Transparent Sheet



## ECOM. Mobile phone store

## Live link - <https://ecomstore-pwn-mca2020.netlify.app/>

#### A DISSERTATION

Submitted in Partial Fulfilment of the Requirements for the Award of the Degree of

MASTER OF COMPUTER APPLICATION (M.C.A) BY

Pawan Kumar Tiwari ROLL NO: 680240600004

Roorkee Institute of Technology (RIT)

(UTTRAKHAND TECHNICAL UNIVERSITY)

2020

INSTITUTE CERTIFICATE

Company CERTIFICATE

## CANDIDATE’S DECLARATION

I hereby declare that the work which is being presented in this project work entitled “Ecom Mobile phone store” is an authentic record of my own work carried out at **Mechlin software tech, mohali** as requirements of six months Industrial Training for the award of the degree of Master in Computer Applications at **Roorkee Institute of Technology (RIT), Roorkee** during the period January 2020 to May 2020 under the supervision and guidance of **Miss. \*\*\*\*\* .**

I have not submitted the matter embodied in this project work anywhere for the award of any degree or diploma.

##### Pawan kumar Tiwari

**Roll No: 680240600004**

**ACKNOWLEDGEMENT**

It is my proud privilege to express my profound gratitude to the entire management of Roorkee Institute of Technology for providing me with the opportunity to avail the excellent facilities and infrastructure. The knowledge and values inculcated have proved to be of immense help at the very start of my career. Special thanks to my faculties for having provided us an excellent infrastructure at **RIT** I am also thankful to **Our HOD MR. Deepak sir** for creating better learning experiences.

I am grateful to **MR. Deepak** for his astute guidance, constant encouragement and sincere support for this project work. I am thankful to all the teaching and non-teaching staff members.

My heartfelt gratitude and sincere thanks to my mentor and industry guide for the project, for making me familiar with the intricacies of software development and aiding me in my work with proficient guidance, unfailing encouragement and immense help.

I am also grateful to **To Mechlin Technology, Mohali** for providing me with an opportunity to work with them and undertake a project of such importance.

I feel pride and privileged in expressing my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavour if I do not place my acknowledgement.

##### Pawan Kumar Tiwari

**Roll No: 680240600004**

**ABSTRACT**

The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e- commerce site is to sell goods and services online. This project deals with developing an e-commerce website for online shopping. It provides the user with a catalogue of different product available for purchase in the

store. In order to facilitate online purchase a shopping cart is provided to the user. The system is implemented using a 2-tier approach, with a backend database, a middle tier REACTjs and a web browser as the front end client. In order to develop online shopping application

use HTML, CSS, JavaScript, REACTjs and database . This is a project with the objective to develop a basic website where consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 2.1 | Agile Methodology | 10 |
| 3.1 | Use Case Diagram | 20 |
| 6.1 | DFD Level 0 | 38 |
| 6.2 | DFD Level 1 | 39 |
| 6.3 | Class Diagram | 40-42 |
| 6.4 | Product List | 46 |
| 6.6 | Customer Product Details (in cart) | 41 |
| 6.7 | Update and Delete Product Details | 48 |
| 6.8 | Add New Payment Mode | 50 |
| 6.9 | Add New Shipping Address | 51 |
| 6.10 | List Shipping Addresses | 51 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No.** |
| 1.1 | Deliverable Chart | 5 |
| 2.1 | Hardware Requirements | 7 |
| 2.2 | Software Requirements | 7 |
| 3.1 | User Details | 16 |
| 3.2 | Role Details | 16 |
| 3.3 | UserRole Details | 17 |
| 3.4 | UserBilling Details | 17 |
| 3.5 | UserPayment Details | 17 |
| 3.6 | UserShipping Details | 18 |
| 3.7 | ProductDetails | 18 |
| 3.8 | Use Case Description of Product Details | 23 |
| 3.9 | Use Case Description of Add To Cart | 24 |
| 3.10 | Use Case Description of Proceed To Payment | 24 |
| 3.11 | Use Case Description of Edit Product In Cart | 25 |
| 3.12 | Use Case Description of Payment Completion | 25 |
| 4.1 | Unit Test Cases | 29 |
| 4.2 | Functional Test Cases | 31 |
| 4.3 | Non- Functional Test Cases | 31-22 |
| 4.4 | Test Report of Functional Testing | 32-33 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| MERN | MongoDB Express.js REACTJS Node.js |
| IoT | Internet of Things |
| AEM | Adobe Experience Manager |
| SRS | Software Requirement Specification |
| SDD | Software Design Document |
| DFD | Data Flow Diagram |
| SDLC | System Development Life Cycle |
| IIS | Internet Information Server |
| I&T | Integration and Testing |
| ST | Software Testing |

**CONTENTS**

### CHAPTER 1: INTRODUCTION Page No.

* 1. [Introduction about the Company 1](#_TOC_250024)
     1. [Company Vision 1](#_TOC_250023)
     2. [Company Mission 1](#_TOC_250022)
     3. [Company location 1](#_TOC_250021)
     4. [Company Highlights 1-2](#_TOC_250020)
     5. [Services 2-3](#_TOC_250019)
  2. [Problem Statement 3](#_TOC_250018)
  3. [Proposed Solution 3](#_TOC_250017)
  4. [Objective and Scope of Project 4](#_TOC_250016)
  5. [Project Deliverables 4-5](#_TOC_250015)

CHAPTER 2: PROJECT DESCRIPTION

* 1. [System Interface 6](#_TOC_250014)
  2. [System Specifications 7](#_TOC_250013)
     1. [H/W Requirement 7](#_TOC_250012)
     2. [S/W Requirement 8-9](#_TOC_250011)
  3. [Methodology 9-11](#_TOC_250010)
     1. [Requirements Phase 10-11](#_TOC_250009)
     2. [Design Phase 11-12](#_TOC_250008)
     3. [Development Phase 12](#_TOC_250007)
     4. [Implementation Phase 13](#_TOC_250006)
     5. [Testing Phase 13](#_TOC_250005)
        1. [Unit Testing 13](#_TOC_250004)
        2. [Integration Testing 14](#_TOC_250003)
        3. [System Testing 14](#_TOC_250002)
  4. [Constraints 14](#_TOC_250001)
  5. [Assumptions & Dependencies 14](#_TOC_250000)

|  |  |
| --- | --- |
| 2.5.1 Assumptions | 14 |
| 2.5.2 Dependencies | 15 |
| 2.6 User Characteristics | 15 |
| **CHAPTER 3: FUNCTIONALITY** |  |
| 3.1 Logical Database Design | 16 |
| 3.1.2 Table Structures | 16-19 |
| 3.2 Use case Description | 19-25 |
| **CHAPTER 4: TESTING** |  |
| 4.1 Testing Activities | 26 |
| 4.1.1 Planning and Control | 26 |
| 4.1.2 Analysis and Design | 27 |
| 4.1.3 Implementation and Execution | 27 |
| 4.1.4 Evaluating Exit Criteria and Reporting | 27-28 |
| 4.1.5 Test Closure activities | 28 |
| 4.2 Unit Testing | 28 |
| 4.2.1 Methodology Used | 28 |
| 4.2.2 Test Cases | 29 |
| 4.3 Integration Testing | 29 |
| 4.3.1 Methodology Used | 28 |
| 4.4 System Testing | 30-31 |
| 4.4.1 Functional Testing | 30 |
| 4.4.1.1 Methodology Used | 30 |
| 4.4.1.2 Test Cases | 30 |
| 4.4.2 Non-Functional Testing | 31 |
| 4.4.2.1 Methodology Used | 31 |
| 4.4.2.2 Test Cases | 31 |
| 4.5 Debugging and Test Reports | 31 |
| 4.5.1 Debugging | 31 |
| 4.5.2 Test Reports: Unit Testing | 32 |

**CHAPTER 5: CONCLUSION AND FUTURE SCOPE**

* 1. Conclusion 34
  2. Limitations of the Project 34
  3. Future Scope for Modification 35

References 35-36

### ANNEXURES

* 1. Data Flow Diagram Level 0 38
  2. Data Flow Diagram level 1 39
  3. Data Flow Diagram level 2 40-42
  4. Class Diagram 43
  5. Screenshot-Login 44
  6. Screenshot-Sign-up 45
  7. Screenshot-Home 46
  8. Screenshot-Admin Add Product 47
  9. Screenshot-Admin List Product 48
  10. Screenshot- Admin Edit Product 48
  11. Screenshot-Admin Product Detail 49
  12. Screenshot- User Product Detail 49
  13. Screenshot-User Payment Mode List 50
  14. Screenshot- User New Payment Mode 50
  15. Screenshot- User Shipping Address List 51
  16. Screenshot- User New Shipping Address 51

Blank Sheet

**CHAPTER 1**

**INTRODUCTION**

#### Introduction about the Company

#### Company Vision

#### Company Mission

#### Company Location

#### Company Highlights

#### Services

#### Problem Statement

#### Proposed Solution

#### Objective and Scope of Project

#### Project Deliverables

* 1. **Introduction about the Company**

TO THE NEW is a premium digital technology company that provides end-to- end Product Engineering and Digital Transformation services to Fortune 500 companies and Silicon Valley start-ups across the globe. We cover the entire gamut of product engineering including user experience design, web & mobile application development, cloud, devOps, big data, testing and infrastructure managed services to transform businesses digitally. At TO THE NEW, design led engineering is at the core of our offerings. Some of the cutting-edge technologies, frameworks and platforms we work on include MEAN, Grails, IoT, Block chain, Bootstrap, AEM, Drupal, Hadoop, AWS, React, Ionic, Roku, iOS, and Android.

* + 1. **Company Vision**

Leveraging our years of extensive experience on emerging digital technologies, to build innovative technology solutions for industry incumbents solving their critical business challenges.

### Company Mission

To leverage the power of experience design, cutting-edge engineering, and cloud to build disruptive web and mobile apps enabling digital transformation for businesses.

### Company Location

To The New located at I-Area, B-37, B Block Road, Sector 67, Noida, Uttar Pradesh 201301.

### Company Highlights

To the New Company Highlights are listed as follows:

* + - * To the New (Intelligrape Software Private Limited) is a private company incorporated on 17 June 2008.
      * To The New announced its strategic move to consolidate its specialized service businesses which used to operate - Ignitee Digital, IntelliGrape Software, Tangerine Digital and Techsailor to form TO THE NEW Digital.
      * In December 2011, TO THE NEW acquired Delhi-based IntelliGrape, a premium technology company that develops web and mobile applications using cutting edge technologies.
      * We already have a large portfolio of clients which boasts of Fortune 500 companies as well as Silicon Valley startups including Time Warner Cable, Sony, Procter & Gamble, Castrol, Airbus, Citi Bank, Samsonite and Mat.se.
      * Technology Stack: Backend Engineering, Frontend Engineering, Mobility, DevOps, Testing, Web Content Management, Data Engineering, Blockchain
      * Present on Facebook, Twitter, Google+, LinkedIn.

### Services

To The New is a company which help Internet-based businesses and product companies design and develop cloud-native web and mobile solutions. We drive digital transformation for businesses by helping them provide a rich, seamless experience to their customers across digital channels resulting in higher engagement, efficiency, and profitability. Industry providing services categorized as under: -

* + - * Digital Transformation- Help companies traverse the complex Digital Transformation Journey leveraging cutting-edge technologies
      * Product Engineering- Build disruptive products with a design-led engineering approach
      * Experience Design- Create delightful experiences that impact customer satisfaction, brand perception and conversion rates.
      * Mobility- Intuitive and experiential mobility solutions that provide compelling user experience.
      * DevOps- Automate end to end delivery pipeline and scale up infrastructure across leading cloud platforms.

### Problem Statement

Before online shopping people had to suffer the rush of the market when they went for shopping. They used to think hundred times to buy anything having the sufficient money for shopping. The problem was the rush, the quarrel at the time of buying the things. The price of the products is also very high, there can be some transport expense also, they have to go for shopping at fixed time, searching for required product is also not easy.

### Proposed Solution

With the advancement of technology brought the new way for shopping. The way of shopping was completely changed with the coming of Internet Technology. People have to fill a simple form on the internet to place their order on any popular shop or shopping-mall for the thing they want to buy. Now they can place their order from the home. This project entitled “ECOM. Mobile phone store” is an implementation of the above description. It means, it implements the E-shopping or in other word shopping through Internet. It lets the user to place their order of any kind of product or service online.

ECOM. Mobile phone store is a web-based is a web-based project which is made for remote- shopping or shopping through Internet. As the technology is being advanced the way of life is changing accordance. Now a day’s we can place the order for anything from our home. There is no need to go the shop of the things we want. The order can be placed online through Internet. The payment, the confirmation of purchasing; we can do everything we want. Now we can think that how the days have been changed with time. People had to stand in rows to wait their terms to buy a particular thing from a popular shop. But what is happening now a day’s; we can extremely surprise that those things can be available on the door-step in few hours.

### Objective and Scope of the Project

* This site is gives all the information about the e-shopping to provide better service for the customer.
* It provides the facility to the customers who want to shop on-line due to lack of time.
* It provides facility to the customer to payment by the cash and cheque and Online Payment.
* It’s providing the full details about the product and related information about the product like cost, size etc.
* With the help of it we can save the time and money also.
* It provides better security and good delivery service to the customer. Online shopping more powerful now a day. Everyone want saving money and also time and online business offer promotions for users. It’s also include bright future and more development. So I like this system for more aggressive and polite way of shopping.

### Project Deliverable

Deliverable is a term used in project management to describe a tangible or intangible object produced as a result of the project that is intended to be delivered to a customer (either internal or external). Deliverables include SRS and Design document.

### SRS

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

### SDD

A software design document (SDD) is a written description of a software product, that a software designer writes in order to give a software development team an overall guidance of 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. Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

The Deliverable Chart is listed in Table 1.1.

### Table 1.1: Deliverable Chart

|  |  |  |
| --- | --- | --- |
| **S No** | **Requirement Phase** | **Deliverables** |
| 1. | Requirement Phase | SRS, Use Case Diagram, Use Case  Description |
| 2. | Design Phase | SDD, ERD, DFD |
| 3. | Development Phase | Code |
| 4. | Testing Phase | Test Cases and Reports |

Blank Sheet

**CHAPTER 2**

**PROJECT DESCRIPTION**

#### System Interface

#### System Specifications

#### H/W Requirement

#### S/W Requirement

#### Methodology

#### Requirements Phase

#### Design Phase

#### Development Phase

#### Implementation Phase

#### Testing Phase

#### Unit Testing

#### Integration Testing

#### System Testing

#### Constraints

#### Assumptions & Dependencies

* + 1. Assumptions
    2. Dependencies
  1. User Characteristics

**2.1. System Interface**

System interface is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analysed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analysing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

1. detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.
   1. **System Specifications**

### Hardware Requirements

Following listed (Table 2.1) hardware configuration are used to develop system:

### Table 2.1: Hardware Requirements

|  |  |
| --- | --- |
| **Type of hardware** | **Hardware requirements** |
| System clock frequency | 2.40 GHz |
| Memory size | 2 GB |
| Processor | Intel PENTIUM |
| RAM | 512 MB or Greater |

* + 1. **Software Requirements**

Following listed (Table 2.2) software configuration are used to develop system:

### Table 2.2: Software Requirements

|  |  |
| --- | --- |
| **Type of software** | **Software requirements** |
| Windows | Windows OS XP (64- bit support) |
| Operating System | Windows 10 |
| Technologies | HTML 5, CSS 3, JavaScript, REACT |
| IDE | **Visual Studio** Code |
| Server | express |
| Database | Mongo DB 4.0.5 |
| Platform | Browser |
| Language | JacaScript |

A brief overview of some of the tools used in the system is given below:

##### MongoDB

MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use MongoDB is free to use. Versions released prior to October 16, 2018 are published under the AGPL. All versions released after October 16, 2018, including patch fixes for prior versions, are published under the Server Side Public.

##### REACTjs

**React** is a declarative, efficient, and flexible **JavaScript** library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”. We'll get to the funny XML-like tags soon. We use components to tell **React** what we want to see on the screen.

### Methodology

System Development Life Cycle (SDLC) approach is adopted for analysis, design and development of the software. Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software’s. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. Agile model of SDLC with sprint in our project as it best suits our system is chosen. Agile methodology is shown in figure 2.1.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

Each iteration involves cross functional teams working simultaneously on various areas like –

* Requirements Phase
* Design Phase
* Development Phase
* Implementation Phase
* Testing Phase

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

### Requirements Phase

It consists of the following sub parts:

* + - * **Structural Requirements:** The structural requirement focuses on the requirement of a particular structure in developing any software product. Basically the structural requirement tells about the how the software product will be designed determining the base outline of the product.



##### Figure 2.1: Agile Methodology

* + - * **Behavioural Requirements:** This type of requirement put emphasis on the basic behaviour of the software after its development. This is one of the necessary areas to be focused on as these requirements will decide how the final product will behave. It covers all the relevant necessity for which the end user is intended to get particular software developed.
      * **Performance Requirements:** Performance requirements show the requirements covering the performance of the developed software product. It is responsible for telling that how the software will perform when encountering real time scenarios. The end user describes all the aspects of the software that how the software is intended to work after being developed.

### Design Phase

Agile Design Model is followed with a Test Driven Design Philosophy. Under this model Unit testing is performed along with Design and development of functional units in parallel. The first step is to quickly add a test, basically just enough code to fail. Next new tests are run to ensure their failure. You then update your functional code to make it pass the new tests. The fourth step is to run tests again. If they fail, we update functional code and retest. Once the tests pass the next step is to start over with a new test. Design phase commences after the requirements finalized and frozen. The design phase attempts to uncover various entities involved in the system and their associated behavior and also the interfaces that would be provided by the system. ER Diagram and data flow diagrams for the systems are developed. Database tables and relationship between them is also defined.

* + - * **Structured method:** Structured design is a conceptualization of problem into several well-organized elements of solution. It is basically concerned with the solution design. Benefit of structured design is, it gives better understanding of how the problem is being solved. Structured design also makes it simpler for designer to concentrate on the problem more

accurately. Structured design is mostly based on ‘divide and conquer’ strategy.

* + - * **Function Oriented Design:** In function-oriented design, the system is comprised of many smaller sub-systems known as functions. These functions are capable of performing significant task in the system. The system is considered as top view of all functions. Function oriented design inherits some properties of structured design where divide and conquer methodology is used.

This design mechanism divides the whole system into smaller functions, which provides means of abstraction by concealing the information and their operation. These functional modules can share information among themselves by means of information passing and using information available globally.

Another characteristic of functions is that when a program calls a function, the function changes the state of the program, which sometimes is not acceptable by other modules. Function oriented design works well where the system state does not matter and program/functions work on input rather than on a state.

### Development Phase

The System is developed as 3 tier architecture with bottom up strategy defined as follows:

* + - * **Data Access Layer:** A data context is defined on this layer which manages connections to the tables in database. Also the data flow is managed by making interaction with the MySQL database.
      * **Service Layer:** This layer acts as a bridge between the Data Access Layer and the Presentation layer. The service layer is basically responsible on order to carry out the coordination between the two layers. It maintains a proper synchronization of the data that is being flowed from the Data layer to the Presentation Layer.
      * **Presentation Layer:** This layer defines generic interface with which user interacts. It is as classes which define functions for presentation of data retrieved from service layer to user.

### Implementation Phase

The System is implemented on to tomcat server present internally in the Hybris. The database is maintained under SQL. User Interacts via a tour website which sends HTTP request to the server which replies back after processing the requested page. Thus the system is implemented as a Client server model. To access the data application connects to the server which replies with requested Data Sets.

### Testing Phase

During this phase a well-structured strategy is implied in order to ensure the proper functioning of the application.

Different types of testing are applied to the environment in order to make the whole application error free.

### Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed.

A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

It is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated.

### Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

### System Testing

System testing is the testing of a complete and fully integrated software product. System test falls under the black box testing category of software testing. White box testing is the testing of the internal workings or code of a software application. In contrast, black box or system testing is the opposite. System test involves the external workings of the software from the user's perspective.

### Constraints

The system has following constraint:

* This project can only be used within the intranet of the organization and this makes it unusable outside the organization's private network.
* No smart (mobile) app has been designed to make it accessible from handheld devices.

### Assumptions and Dependencies

* + 1. **Assumptions**
       - The adequate knowledge and skills acquired for the deployment of the setup and the understanding to resolve any issue cropping up in the development process is assumed to be a major prerequisite during the entire process.
       - Besides, to reproduce the issue raised, inputs from the customer end are required in the form of configuration specification or scenario description so as to assist the resolution of the same.
       - For testing purposes, active participation of the customer is also assumed so as to specify the diverse testing scenarios and to report the issue faced and debug the same.

### Dependencies

* + - * A successful network connection with the server.

### User Characteristics

The delivery of the product is given to the customers who deploy the setup at their site and generate the interface according to the kind of specification with the basic knowledge of the terminology used in the whole process of adjudicating the scenarios and the setup judiciously.

Blank Sheet

**CHAPTER 3**

**FUNCTIONALITY**

Use case Description

**Use Case Description**

In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known as an actor) and a system to achieve a goal. The actor can be a human or other external system. In systems engineering use cases are used at a higher level than within software engineering often representing missions or stakeholder goals.

Use case have following:

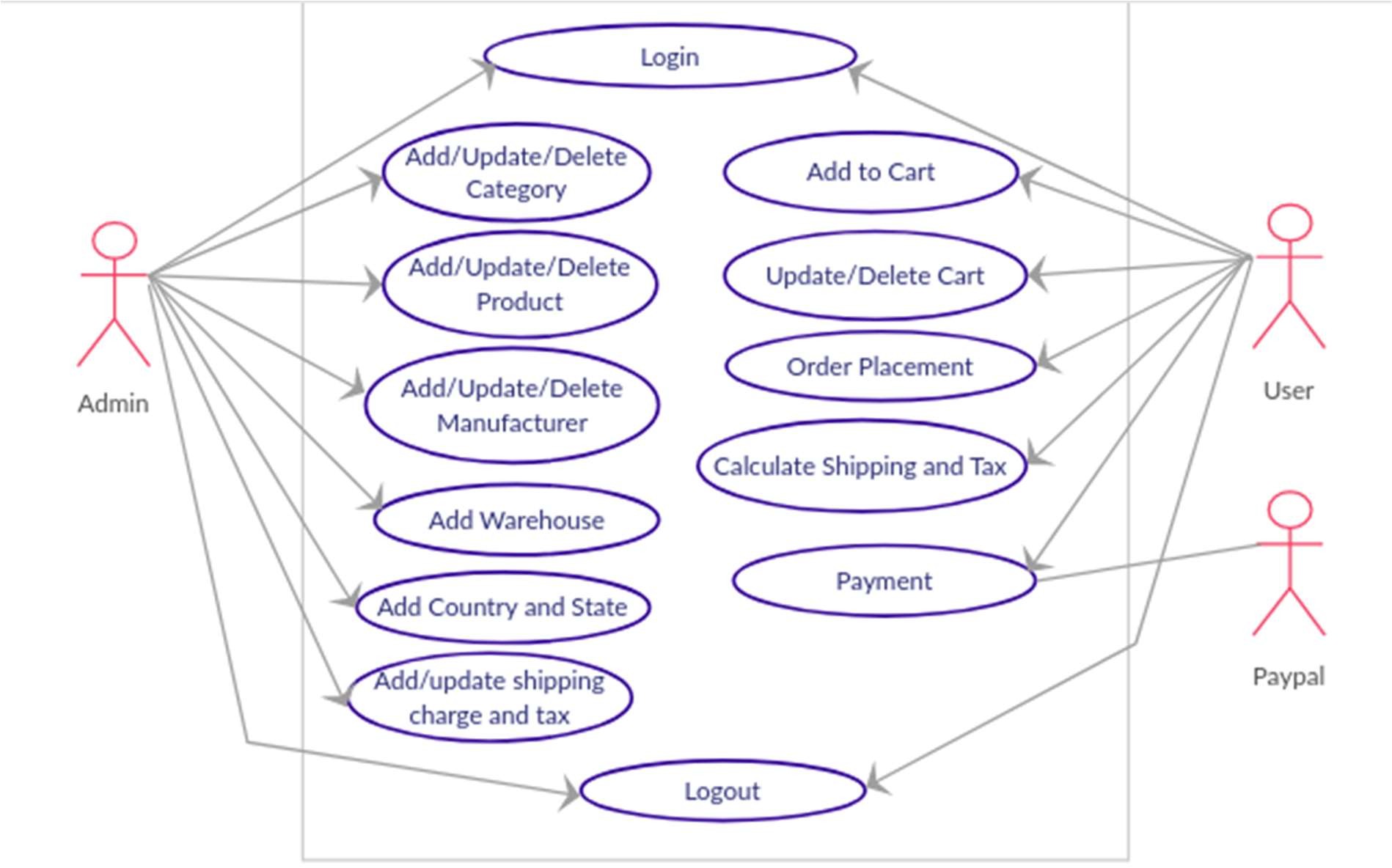
No. of Actors: Multiple No. of Use Cases: 09

##### Use Cases associated with Employee Reward and Recognition System: -

* Login
* Forget Password
* Register
* Search Product
* Product Details
* Add to Cart
* Proceed to Payment
* Edit Product in Cart
* Payment Completion

The use case of system is shown in Figure 3.1.

The use case descriptions are demonstrated in Table 3.6 to 3.15.



##### Figure 3.1: Use Case Diagram

**Table 3.7: Use Case Description of Login**

|  |  |
| --- | --- |
| **Use Case 01** | **Login** |
| Objective | This use case describes how the user will log into the  system. |
| Users | Admin, User |
| Pre-Condition | Should have login credentials. |
| Post Condition | Login page is loaded in the portal. |
| Basic Flow | * The actor enters username and password and clicks |

|  |  |
| --- | --- |
|  | on Login button.   * The system validates username and password.   + The system displays home page. |
| Alternative Flow | Actor is displayed with error message when trying to login using invalid credentials. |

**Table 3.8: Use Case Description of Register**

|  |  |
| --- | --- |
| **Use Case 02** | **Register** |
| Objective | This use case describes how the user will register into  the system. |
| Users | User |
| Pre-Condition | None |
| Post Condition | Login page is loaded in the portal |
| Basic Flow | * The actor enter the username and email id, and click on Register button.   + System validate the email id. * One-time email link with auto generated password send to given email, which is required to click by user, that make user logged in and redirected to   Edit user information page   * + Edit user information page is to complete login process by entering its own password and other   required credentials.   * + The actor enters first name, last name, email & password and clicks on Save button.     - The system displays home page |
| Alternative Flow | Actor is displayed with error message when trying to login using invalid credentials. |

**Table 3.9: Use Case Description of Forget Password**

|  |  |
| --- | --- |
| **Use Case 03** | **Forget Password** |
| Objective | This use case describes how the user will recover  password |
| Users | Users |
| Pre-Condition | User Must Remember their email Id |
| Post Condition | User Able to Set new Password and login |
| Basic Flow | * User Clicks on Forget Password Tab, and enter   their registered email.   * One-time link is sent to that email, which user clicks, and redirect it to the reset password page. * User reset its password and redirected to the login Page. |
| Alternative Flow | Actor is displayed with error message when trying to login using invalid credentials. |

**Table 3.10: Use Case Description of Search Product**

|  |  |
| --- | --- |
| **Use Case 04** | **Search Product** |
| Objective | This use case describes how logged in user can search  products. |
| Users | Users |
| Pre-Condition | Users must be logged in |
| Post Condition | List of product Shown |
| Basic Flow | * User Clicks the Product Shown on Home page, or search the product form search box. * If product Available that particular product of list of related product shown as List. |
| Alternative Flow | users is displayed with error message when trying to search unavailable product. |

**Table 3.11: Use Case Description of Product Details**

|  |  |
| --- | --- |
| **Use Case 05** | **Product Details** |
| Objective | This use case describes how the user see details of  listed Product by clicking the links available with product |
| Users | User |
| Pre-Condition | User must be logged in |
| Post Condition | * Product Details shown * Add To Cart button With product Details   Available. |
| Basic Flow | * User click on product image or name of the listed   product.   * + Product Details and Add to Cart Button Shown. * The actor click on add to cart button to add product   to cart |
| Alternative Flow | none |

**Table 3.12: Use Case Description of Add to Cart**

|  |  |
| --- | --- |
| **Use Case 06** | **Add To Cart** |
| Objective | This use case describes how the can add product to  cart |
| Users | Users |
| Pre-Condition | Users must be logged in |
| Post Condition | * Product Added to Cart if available in stock. * List of Product Added to Cart shown in list |

|  |  |
| --- | --- |
| Basic Flow | * User Click Add to Cart Button   + Product Added to Cart * User Can Set Quantity of Product |
| Alternative Flow | An Error message is displayed if Product is out of stock. |

**Table 3.13: Use Case Description of Proceed to Payment**

|  |  |
| --- | --- |
| **Use Case 07** | **Proceed to Payment** |
| Objective | This use case describes how the can buy Product  Added to Cart and Proceed to Payment. |
| Users | Users |
| Pre-Condition | Users must be logged in |
| Post Condition | Add and select Payment mode page shown |
| Basic Flow | * User click on Buy button.   + Payment Page Shown. * User Can Add Delivery Address. * User Can add and select payment mode |
| Alternative Flow | Users is displayed with error message when invalid Payment process followed such as wrong credit Card number. |

**Table 3.14: Use Case Description of Edit Product in Cart**

|  |  |
| --- | --- |
| **Use Case 08** | **Edit Product in Cart** |
| Objective | This use case describes how the user can edit Product  available in Cart |
| Users | User |
| Pre-Condition | User must be logged in |
| Post Condition | Either product removed from Cart or Quantity Will be |

|  |  |
| --- | --- |
|  | Increased. |
| Basic Flow | User can Click on Remove Product From Cart or Increase quantity of Product. |
| Alternative Flow | No Quantity will Increase if Product is out of Stock |

**Table 3.15: Use Case Description of Complete Payment**

|  |  |
| --- | --- |
| **Use Case 09** | **Complete Payment** |
| Objective | This use case describes how the user can Pay for the  Product. |
| Users | User |
| Pre-Condition | User must be logged in |
| Post Condition | Payment Completion message shown to user, and Date  of delivery shown. |
| Basic Flow | * User Add Card Details. * User Add Delivery Address.   + User Complete payment |
| Alternative Flow | Error message shown for invalid credentials. |

Blank Sheet

**CHAPTER 4**

**TESTING**

#### Testing Activities

* + 1. Planning and Control
    2. Analysis and Design
    3. Implementation and Execution
    4. Evaluating Exit Criteria and Reporting
    5. Test Closure activities
  1. Unit Testing
     1. Methodology Used
     2. Test Cases
  2. Integration Testing
     1. Methodology Used
  3. System Testing
     1. Functional Testing
        1. Methodology Used
        2. Test Cases
     2. Non-Functional Testing
        1. Methodology Used
        2. Test Cases
  4. Debugging and Test Reports
     1. Debugging
     2. Test Reports: Unit Testing
  5. **Testing Activities**

Testing is a process rather than a single activity. Testing must be planned and it requires discipline to act upon it. The quality and effectiveness of software testing are primarily determined by the quality of the test processes used. The activities of testing can be divided into the following basic steps:

* + - Planning and Control
    - Analysis and Design
    - Implementation and Execution
    - Evaluating exit criteria and Reporting
    - Test Closure activities

##### Planning and Control

Test Planning: Test planning involves producing a document that describes an overall approach and test objectives. It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and designing the test environment. Completion or exit criteria must be specified so that we know when testing (at any stage) is complete.

##### Purpose

* + - * To determine the scope and risks and identify the objectives of testing.
      * To determine the required test resources like people, test environments etc.
      * To schedule test analysis and design tasks, test implementation, execution and evaluation.

##### Control

This is the activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan. It involves taking actions necessary to meet the mission and objectives of the project.

##### Analysis and Design

Test analysis and Test Design has the following major tasks:

* + - * To review the test basis. The test basis is the information on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces.
      * To identify test conditions.
      * To design the tests.
      * To design the test environment set-up and identify the required infrastructure and tools.

##### Implementation and Execution

Test execution involves actually running the specified test on a computer system either manually or by using an automated test tool. It is a Fundamental Test Process in which actual work is done.

Test implementation has the following major task: -

* + - * To develop and prioritize test cases by using techniques and create test data for those tests.
      * To create test suites from the test cases for efficient test execution. Test suite is a collection of test cases that are used to test a software program.
      * To re-execute the tests that previously failed in order to confirm a fix.
      * To log the outcome of the test execution. A test log is the status of the test case (pass/fail).
      * To compare actual results with expected result.

##### Evaluating Exit Criteria and Reporting

Evaluating exit criteria is a process defining when to stop testing. It depends on coverage of code, functionality or risk. Basically it also depends on business risk, cost and time and vary from project to project. Exit criteria come into picture, when maximum test cases are executed with certain pass percentage. Bug rate falls below certain level. When we achieve the deadlines.

Evaluating exit criteria has the following major tasks: -

* + - * To assess if more test is needed or if the exit criteria specified should be changed.
      * To write a test summary report for stakeholders.

##### Test Closure Activities

Test closure activities are done when software is ready to be delivered. The testing can be closed for the other reasons also like:

* + - * When a project is cancelled.
      * When some target is achieved.
      * When a maintenance release or update is done.
  1. **Unit Testing**

It is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class.

##### Methodology Used

Unit testing was carried out at the developer environment only. Manual testing was done. The developers review their code to check whether their respective units under tests behave as expected.

##### Test Cases

Unit test cases are listed in Table 4.1.

##### Table 4.1: Unit Test Cases

**Test Id Test Case Name Test Case Description**



Login only with correct

1.

credentials.

With correct username and password, the admin should be easily logged in.

Error on Invalid

2.

username and password

While retrieving username and password, if it doesn’t match to the existing data values then, an error should be displayed.

* + - 1. List the Products

When User Click or Search for product, related list must be shown, if not Product not found message should be shown.

* + - 1. Details of Each Product

When User Clicks on Product User must able to see Details related to that product.

* + - 1. Add to Cart

When User Click on Add To Cart Button, Product must be added to Cart

* + - 1. Cart History

To check whether the portal can get the complete list of product added to cart.

* + - 1. Purchase History

To check whether the portal can get the complete order history.

### Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are

combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

##### 4.3.1 Methodology Used

Integration testing was carried out using bottom up approach where the lowest level components are tested first, and then used to facilitate the testing of higher level components. The process was repeated until the component at the top of the hierarchy was reached.

### System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

##### Functional Testing

Functional testing is primarily is used to verify that a piece of software is providing the same output as required by the end-user or business. Typically, functional testing involves evaluating and comparing each software function with the business requirements. Software is tested by providing it with some related input so that the output can be evaluated to see how it conforms, relates or varies compared to its base requirements.

##### Methodology Used

Under this the whole system was tested under the development team. Basically all functionalities as per requirements are tested here.

##### Test Cases

Functional test cases are listed in Table 4.2.

##### Table 4.2: Functional Test Cases



User should be relocated to error page with access violation message.

Invalid user trying to access the Details.

3.

Flow of screen should be as per requirements.

Flow of screens as per requirement.

2.

Admin should have complete control over the system.

Admin has complete control.

1.

**Test Case Description**

**Test Case Name**

**Test Id**

* + 1. **Non-Functional Testing**

Non-functional testing is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It is designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing.

##### Methodology Used

Under this the whole system was tested under the development team. Basically all functionalities as per requirements are tested here.

##### Test Cases

Non-functional test cases are listed in Table 4.3.

##### Table 4.3: Non-Functional Test Cases



The update at Interface Layer should be as it was mapped to the database via

Check for accuracy while displaying

1.

**Test Case Description**

**Test Case Name**

**Test Id**

|  |  |  |
| --- | --- | --- |
|  | information | Business Layer. |
| 2. | System stability state | System’s ability to recover and save last state. |

* 1. **Debugging and Test Reports**
     1. **Debugging**

In computers, debugging is the process of locating and fixing or by passing bugs (errors) in computer program code or the engineering of a hardware device. To debug a program or hardware device is to start with a problem, is to late the source of the problem, and then fix it. A user of a program that does not know how to fix the problem may learn enough about the problem to be able to avoid it until it is permanently fixed. They imply that they fixed it so that the bugs no longer exist. For complex products, debugging is done as the result of the unit test for the smallest unit of a system, again at component test when parts are brought together, again at system test when the product is used with other existing products, and again during customer beta test, when users try the product out in a real world situation.

### Test Reports: Unit Testing

Test report of functional testing is listed in Table 4.4.

##### Table 4.4: Test Report of Functional Testing



Pass

Logged In

Should Login

Valid Data

With correct username and password, the admin should be

Login only with correct credentials.

1.

**Actual**

**Expected**

**Test Case Status**

**Test Results**

**Test Case Input**

**Test Case Description**

**Test Case Name**

**Test Id**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | easily logged in. |  |  |  |  |
| 2. | Error on Invalid username and password | While retrieving username and password, if it doesn’t match to the existing data values then, an error should be displayed. | Invalid Data | Error Should Appear | Error Appears | Pass |
| 3. | List the Products | When User Click or Search for product, related list must be shown, if not Product not found message should be shown. | Valid Data | Product List Must Appear | List Appear | Pass |
| 4. | Details of Each Product | When User Clicks on Product User must able to see Details related to that product. | Valid Data | Product Details must Appear | Details Appear | Pass |
| 5. | Add to Cart | When User Click on Add To Cart Button, Product must be added to Cart | Valid Data | Product must added to cart | Product added to cart | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6. | Cart History | To check whether the portal can get the complete list of product added to cart. | Valid Data | List of added product must be shown when user add product | List available | Pass |
| 7. | Purchase History | To check whether the portal can get the complete order history | Valid data | Order details must be returned | Order details returned | Pass |

Blank Sheet

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE**

#### Conclusion

* 1. Limitations of the Project
  2. Future Scope for Modification
  3. References
  4. **Conclusion**

The Internet has become a major resource in modern business, thus online shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, online shopping generates a new business opportunities and for the customer, it makes the comparative shopping possible. A god shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a numbers of features that are designed to make the customer more comfortable. This project helps in understanding the creation of an interactive web page and technologies user to implement it. During the course of the project, we have across the wide variety of problems and difficulties. We have learned the appropriate intricate working behind the dynamic website, how tricky data manipulation can be occurred sometimes but we have done everything Successfully, the system has been designed in response to the system analysis. All possible error in the program have been eliminated. Necessary validation techniques have been used and normal, abnormal and extremely data was used to test the system. However, doing this project has been a good boost to our confidence as the future IT member of our global village.

* 1. **Limitation of the Project**

The software we have developed have some limitation. We are trying to fulfil all requirements. But some tasks are not done here. Real time services like voice, chatting and video conferencing are not done here. The system is developed for only cash on delivery as payment method. PayPal not added just added to show it. Upcoming product will display only website but not added messaging service that when any product will come will send message to customer mobile that new product available.

### Future Scope for Modification

If its limitation can be solved, then it would be more effective. In future we will try to develop our project so that we able to give these facilities and credit card validation is not done. The administrator of the website can be given more functionality, looking specific customer generate invoice from admin panel of customer till present for taking decision. Adding SMS gateway to confirm that customer order successfully submitted, and also the current stats of order that means any state change of order customer will get a notification.

### References/Bibliography

* + 1. Functional Interfaces, Lambda Expression, Stream, Method References. Retrieved from [www.tutorialspoint.com/java8](http://www.tutorialspoint.com/java8)
    2. Greg L. TurnQuist (2014). Securing your Application with Spring Boot (Spring Security) of Learning Spring Boot. Birmingham, UK: Packt Publishing Ltd. Retrieved from https://heapy.io/t/me/kotlin\_lang/files/Learning%20Spring%20Boot%202

.0%20-%20Second%20Edition.pdf

* + 1. HTML, CSS, BOOTSTRAP, JQUERY. Retrieved from [www.w3schools.com](http://www.w3schools.com/)
    2. James Rum Baugh, Grady Booch, Ivar Jacobson (2005). Use Case View of The Unified Modelling Language Reference Model. India: Pearson Education Inc. Retrieved from [http://www.temida.si/~bojan/IPIT\_2014/literatura/UML\_Reference\_Man](http://www.temida.si/%7Ebojan/IPIT_2014/literatura/UML_Reference_Man) ual.pdf
    3. Kathy Sierra, Bert Bates (2005). Multithreading, Serialization, Collection, Generics of Head First Java: Brain-Friendly Guide. United States of America: O’Reilly Media, Inc. Retrieved from https://s3-ap-southeast- 1.amazonaws.com/tv-prod/documents/5570-HeadFirstJava2ndEdition.pdf
    4. Peter Membrey, David Hows and Eelco Plugge (2014). MongoDB, Installing MongoDB, The Data Model, Working with Data of MongoDB Basics. New York: Springer Science and Business Media. Retrieved from

https://urotrecra.firebaseapp.com/aa633/mongodb-basics-by-david-hows- peter-membrey-eelco-plugge-b00olet4hq.pdf

* + 1. Spring Boot Tutorial. Retrieved from [www.baeldung.com/spring-boot](http://www.baeldung.com/spring-boot)
    2. Thymeleaf. Retrieved from [www.thymeleaf.org/documentation.html](http://www.thymeleaf.org/documentation.html)
    3. To The New. Retrieved from [www.tothenew.com](http://www.tothenew.com/)

Blank Sheet

# ANNEXURES

* 1. Data Flow Diagram Level 0 A-2 Data Flow Diagram level 1 A-2 Data Flow Diagram level 2 A-3 Class Diagram

A-4 Screenshot-Login A-5 Screenshot-Sign-up A-6 Screenshot-Home

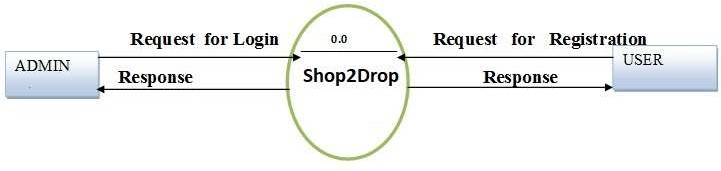
A-7 Screenshot-Admin Add Product A-8 Screenshot-Admin List Product A-9 Screenshot- Admin Edit Product

A-10 Screenshot-Admin Product Detail A-11 Screenshot- User Product Detail

A-12 Screenshot-User Payment Mode List A-13 Screenshot- User New Payment Mode

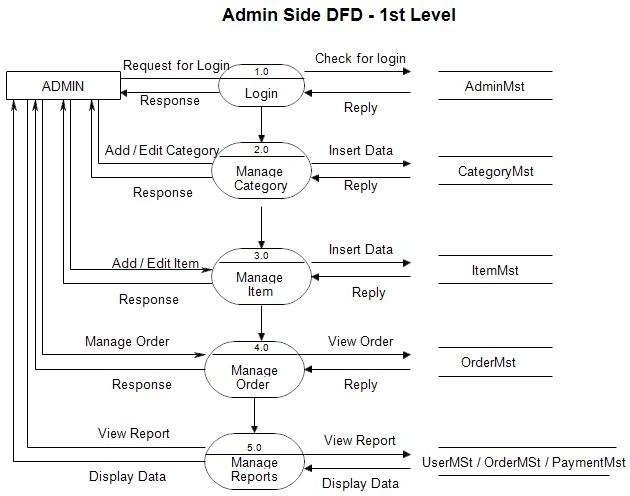
A-14 Screenshot- User Shipping Address List A-15 Screenshot- User New Shipping Address

### A-1: DFD Level 0



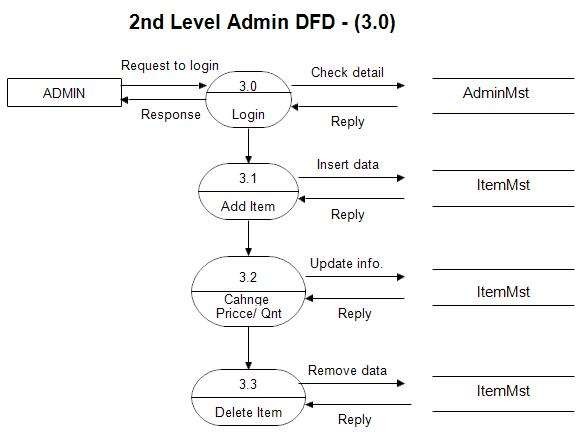
##### Figure 6.1: DFD Level 0

**A-2: DFD Level 1**



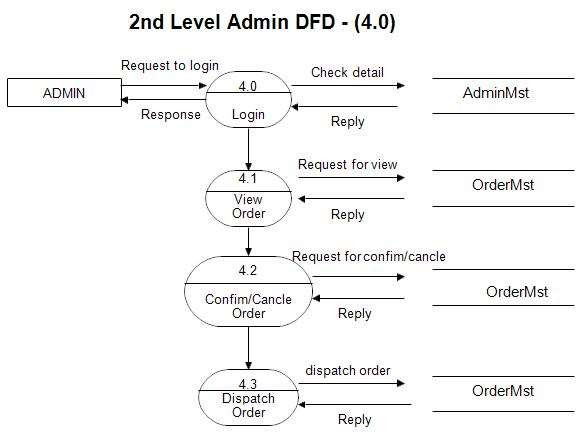
**Figure 6.2: DFD Level 1 Admin Side**

**A-3: DFD Level 2**



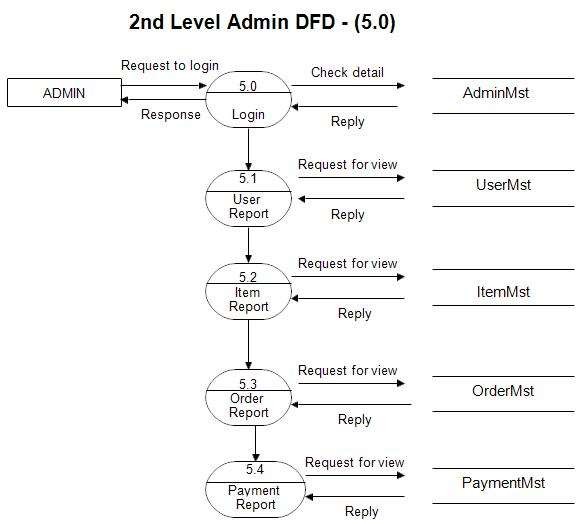
**Figure 6.3: DFD Level 2 Admin Side**

**A-4: DFD Level 2**



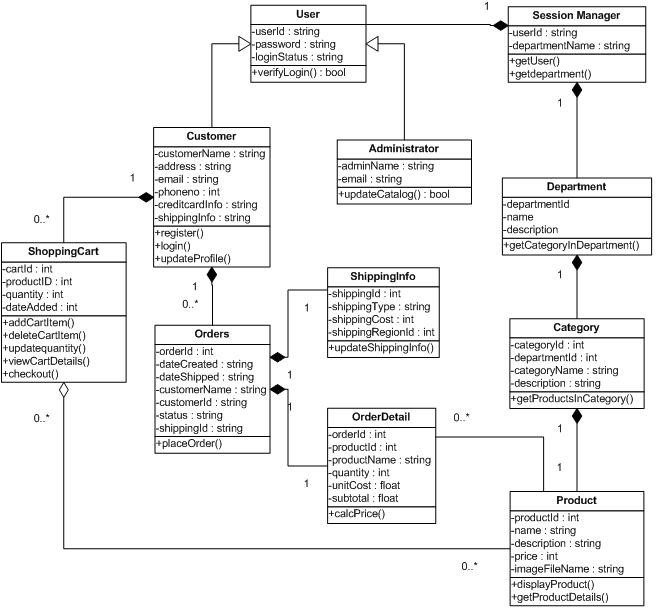
**Figure 6.4: DFD Level 2 Admin Side**

**A-5: DFD Level 2**



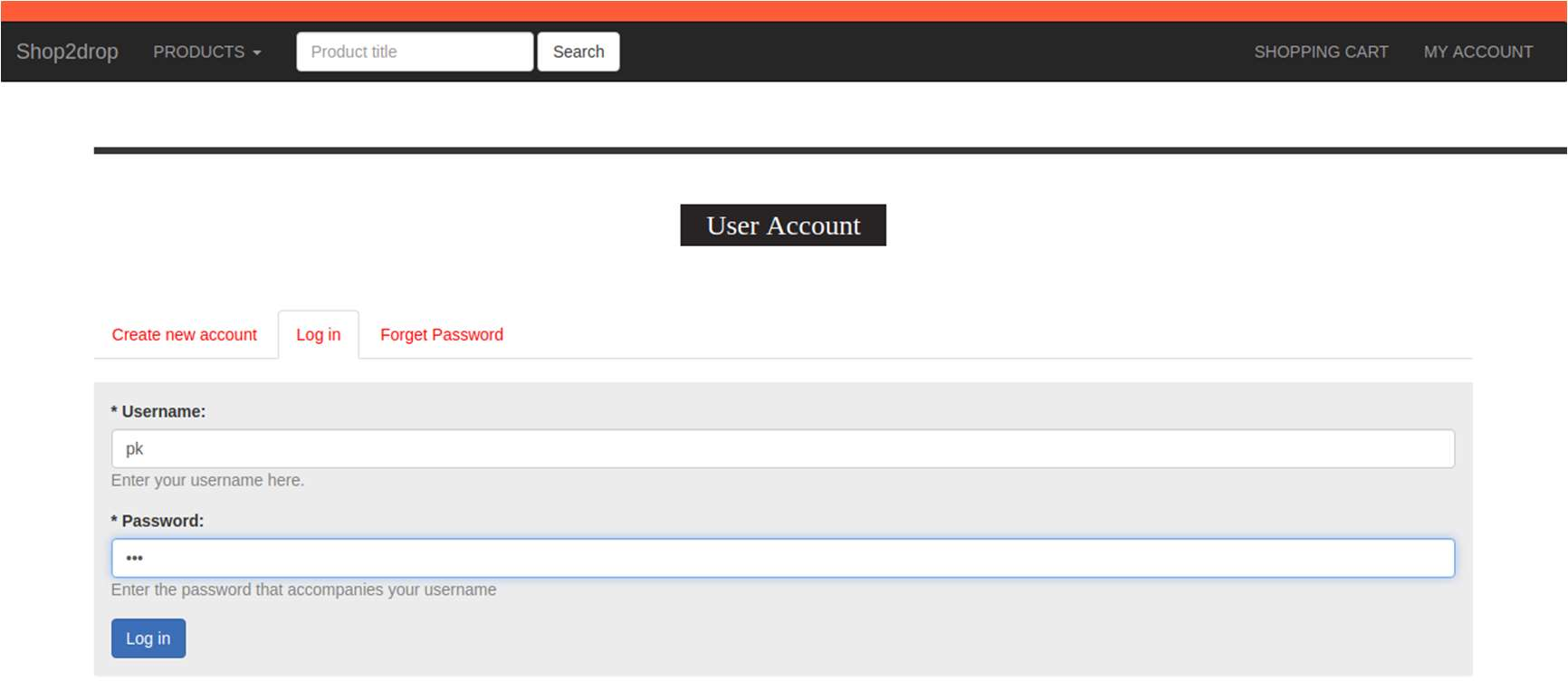
**Figure 6.5: DFD Level 2 Admin Side**

**A-3: Class Diagram**



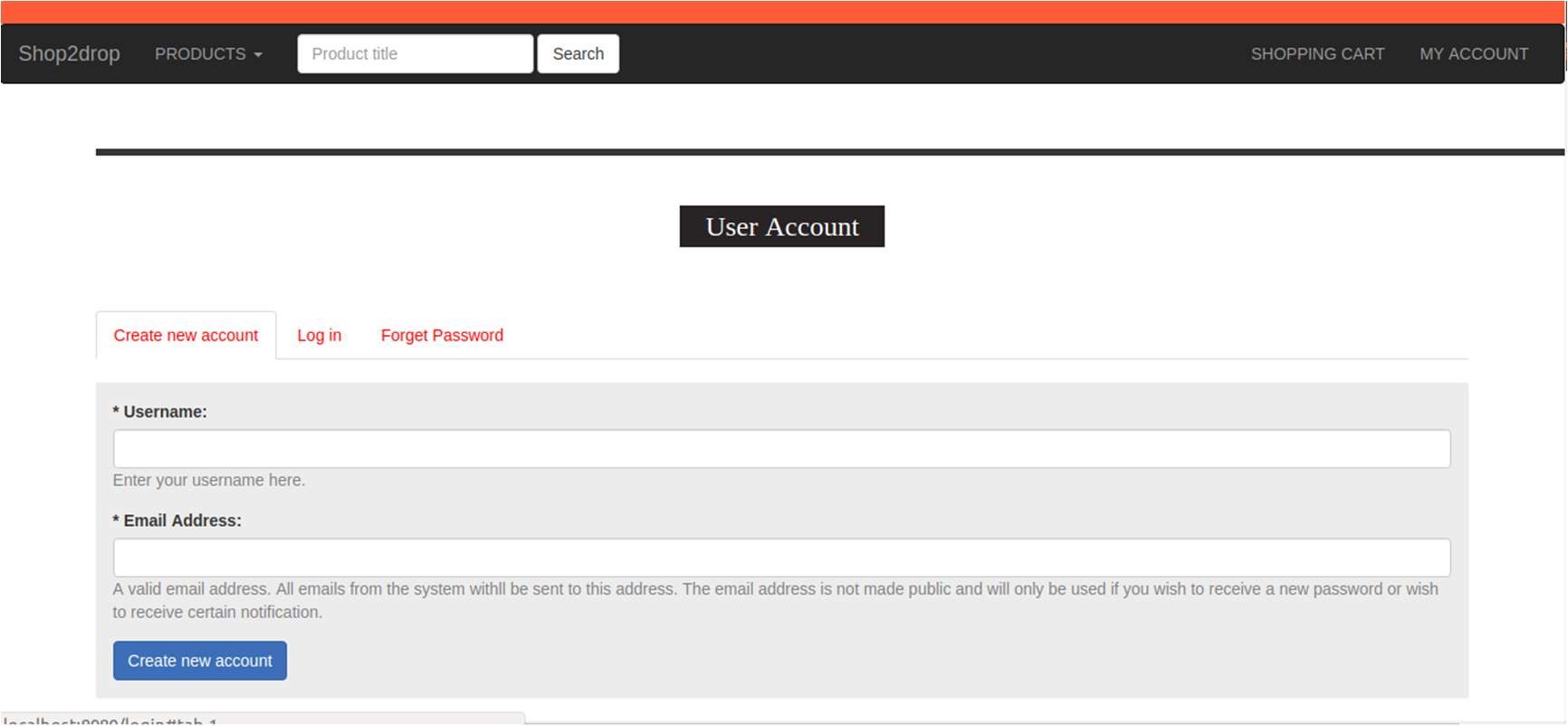
**Figure 6.6: Class Diagram**

**A-4: Screenshot-Login**



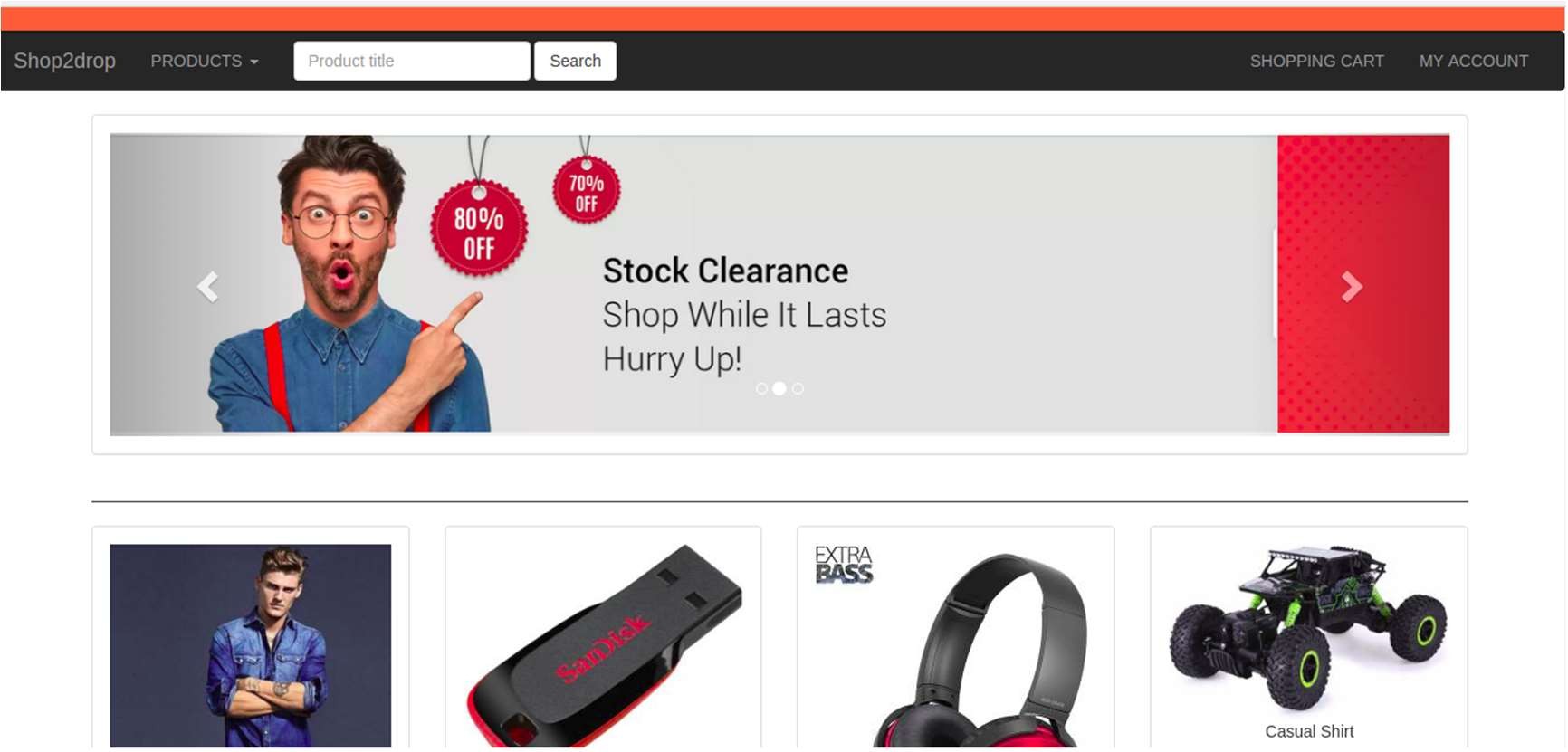
**Figure 6.7: Login**

**A-5: Screenshot-Sign-Up/Create New User**



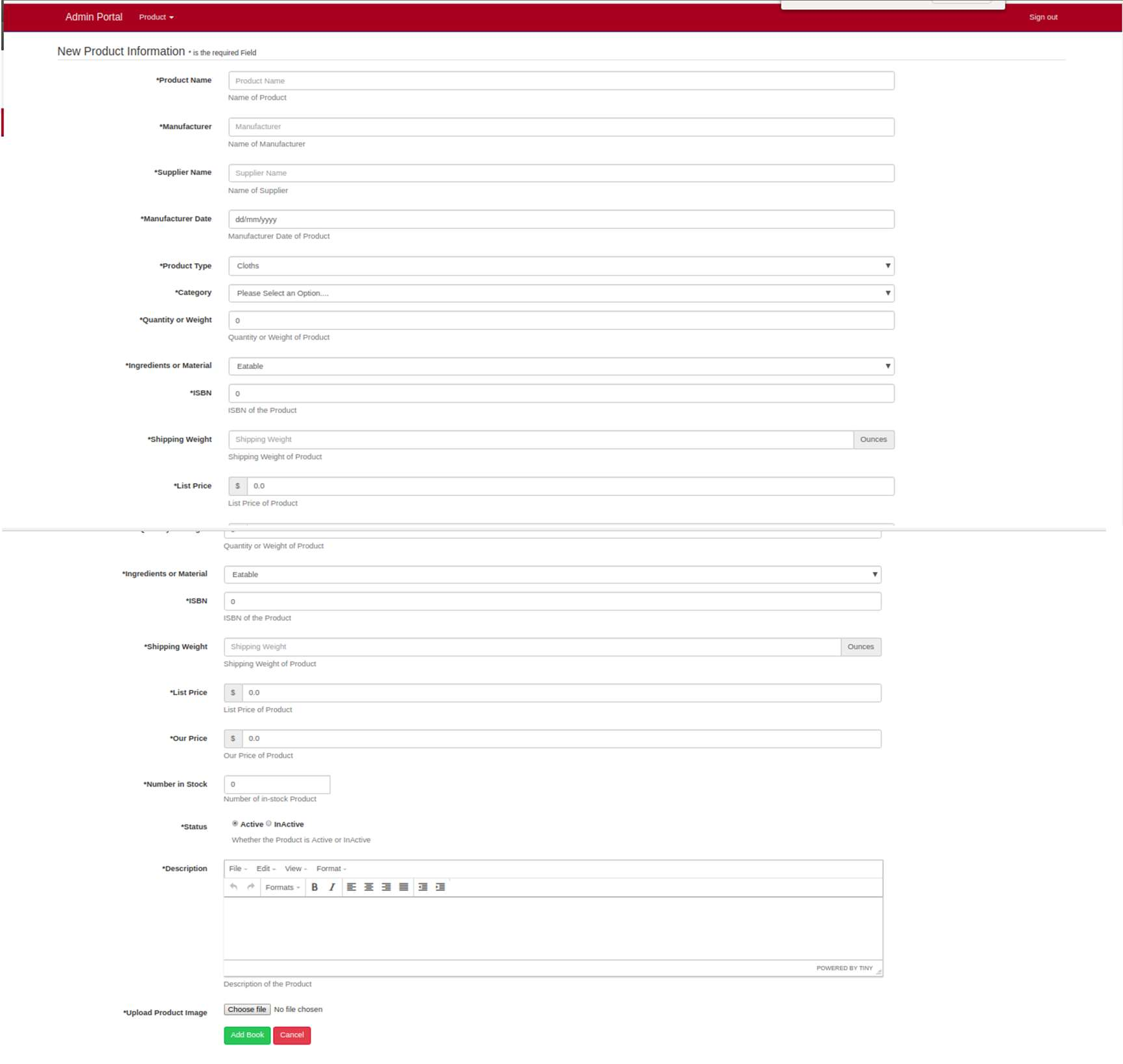
**Figure 6.8: Sign-up/Create New User**

**A-6: Screenshot-Home**



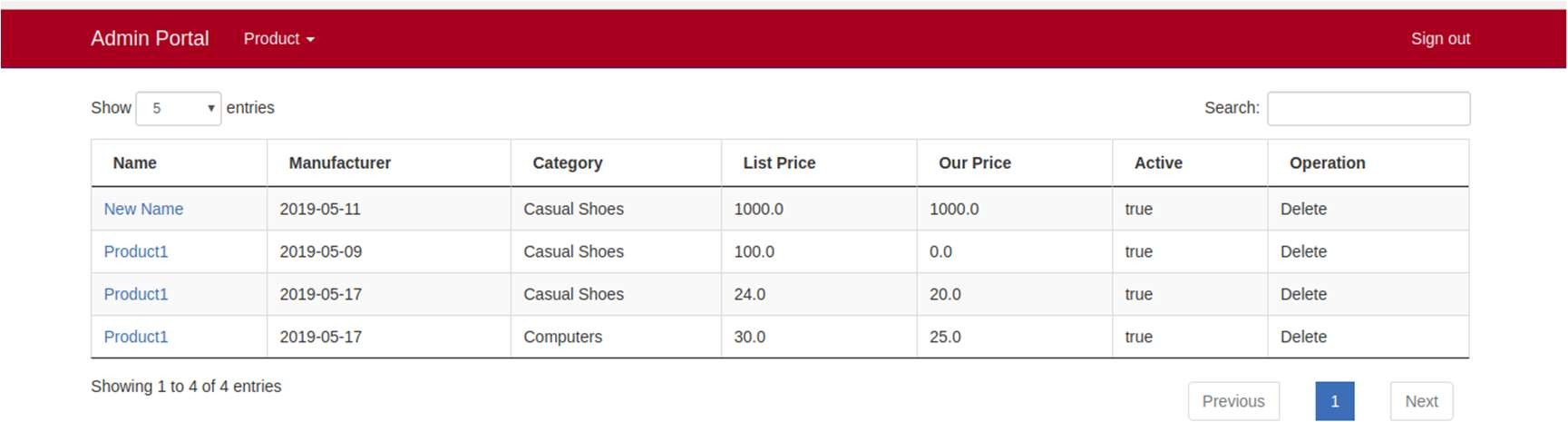
**Figure 6.9: Home**

**A-7: Screenshot-Admin Add Product**



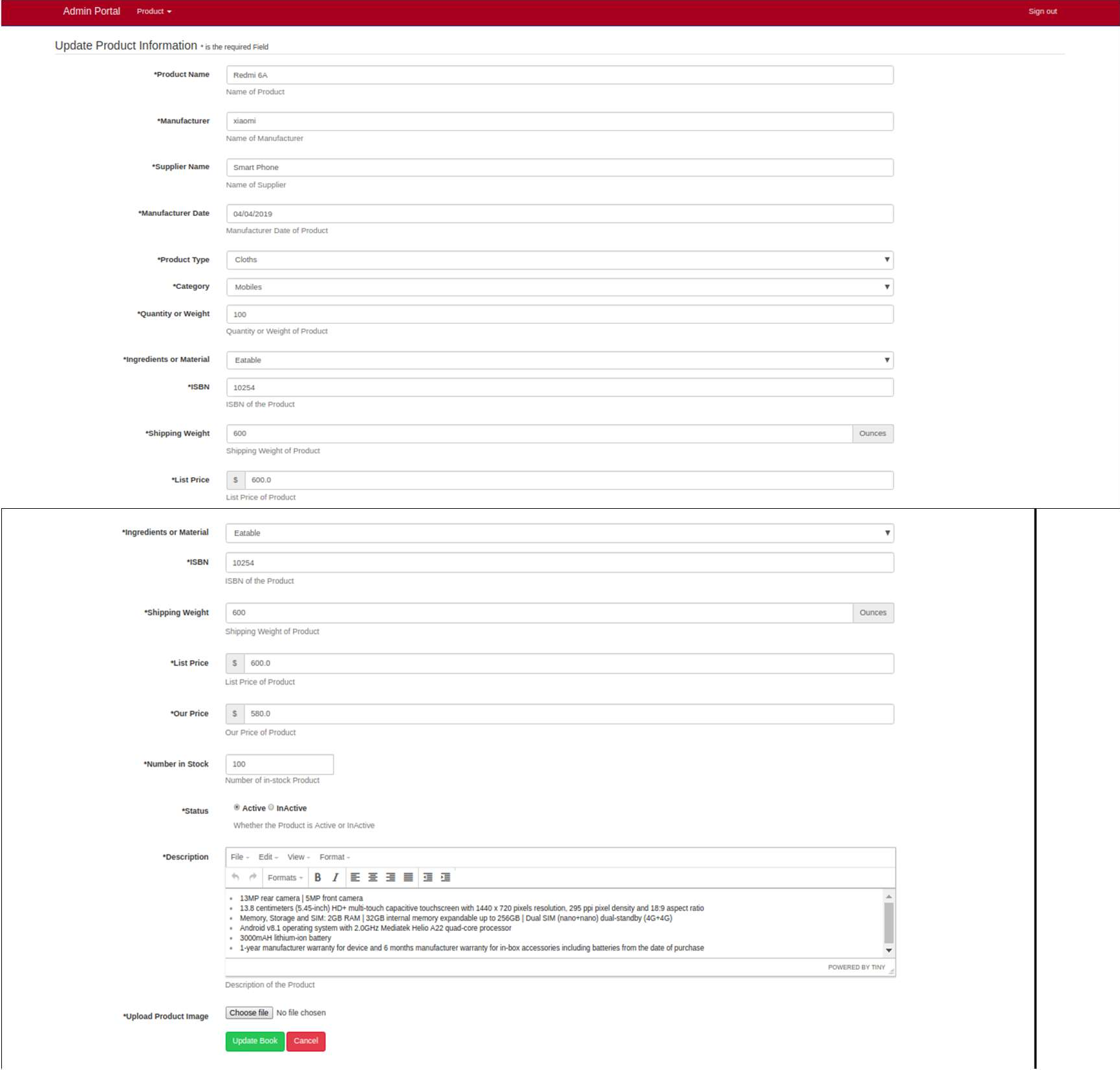
**Figure 6.10: Admin Add Product**

**A-8: Screenshot-Admin Product List**



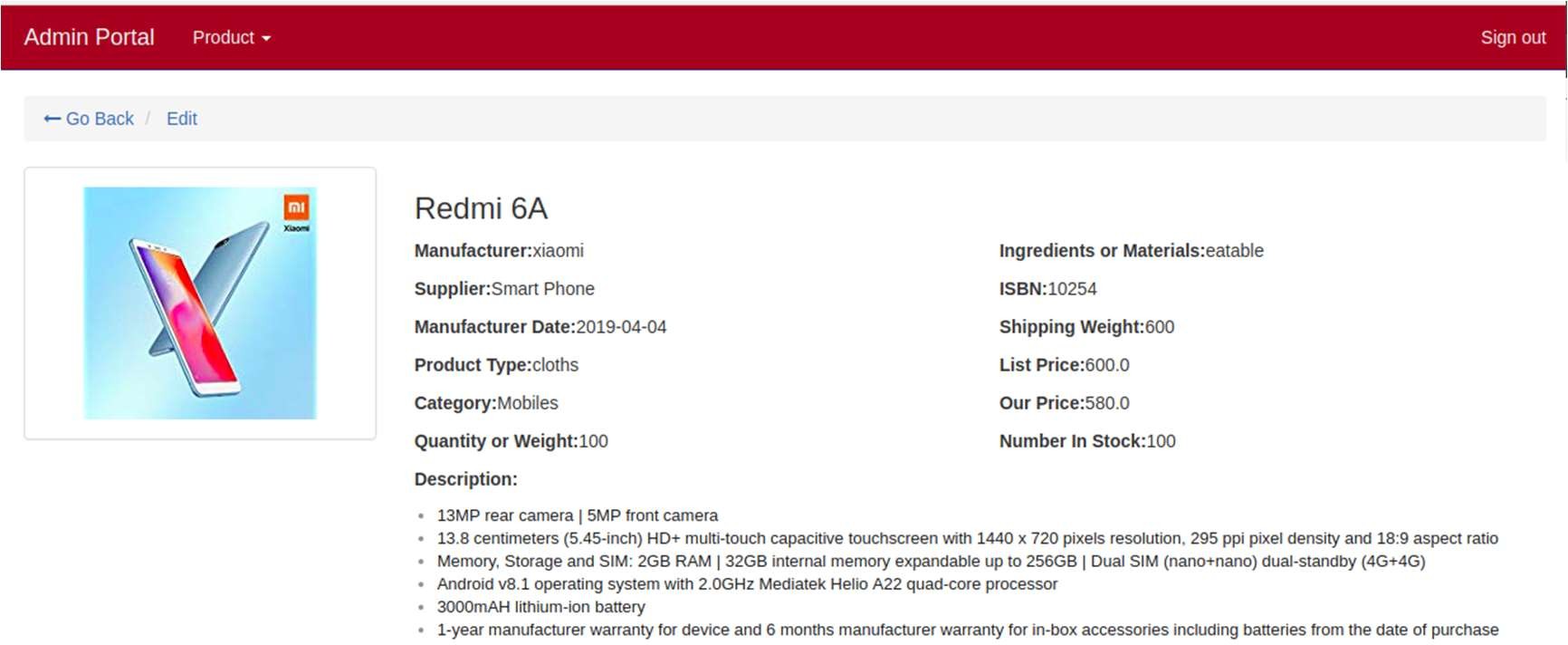
**Figure 6.11: Admin Product List**

**A-9: Screenshot-Admin Edit Product**



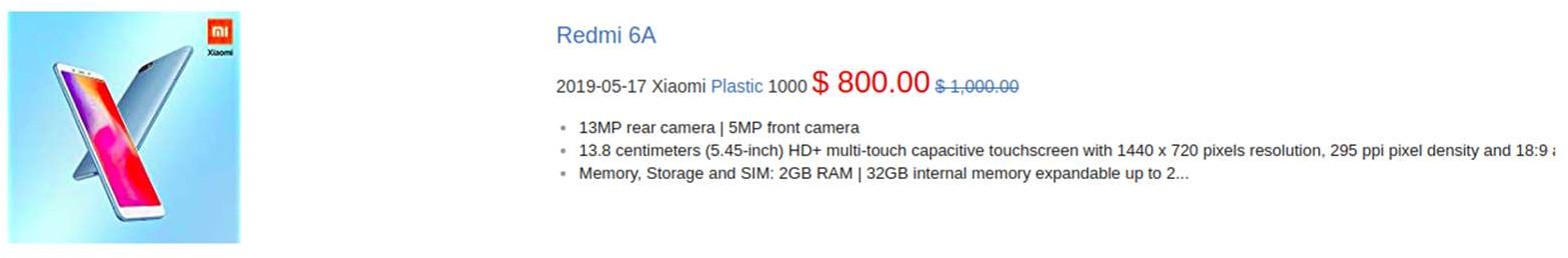
**Figure 6.12: Admin Edit Product**

### A-10: Screenshot-Admin Product Details



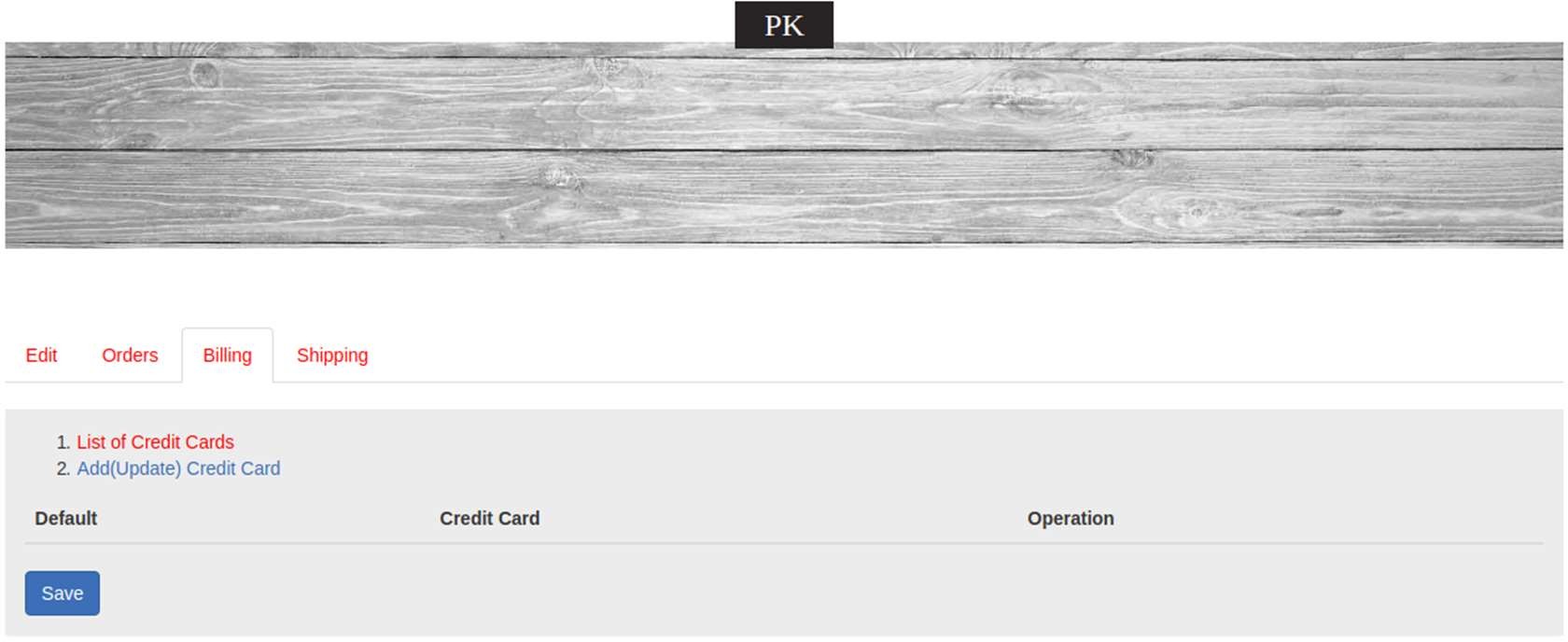
##### Figure 6.13: Admin Product Details

**A-11: Screenshot-User Product Details**



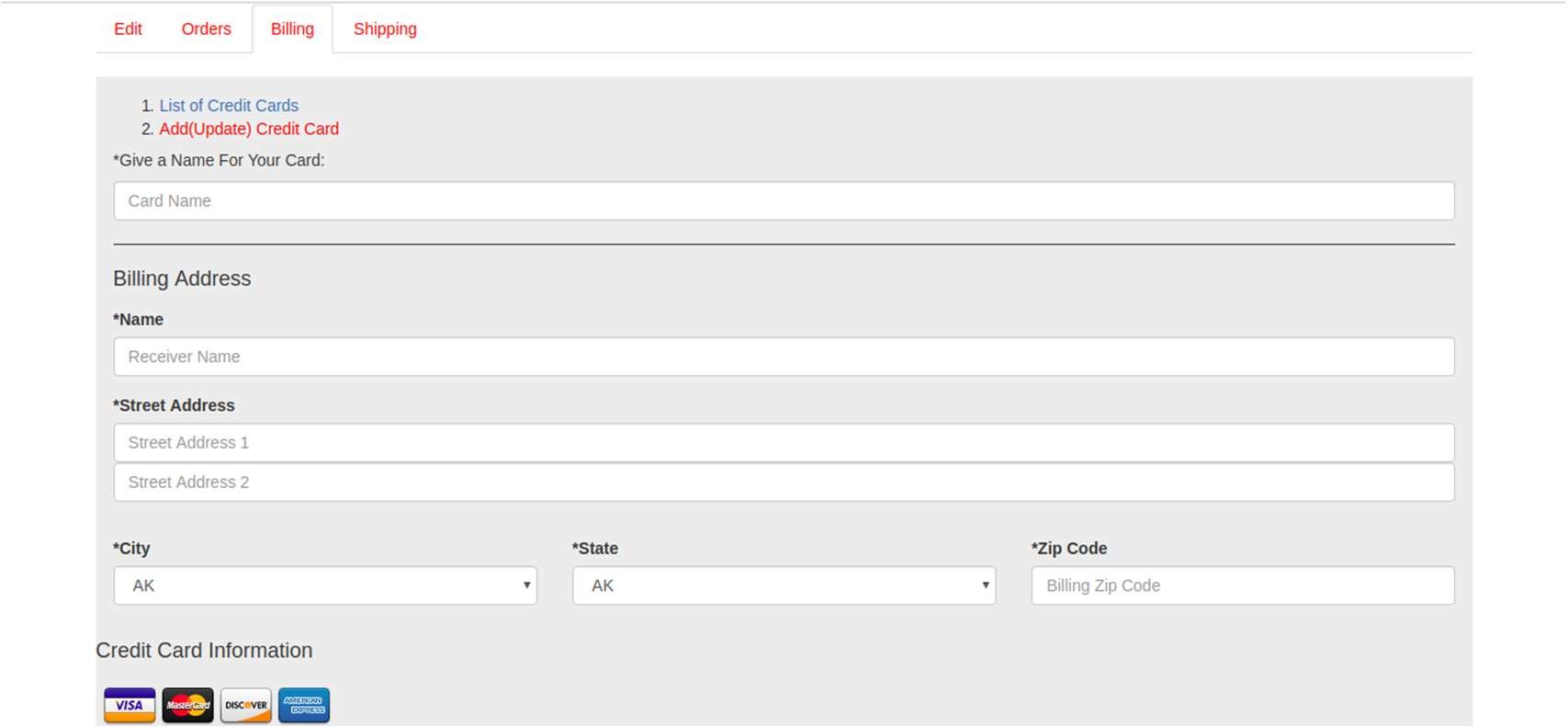
**Figure 6.14: User Product Details**

**A-12: Screenshot-User Payment Mode List /Credit or Debit Card**



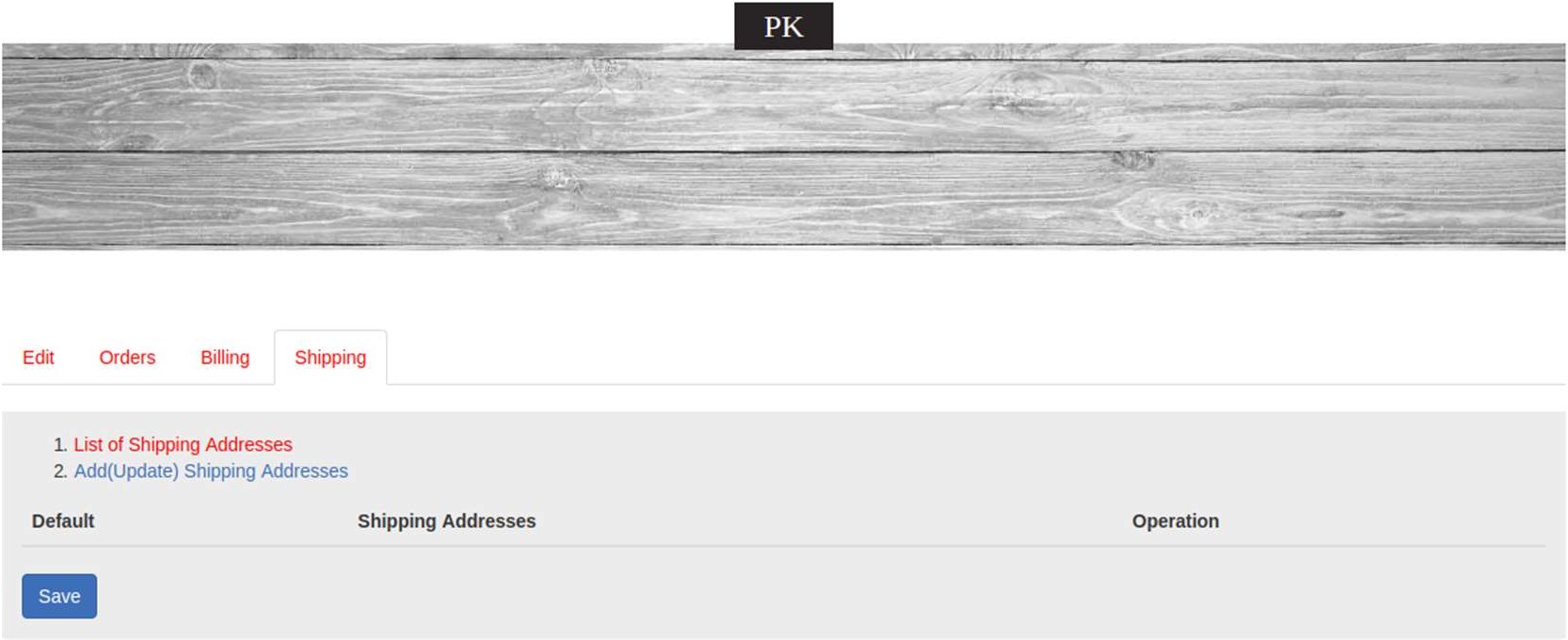
**Figure 6.15: User Payment Mode List /Credit or Debit Card**

### A-13: Screenshot-User Add New Payment Mode List /Credit or Debit Card



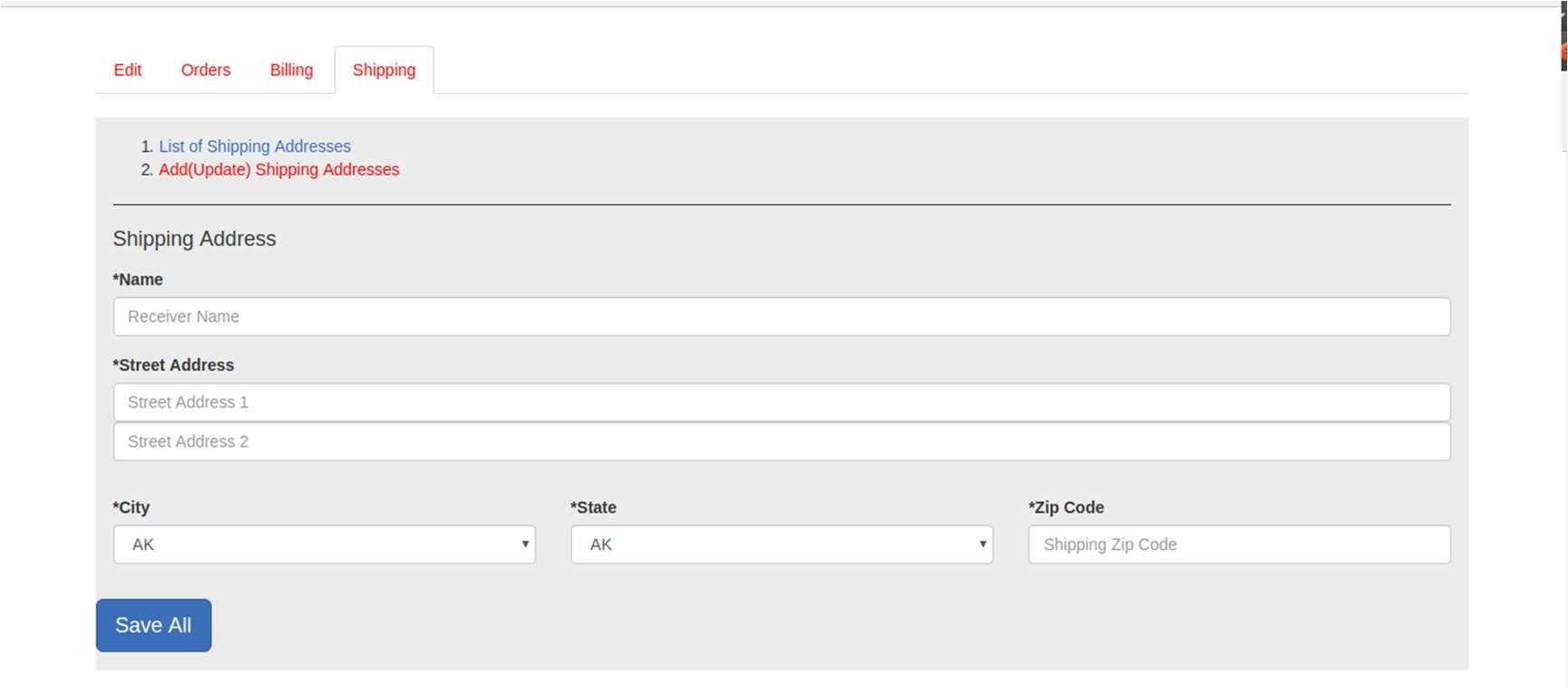
**Figure 6.16: User Add New Payment Mode List /Credit or Debit Card**

**A-14: Screenshot-User Shipping Address List**



**Figure 6.17: User Shipping Address List**

### A-15: Screenshot- User Add New Shipping Address



**Figure 6.18: User Add New Shipping Address**

Blank Sheet