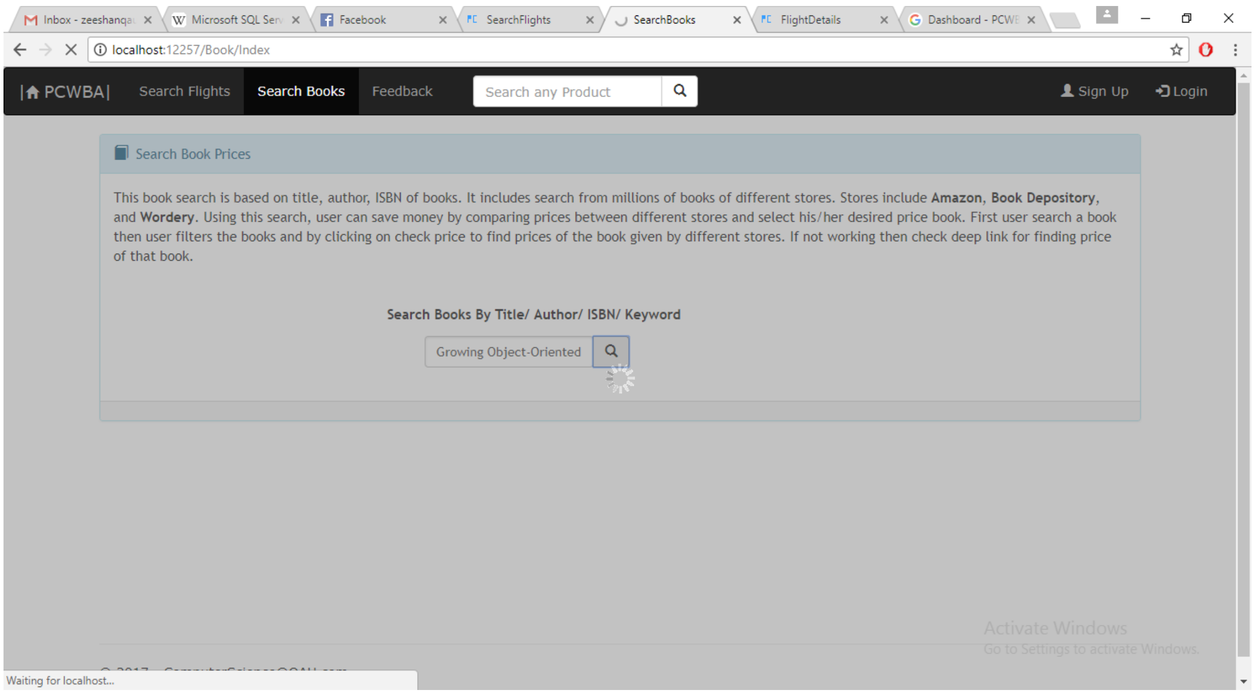
Search Books By Title/ Author/ ISBN/ Keyword

Software

- Growing Object-Oriented **Software**, Guided by Tests
- MYOB **Software** for Dummies 8th Australian Edition
- Soft Skills:The **Software** Developer's Life Manual
- Foundations of **Software** Testing ISTQB Certification

© 2017 - ComputerScience@QAU.com

4.4.10 SC-10: Search Books-III



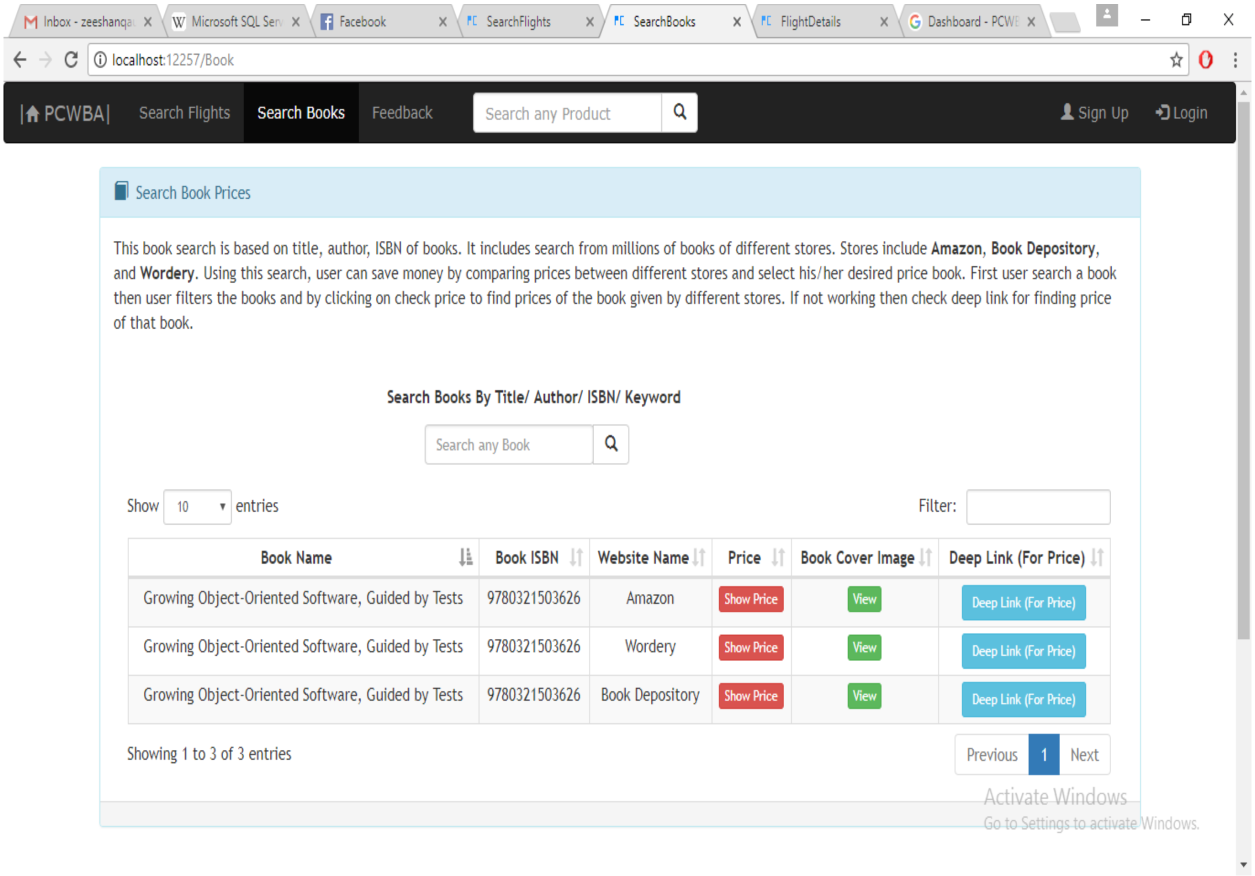
Search Book Prices

This book search is based on title, author, ISBN of books. It includes search from millions of books of different stores. Stores include **Amazon**, **Book Depository**, and **Wordery**. Using this search, user can save money by comparing prices between different stores and select his/her desired price book. First user search a book then user filters the books and by clicking on check price to find prices of the book given by different stores. If not working then check deep link for finding price of that book.

Search Books By Title/ Author/ ISBN/ Keyword

Growing Object-Oriented

4.4.11 SC-11: Search Book Result



Search Book Prices

This book search is based on title, author, ISBN of books. It includes search from millions of books of different stores. Stores include **Amazon**, **Book Depository**, and **Wordery**. Using this search, user can save money by comparing prices between different stores and select his/her desired price book. First user search a book then user filters the books and by clicking on check price to find prices of the book given by different stores. If not working then check deep link for finding price of that book.

Search Books By Title/ Author/ ISBN/ Keyword

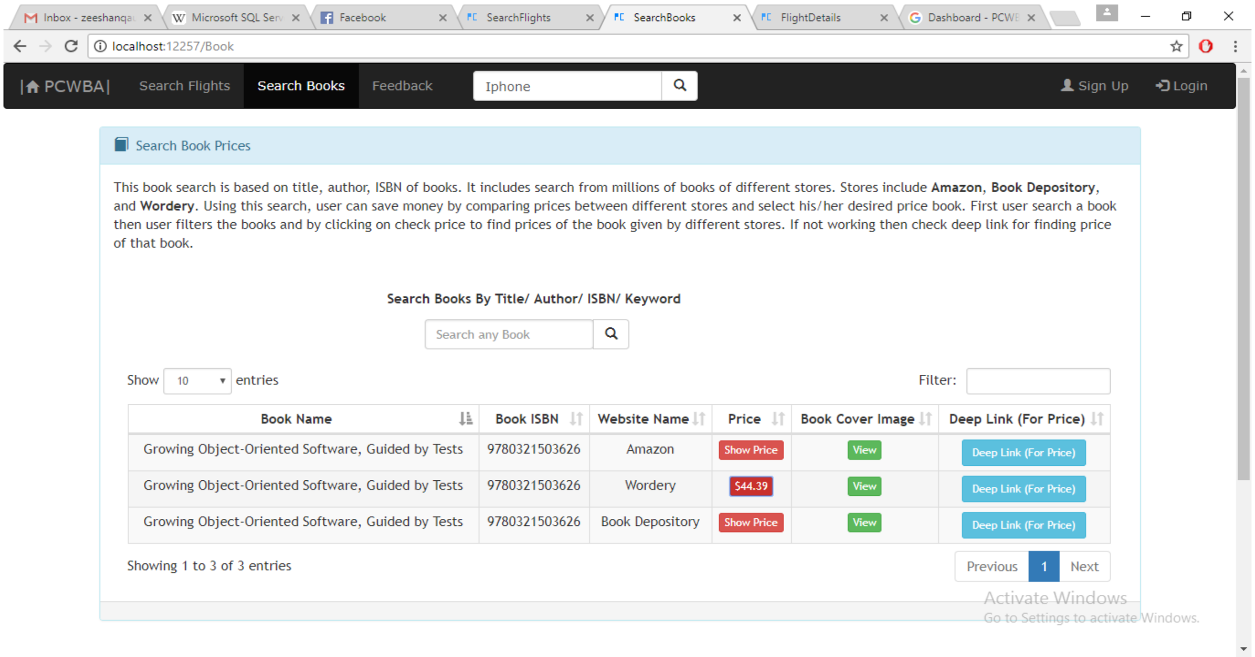
Search any Book

Show 10 entries   Filter:

Book Name	Book ISBN	Website Name	Price	Book Cover Image	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Amazon	Show Price	View	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Wordery	Show Price	View	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Book Depository	Show Price	View	Deep Link (For Price)

Showing 1 to 3 of 3 entries   Previous 1 Next

4.4.12 SC-12: Book Price Details



Search Book Prices

This book search is based on title, author, ISBN of books. It includes search from millions of books of different stores. Stores include **Amazon**, **Book Depository**, and **Wordery**. Using this search, user can save money by comparing prices between different stores and select his/her desired price book. First user search a book then user filters the books and by clicking on check price to find prices of the book given by different stores. If not working then check deep link for finding price of that book.

Search Books By Title/ Author/ ISBN/ Keyword

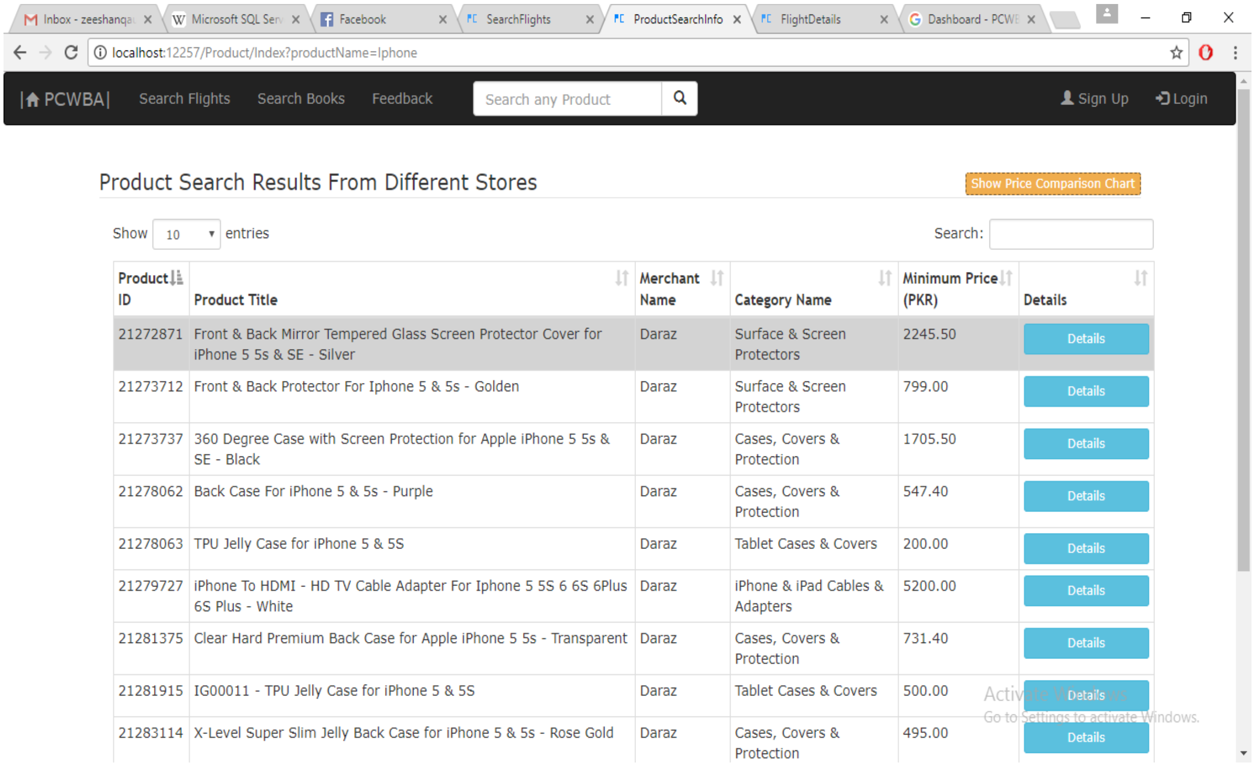
Search any Book

Show 10 entries   Filter:

Book Name	Book ISBN	Website Name	Price	Book Cover Image	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Amazon	Show Price	View	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Wordery	$44.39	View	Deep Link (For Price)
Growing Object-Oriented Software, Guided by Tests	9780321503626	Book Depository	Show Price	View	Deep Link (For Price)

Showing 1 to 3 of 3 entries   Previous 1 Next

4.4.13 SC-13: Search Other Product Result



localhost:12257/Product/Index?productName=Iphone

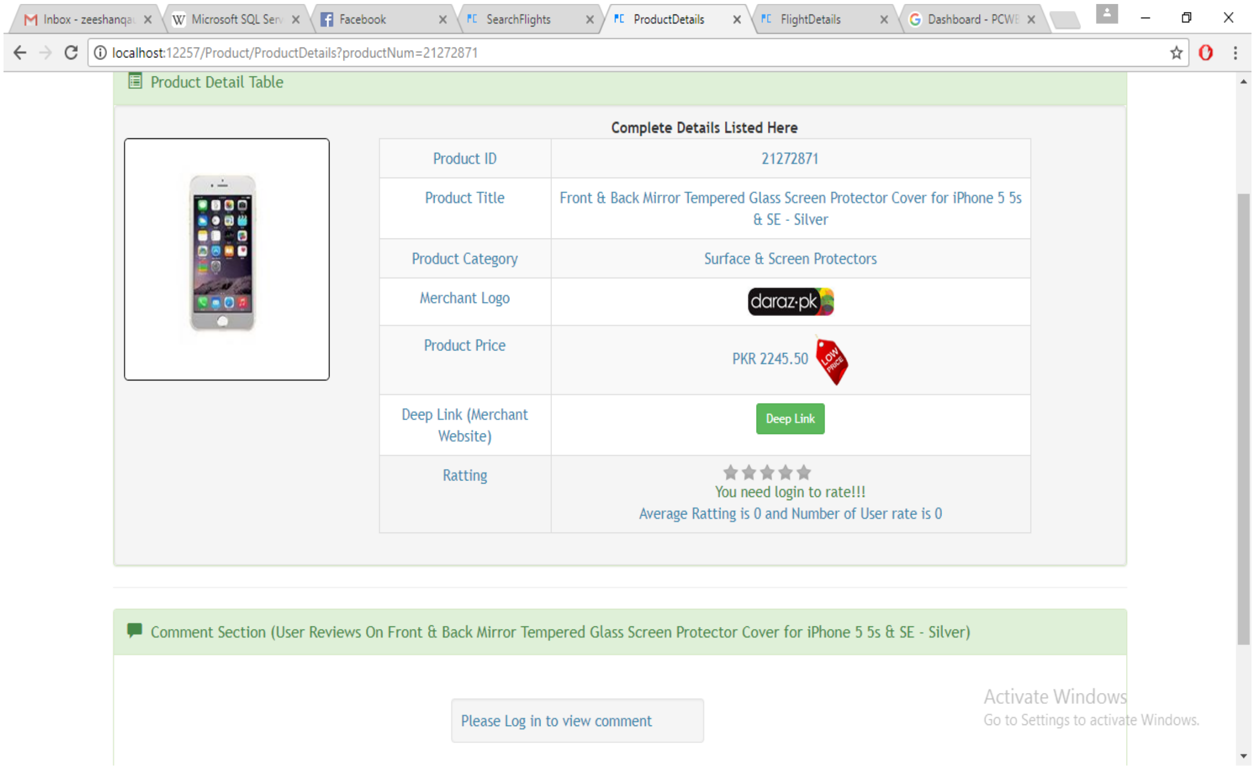
Product Search Results From Different Stores

Show 10 entries

Search:

Product ID	Product Title	Merchant Name	Category Name	Minimum Price (PKR)	Details
21272871	Front & Back Mirror Tempered Glass Screen Protector Cover for iPhone 5 5s & SE - Silver	Daraz	Surface & Screen Protectors	2245.50	Details
21273712	Front & Back Protector For Iphone 5 & 5s - Golden	Daraz	Surface & Screen Protectors	799.00	Details
21273737	360 Degree Case with Screen Protection for Apple iPhone 5 5s & SE - Black	Daraz	Cases, Covers & Protection	1705.50	Details
21278062	Back Case For iPhone 5 & 5s - Purple	Daraz	Cases, Covers & Protection	547.40	Details
21278063	TPU Jelly Case for iPhone 5 & 5S	Daraz	Tablet Cases & Covers	200.00	Details
21279727	iPhone To HDMI - HD TV Cable Adapter For Iphone 5 5S 6 6S 6Plus 6S Plus - White	Daraz	iPhone & iPad Cables & Adapters	5200.00	Details
21281375	Clear Hard Premium Back Case for Apple iPhone 5 5s - Transparent	Daraz	Cases, Covers & Protection	731.40	Details
21281915	IG00011 - TPU Jelly Case for iPhone 5 & 5S	Daraz	Tablet Cases & Covers	500.00	Details
21283114	X-Level Super Slim Jelly Back Case for iPhone 5 & 5s - Rose Gold	Daraz	Cases, Covers & Protection	495.00	Details

4.4.14 SC-14: Search Product Details



localhost:12257/Product/ProductDetails?productNum=21272871

Product Detail Table

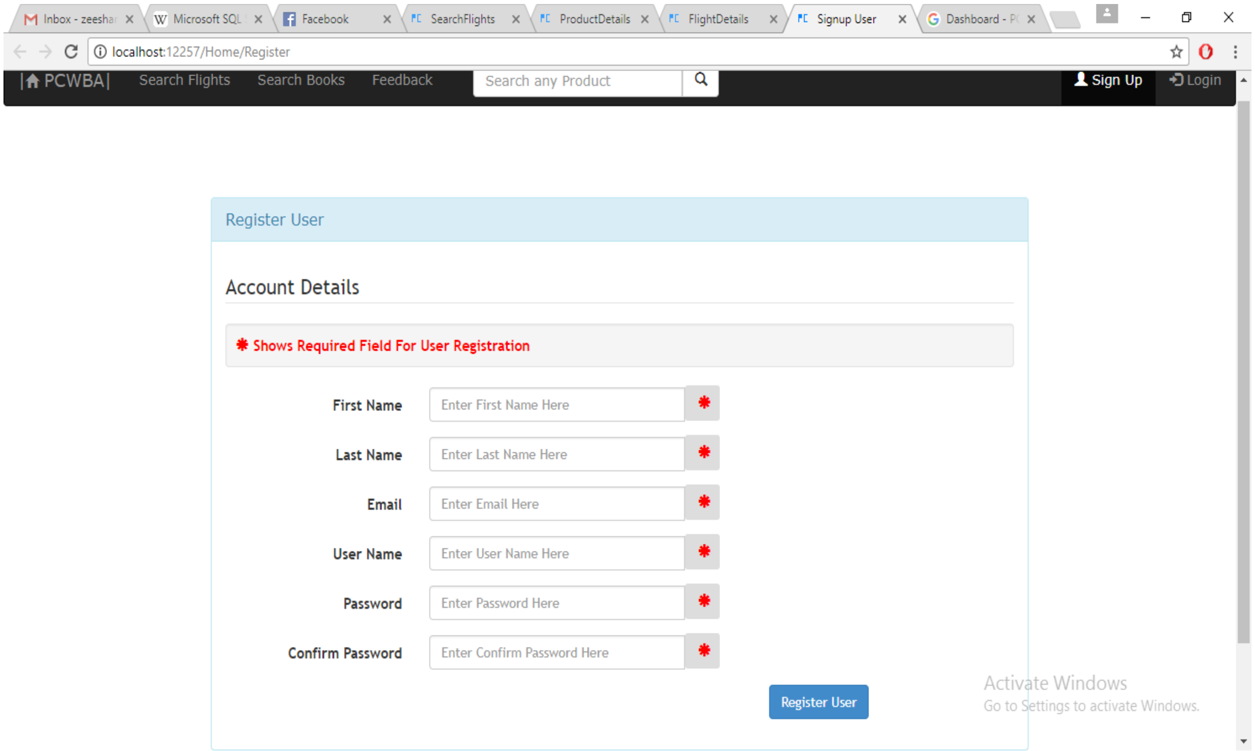
Complete Details Listed Here

Product ID	21272871
Product Title	Front & Back Mirror Tempered Glass Screen Protector Cover for iPhone 5 5s & SE - Silver
Product Category	Surface & Screen Protectors
Merchant Logo	daraz.pk
Product Price	PKR 2245.50
Deep Link (Merchant Website)	Deep Link
Ratting	★★★★★ You need login to rate!!! Average Ratting is 0 and Number of User rate is 0

Comment Section (User Reviews On Front & Back Mirror Tempered Glass Screen Protector Cover for iPhone 5 5s & SE - Silver)

Please Log in to view comment

4.4.15 SC-15: Register User



Register User

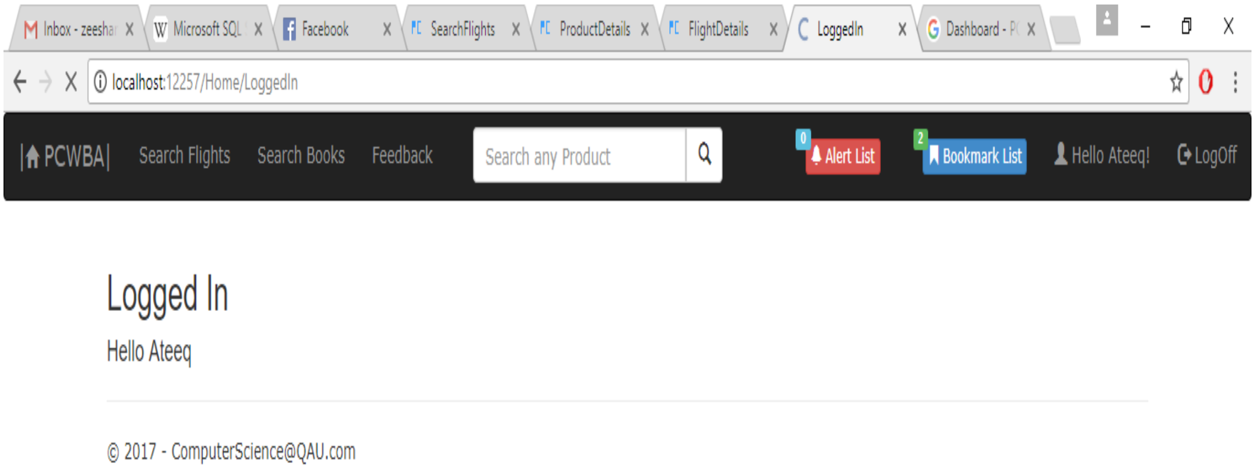
Account Details

✱ Shows Required Field For User Registration

First Name	Enter First Name Here	✱
Last Name	Enter Last Name Here	✱
Email	Enter Email Here	✱
User Name	Enter User Name Here	✱
Password	Enter Password Here	✱
Confirm Password	Enter Confirm Password Here	✱

Register User

4.4.16 SC-16: Login

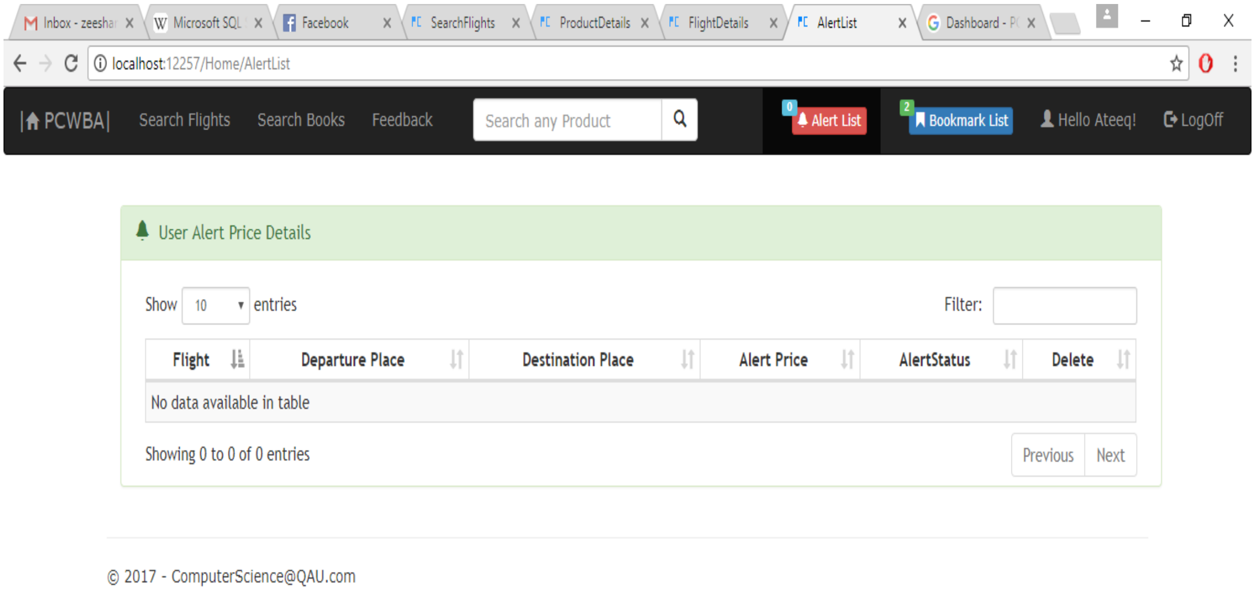


Logged In

Hello Ateeq

© 2017 - ComputerScience@QAU.com

4.4.17 SC-17: User Alert



User Alert Price Details

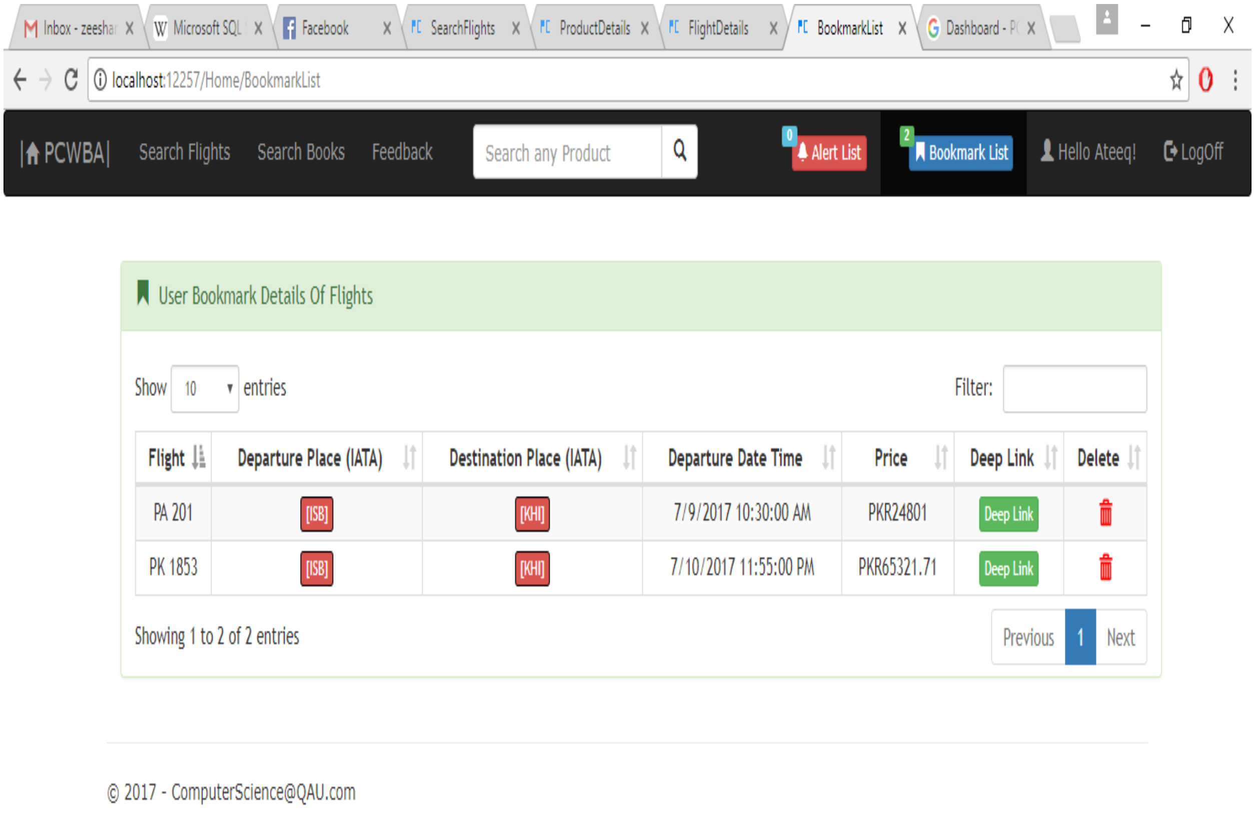
Show 10 entries Filter:

Flight	Departure Place	Destination Place	Alert Price	AlertStatus	Delete
No data available in table					

Showing 0 to 0 of 0 entries

© 2017 - ComputerScience@QAU.com

4.4.18 SC-18: Bookmark Product



User Bookmark Details Of Flights

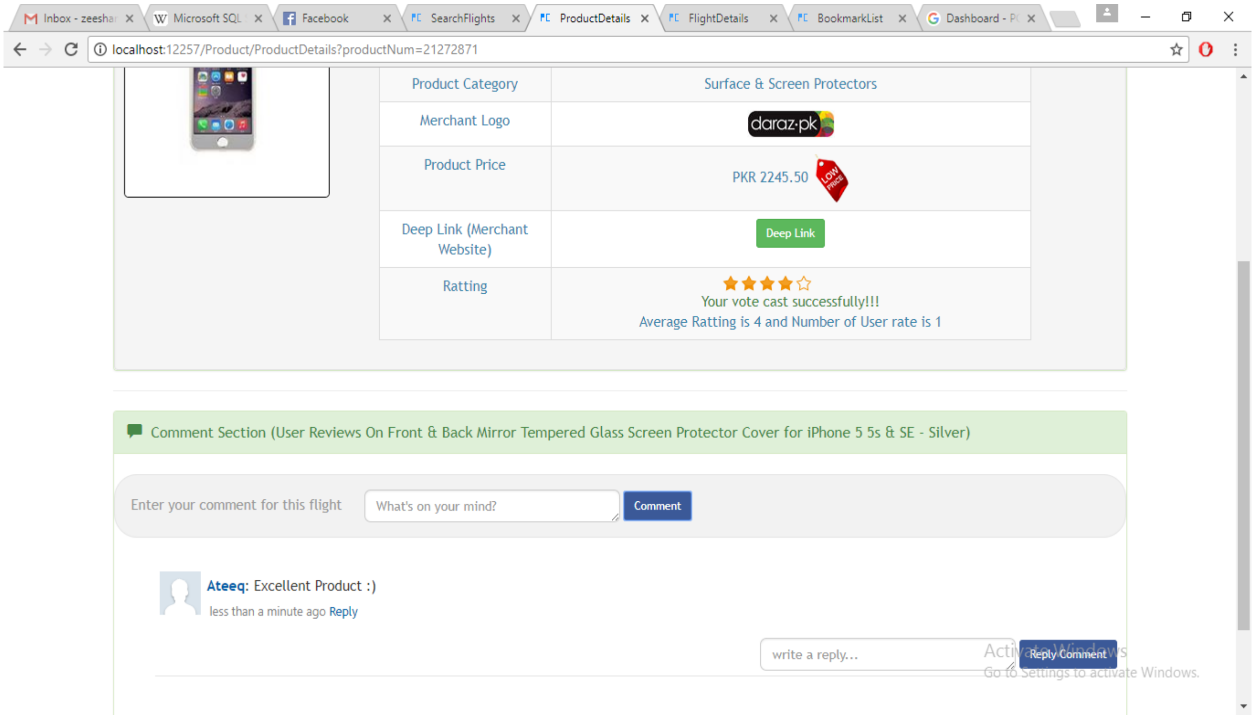
Show 10 entries Filter:

Flight	Departure Place (IATA)	Destination Place (IATA)	Departure Date Time	Price	Deep Link	Delete
PA 201	[ISB]	[KHI]	7/9/2017 10:30:00 AM	PKR24801	Deep Link	🗑
PK 1853	[ISB]	[KHI]	7/10/2017 11:55:00 PM	PKR65321.71	Deep Link	🗑

Showing 1 to 2 of 2 entries

© 2017 - ComputerScience@QAU.com

4.4.19 SC-19: Comment and Rate Product



localhost:12257/Product/ProductDetails?productNum=21272871

Product Category	Surface & Screen Protectors
Merchant Logo	daraz·pk
Product Price	PKR 2245.50
Deep Link (Merchant Website)	Deep Link
Ratting	★★★★☆ Your vote cast successfully!!! Average Ratting is 4 and Number of User rate is 1

Comment Section (User Reviews On Front & Back Mirror Tempered Glass Screen Protector Cover for iPhone 5 5s & SE - Silver)

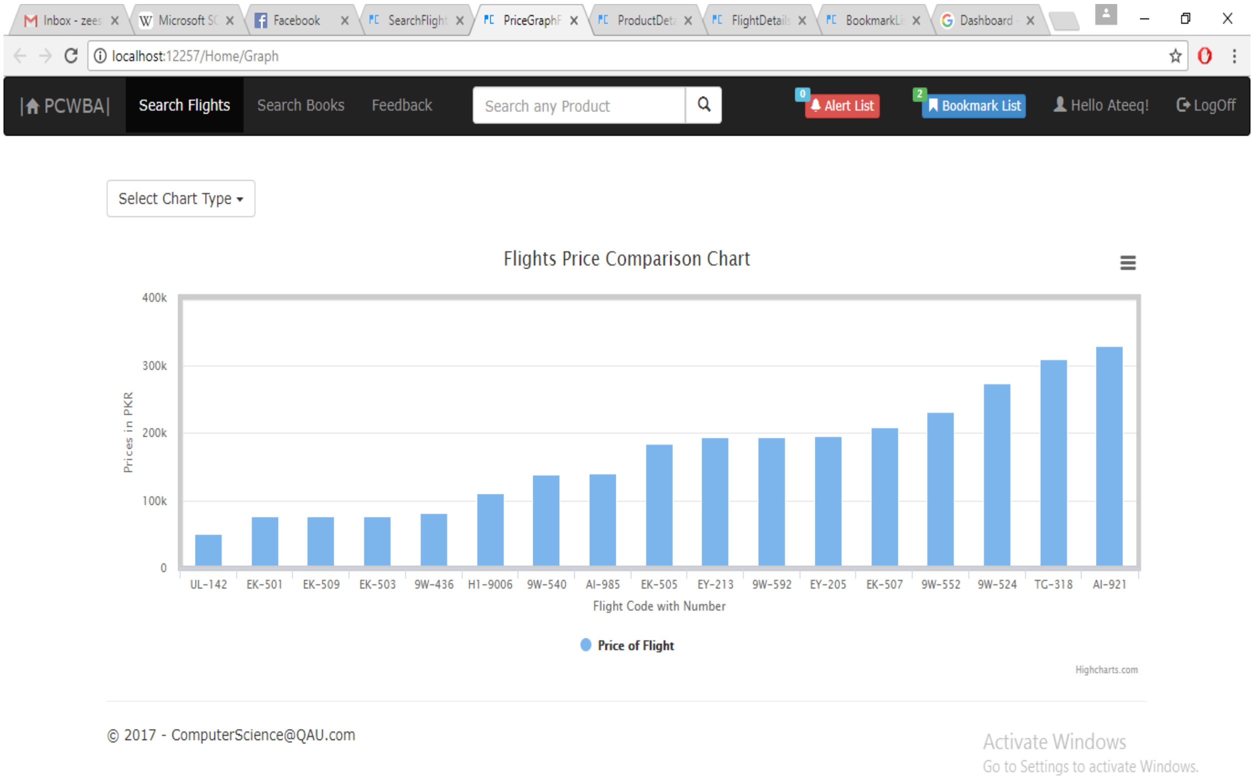
Enter your comment for this flight What's on your mind? Comment

Ateeq: Excellent Product :)
less than a minute ago Reply

write a reply... Reply Comment

Activate Windows
Go to Settings to activate Windows.

4.4.20 SC-20: Price Comparison Graph

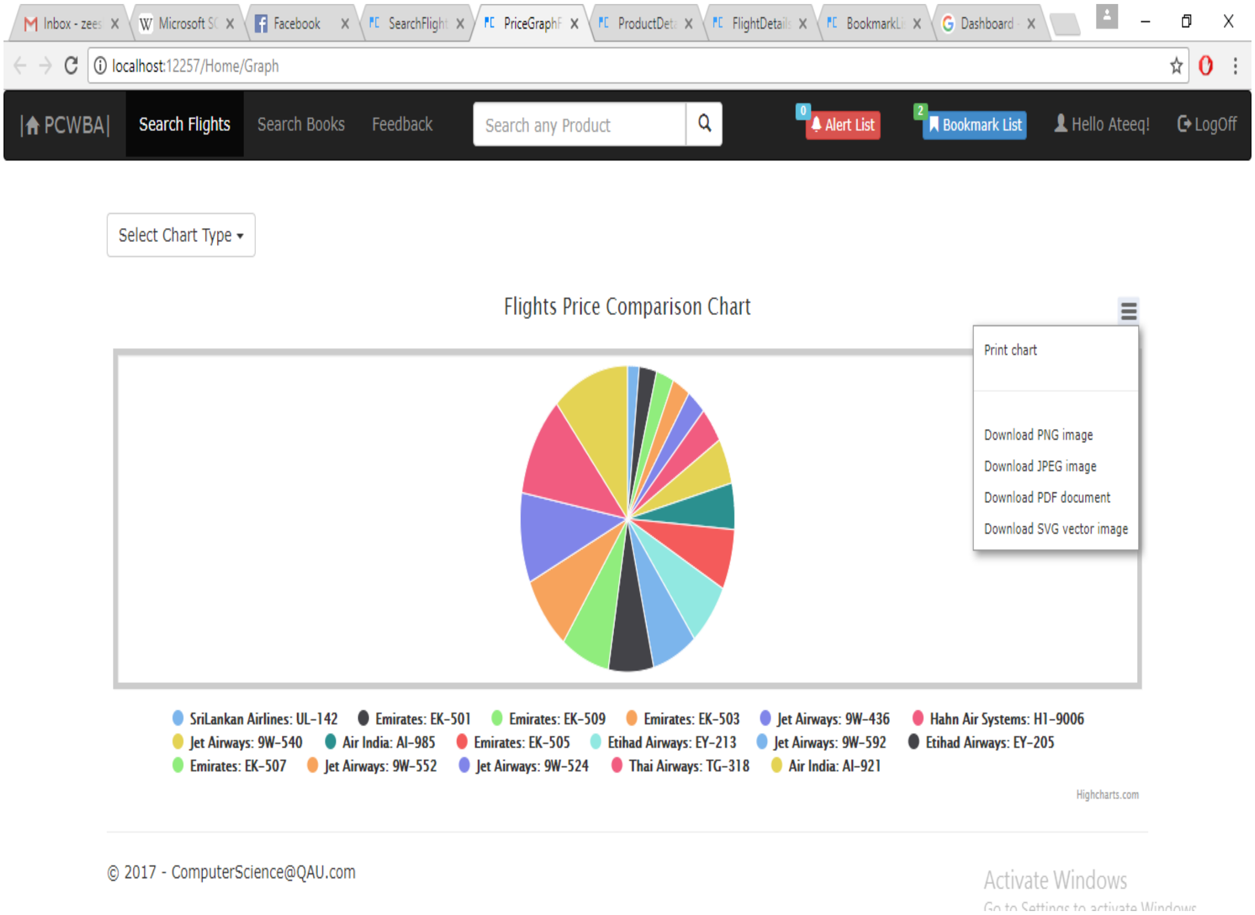


localhost:12257/Home/Graph

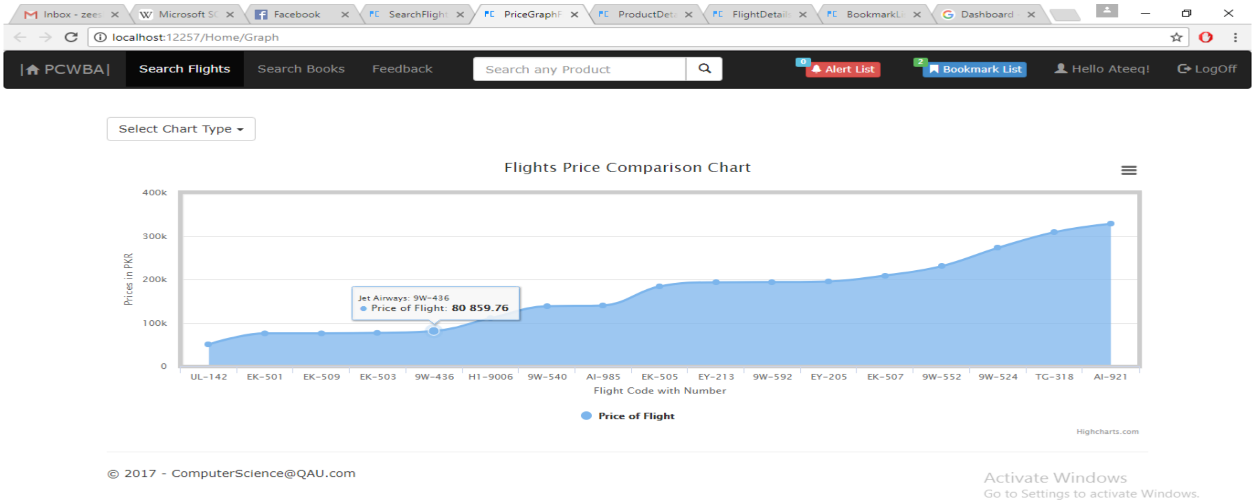
© 2017 - ComputerScience@QAU.com

Activate Windows
Go to Settings to activate Windows.

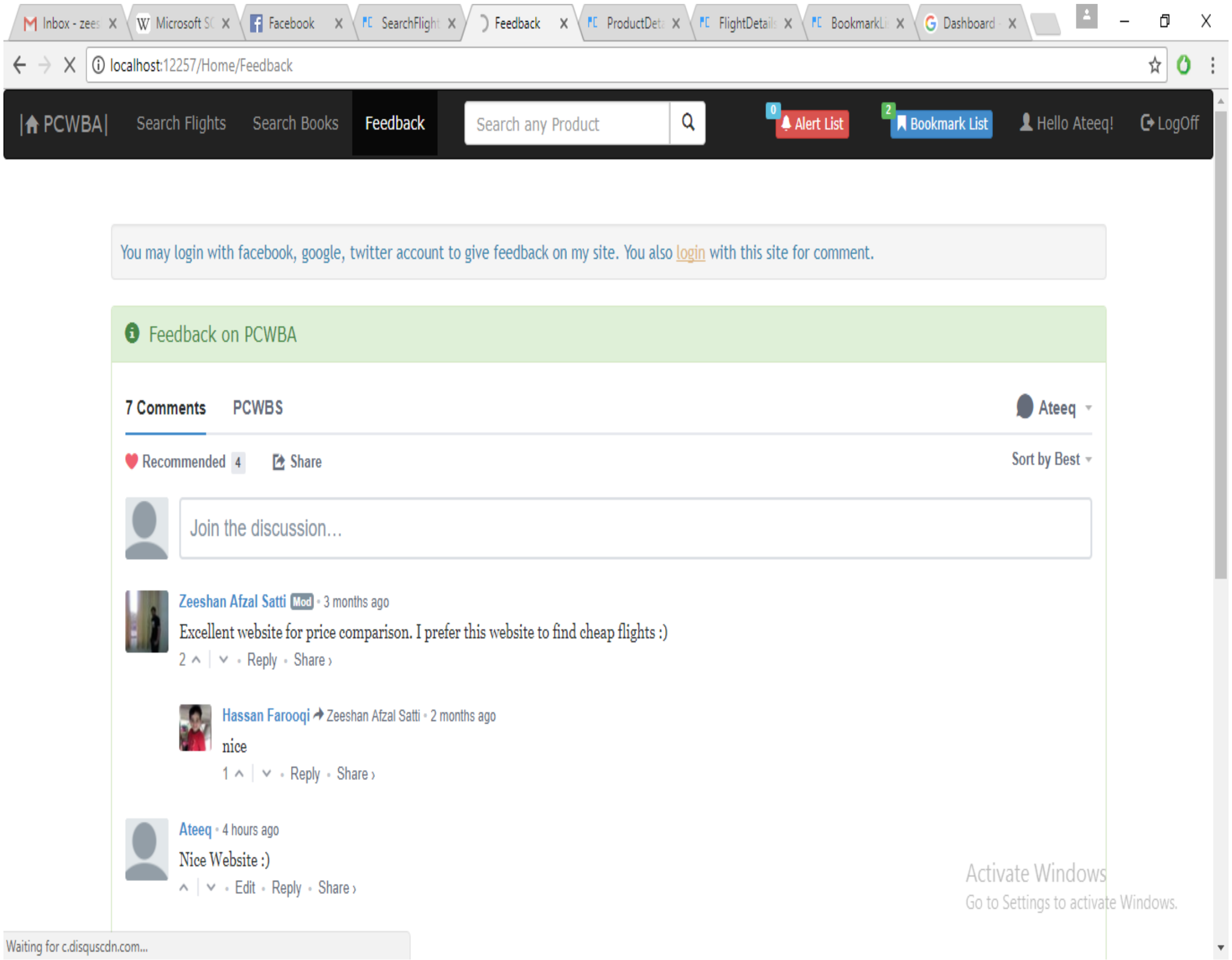
4.4.21 SC-21: Price Comparison Graph-II



4.4.22 SC-22: Price Comparison Graph-III



4.4.23 SC-23: Price Comparison Graph-III



You may login with facebook, google, twitter account to give feedback on my site. You also login with this site for comment.

Feedback on PCWBA

7 Comments PCWBS Ateeq

Recommended 4 Share Sort by Best

Join the discussion...

Zeeshan Afzal Satti Mod • 3 months ago
Excellent website for price comparison. I prefer this website to find cheap flights :)
2 ^ | v • Reply • Share ›

Hassan Farooqi → Zeeshan Afzal Satti • 2 months ago
nice
1 ^ | v • Reply • Share ›

Ateeq • 4 hours ago
Nice Website :)
^ | v • Edit • Reply • Share ›

Activate Windows
Go to Settings to activate Windows.
Waiting for c.disquscdn.com...

Chapter 5

Software Test Document

5.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Testing documentation involves the documentation of artifacts that should be developed before or during the testing of Software. This chapter specifying the test approach, test plan which include features to be tested and features not to be tested, testing tools and techniques and test cases that involve a set of steps that can be used while performing testing tasks.

5.1.1 Test Approach

Test approach is implementation of the test strategy for a specific project. It describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. The Test Approach describe the sources (project documentation) for requirements that will be used to drive the test design. This includes the modeling outputs (e.g., outputs from the process modeling activities such as use cases and task scenarios) from which test cases are developed. The choice of test approaches or test strategy is one of the most powerful factor in the success of the test effort and the accuracy of the test plans and estimates. The testing approach for PCWBA (Price Comparison Web-based Application) is **Unit Testing**. Unit test is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. So, we construct PCWBA in step by step manner, that’s why unit testing is suitable to test because the main focus of unit testing to segregate each part of the program and test that the individual parts are working correctly. This means that for any function or procedure when a set of inputs are given then it should return the proper values. It should handle the failures gracefully during the course of execution when any invalid input is given. White Box Testing method is used for executing the unit test.

5.2 Test Plan

A test plan describes how testing will be accomplished on a software product, together with the resources and schedule needed. It is the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. It also defines the size of the test effort.

5.2.1 Feature to be Tested

All the major functions (functional requirements) of PCWBA (Price Comparison Web-based Application) are to be tested. Following are the list of features to be tested:

- Login
- Signup
- Search Product
- Show User History
- Show Product History
- Compare Products
- Customer Comments and Ratting
- Price Alert
- Add Product To Cart
- Order Product
- Approved User Account Request
- Approved Edit User Account Request
- Approved Delete User Account Request

5.2.2 Features not to be Tested

All the features are tested to increase reliability of PCWBA (Price Comparison Web-based Application). So all features are to be tested.

5.2.3 Testing Tools and Environment

Test Environment consists of elements that support test execution with software, hardware and network configured. **Manual test environment** is used in PCWBA (Price Comparison Web-based Application). Manual testing is crucial for testing software applications more thoroughly. **ApTest Manager** is a web-based tool for managing manual testing. ApTest Manager solves challenges by recording and tracking the tests used in your software and hardware testing projects and the results of running them. Testing information becomes instantly available to everyone - always accurate, always current. It also allows you complete access to monitor the progress of projects during every phase from start to finish.

5.3 Test Cases

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. It is good practice to make your test cases atomic.

5.3.1 Login Test Case

Test Case ID	TC-001
Actors	Registered user/Admin
Purpose	Login to account

Input	Frequency	Expected Output	Observed Output	Verdict
User enters his correct email and password.	2	User successfully login.	User successfully login.	Pass
User enters his incorrect email and password.	1	User not exist.	User not exist.	Pass
User submit login form with empty email and empty password.	4	Empty fields are not allowed.	Empty fields are not allowed.	Pass

5.3.2 Sign up Test Case

Test Case ID	TC-002
Actors	Non-Registered User
Purpose	Register user

Input	Frequency	Expected Output	Observed Output	Verdict
User enter data in all the required fields with correct information.	5	User successfully register in system.	User successfully register in system.	Pass
User enter incorrect information like wrong mail syntax (zeeshanqau2013@gmail).	2	Enter valid email for proceed.	Enter valid email for proceed.	Pass
User submit correct information but user of that name already exist.	2	Username already exist in our system.	Username already exist in our system.	Pass

User send some malicious data and some form.	6	Entered data contain malicious entries.	Entered data contain malicious entries.	Pass

5.3.3 Search Product Test Case

Test Case ID	TC-003
Actors	Registered user
Purpose	Search a product

Input	**Frequency**	**Expected Output**	**Observed Output**	**Verdict**
User enter the name of product.	7	User successfully found and view the details of the product.	User successfully found and view the details of the product.	Pass
User enter his incorrect product name.	2	Product not exist.	Product not exist.	Pass
User submit form with empty product name.	3	All fields are required to submit form.	All fields are required to submit form	Pass

5.3.4 Approve User Account Test Case

Test Case ID	TC-004
Actors	Admin
Purpose	Approve account

Input	**Frequency**	**Expected Output**	**Observed Output**	**Verdict**
Admin selects the user to approve account.	3	User create account request is approved.	User create account request is approved.	Pass

5.3.5 Set Price Alert Test Case

Test Case ID	TC-005
Actors	Registered user

Purpose set price alert

Input	Frequency	Expected Output	Observed Output	Verdict
User entered an alert price for specific product and submit it.	3	User successfully avail the set price alert for that product.	Price alert for a product is set successfully.	Pass
User entered an invalid price for alert and submit it.	2	Invalid price entered by a user.	Price must be a valid.	Pass

5.3.6 View Product History Test Case

Test Case ID TC-006
Actors Registered user
Purpose View product history

Input	Frequency	Expected Output	Observed Output	Verdict
User selects a month to view history of product.	4	User successfully view the history of product.	User successfully view the history of product.	Pass
User not selects any month or not choose any other option to view history of product.	2	History not exist from that month to current month.	History not exist from that month to current month.	Pass

5.3.7 Rate Product Test Case

Test Case ID TC-007
Actors Registered user
Purpose Rate product

Input	Frequency	Expected Output	Observed Output	verdict
User selects ratting for a product.	3	User successfully rate the product.	User successfully rate the product.	Pass

User submit ratting without rate a product.	2	Rate a product for submitting ratting.	Rate a product for submitting ratting.	Pass

5.3.8 Delete Account Test Case

Test Case ID	TC-008
Actors	Admin
Purpose	Delete account

Input	**Frequency**	**Expected Output**	**Observed Output**	**Verdict**
Admin selects the user to delete account.	3	User delete account request approved.	User delete account request approved.	Pass

5.3.9 Comment Product Test Case

Test Case ID	TC-009
Actors	Registered user
Purpose	Comment product

Input	**Frequency**	**Expected Output**	**Observed Output**	**Verdict**
User writes comment on a product and submit it.	5	User successfully write comment on product.	User successfully write a comment on product.	Pass
User submit empty comment on a product.	1	Comment must be at least 100 words.	Comment must be at least 100 words.	Pass

5.3.10 View User History Test Case

Test Case ID	TC-0010
Actors	Registered user
Purpose	View user history

Input	Frequency	Expected Output	Observed Output	Verdict
User select a month to view his/her history.	2	User successfully view his/her history.	User successfully view his/her history.	Pass
User not select any month or not choose any other option to view history of product.	1	History not exist from that month to current month.	History not exist from that month to current month.	Pass

5.3.11 Edit Account Test Case

Test Case ID TC-0011
Actors Admin
Purpose Edit account

Input	Frequency	Expected Output	Observed Output	Verdict
Admin selects the user to edit account.	5	User edit account request is approved.	User edit account request is approved	Pass

5.3.12 Logout User Test Case

Test Case ID TC-0012
Actors Registered user
Purpose Logout User

Input	Frequency	Expected Output	Observed Output	Verdict
User selects a logout button.	4	User successfully logout from system and session destroyed.	User successfully logout from system and session destroyed	Pass

5.3.13 Order Product Test Case

Test Case ID	TC-0013
Actors	Registered user
Purpose	Order Product

Input	Frequency	Expected Output	Observed Output	Verdict
User selects a product to order and fill the required fields with valid entries.	2	User successfully ordered that product.	User successfully ordered that product.	Pass
User selects product to order but submit order with empty fields.	1	Empty fields are not allowed.	Empty fields are not allowed.	Pass

5.3.14 Add Product in Cart Test Case

Test Case ID	TC-0014
Actors	Registered user
Purpose	Add product in cart

Input	Frequency	Expected Output	Observed Output	Verdict
User selects a product and submit to add product in cart	6	User successfully add product in cart.	User successfully add product in cart	Pass
User selects product to add in cart. Selected product already present in cart.	2	Product already present in cart.	Product already present in cart.	Pass

Chapter 6

Conclusion and Future Enhancements

6.1 Introduction

In last chapter the overview of implemented system is given. It is described that who much functionalities can be done provided by the system. At last future work and enhancements, are a given. It describes what future enhancements are possible and what will be the effect.

6.2 Conclusion

The basic purpose of Price Comparison Web-based Application is to provide a platform to the people for comparing prices of same product from different companies/vendors. Basically, this web-based application is a price comparison portal for online comparison of prices. This application provide analysis on the price history of products. User view prices comparison of same product from different vendors in the form of graph. It save user times and also user money. Customer can also provide feedback and rate the product about their experience. The System is completely developed and testing is performed from different aspects to develop confidence on system.

6.3 Future Enhancements

It is possible to enhance the application in future. Here are some features that can be added to application:

- Price alert is till now for mobile prices and flight prices. In future, this feature is extend to other category products.
- There is some push notification on special sale packages offered by different companies like well.pk, daraz.pk etc.
- In future, more stores is added and different web crawlers programs is written to get data from those stores.
- In future, there is a message box through which user can communicate with admin in case of query or some other types.
- This could be android app in future.

References:

[Books]

[1] Pratap Mohapatra, Software Engineering A Lifecycle Approach, New Age Publications, 2010.

[2] Roger S. Pressman, Software Engineering - A Practitioner's Approach, McGraw Hill, 7th Edition, 2010.

[3] Ian Somerville, Software Engineering, Addison Wesley, Ninth Edition, 2010.

[4] Applying UML and Patterns, Craig Larman, Second Edition.

[5] Thomas M. Connolly and Carolyn E. Begg, Database Systems, A Practical Approach to Design, Implementation and Management, Fourth Edition, 2005.