

Contents

1	INTRODUCTION / OBJECTIVES	1
1.1	Background	1
1.2	Purpose & Motivation	2
1.3	Objectives	2
1.4	Project Category	3
2	SYSTEM ANALYSIS	4
2.1	Identification of Need	4
2.2	Preliminary Investigation	4
2.3	Feasibility Study	4
2.4	Project Planning	7
2.5	Project Scheduling	8
2.6	Software Requirement Specifications	10
2.6.1	Problem Definition	10
2.6.2	User Module	10
2.6.3	User Module - Registration	10
2.6.4	User Module - Sign-On	11
2.6.5	User Module - Search Products	12
2.6.6	Shopping Cart	13
2.6.7	Place Order	15
2.6.8	User Module - Make Payment	17
2.6.9	User Module - Invoice	19
2.6.10	User Module - Notifications	20
2.6.11	Admin Module	21
2.6.12	Admin Module - View Order data	21
2.6.13	Admin Module - Manage Products	22
2.6.14	Admin Module - Add Product	23
2.6.15	Reports	25
2.7	Software Engineering Paradigm applied	25
2.7.1	The Personal Software Process (PSP)	25
2.7.2	Notes on the Go Programming Language Design ^[4]	27
2.8	Data Models & Diagrams	31
3	SYSTEM DESIGN	37
3.1	Modularization Details	37
3.2	Database Design	39
3.3	User Interface Design	45
4	CODING	79
5	SYSTEM SECURITY MEASURES	92
6	COST ESTIMATION OF THE PROJECT	97
7	REPORTS	99
8	PROJECT CHART, GANTT CHART	100

9 FUTURE SCOPE AND FURTHER ENHANCEMENT	103
10 BIBLIOGRAPHY	104
11 APPENDICES	105
12 GLOSSARY	108

1 INTRODUCTION / OBJECTIVES

Electronic Commerce (also e-commerce) commonly written as E-commerce or ecommerce is the trading or facilitation of trading using Computer Network such as Internet or Social Networks. E-commerce draws on technologies like Mobile Commerce, Electronics Funds Transfer, Supply Chain Management, Internet Marketing, Online Transaction Processing, Electronic Data Interchange (EDI), Inventory Management Systems and automated data collection systems. Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction's life cycle though it may also use other technologies like email.

Online Shopping is a form of electronic commerce which allows consumers to directly buy goods and services from a seller over the Internet using a Web Browser. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping Search Engine which displays the same product's availability and pricing at different vendors. Customers can shop online using a range of different computers and devices including desktop computers, laptops, tablet computers and smart-phones.

1.1 Background

An online shop evokes the physical analogy of buying products and services at regular "bricks-and-mortar" retailer or shopping center; the process is called Business-to-consumer (B2C) online shopping. A typical online store enables the customer to browse the firm's range of products and services, view photos or images of the product along with the information about the product specification, features and prices.

Online stores typically enable shoppers to use "search" features to find specific models, Brands or Items. Online customers must have valid access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Interac-enabled debit card, or a service such as Paypal. For physical products (e.g., paperback books or clothes), the e-tailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-tailer typically sends the file to the customer over the Internet. The largest of these online retailing corporations are Alibaba, Amazon.com and eBay.

English entrepreneur Michael Aldrich was a pioneer of online shopping in 1979^[1]. His system connected a modified domestic TV to a real-time transaction processing computer via a domestic telephone line. The first World Wide Web server and browser, created by Tim Berners-Lee in 1990, opened for commercial use in 1991. Thereafter, subsequent technological innovations emerged in 1994: online banking, the opening of an online pizza shop by Pizza Hut, Netscape SSL v2 encryption standard for secure data transfer and Intershop's first Online shopping system. The first retail transaction over the Web was ei-

ther by NetMarket or Internet Shopping Network in 1994. Immediately after, Amazon.com launched its online shopping site in 1995 and eBay was also introduced in 1995. Alibaba's sites Taobao and Tmall were launched in 2003 and 2003 respectively.

Mobile Phone online buying platforms can be broadly classified into 2 types

- Owned by Retailer to sell own products
- Marketplace, which allows various merchants to showcase and sell their products. The retailer only manages the marketplace.

This project describes a minimal implementation of a platform which the retailer can use to sell own products i.e. the retailer is accountable and responsible for the product inventory.

1.2 Purpose & Motivation

The main purpose of this project is to create an online store to buy mobile phones. The site will allow users to search mobile phones from the products listing page. Users can add selected products to a shopping cart and checkout by making payment. Users will receive an order copy of their invoice.

The retailer website will be managed by an Admin. Admin will have additional functionality such as managing product catalog and generating reports.

Motivation to work on this project includes

- Working on a project in the Retail domain
- To gain knowledge of the working of a good user friendly website that facilitates online transactions using a database
- Interest in technologies such as Golang, Javascript, HTML, CSS and SQL for web development
- Explore data analytics that can be implemented using Golang

1.3 Objectives

The Key objectives of the Project include

- Implementation of an Admin module for managing a website facilitating buying of mobile phones using online transactions.
- Develop and host a website which allows users to search and explore mobile phones
- Implement the shopping cart feature for the site that allows users to add selected products and tag it to a single order
- Implement the online payment module (Credit Cards Only)

- Explore technologies such as Golang, Javascript, HTML, CSS and SQL for web development
- Explore data analytics that can be implemented using Golang

1.4 Project Category

This project can be categorized as a web development project that uses concepts of Internet technologies and web design, web security and RDBMS. Though Golang is not an OOP language, OOP has been achieved by making use of interfaces allowed in Golang. Network security has also been implemented for the website using Transport Layer Security (TLS)

2 SYSTEM ANALYSIS

2.1 Identification of Need

Small scale retailers or start-up retailers in the mobile phone category would like to get their website up and running with a minimum investment in hardware and technology. The technology should be chosen in a way that allows to scale up later if required.

Large retailers may want to look at alternative technologies that make possible to lower development time and cost translating to better Project Management and increased efficiency.

Mobile Phones is an indicative category. The Online selling platform can be deployed for the purpose of selling almost any kind of product categories and services online. Mobile Phone Category has been chosen for this project due to the limited and universally well-understood feature set of the products of this category that allows to describe the project implementation without delving too much into the business aspects of the products.

2.2 Preliminary Investigation

India was one of the fastest growing retail e-commerce markets in 2015, growing at the rate of 129.5 per cent Y-o-Y. Declining broadband subscription prices and the launch of 4G services has become the driving forces of e-commerce in the country. India will see more people come online than any other country in the next 15 years. With the penetration of digital devices and social media in the interiors of the country, online sellers have been presented with an unprecedented opportunity of growth, becoming extremely attractive to investors. E-commerce is expected to acquire 4.8% market share in total retail sales by 2019.

Among e-tail categories, mobile phone and mobile accessories continue to be the top contributor to the overall pie^[2]. Retailers would be willing to invest in technology and even maintain it in-house provided it is low-cost, easy to maintain and of course effective in serving their business needs. This allows retailers to manage their own product catalog, respond to market dynamics by carrying out promotional campaigns, manage the look and feel of their website and quickly go-live and roll-out the changes. Though all of this could also be achieved through an intermediate service provider, the retailer in that case would not be able to manage costs and time as they would like to.

2.3 Feasibility Study

The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope. This project does not aim to build a full scale website that is ready for industrial deployment. The project aims to explore web development

using Golang by creation of an online mobile store. The full scale implementation of the project is constrained by time, resource and cost and is in fact not really necessary for an academic project of this kind.

The scope of the project shall be limited to

- User Registration and Sign-on
- Manage Products Catalog
- Order Management
 - Shopping Cart
 - Place Order
 - Cancel Order
- Payment Gateway Integration
- Notifications
- Analytics & Reports

To deploy a website with the basic benchmarks as stated above the following tools, platforms, hardware and software were used.

The development environment was set up on a i686 computer loaded with a 32-bit Linux operating system. The host environment shall be the same i686 computer with the 32-bit Linux operating system i.e., the development server and the host server are one and the same machine.

Later, after the development process is complete, the option of deploying the web app on a remote host or Google App Engine may be explored for demo purpose.

Table 2: Software & Hardware Requirements

Sr. No.	Tools & Technologies	Description
1.0	Go ver 1.7 linux/386	Tool for managing Go source code.Go (also commonly referred to as golang) is an open source systems programming language developed at Google

2.0	net/http package	Package http of Golang that provides http client and server implementations
3.0	SQLite3 Database	SQLite is a self-contained high-reliability, full-featured, public-domain, SQL database engine.
4.0	SQLite Manager	SQLite Manager is a Database Management System for SQLite database and is available as a firefox addon that can be used in the browser.
5.0	Emacs ver. 24.3	Development Environment
6.0	go-mode	Emacs package for GoLang
7.0	Geany ver 1.22	A lightweight IDE with support for HTML, CSS, Javascript and JQuery
8.0	Github	Repository Management Cloud
9.0	Git	Version Control System
10.0	HTML5	HTML 5 is the markup language used for structuring and designing content on the world wide web

11.0	CSS	A declarative stylesheet language for structured documents
12.0	Javascript	Javascript is a high-level, dynamic, untyped and interpreted programming language for front-end web functionality
13.0	JQuery ver 1.11.3	JQuery is a cross-platform javascript library designed to simplify the client side scripting of HTML
14.0	Crunchbang Linux Waldorf 11.0	Operating System
15.0	stripe-go	Golang package for Stripe API. Stripe is a Payment Gateway Service provider

2.4 Project Planning

Software Development

- The online Mobile Store shall be a minimalist functional website with functionalities as described in the scope
- Server Code shall be developed entirely using Golang
- Web server provided by Golang's "net/http" package shall be used to serve the pages
- Front-end development shall be done using HTML 5, CSS3, Javascript and JQuery.
- Shopping cart shall be developed entirely using javascript. The shopping cart will not be persistent and will be valid only for the User Session.
- Stripe Checkout shall be used for Payment Gateway. API integration shall be done with the Go code.

- Personal Software Process (PSP) Software Development Methodology shall be followed

Deployment & Hosting

- The website shall be hosted on the local server on which the development is done. Internet connectivity will be required for the web application to work
- Options to deploy the web app on Google Cloud App Engine shall be explored for demo purpose

Testing

- Test Scenarios, Test Cases & Testing

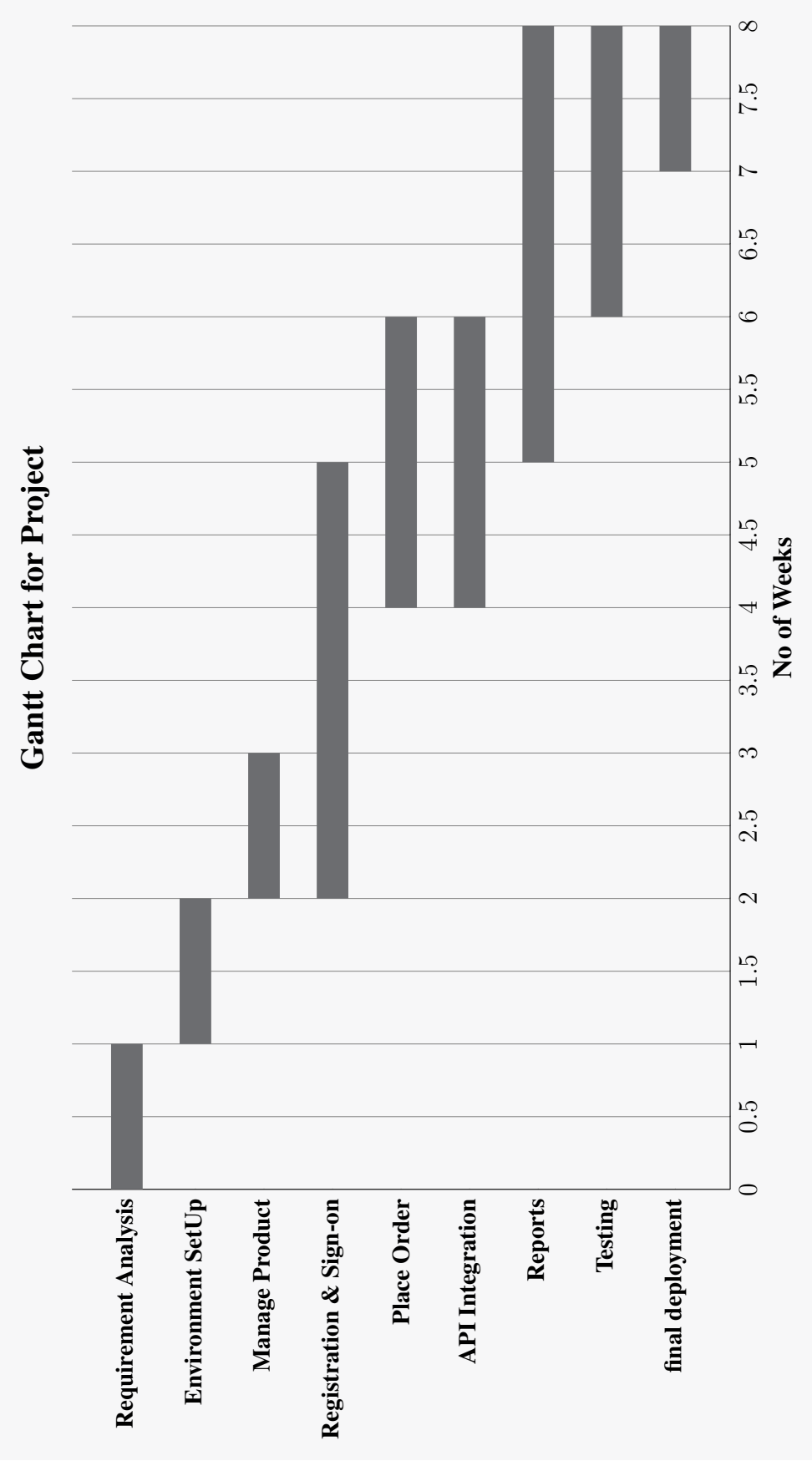
Resources & Cost

- This is an academic project and a single resource i.e., I myself shall be working on all aspects of the Project.
- There is no financial cost attributed to the project other than the remuneration bill for the Project Guide to be footed by IGNOU.

2.5 Project Scheduling

Major deliverables of this project are:

- Setting up the development environment
- Code implementation of key functions i.e Product Catalogue, Sign-On, Order management & Reports
- Payment API Integration
- Test Scenarios, Test Cases & Testing
- Deployment & Hosting



2.6 Software Requirement Specifications

2.6.1 Problem Definition

Mobile Phone Retailer requires to present inventory to customers online and facilitate users to search, select and place orders for mobile phones as well as make payments online. Retailer should be able to manage the platform that allows the buying of mobile phones online.

2.6.2 User Module

Users can register, login, search for products, add selected products to shopping cart and place orders after check-out

2.6.3 User Module - Registration

Users can register on the customer website by clicking on a link that directs them to the Sign-up or registration form. Users have to provide the username with which they register, first and last name, an email address and confirm the password they want to create for the username

Table 2: Requirements - User Module - Registration

Req. ID	Description	UI Element & Datatype	Validation
FR1	User should be able to navigate to the sign-up / registration form from the landing page	Hyperlink	User should have a secure connection to the website
FR2	User should be able to create a unique username for registration purpose	Input text box String	should be unique should be alphanumeric

FR3	User should be able to create a provide First Name, Last name and email address for registration purpose. Email address should be unique	Input text boxes String	Should be string (alpha) Email Address should be a valid email ID and unique
FR4	User should be able to create a create a password for the username	Password box string	should be alphanumeric User is required to reconfirm
FR5	User should receive error alerts for invalid input	Alert boxes String	Empty space is not allowed Fields that require unique value shall only accept such values Fields not meeting the criteria described in the regex pattern for the field shall be invalidated

2.6.4 User Module - Sign-On

User can sign-on using the username and password

Table 2: Requirements - User Module - Registration

Req. ID	Description	UI Element & Datatype	Validation
FR6	User connection should be secure		Transport Layer Security (TLS) should be implemented Secure connection via HTTPS
FR7	User should be able to log into the website using the username and password created at the time of registration	Input text boxes String	User should authenticate by providing valid password New session shall be generated for every successful login
FR8	User should receive error alerts for invalid input	Alert boxes string	Username cannot be empty and should be valid Password should correspond to the correct password of the username maintained at the server.

2.6.5 User Module - Search Products

User can search for specific products in the product catalog by description By default all products shall be listed in the main view.

Table 2: Requirements - User Module - Product Listing & Search

Req. ID	Description	UI Element & Datatype	Validation
FR9	All products should be listed in the main view . Listings should include the image, name and price of the product	Info Panel Image File, String	data object should be present in the product catalog
FR10	User should be able to search the product by entering string that shall be matched with the product description. User should be able to view only such products that fulfill the search criteria	Search Box string	Search string cannot be empty
FR11	User should be able to select a product by adding it to the shopping cart	Button Each product info panel shall have its own button	A single unit of the product shall be added to the shopping cart each time the button is clicked

2.6.6 Shopping Cart

Selected products can be added to shopping cart.Shopping Cart is not Persistent i.e it is valid only for the session

Table 2: Requirements - User Module - Shopping Cart

Req. ID	Description	UI Element & Datatype	Validation
FR12	User should be able to view the list of products added to the shopping cart. The list should include the image, name and price of the product.	Dropdown list (Tabular) image File, String	Details of the Product should be similar to the selected product in all aspects
FR13	User should be able to change the quantity of the product added to the shopping cart	Input Number Box Integer	Minimum value shall be 1
FR14	User should be able to view the total cost per product in the shopping cart	Output text Number	Total Cost cannot be Zero Total Cost = Quantity * Price Per Unit

FR15	User should be remove the product from the shopping cart	Button	<p>Quantity should be decremented by 1 each time this button is clicked</p> <p>Entire line item should get removed if the product quantity is 1 when this button gets clicked. i.e., it is equivalent to removing the product from the shopping cart</p>
FR16	User should be able to checkout and directly place an order from the shopping cart	Button	User session should be valid

2.6.7 Place Order

Users can trigger orders after checkout by making payment.

Table 2: Requirements - User Module - Place Order

Req. ID	Description	UI Element & Datatype	Validation
---------	-------------	-----------------------	------------

FR17	User should be able to view the list of products in the order basket. The list should include the image, name, quantity and price of the product.	Order Data info panel Image file, String, Integer	List of products should correspond to the ones added to the shopping cart Product details must be equal in all respects to the ones added to the shopping cart
FR18	User should be able to view the total cost per product in the order basket	Output Text Number	Total Cost cannot be Zero $\text{Total Cost} = \text{Quantity} * \text{Price Per Unit}$

FR19	User should be able to view the overall order quantity of products and the overall order cost	Output Text Integer, Number	Overall Order Quantity cannot be Zero. Overall order cost cannot be zero. Overall Order Quantity = Sum of total quantity of individual product items of the order basket Overall Total Cost = Sum of total cost of individual product items of the order basket
FR20	User should be able to initiate payment for the order	Button	User session should be valid
FR21	User should be able to navigate back to the product listings page	Link	User session should be valid

2.6.8 User Module - Make Payment

Users can use the Payment Gateway integrated into the website using API

Table 2: Requirements - User Module - Make Payment

Req. ID	Description	UI Element & Datatype	Validation
---------	-------------	-----------------------	------------

FR22	User should be able to provide cardholder's email address in the Stripe Payment form	Input Text box String	email address field cannot be empty email address should be a valid email address
FR23	User should be able to provide billing & shipping address in the Stripe Payment form	Input text boxes Dropdown list String	Address line 1 field cannot be empty Pincode field cannot be empty Town / City field cannot be empty Country Field cannot be empty
FR24	User should be able to provide card details i.e, the card number, card expiry date and CVV number in the Stripe Payment form	Input boxes datetime widget Number	Card Number field cannot be empty card expiry date cannot be blank or a previous date CVV Number field cannot be blank
FR25	User should be able save input card details to the Stripe server	Checkbox boolean	Card details should be filled out for the checkbox to get enabled

FR26	User should be able to exit the Stripe payment form	Exit form button	User should be navigated to the make payment page
------	---	------------------	---

2.6.9 User Module - Invoice

Users can receive invoice copy on their registered email addresses as well on the card-holder's email address

Table 2: Requirements - User Module - Invoice

Req. ID	Description	UI Element & Datatype	Validation
FR27	User should receive order invoice copy cum payment receipt upon a successful order placement	Info Panel	User session should be valid Invoice copy cum payment receipt cannot be empty.
FR28	User should receive a unique order ID in the order invoice cum payment receipt for each new order placed.	Output text String	should be unique

FR29	The Order invoice cum payment receipt should contain details of the products ordered that include the image, name, quantity, price per unit and the total price of the product.	Info Panel Image file, String	Products details in the Invoice copy cum payment receipt cannot be empty.
FR30	The Order invoice cum payment receipt should contain the total quantity of products ordered and the total order value	Output text Integer, Number	the total quantity of products ordered and the total order value in the Invoice copy cum payment receipt cannot be empty.
FR31	User should receive error alerts in case of a failed transaction	Alert Message string	user session should be valid

2.6.10 User Module - Notifications

User can receive notifications regarding sign-in, selected products, payment success and order confirmation.

Table 2: Requirements - User Module - Invoice

Req. ID	Description	UI Element & Datatype	Validation
---------	-------------	-----------------------	------------

FR32	User should receive email notification on successful order placement	Email message delivered to user inbox a	<p>Email notification shall be sent to the registered email address of the customer</p> <p>Email notification shall be sent to the email address of the cardholder making the payment</p>
------	--	---	---

2.6.11 Admin Module

Administrator of the website can manage the product catalog and view order data

2.6.12 Admin Module - View Order data

Administrator user of the website should be able to view the orders being placed on the customer web portal

Table 2: Requirements - Admin Module - View Order Data

Req. ID	Description	UI Element & Datatype	Validation
FR33	Admin user should be able to view order data i.e, Order ID, Order Amount and the Order timestamp	<p>Output text</p> <p>String</p>	None of the fields can be invalid or empty

FR34	Admin user should be able to navigate to the Manage Products page	link	Admin user session should be valid
------	---	------	------------------------------------

2.6.13 Admin Module - Manage Products

Admin can create, edit and delete products maintained in the product catalog.
User is presented with the list managed by Admin

Table 2: Requirements - Admin Module - Manage Products

Req. ID	Description	UI Element & Datatype	Validation
FR35	Admin user should be able to navigate to the 'Add Product Form' by clicking a link or button	Button	Admin user session should be valid
FR36	Admin user should be able to view the product details such as the image, name, quantity and price of the products in the product catalog	Info table Image file, String	Product details should correspond to the ones maintained at the backend database

FR37	Admin user should be able to delete a product from the product catalog	Button	Line item for the product should get removed from the table view Product should exist in the backend database for it to get removed
FR38	Admin user should be able to update a product of the product catalog by navigating to the 'Update Product Catalog' form	Button	Product details should be fetched from the backend database and should get displayed in the update Product Form
FR39	Admin user should be able to navigate to the Admin main view i.e. 'View Order Data'	link	Admin user session should be valid

2.6.14 Admin Module - Add Product

Table 2: Requirements - Admin Module - Add Products

Req. ID	Description	UI Element & Datatype	Validation
---------	-------------	-----------------------	------------

FR40	Admin user should be able to create a product by adding a product name, product quantity, and image	Input Text boxes File Upload utility String, Image file	Product name should be unique image file name should be unique image quantity cannot be zero
FR41	Admin user should be able upload an image for the product being added	File Upload utility Image file	image file cannot be empty. image filename cannot be empty and should be valid
FR42	A unique product ID should get generated for each new product created	UUID UUID	Product ID should unique
FR43	Admin user should be able logoff, go back to the previous page, directly go to the Manage Product page or the View Order data page using links or buttons	Link	Admin user session should be valid

FR44	Admin user should receive error alerts for invalid inputs	Alert boxes string	<p>Empty space is not allowed Fields that require unique value shall only accept such values</p> <p>Fields not meeting the criteria described in the regex pattern for the field shall be invalidated</p>
------	---	--------------------	---

2.6.15 Reports

Admin can generate basic report of products purchased on the website

2.7 Software Engineering Paradigm applied

2.7.1 The Personal Software Process (PSP)

The PSP is a structured software development process that is intended (planned) to help software engineers better understand and improve their performance by tracking their predicted and actual development of code. The PSP was created by Watts Humphrey to apply the underlying principles of the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) to the software development practices of a single developer. It claims to give software engineers the process skills necessary to work on a Team software process (TSP) team ^[3].

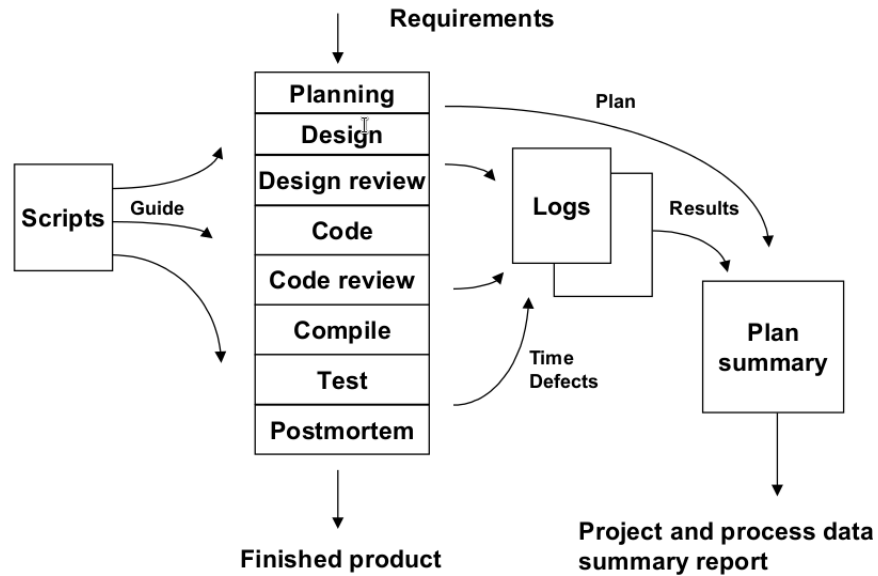


Figure 1: PSP Process Flow Diagram

The structure of the PSP process is shown conceptually in the Figure. Starting with a requirements statement, the first step in the PSP process is planning. There is a planning script that guides this work and a plan summary for recording the planning data. While the engineers are following the script to do the work, they record their time and defect data on the time and defect logs. At the end of the job, during the postmortem phase (PM), they summarize the time and defect data from the logs, measure the program size, and enter these data in the plan summary form. When done, they deliver the finished product along with the completed plan summary form.

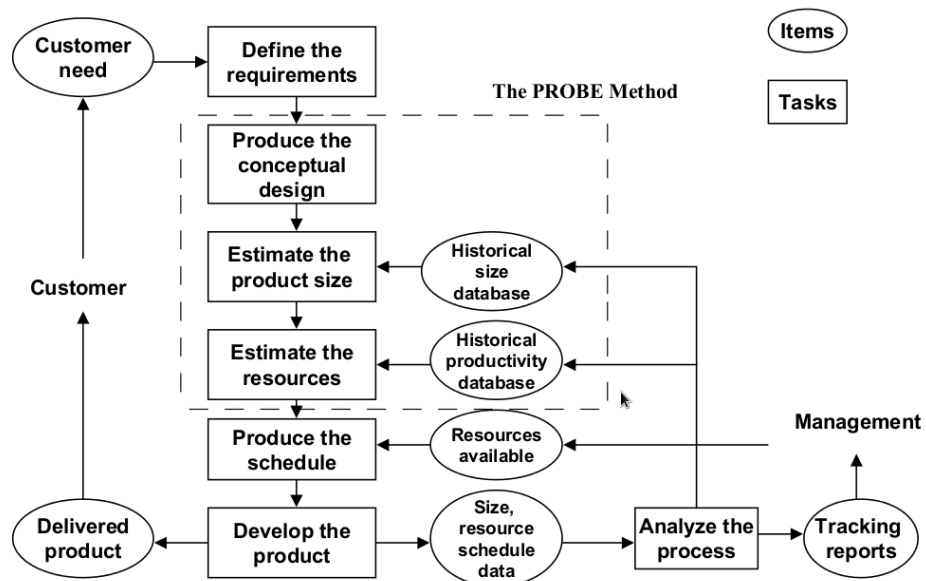


Figure 2: PSP Planning Process Diagram

Requirements

Engineers start planning by defining the work that needs to be done in as much detail as possible. If all they have is a one-sentence requirements statement, then that statement must be the basis for the plan

Conceptual Design.

To make an estimate and a plan, engineers first define how the product is to be designed and built. However, since the planning phase is too early to produce a complete product design, engineers produce what is called a conceptual design. This is a first, rough guess at what the product would look like if the engineers had to build it based on what they currently know. Later, during the design phase, the engineers examine design alternatives and produce a complete product design.

2.7.2 Notes on the Go Programming Language Design ^[4]

The Go programming language was conceived in late 2007 as an answer to some of the problems that were being seen in developing software infrastructure at Google. Today's programs may comprise of tens of millions of lines of code, are worked on by hundreds or even thousands of programmers, and are updated literally every day. To make matters worse, build times, even on large compilation clusters, have stretched to many minutes, even hours.

Go was designed and developed to make working in this environment more productive. Besides its better-known aspects such as built-in concurrency and garbage collection, Go's design considerations include rigorous dependency management, the adaptability of software architecture as systems grow, and robustness across the boundaries between components.

Go is a compiled, concurrent, garbage-collected, statically typed language developed at Google. It is an open source project: Google imports the public repository rather than the other way around. Go is efficient, scalable, and productive.

The goals of the Go project were to eliminate the slowness and clumsiness of software development at Google, and thereby to make the process more productive and scalable. The language was designed by and for people who write, read and debug and maintain large software systems.

The primary considerations in developing the Go programming language were:

- It must work at scale, for large programs with large numbers of dependencies, with large teams of programmers working on them.
- It must be familiar, roughly C-like. The need to get programmers productive quickly in a new language means that the language cannot be too radical.
- It must be modern. C, C++, and to some extent Java are quite old, designed before the advent of multi-core machines, networking, and web

application development. There are features of the modern world that are better met by newer approaches, such as built-in concurrency.

Dependencies in Go

Dependencies are defined, syntactically and semantically, by the language. They are explicit, clear, and “computable“, which is to say, easy to write tools to analyze.

- language defines that unused dependencies are a compile-time error
- Package invocations are transitive and restricted to the other packages imported by the package thus improving compilation time.
- To make compilation even more efficient, the object file is arranged so the export data is the first thing in the file, so the compiler can stop reading as soon as it reaches the end of that section.
- Another feature of the Go dependency graph is that it has no cycles. The language defines that there can be no circular imports in the graph, and the compiler and linker both check that they do not exist.

Packages

- The design of Go’s package system combines some of the properties of libraries, namespaces, and modules into a single construct.
- An important property of Go’s package system is that the package path, being in general an arbitrary string, can be co-opted to refer to remote repositories by having it identify the URL of the site serving the repository.

Syntax

- Go was therefore designed with clarity and tooling in mind, and has a clean syntax.
- Unlike C and Java and especially C++, Go can be parsed without type information or a symbol table; there is no type-specific context. The grammar is easy to reason about and therefore tools are easy to write.
- Function syntax is straightforward for simple functions. A method is just a function with a special parameter, its receiver, which can be passed to the function using the standard ”dot” notation. Go has first-class functions and closures. Finally, in Go functions can return multiple values.
- One feature missing from Go is that it does not support default function arguments. One mitigating factor for the lack of default arguments is that Go has easy-to-use, type-safe support for variadic functions.

Naming

- Go takes an unusual approach to defining the visibility of an identifier. The case of the initial letter of the identifier determines the visibility.

If the initial character is an upper case letter, the identifier is exported (public); otherwise it is not.

- Another simplification is that Go has a very compact scope hierarchy:
 - universe (predeclared identifiers such as int and string)
 - package (all the source files of a package live at the same scope)
 - file (for package import renames only; not very important in practice)
 - function
 - block
- The rules for naming provide an important property for scaling because they guarantee that adding an exported name to a package can never break a client of that package. The naming rules decouple packages, providing scaling, clarity, and robustness.

Semantics

- The semantics of Go statements is generally C-like. That said, Go makes many small changes to C semantics, mostly in the service of robustness. These include:
 - there is no pointer arithmetic
 - there are no implicit numeric conversions
 - array bounds are always checked
 - there are no type aliases (after type X int, X and int are distinct types not aliases)
 - ++ and – are statements not expressions
 - assignment is not an expression
 - it is legal (encouraged even) to take the address of a stack variable
- There are some much bigger changes too, stepping far from the traditional C, C++, and even Java models. These include linguistic support for:
 - concurrency
 - garbage collection
 - interface types
 - reflection
 - type switches

Concurrency

Concurrency is important to the modern computing environment with its multi-core machines running web servers with multiple clients, what might be called the typical Google program. This kind of software is not especially well served by C++ or Java, which lack sufficient concurrency support at the language level.

Garbage Collection

Go has no explicit memory-freeing operation: the only way allocated memory returns to the pool is through the garbage collector. The language is much easier to use because of garbage collection.

Composition not Inheritance

Go takes an unusual approach to object-oriented programming, allowing methods on any type, not just classes, but without any form of type-based inheritance like sub-classing. This means there is no type hierarchy. Instead, Go has interfaces. In Go an interface is just a set of methods. A type will usually satisfy many interfaces, each corresponding to a subset of its methods. Go encourages composition over inheritance, using simple, often one-method interfaces to define trivial behaviors that serve as clean, comprehensible boundaries between components.

Error

Go does not have an exception facility in the conventional sense, that is, there is no control structure associated with error handling. The key language feature for error handling is a pre-defined interface type called `error` that represents a value that has an `Error` method returning a string.

2.8 Data Models & Diagrams

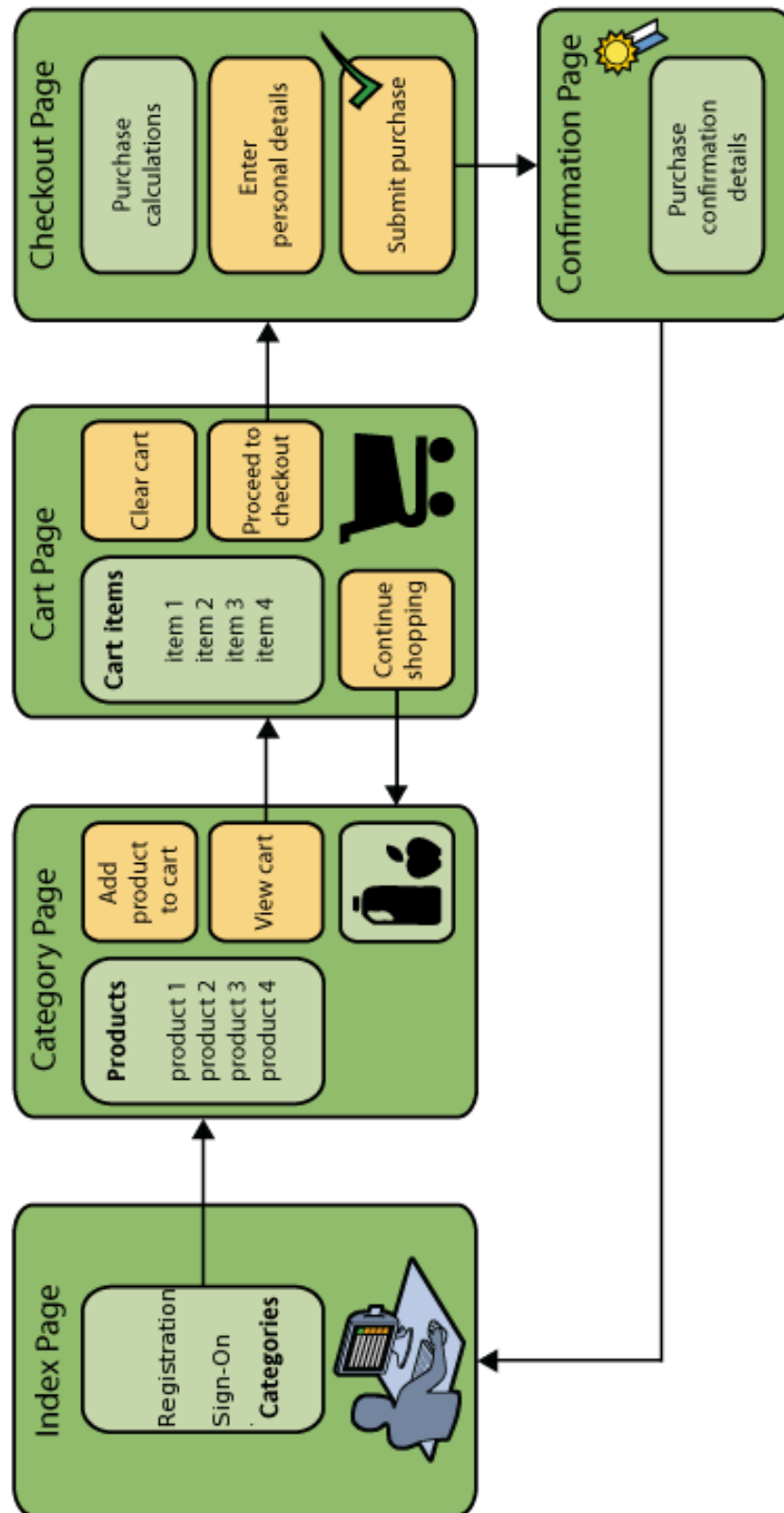


Figure 3: Process Flow Diagram

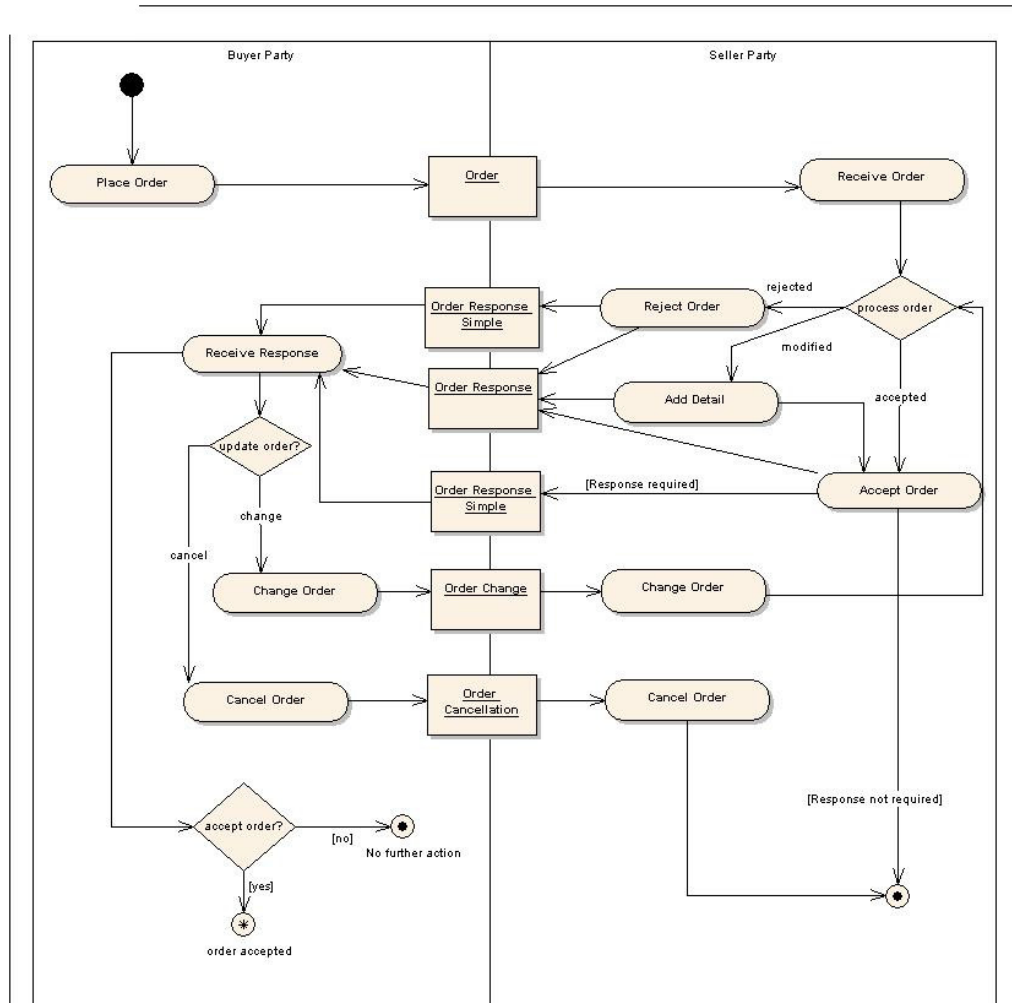


Figure 4: Ordering Process Flow Diagram

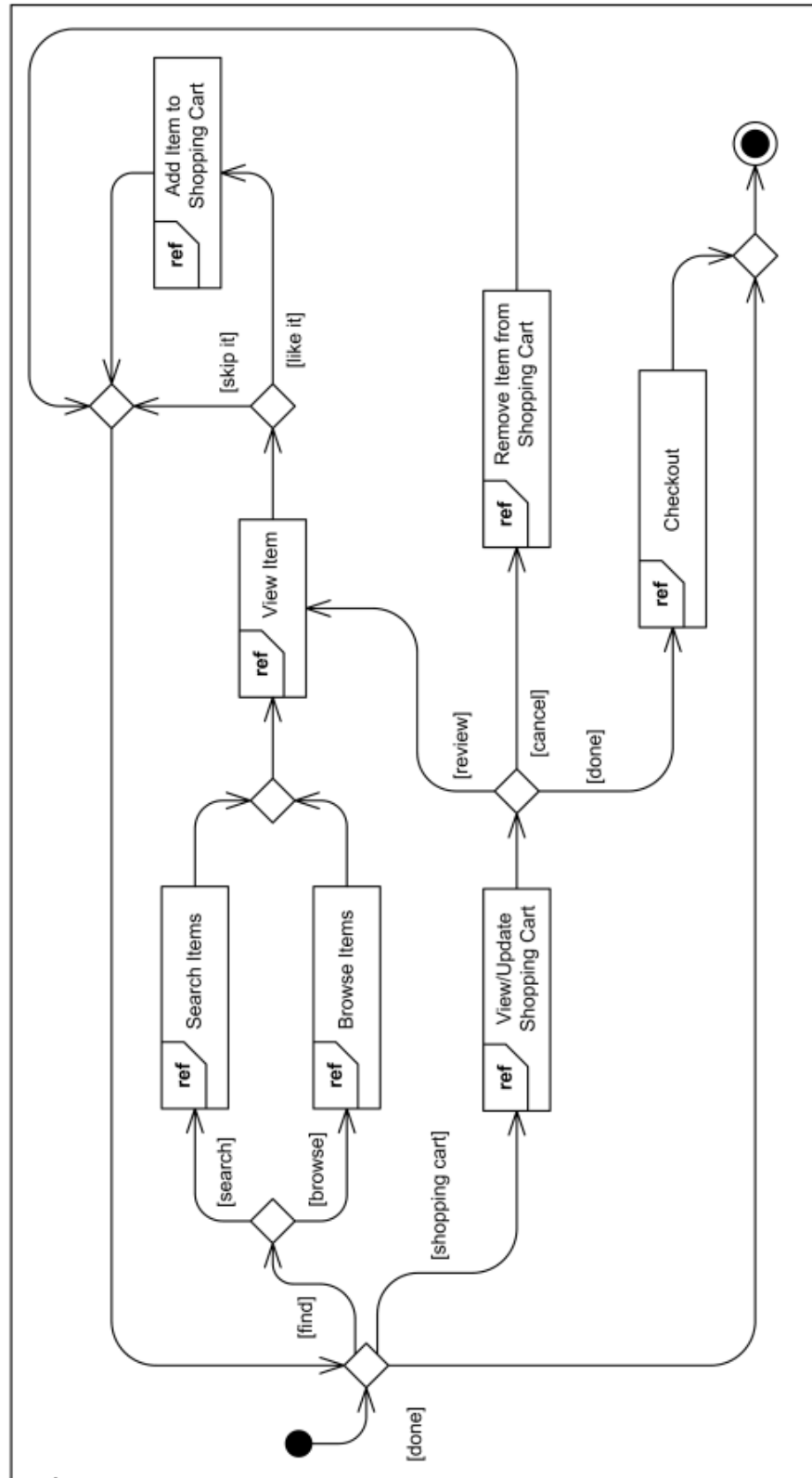
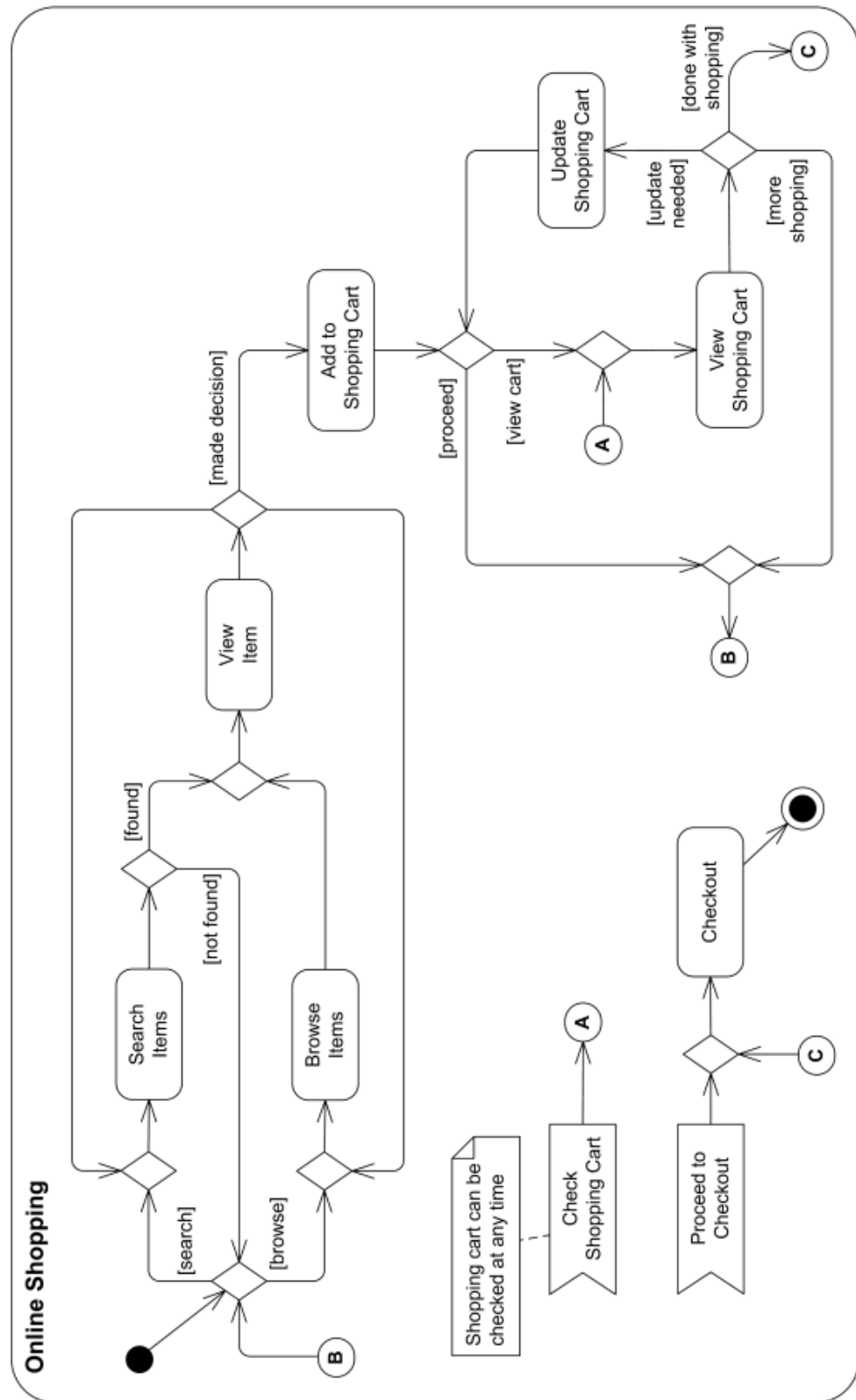


Figure 5: Shopping Cart Process Flow Diagram



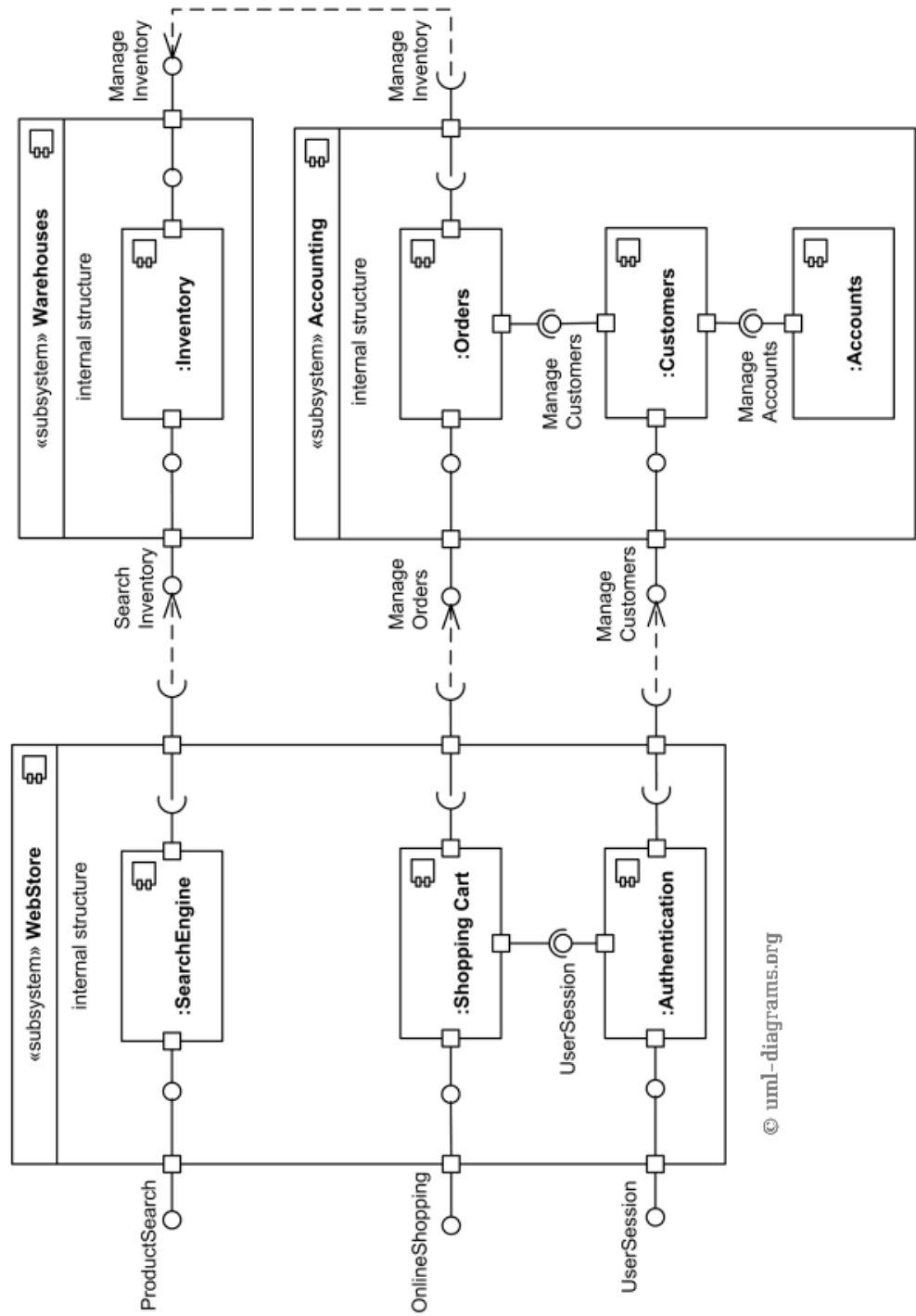


Figure 7: Website Component Diagram

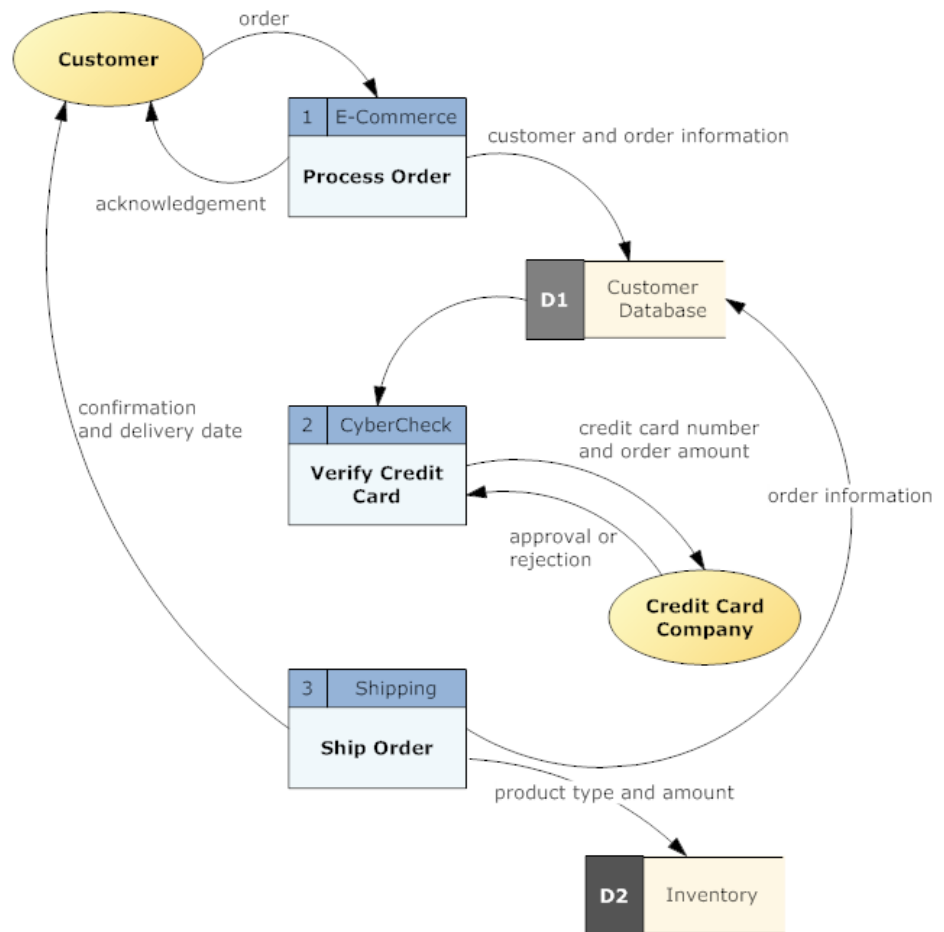


Figure 8: Data Flow Diagram

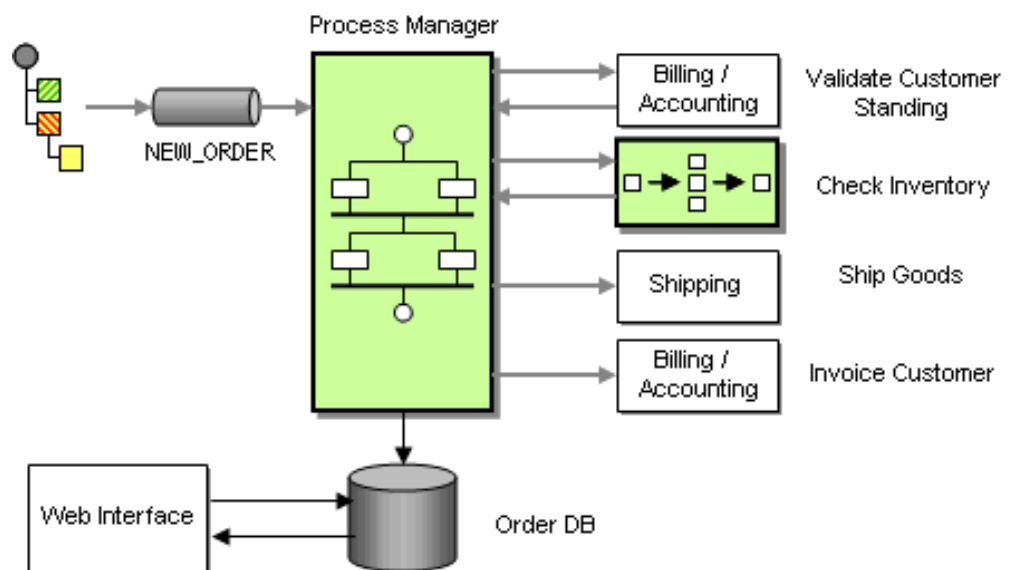


Figure 9: Architecture Diagram

3 SYSTEM DESIGN

3.1 Modularization Details

Table 2: Software & Hardware Requirements

Sr. No.	Module	Description
1.0	User Registration & Sign-On	Users can register on the website. Users need to authenticate during login
1.0	Manage Session	Unique session is created for every user after login. Session is destroyed after user logs out
2.0	Shopping Cart	Javascript module for users to add items to shopping cart
3.0	Product Search	Javascript function for users to be able to search for products using the name string
4.0	Place Order	Triggers order for the selected items in the Cart. A unique Order ID is generated and details are saved to the Database
5.0	Stripe Payment Gateway	API to receive payments for Orders
6.0	Email Notification for Order Confirmation	Automated email notification on successful order and successful payment for the order

7.0	Manage Products	Utility to add products to the database that includes the File Upload Handler
8.0	File Upload Handler	Utility to add Product image when uploading details of a new product
9.0	View Orders	Admin user can query and view details of Placed orders
10.0	View Specific Order	Admin user can query and view details of a specific order

3.2 Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database.

Table	Children	Parents	Columns
orders			10
products			5
sqlite_stat1			3
users			6
4 Tables			24

Figure 10: List of Database Tables

orders	products	sqlite_stat1	users
oid	puid	tbl	uuid
userid	pname	idx	firstname
token	quantity	stat	lastname
orderdetail	price		username
shipdetail	image		email
totalamount			password
chargedamount			
chargeid			
chargestatus			
timestamp			

Figure 11: Utility Tables with data Elements

users.sqlite3 contains 24 columns - click on heading to sort:

Table	Column	Type	Size	Nulls	Auto	Default
orders	chargedamount	integer	2000000000,10			
orders	chargeid	text	2000000000,10			
orders	chargestatus	text	2000000000,10			
orders	orderdetail	blob	2000000000,10			
orders	oid	text	2000000000,10			
orders	shipdetail	blob	2000000000,10			
orders	timestamp	text	2000000000,10	√		null
orders	token	text	2000000000,10			
orders	totalamount	text	2000000000,10			
orders	userid	text	2000000000,10			
products	image	text	2000000000,10	√		null
products	pname	text	2000000000,10			
products	price	text	2000000000,10			
products	puid	text	2000000000,10			
products	quantity	text	2000000000,10			
sqlite_stat1	idx		2000000000,10	√		null
sqlite_stat1	stat		2000000000,10	√		null
sqlite_stat1	tbl		2000000000,10	√		null
users	email	text	2000000000,10			
users	firstname	text	2000000000,10			
users	lastname	text	2000000000,10			
users	password	text	2000000000,10			
users	username	text	2000000000,10			
users	uuid	text	2000000000,10			

Figure 12: Data Elements type and size

Table users.sqlite3.orders

☒ Related columns ☒ Constraints ☐ Comments ☒ Legend

Column	Type	Size	Nulls	Auto	Default	Children	Parents
oid	text	2000000000,10					
userid	text	2000000000,10					
token	text	2000000000,10					
orderdetail	blob	2000000000,10					
shipdetail	blob	2000000000,10					
totalamount	text	2000000000,10					
chargedamount	integer	2000000000,10					
chargeid	text	2000000000,10					
chargestatus	text	2000000000,10					
timestamp	text	2000000000,10	√		null		

Analyzed at Wed Feb 22 10:45 IST 2017

Indexes:

Column(s)	Type	Sort	Constraint Name
oid	Must be unique	Asc	sqlite_autoindex_orders_1

Figure 13: Order table

Table users.sqlite3.products

☒ Related columns ☒ Constraints ☐ Comments ☒ Legend

Column	Type	Size	Nulls	Auto	Default	Children	Parents
puid	text	2000000000,10					
pname	text	2000000000,10					
quantity	text	2000000000,10					
price	text	2000000000,10					
image	text	2000000000,10	√		null		

Analyzed at Wed Feb 22 10:45 IST 2017

Indexes:

Column(s)	Type	Sort	Constraint Name
puid	Must be unique	Asc	sqlite_autoindex_products_1

Figure 14: Product table

Table users.sqlite3.users

☒ Related columns ☒ Constraints ☐ Comments ☒ Legend

Column	Type	Size	Nulls	Auto	Default	Children	Parents
uid	text	2000000000,10					
firstname	text	2000000000,10					
lastname	text	2000000000,10					
username	text	2000000000,10					
email	text	2000000000,10					
password	text	2000000000,10					

Analyzed at Wed Feb 22 10:45 IST 2017

Figure 15: User table

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

JSON is a format that encodes objects in a string. Serialization means to convert an object into that string, and deserialization is its inverse operation. When transmitting data or storing them in a file, the data are required to be byte strings.

The 'orderdetails' and 'shipdetails' fields are maintained as JSON data (Blob type) in the 'Orders' table of the database

```

▼ array [2]
  ▼ 0 {5}
    pid : d6742e4e-2ad6-43c5-97f4-e8a7b00684e2
    image : 1Appleiphone7.jpeg
    name : iphone7
    price : 70000
    count : 1
  ▼ 1 {5}
    pid : 12d3d8fc-66b6-45f9-a91b-d400b91c32aa
    image : 2SamsungGalaxys7.jpeg
    name : SamsungGalaxy7
    price : 70000
    count : 1

```

Figure 16: Orderdetails Field JSON Structure

```

▼ array {6}
  eladdress : rejoy_nair@yahoo.com
  shaddresscity : Kharghar
  shaddresscountry : India
  shaddressline : 403, Ahilya dutt, sec-13
  shaddressname : Rejoy
  shaddresszip : 410210

```

Figure 17: Shipdetails Field JSON Structure

```
▼ array [6]
  ▼ 0 {3}
    chargedamount : 140000
    ould : 8c902844-2075-4dfc-a1e9-89e0b95cd8ae
    timestamp : 18.02.2017 11:45:39
  ▼ 1 {3}
    chargedamount : 70000
    ould : 69c1f199-7d27-4076-aa70-af1754be1bd4
    timestamp : 21.02.2017 10:00:17
  ▼ 2 {3}
    chargedamount : 70000
    ould : e3767864-14de-42d6-b3a5-9122e3b9aa8f
    timestamp : 21.02.2017 11:44:59
  ▼ 3 {3}
    chargedamount : 70000
    ould : a589b558-3dd7-4cf9-9673-718e8f1dbc5b
    timestamp : 21.02.2017 12:12:41
  ▼ 4 {3}
    chargedamount : 70000
    ould : 2a7f5bf6-ba13-4234-b345-7ab8f379a9d8
    timestamp : 23.02.2017 16:23:13
  ▼ 5 {3}
```

Figure 18: JSON Array View Order

Figure 19: Order table viewed in SQLite Manager

3.3 User Interface Design

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).⁵

A good e-commerce site should present the following factors to users for better usability

- Simple navigation from home page to information and order links for specific products
- Obvious shopping links or buttons
- Effective categorical Organization of products
- Easy scanning and selecting items on list
- Consistent layout of product information
- Minimal or effective security notification or messages
- Knowing when an item was saved or not saved in the shopping cart

Login

User Name

Password

Submit

Sign Up

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 20: Customer Login Screen

Table 2: Customer Login Screen

Sr. No.	Input Element	Description & Behaviour
1.0	Username Input Field	Textbox to accept registered username. Email ID or any other unique attribute shall not be accepted
2.0	Password Input Field	Textbox to accept password associated with the Username
3.0	Submit Button	To post the Username and password entered by the user to the server
4.0	Alert	To notify User of incorrect or empty Username and / or password or an expired User session

Login

Username or Password field are empty!

User Name

Password

Submit

Sign Up

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 21: Customer Login Screen - Alert

User Sign in - Iceweasel

https://127.0.0.1:8091/signup

Web Search CrunchBang click here to go to tl... Shop Donate

Sign Up

User Name

First Name

Last Name

Email

Password

Confirm Password

Submit

Sign In!

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 22: Customer Sign-up Screen

Table 2: Customer Signup Screen

Sr. No.	Input Element	Description & Behaviour
1.0	Username Input Field	Textbox to accept username that the user wishes to register with. Username will have to be unique
2.0	First Name Input Field	Textbox to accept first name that the user wishes to register with.
3.0	Last Name Input Field	Textbox to accept last name that the user wishes to register with.
4.0	Email Input Field	Textbox to accept email address that the user wishes to register with.
5.0	Password Input Field	Textbox to create a password for the username that the user wishes to register with.
6.0	Confirm Password Input Field	Textbox to reconfirm the entered password at the time of registration
7.0	Submit Button	To post the details contained in the sign-up form as entered by the user to the server
8.0	Alert	To notify User of incorrect or empty or invalid input entries for each of the fields of the sign-up form

User Sign in - Iceweasel

https://127.0.0.1:8091/signup

Web Search CrunchBang click here to go to tl... Shop Donate

User Name

Please enter a valid Username!

First Name

Please enter a valid First Name

Last Name

Please enter a valid Last Name

Email

Please enter a valid Email Address!

Password

Please enter a Password!

Confirm Password

Submit

Figure 23: Customer Sign-up Screen Validations

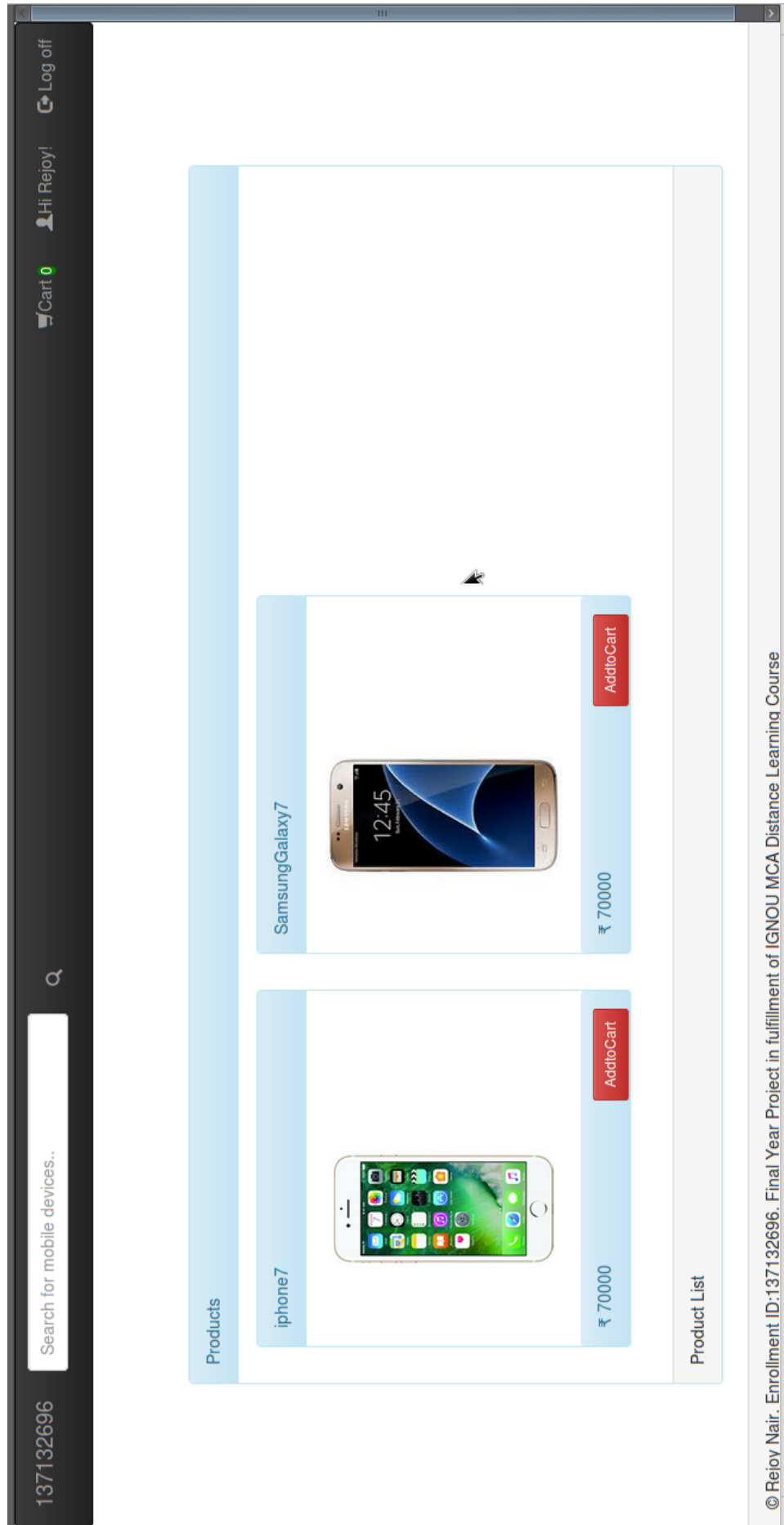


Figure 24: Customer Login Main View

Table 2: Customer Login Main View

Sr. No.	Input Element	Description & Behaviour
1.0	Product Info Panel	Panel that contains description of the product such as name and price along with its image
2.0	Product Info Panel - Add to Cart Button	''Add to Cart'' button on the product info panel that allows customers to add products to the cart
3.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
4.0	Search Box	Search box to filter products based on Product description
5.0	Shopping Cart	Icon which shows the count of the products added to the shopping cart. Clicking on the shopping cart displays the dropdown that contains the list of the products added to the cart along with the image, price, quantity and total amount per item
6.0	Shopping Cart-Quantity	Number input type that can be increased or decreased. Min Value is 1

7.0	Shopping Cart-Button	<p>``-`` button to remove a single unit of the item from the total quantity.</p> <p>``+`` to add a single unit to the existing quantity.</p> <p>``X`` to completely remove the item from the shopping cart list</p>
8.0	Alert	To notify User of a product being added to the shopping cart
9.0	User Profile	Link to the user Profile Page. By defaults shows the user first name.
10.0	Logout	Allows the user to logout from the web portal

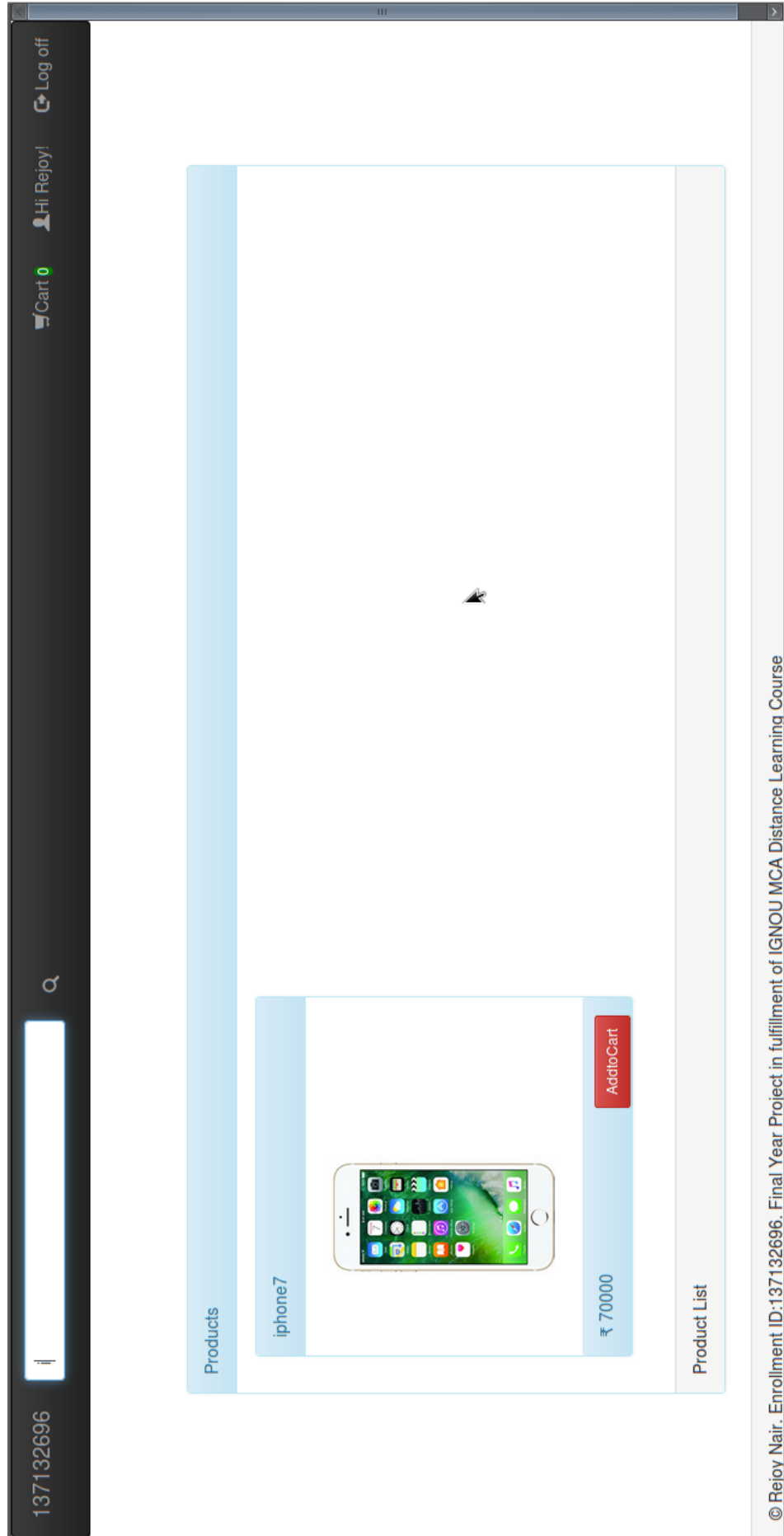


Figure 25: Customer Login Main View - Search

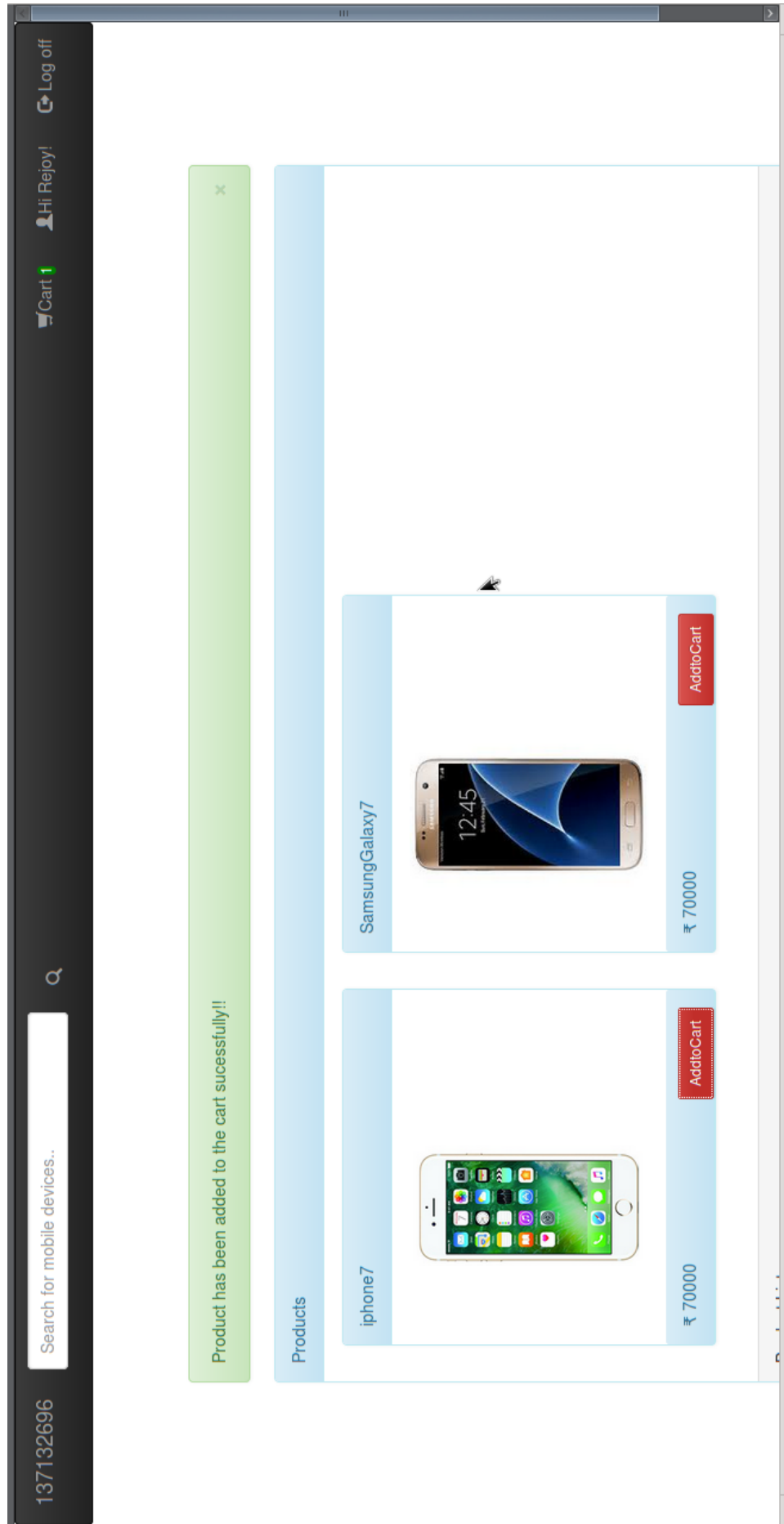


Figure 26: Customer Login Main View - Alert

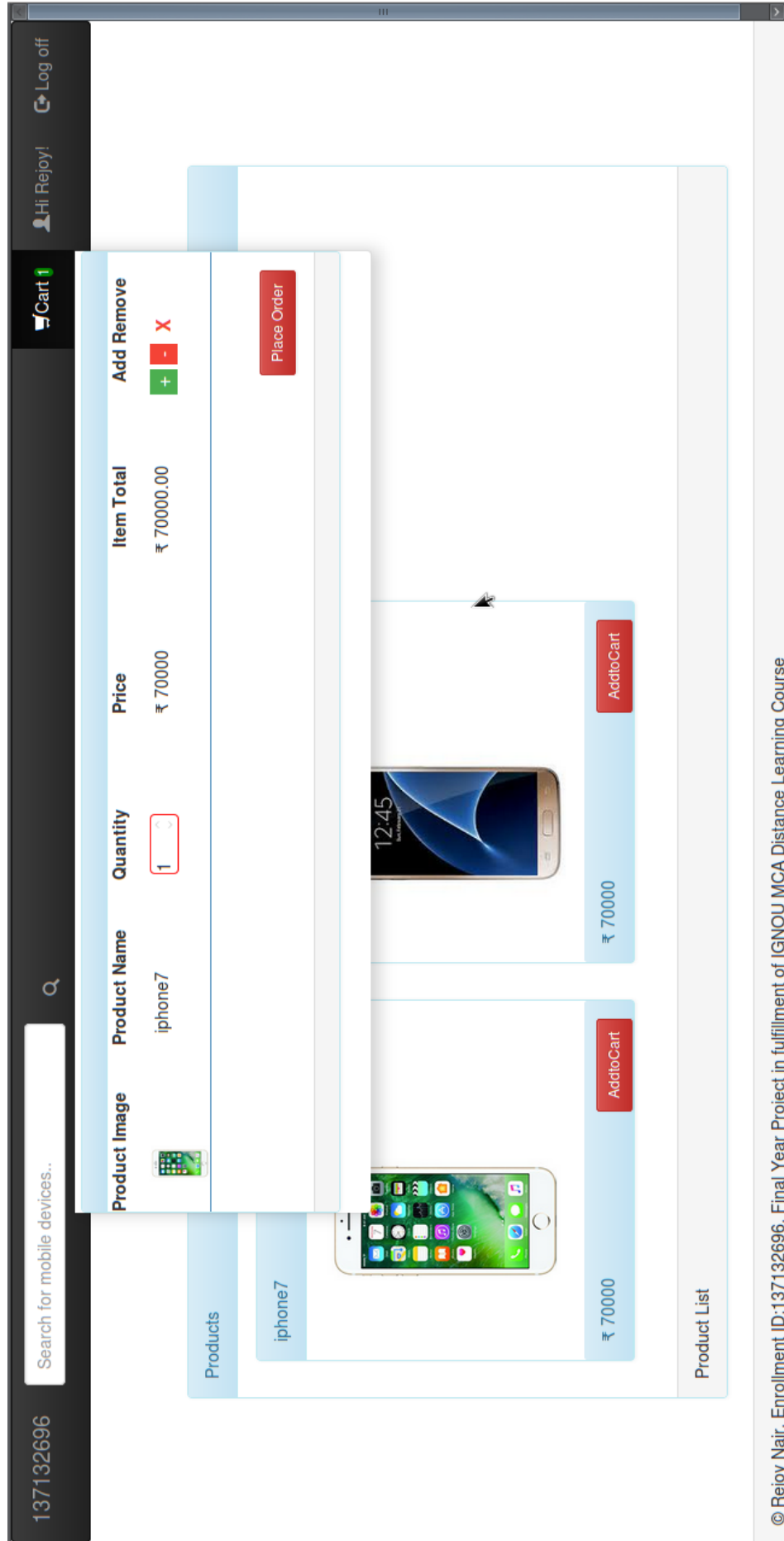


Figure 27: Customer Login Main View - Shopping Cart


Table 2: Customer Login Main View

Sr. No.	Input Element	Description & Behaviour
1.0	Product Image	Image of the Product added to the shopping cart
2.0	Product Name	Name of the Product added to the shopping cart
3.0	Quantity	Input number box to increase or decrease the quantity of the product added to the shopping cart
4.0	Price	Price of the product added to the shopping cart
5.0	Item Total	Total cost per line item
6.0	``-`` button	Button to remove a single unit of the product added to the shopping cart
7.0	``+`` button	Button to increase the quantity by a single unit of the product already added to the shopping cart
8.0	``X``	Button to completely remove the product added to the shopping cart
9.0	Place Order	Button to place an order with the items added to the shopping cart

137132696

Back

Log off

Product Image	Product Name	Quantity	Price	Item Total
	iphone7	1	₹ 70000	₹ 70000.00
			1	₹ 70000.00

Pay with Card

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 28: Customer Place Order Screen

Table 2: Customer Place Order Screen

Sr. No.	Input Element	Description & Behaviour
1.0	Product Image	Image of the Product added to the shopping cart
2.0	Product Name	Name of the Product added to the shopping cart
3.0	Quantity	Input number box to increase or decrease the quantity of the product added to the shopping cart
4.0	Price	Price of the product added to the shopping cart
5.0	Item Total	Total cost per line item
6.0	Total Count	Total count of products added to shopping cart
7.0	Grand Total	Total Order value
8.0	Pay with Card	Button to call the Stripe's Payment Gateway API
9.0	Back Button	Button to go back to the previous page i.e. the main view
10.0	Logoff Button	Button to logoff from the customer web portal

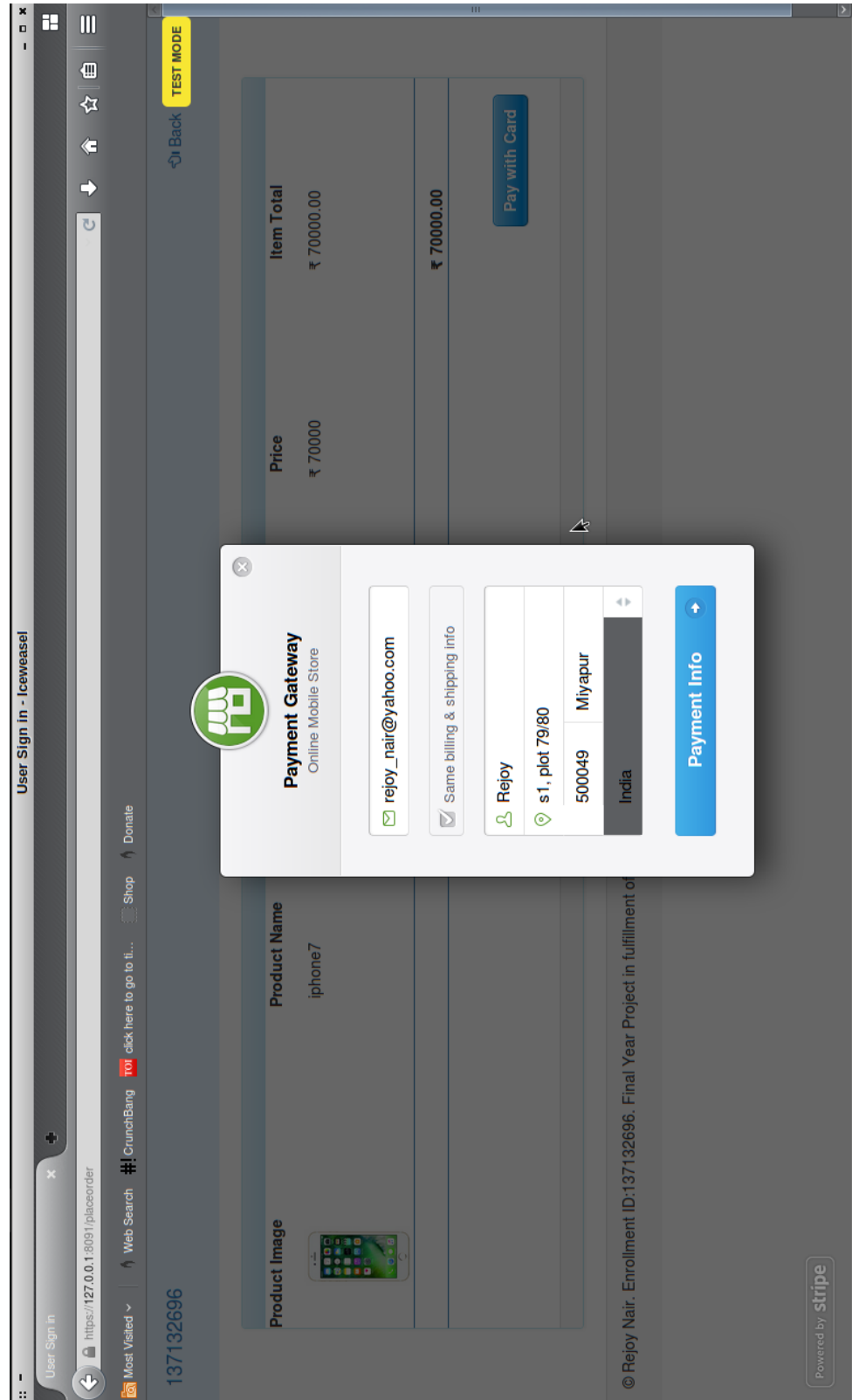


Figure 29: Stripe Payment Gateway Submit Form-1

Table 2: Stripe Payment Gateway Submit Form-1

Sr. No.	Input Element	Description & Behaviour
1.0	Label	Label on top of the form
2.0	Email Address	Email Address of the card Holder. This can be different from the registered email address of the customer
3.0	Checkbox - Billing and Shipping info	Checkbox to indicate if the billing and shipping addresses are one and the same. Default is checked.
4.0	Name of the Card Holder	Text box to enter the name of the card holder
5.0	Address Line 1	Text box to enter the address line of the card holder
6.0	Pincode	Text box to enter the pincode corresponding to the address of the card holder
7.0	Town / City Name	Town / city name for the entered pincode. This field is auto-populated based on the entered pincode.
8.0	Payment Info	Button to navigate to the next section of the form where the card information shall be sought
9.0	Grand Total	Total Order value
10.0	Pay with Card	Button to call the Stripe's Payment Gateway API

11.0	Card Number	Text box to enter the card number of the card holder
12.0	Card Expiry date	Calendar widget to enter the card number of the card holder
13.0	Card CVV Number	Input box to enter the CVV number associated with the Card Number of the card holder
14.0	'Remember me' Checkbox	Checkbox to save the cardholder and the card details on Stripe's server
15.0	Pay	Button to post card holder information and card details to Stripe's server

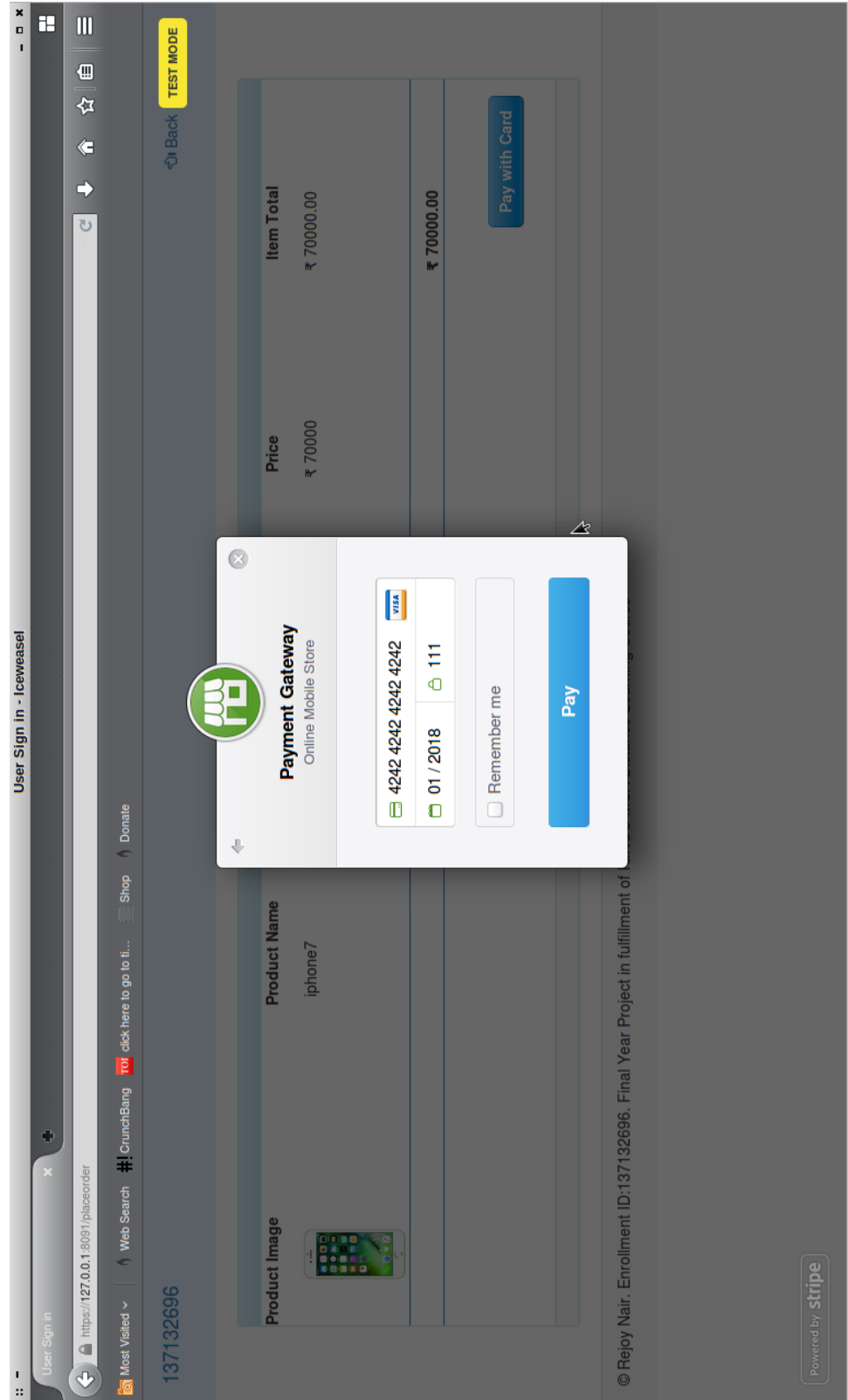


Figure 30: Stripe Payment Gateway Submit Form-2

137132696

Home Log off

Order Invoice / Payment Receipt

Order ID:

2a7f5bf6-ba13-4234-b345-7ab8f379a9d8

Shipping Address

Rejoy
s1, plot 79/80
Miyapur
500049
India

Email Address:

rejoy_nair@yahoo.com

Ordered Items:



Product: iphone7

Qty: 1

Price: ₹ 70000

Total Amount:

₹ 70000

Order Date/Time:

23.02.2017 16:23:13

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 31: Order Confirmation Screen

Table 2: Order Invoice / Payment Receipt

Sr. No.	Input Element	Description & Behaviour
1.0	Order ID	Output text displaying the Order ID field value of the Order Invoice cum payment receipt of the Order placed by the customer
2.0	Shipping Address	Output text displaying the Shipping Address field value of the Order Invoice cum payment receipt of the Order placed by the customer
3.0	Email Address	Output text displaying the Cardholder's Email Address field value of the Order Invoice cum payment receipt of the Order placed by the customer
4.0	Ordered Items - Product Image	Image of the Ordered Product corresponding to the Order Invoice cum payment receipt of the Order placed by the customer
5.0	Ordered Items - Product Name	Output text displaying the Name of the Ordered Product corresponding to the Order Invoice cum payment receipt of the Order placed by the customer

6.0	Ordered Items - Ordered Quantity	Output text displaying the Quantity of the Ordered Product corresponding to the Order Invoice cum payment receipt of the Order placed by the customer
7.0	Ordered Items - Amount	Output text displaying the Per Unit Cost of the Ordered Product corresponding to the Order Invoice cum payment receipt of the Order placed by the customer
8.0	Total Amount	Output text displaying the Total Order Amount corresponding to the Order Invoice cum payment receipt of the Order placed by the customer
9.0	Order DateTime	Output text displaying the local date-time corresponding to the Order Invoice cum payment receipt of the Order placed by the customer
10.0	Home Button	Button to navigate the customer back to the main page of the customer web portal
11.0	Logoff Button	Button to logoff from the customer web portal

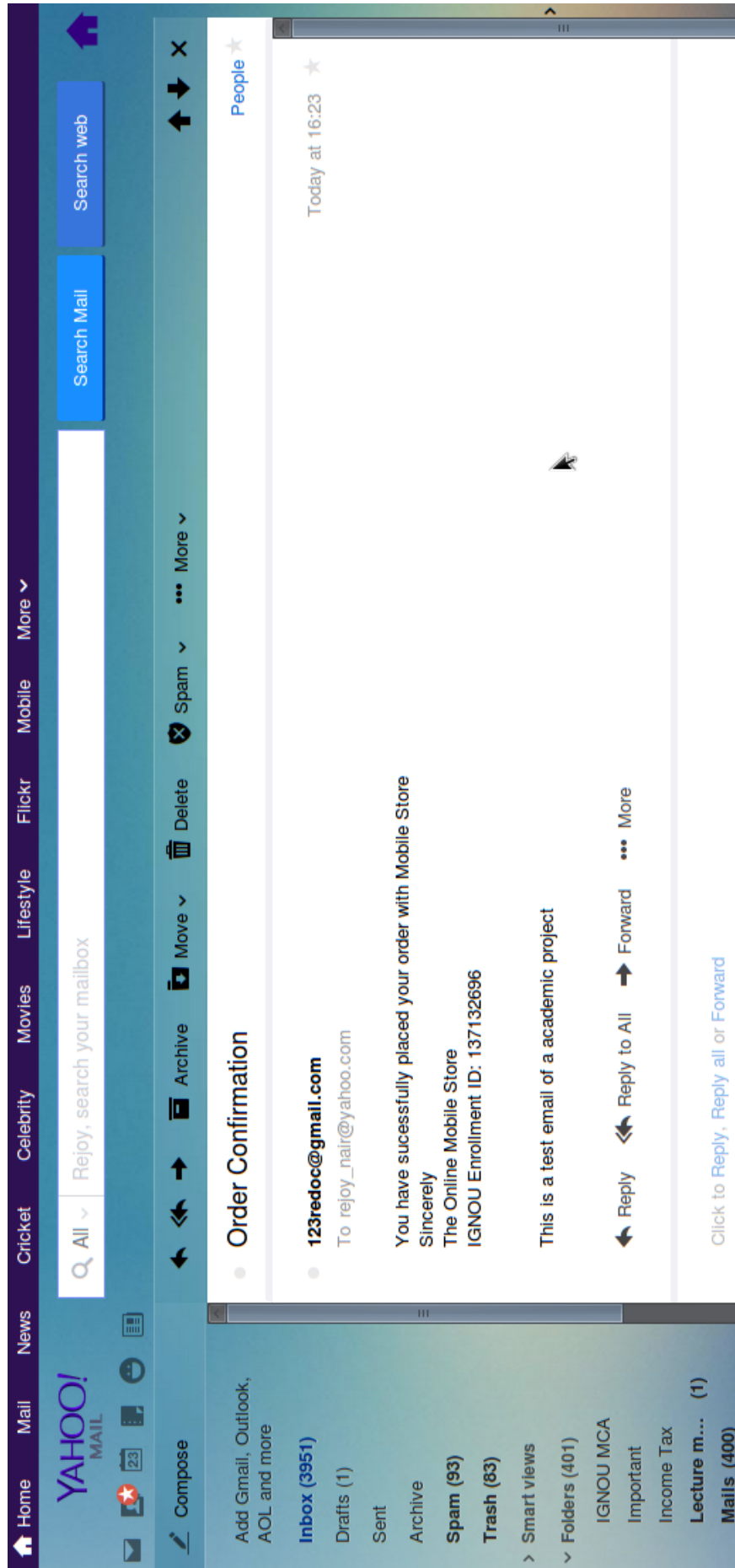


Figure 32: Email Notification in Inbox

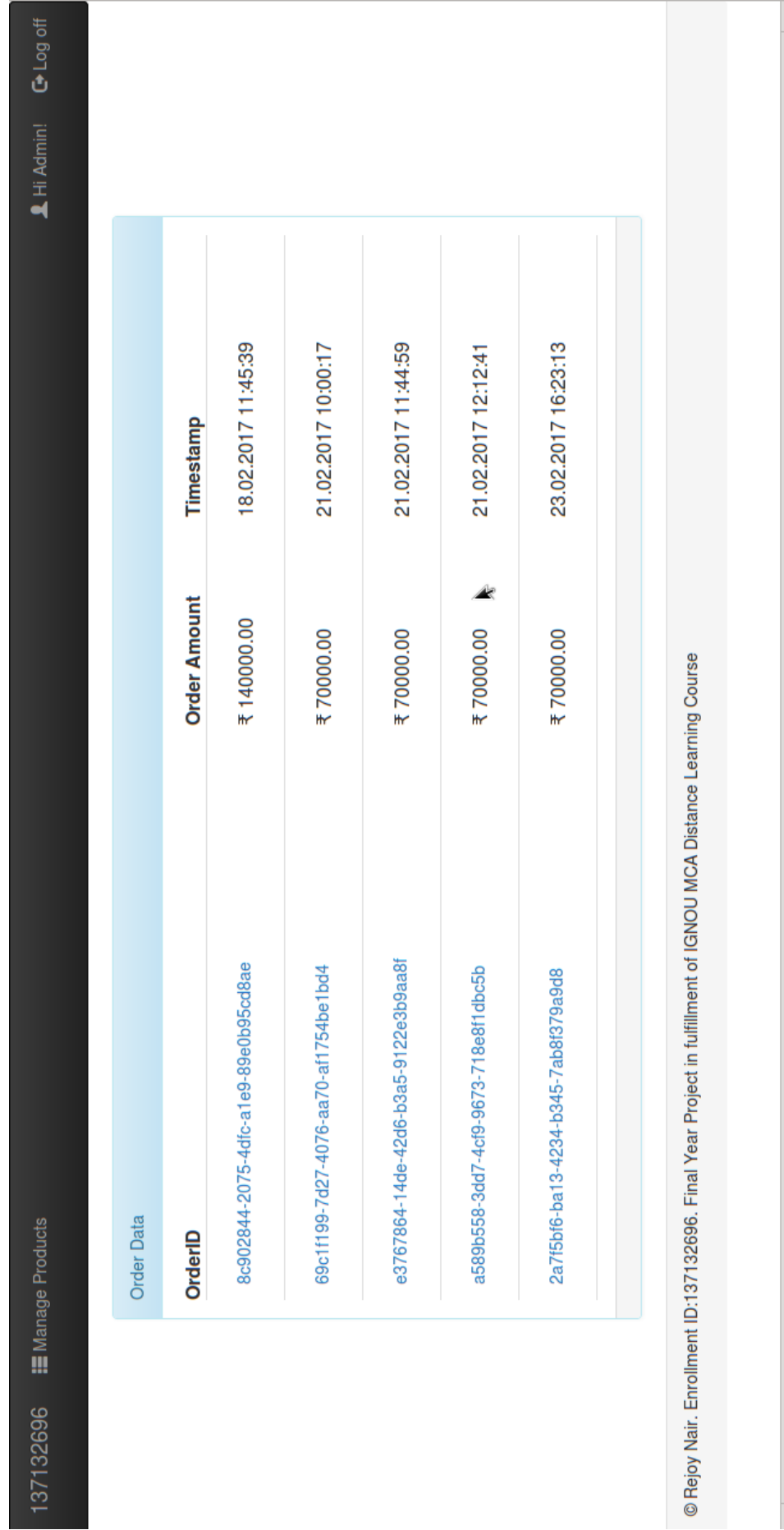


Figure 33: Admin User Main View

Table 2: Admin Main View - Order data

Sr. No.	Input Element	Description & Behaviour
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
2.0	Manage Product	Link to navigate the Admin user to the Manage Products Page
3.0	User Profile icon	Displays the text 'Admin' indicating Admin user login
4.0	Log-off	Link to log-off the Admin User from the Admin User portal
5.0	Order Data - Order ID	Output text displaying the Order ID of the successfully placed orders displayed in the order data table
6.0	Order Data - Order Amount	Output text displaying the total amount corresponding to each order ID of the successfully placed orders displayed in the order data table
7.0	Order Data - Timestamp	Output text displaying the timestamp corresponding to each order ID of the successfully placed orders displayed in the order data table

137132696

View Orders



Hi Admin! Log off

Add Product

Click on the button to add a new product

Add Product

Manage Products

Product Image	Product Name	Product Quantity	Product Price	Delete / Update
	iphone7	100	₹ 70000	<div>DeleteUpdate</div>
	SamsungGalaxy7	100	₹ 70000	<div>DeleteUpdate</div>

© Rajou Neir Enrollment ID-137132696 Final Year Project in fulfillment of ICNOLMCA Distance Learning Course

Figure 34: Admin - Manage Products

Table 2: Admin Manage Products

Sr. No.	Input Element	Description & Behaviour
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
2.0	View Order	Link to navigate the Admin user back to the Admin Main View i.e., the View Order Data Page
3.0	User Profile icon	Displays the text 'Admin' indicating Admin user login
4.0	Log-off	Link to log-off the Admin User from the Admin User portal
5.0	Add Product	Button that navigates the admin user to the add product form
6.0	Manage Products - Product Image	Image of the Product that exists in the backend database
7.0	Manage Products - Product Name	Output text displaying the name of the Product that exists in the backend database
8.0	Manage Products - Product Quantity	Output text displaying the quantity of the Product as maintained in the backend database
9.0	Manage Products - Product Price	Output text displaying the price of the Product as maintained in the backend database

10.0	Manage Products - Delete Button	Button to delete the record of the corresponding product from the database
11.0	Manage Products - Update Button	Button to navigate the user to the update product form to update the record of the corresponding product from the database

137132696

Manage Products

View Orders

Back

Log off

Add Product Form

Product Name

Quantity

Price

Upload Image of the Product

Browse ...

No files selected.

Submit

© Rejoy Nair. Enrollment ID:137132696. Final Year Project in fulfillment of IGNOU MCA Distance Learning Course

Figure 35: Admin - Add Product Form

Table 2: Admin Manage Products

Sr. No.	Input Element	Description & Behaviour
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
2.0	View Order	Link to navigate the Admin user back to the Admin Main View i.e., the View Order Data Page
3.0	Manage Products	Link to navigate the Admin user back to the Admin - Manage Products Page
4.0	Back Button	Navigates the admin user back to the previous page
4.0	Log-off	Link to log-off the Admin User from the Admin User portal
5.0	Add Product - Product Name	Text field to input the name of the product to be added
6.0	Add Product - Quantity	Input field to input the name of the product to be added
7.0	Add Product - Price	Input field to input the price of the product to be added
8.0	Add Product - Upload Image File	File Upload utility to upload the image of the product to be added
9.0	Add Product - Submit Button	Button to post the Add Product form details to the server

10.0	Add Product - Alerts	Alerts to notify the admin users of fields that fail validation criteria
------	----------------------	--

User Sign in - Iceweasel

137132696

Manage Products View Orders

Web Search Most Visited

click here to go to l... #! CrunchBang Shop Donate

Back Log off

Add Product Form

Product Name

Please enter a valid Product Name

Quantity

Please enter a valid Quantity!

Price

Please enter a valid Price

Figure 36: Admin - Add Product Form Validations - I

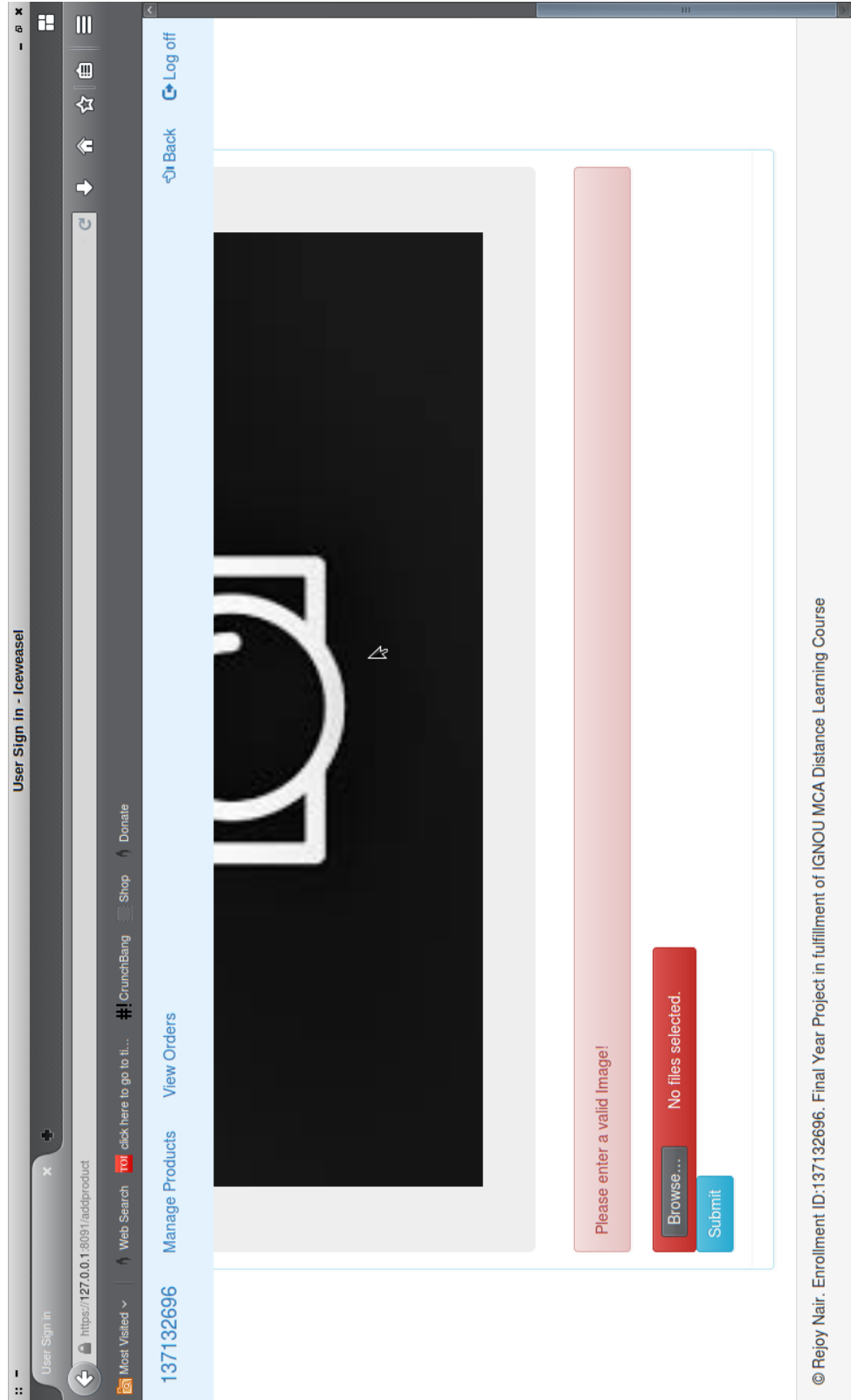


Figure 37: Admin - Add Product Form Validation -II

4 CODING

```
1 package main
2
3 import (
4     "log"
5     "net/smtp"
6 )
7
8 func orderEmail(el string) {
9     from := "123redoc@gmail.com"
10    password := "something@1"
11
12    auth := smtp.PlainAuth("", from, password, "smtp.gmail.com↵
13    ")
14    to := el
15    msg := "From: " + from + "\r\n" +
16           "To: " + to + "\r\n" +
17           "Subject: Order Confirmation" + "\r\n\r\n" +
18           "You have sucessfully placed your order with ↵
19           Mobile Store" + "\r\n" +
20           "Sincerely" + "\r\n" +
21           "The Online Mobile Store" + "\r\n" +
22           "IGNOU Enrollment ID: 137132696" + "\r\n" + "\r\n\r\n↵
23           n" +
24           "This is a test email of a academic project" + "\r\n↵
25           "
26
27    /* Ports 465 and 587 are intended for email client to ↵
28       email server communication (sending email). Port 465 ↵
29       is for smtps. SSL encryption is started automatically ↵
30       before any SMTP level communication. Port 587 is for ↵
31       msa. It is almost like standard SMTP port. */
32
33    err := smtp.SendMail("smtp.gmail.com:587", auth, from, []↵
34        string{to}, []byte(msg))
35    if err != nil {
36        log.Printf("Error: %s", err)
37        return
38    }
39    log.Print("message sent")
40 }
41
42 /*
43
44 func main() {
45     el := "rejoy_nair@yahoo.com"
46     orderEmail(el)
47 }
48 */
```

```
1 {{ define "header" }}
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge,
   chrome=1">
```

```

6 <title>User Sign in</title>
7 <meta name="description" content="">
8 <meta name="viewport" content="width=device-width,
  initial-scale=1">
9
10 <link rel="stylesheet" href="/initializr/css/
  bootstrap.min.css">
11 <link rel="stylesheet" href="/initializr/css/
  bootstrap-theme.min.css">
12 <link rel="stylesheet" href="/initializr/css/main.css">
13
14 <script src="/initializr/js/vendor/
  modernizr-2.8.3-respond-1.4.2.min.js"></script>
15
16 <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.3/
  jquery.min.js"></script>
17 <script>window.jQuery || document.write('<script src="js/
  vendor/jquery-1.11.3.min.js"></script>')</script>
18
19 <script src="/initializr/js/vendor/bootstrap.min.js">
  </script>
20
21 <script src="/initializr/js/main.js"></script>
22
23 </head>
24 <body>
25 {{end}}
26
27
28
29 {{define "footer"}}
30 <footer class="panel-footer" ><p>&copy; Rejoy Nair.
  Enrollment ID:137132696. Final Year Project in
  fulfillment of IGNOU MCA Distance Learning Course</p></
  footer>
31 <!-- <footer class="panel-footer" style="
  position:absolute;bottom:0;width:100%;" ><p>&copy; Rejoy
  Nair. Enrollment ID:137132696. Final Year Project in
  fulfillment of IGNOU MCA Distance Learning Course</p></
  footer> -->
32
33
34 <!-- Google Analytics: change UA-XXXXX-X to be your site's
  ID. -->
35 <script>
36     (function(b,o,i,l,e,r){
37         b.GoogleAnalyticsObject=l;b[l]||(b[l]=
38         function(){(b[l].q=b[l].q||[]).push(arguments)
39         });b[l].l=+new Date;
40         e=o.createElement(i);r=o.getElementsByTagName(i
41         )[0];
42         e.src="//www.google-analytics.com/analytics.js"
43         ;
44         r.parentNode.insertBefore(e,r)}(window,document
45         , 'script', 'ga'));
46         ga('create','UA-XXXXX-X','auto');ga('send','
47         pageview');
48 </script>

```

43 </body>
44 </html>
45 {{end}}

Table 2: Test Cases for Req. ID FR1

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR2

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR3

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR4

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR5

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR6

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR7

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR8

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR9

--

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR10

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR11

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR12

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR13

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR14

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR11

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR15

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR16

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR17

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR18

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR19

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR20

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR21

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR22

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR23

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR24

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR25

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR26

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR27

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR28

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR29

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR30

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR31

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR32

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR33

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR34

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR35

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR36

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR37

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR38

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR39

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR40

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

Table 2: Test Cases for Req. ID FR41

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR42

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR43

Sr. No	Test Cases	Expected Behavior & Status
1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696

Table 2: Test Cases for Req. ID FR44

Sr. No	Test Cases	Expected Behavior & Status

1.0	Banner	Banner to place the brand logo. Currently contains the enrollment ID 137132696
-----	--------	---

FR1 FR2 FR3 FR4 FR5
 FR6 FR7 FR8
 FR9 FR10 FR11
 FR12 FR13 FR14 FR15 FR16
 FR17 FR18 FR19 FR20 FR21
 FR22 FR23 FR24 FR25 FR26
 FR27 FR28 FR29 FR30 FR31
 FR32
 FR33 FR34
 FR35 FR36 FR37 FR38 FR39
 FR40 FR41 FR42 FR43 FR44

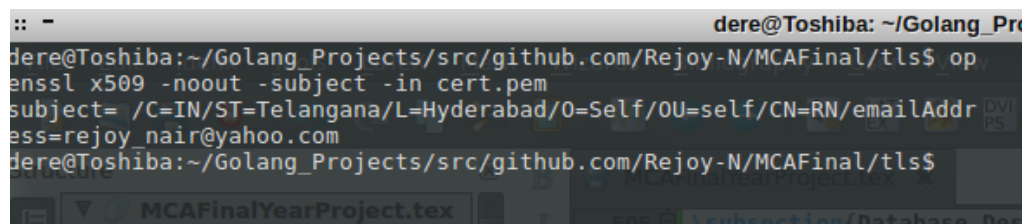
5 SYSTEM SECURITY MEASURES

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), both frequently referred to as “SSL“, are cryptographic protocols that provide communications security over a Computer Network ⁶.

TLS has been implemented in the project using the OpenSSL Client 1.0.1e
Command use to generate the TLS cert keys

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days  
365 -nodes
```

Data contained in the cert key



```
dere@Toshiba: ~/Golang_Pr  
dere@Toshiba:~/Golang_Projects/src/github.com/Rejoy-N/MCAFinal/tls$ op  
enssl x509 -noout -subject -in cert.pem  
subject= /C=IN/ST=Telangana/L=Hyderabad/O=Self/OU=self/CN=RN/emailAddr  
ess=rejoy_nair@yahoo.com  
dere@Toshiba:~/Golang_Projects/src/github.com/Rejoy-N/MCAFinal/tls$
```

Figure 38: TLS Certificate Details - 1

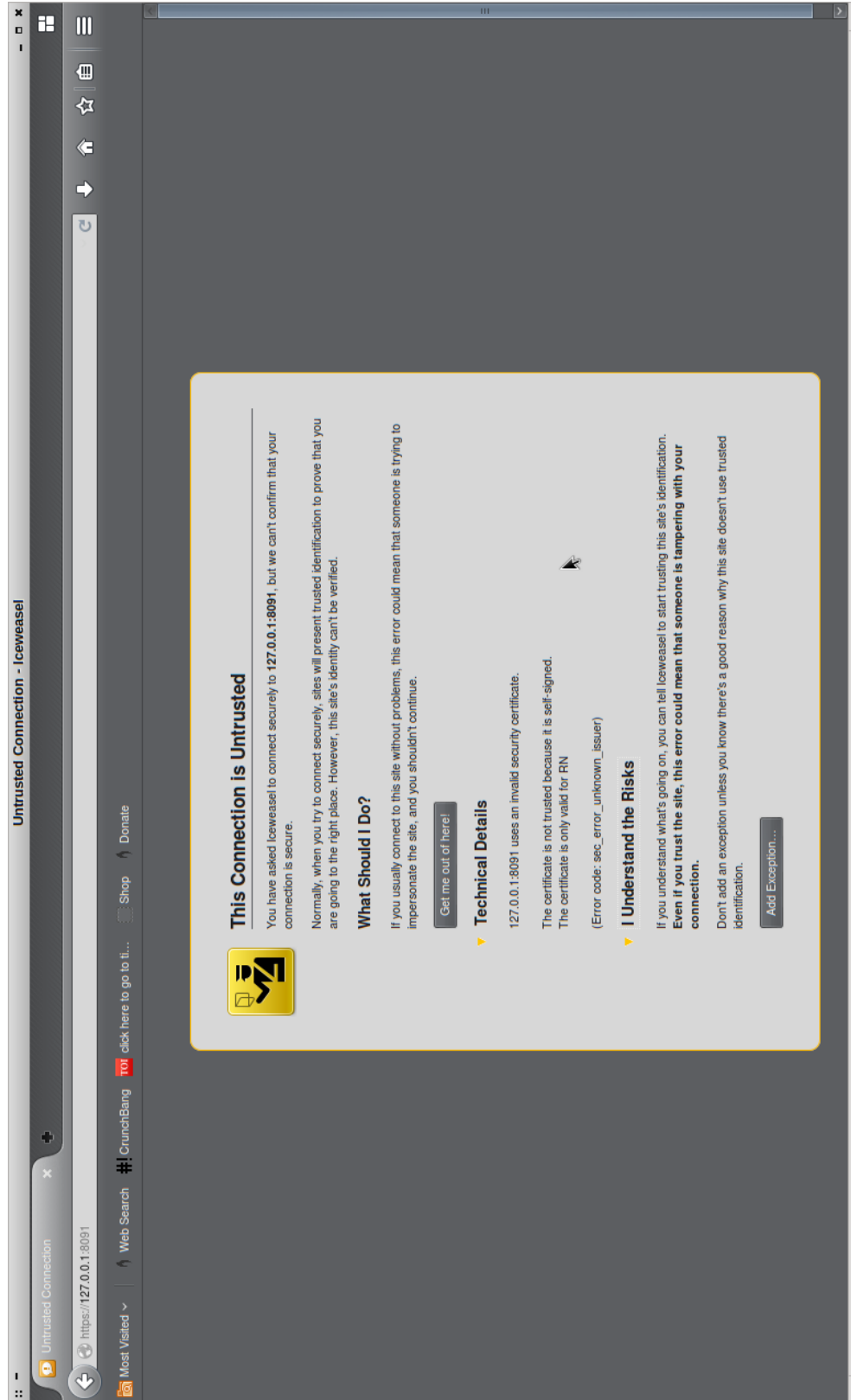


Figure 39: TLS Certificate Details - 2

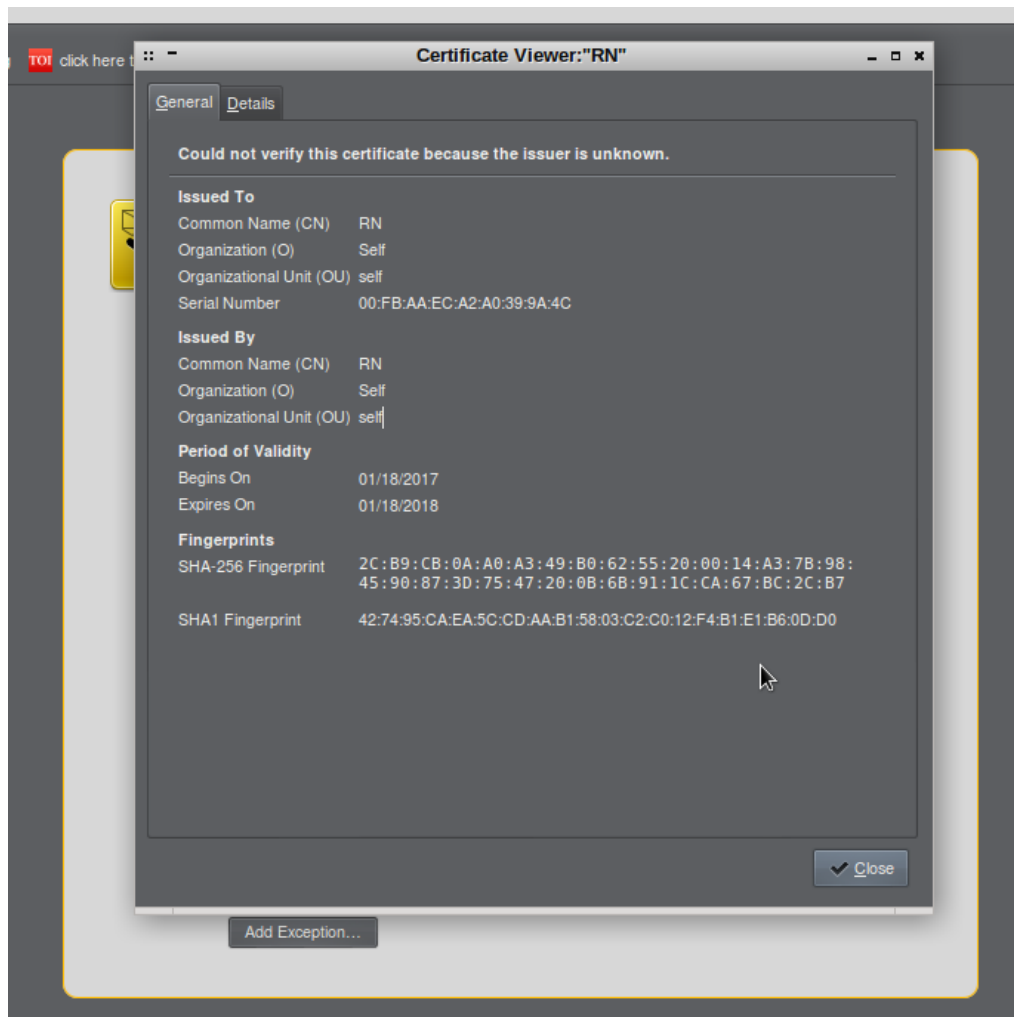


Figure 40: TLS Certificate Details - 3

The Transport Layer Security protocol aims primarily to provide privacy and data integrity between two communicating computer applications. When secured by TLS, connections between a client (e.g., a web browser) and a server have one or more of the following properties:

- The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret negotiated at the start of the session. The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted. The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).
- The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).

- The connection ensures integrity because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

In addition to the properties above, careful configuration of TLS can provide additional privacy-related properties such as forward secrecy, ensuring that any future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.

The TLS protocol comprises two layers: the TLS record protocol and the TLS handshake protocol.

Since applications can communicate either with or without TLS (or SSL), it is necessary for the client to indicate to the server the setup of a TLS connection. One of the main ways of achieving this is to use a different port number for TLS connections, for example port 443 for HTTPS.

Once the client and server have agreed to use TLS, they negotiate a stateful connection by using a handshaking procedure. The protocols use a handshake with an asymmetric cipher to establish cipher settings and a shared key for a session; the rest of the communication is encrypted using a symmetric cipher and the session key. During this handshake, the client and server agree on various parameters used to establish the connection's security:

- The handshake begins when a client connects to a TLS-enabled server requesting a secure connection and presents a list of supported cipher suites (ciphers and hash functions).
- From this list, the server picks a cipher and hash function that it also supports and notifies the client of the decision.
- The server usually then sends back its identification in the form of a digital certificate. The certificate contains the server name, the trusted Certificate Authority (CA) and the server's public encryption key.
- The client confirms the validity of the certificate before proceeding.
- To generate the session keys used for the secure connection, the client either:
 - encrypts a random number with the server's public key and sends the result to the server (which only the server should be able to decrypt with its private key); both parties then use the random number to generate a unique session key for subsequent encryption and decryption of data during the session
 - uses Diffie-Hellman key exchange to securely generate a random and unique session key for encryption and decryption that has the additional property of forward secrecy: if the server's private key is disclosed in future, it cannot be used to decrypt the current

session, even if the session is intercepted and recorded by a third party.

This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the session key until the connection closes. If any one of the above steps fail, the TLS handshake fails, and the connection is not created.

Digital Certificates A digital certificate certifies the ownership of a public key by the named subject of the certificate, and indicates certain expected usages of that key. This allows others (relying parties) to rely upon signatures or on assertions made by the private key that corresponds to the certified public key.

TLS typically relies on a set of trusted third-party certificate authorities to establish the authenticity of certificates. Trust is usually anchored in a list of certificates distributed with user agent software and can be modified by the relying party. As a consequence of choosing X.509 certificates, certificate authorities and a public key infrastructure are necessary to verify the relation between a certificate and its owner, as well as to generate, sign, and administer the validity of certificates.

6 COST ESTIMATION OF THE PROJECT

Estimate Product Size and Resources

The correlation of program size with development time is only moderately good for engineering teams and organizations. However, for individual engineers, the correlation is generally quite high. Therefore, the PSP starts with engineers estimating the sizes of the products they will personally develop. Then, based on their personal size and productivity data, the engineers estimate the time required to do the work. In the PSP, these size and resource estimates are made with the PROBE method ⁷.

Size Estimating with PROBE PROBE stands for PROxy Based Estimating and it uses proxies or objects as the basis for estimating the likely size of a product. With PROBE, engineers first determine the objects required to build the product described by the conceptual design. Then they determine the likely type and number of methods for each object. They refer to historical data on the sizes of similar objects they have previously developed and use linear regression to determine the likely overall size of the finished product. The example object size data in figure below show the five size ranges the PSP uses for objects. Since object size is a function of programming style, the PROBE method shows engineers how to use the data on the programs they have personally developed to generate size ranges for their personal use. Once they have estimated the sizes of the objects, they used linear regression to estimate the total amount of code they plan to develop. To use linear regression, the engineers must have historical data on estimated versus actual program size for at least three prior programs.

C++ Object Sizes in LOC per Method					
Category	Very Small	Small	Medium	Large	Very Large
Calculation	2.34	5.13	11.25	24.66	54.04
Data	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Logic	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09
Text	3.75	8.00	17.07	36.41	77.66

Figure 41: C++ Object Size Data

Any reference material for Object Size Data for Go Programs was not found. Hence the C++ object size data was used in the estimation.

Size Measures : Lines of Code (LOC) Since the time it takes to develop a product is largely determined by the size of that product, when using the PSP, engineers first estimate the sizes of the products they plan to develop. Then, when they are done, they measure the sizes of the products they produced. This provides the engineers with the size data they need to make accurate size estimates. However, for these data to be useful, the size measure must correlate with the development time for the product. While lines of code (LOC) is the principal PSP size measure, any size measure can be used that provides a reasonable correlation between development time and product size. It should also permit automated measurement of actual product size.

The PSP uses the term “logical LOC” to refer to a logical construct of the programming language being used. Since there are many ways to define logical LOC, engineers must precisely define how they intend to measure LOC. In this project LOC is simply the actual number of lines of codes as provided by the IDE.

7 REPORTS

8 PROJECT CHART, GANTT CHART

There cannot be a concept of a The program (or project) evaluation and review technique (PERT) chart for software development work carried out by a single developer as there is no other option but to carry out the work sequentially. In that sense, the PERT chart and the Gantt chart would be the same. PSP however has methods are introduced in a series of seven process versions. These versions are labeled PSP0 through PSP3, and each version has a similar set of logs, forms, scripts, and standards, as shown in Figure. The process scripts define the steps for each part of the process, the logs and forms provide templates for recording and storing data, and the standards guide the engineers as they do the work.

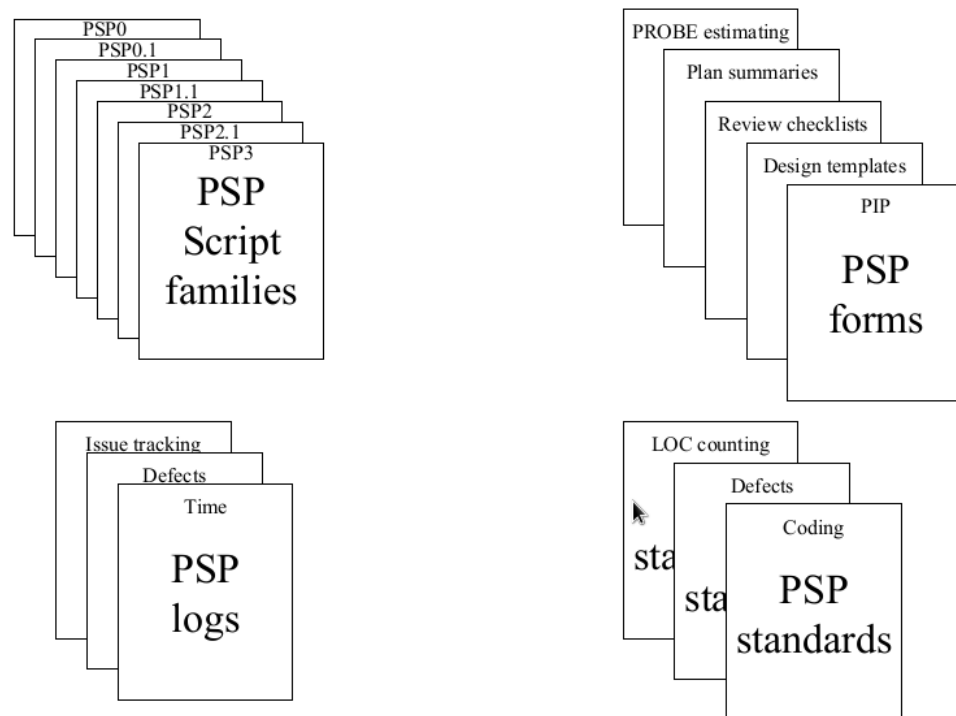


Figure 42: PSP Process Elements

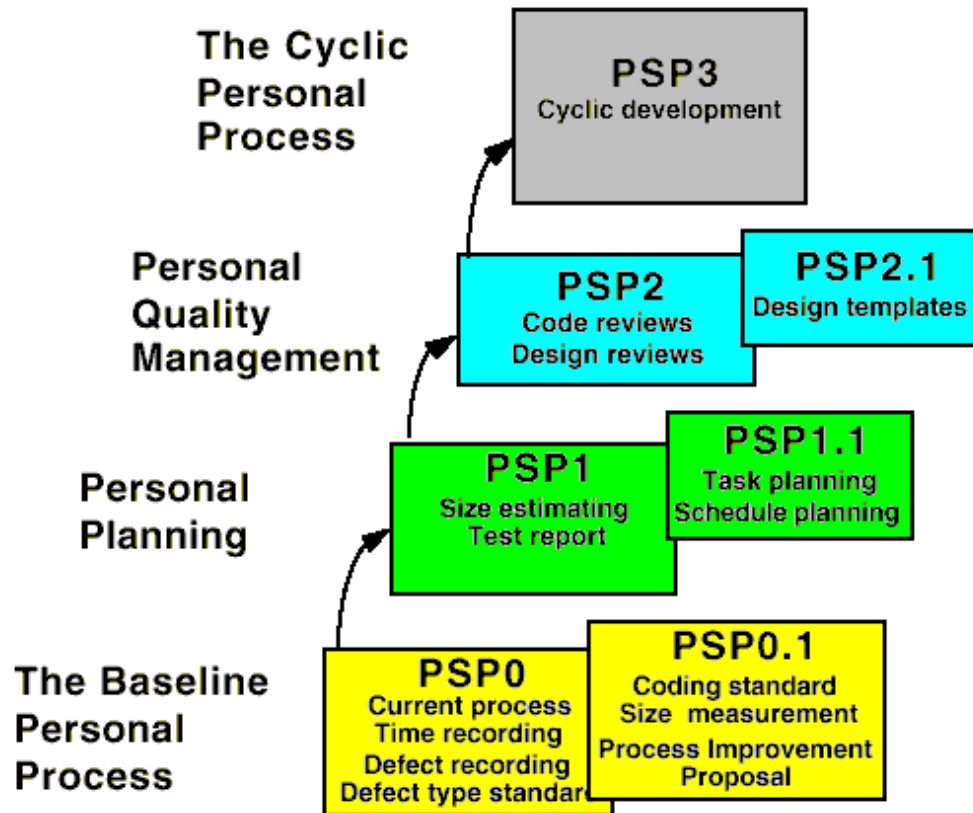
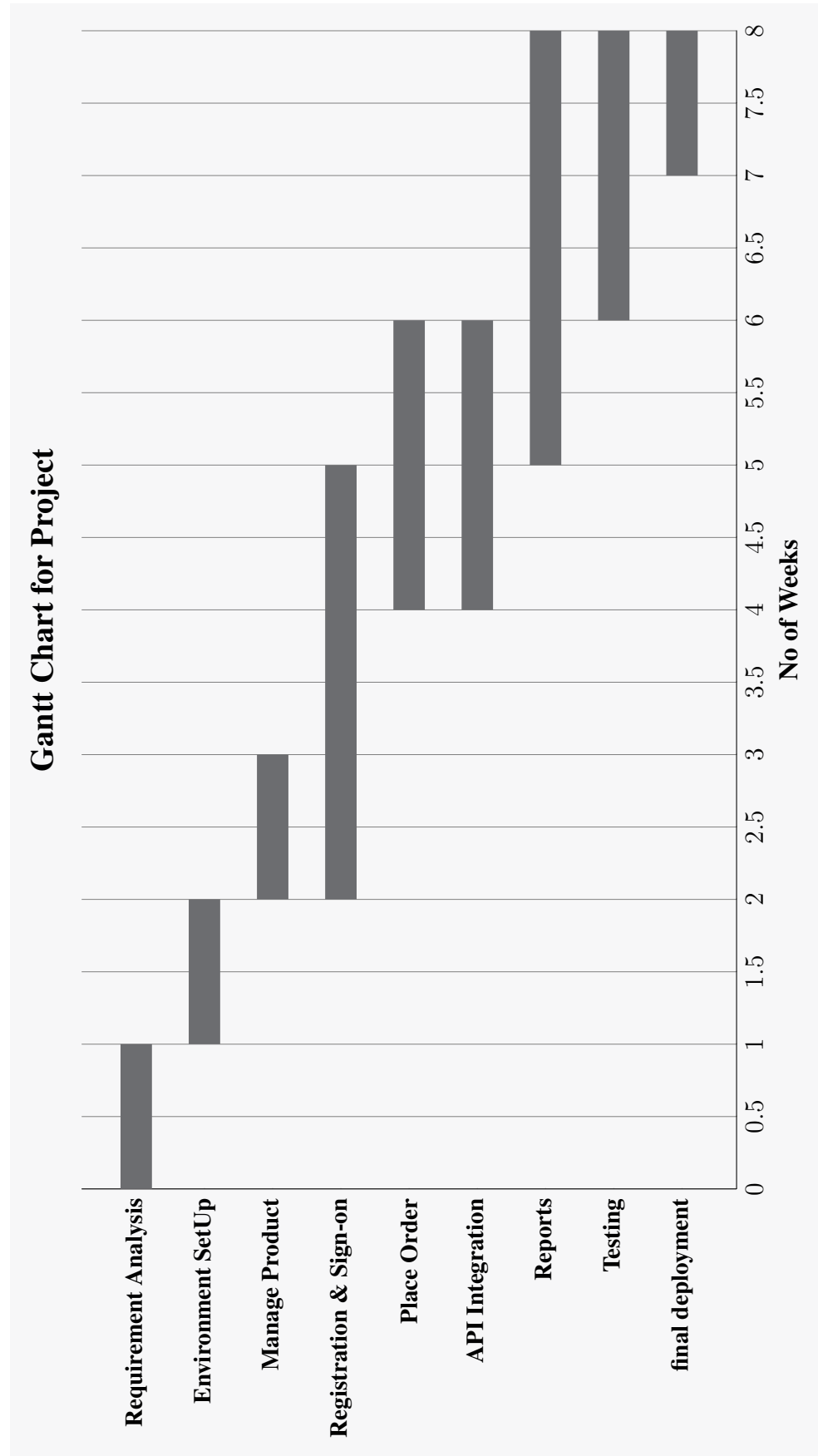


Figure 43: PSP Process Elements in the PSP Process



9 FUTURE SCOPE AND FURTHER ENHANCEMENT

The existing website can be developed further by adding more functionalities viz.,

- Forgot Password functionality that allows users to reset password
- Single-Sign On
- Use of apt Keywords in Meta-tags that allow fo better results in Google Search
- Filter Products
- Persistent Cart
- Cancel Order
- Delete Product
- Update Product
- Add more Product attributes
- Different Modes of Payment Like Digital Wallets
- Responsive Web Design (RWD)
- Location services based on Wifi or GPS
- Rate & Review Product
- Chat Support / Live Help
- Email Support

10 BIBLIOGRAPHY

- [1] https://en.wikipedia.org/wiki/Online_shopping
- [2] <http://www.indiaretailing.com/2016/08/30/retail/online-shopping-trends-facts-figures-on-indian-e-comm-sector>
- [3] https://en.wikipedia.org/wiki/Personal_software_process
- [4] <https://talks.golang.org/2012/splash.article>
- [5] https://en.wikipedia.org/wiki/User_interface_design
- [6] https://en.wikipedia.org/wiki/Transport_Layer_Security
- [7] Pittsburgh, PA 15213-3890. The Personal. Software. Process. SM. (PSP. SM.) CMU/SEI-2000-TR-022. ESC-TR-2000-022. Watts S. Humphrey. November 2000.

11 APPENDICES

List of Figures

Figure 1	PSP Process Flow Diagram	26
Figure 2	PSP Planning Process Diagram	26
Figure 3	Process Flow Diagram	31
Figure 4	Ordering Process Flow Diagram	32
Figure 5	Shopping Cart Process Flow Diagram	33
Figure 6	Shopping Cart Activity Flow Diagram	34
Figure 7	Website Component Diagram	35
Figure 8	Data Flow Diagram	36
Figure 9	Architecture Diagram	36
Figure 10	List of Database Tables	39
Figure 11	Utility Tables with data Elements	39
Figure 12	Data Elements type and size	40
Figure 13	Order table	40
Figure 14	Product table	41
Figure 15	User table	41
Figure 16	Orderdetails Field JSON Structure	42
Figure 17	Shipdetails Field JSON Structure	42
Figure 18	JSON Array View Order	43
Figure 19	Order table viewed in SQLite Manager	44
Figure 20	Customer Login Screen	46
Figure 21	Customer Login Screen - Alert	48
Figure 22	Customer Sign-up Screen	49
Figure 23	Customer Sign-up Screen Validations	51
Figure 24	Customer Login Main View	52
Figure 25	Customer Login Main View - Search	55
Figure 26	Customer Login Main View - Alert	56
Figure 27	Customer Login Main View - Shopping Cart	57
Figure 28	Customer Place Order Screen	59
Figure 29	Stripe Payment Gateway Submit Form-1	61
Figure 30	Stripe Payment Gateway Submit Form-2	64
Figure 31	Order Confirmation Screen	65
Figure 32	Email Notification in Inbox	68
Figure 33	Admin User Main View	69
Figure 34	Admin - Manage Products	71
Figure 35	Admin - Add Product Form	74
Figure 36	Admin - Add Product Form Validations - I	77
Figure 37	Admin - Add Product Form Validation -II	78
Figure 38	TLS Certificate Details - 1	92
Figure 39	TLS Certificate Details - 2	93
Figure 40	TLS Certificate Details - 3	94
Figure 41	C++ Object Size Data	97
Figure 42	PSP Process Elements	100
Figure 43	PSP Process Elements in the PSP Process	101

List of Tables

Table 2	Software & Hardware Requirements	5
Table 2	Requirements - User Module - Registration	10
Table 2	Requirements - User Module - Registration	11
Table 2	Requirements - User Module - Product Listing & Search	12
Table 2	Requirements - User Module - Shopping Cart	14
Table 2	Requirements - User Module - Place Order	15
Table 2	Requirements - User Module - Make Payment	17
Table 2	Requirements - User Module - Invoice	19
Table 2	Requirements - User Module - Invoice	20
Table 2	Requirements - Admin Module - View Order Data	21
Table 2	Requirements - Admin Module - Manage Products	22
Table 2	Requirements - Admin Module - Add Products	23
Table 2	Software & Hardware Requirements	37
Table 2	Customer Login Screen	47
Table 2	Customer Signup Screen	50
Table 2	Customer Login Main View	53
Table 2	Customer Login Main View	58
Table 2	Customer Place Order Screen	60
Table 2	Stripe Payment Gateway Submit Form-1	62
Table 2	Order Invoice / Payment Receipt	66
Table 2	Admin Main View - Order data	70
Table 2	Admin Manage Products	72
Table 2	Admin Manage Products	75
Table 2	Test Cases for Req. ID FR1	81
Table 2	Test Cases for Req. ID FR2	81
Table 2	Test Cases for Req. ID FR3	81
Table 2	Test Cases for Req. ID FR4	81
Table 2	Test Cases for Req. ID FR5	81
Table 2	Test Cases for Req. ID FR6	82
Table 2	Test Cases for Req. ID FR7	82
Table 2	Test Cases for Req. ID FR8	82
Table 2	Test Cases for Req. ID FR9	82
Table 2	Test Cases for Req. ID FR10	83
Table 2	Test Cases for Req. ID FR11	83
Table 2	Test Cases for Req. ID FR12	83
Table 2	Test Cases for Req. ID FR13	83
Table 2	Test Cases for Req. ID FR14	84
Table 2	Test Cases for Req. ID FR11	84
Table 2	Test Cases for Req. ID FR15	84
Table 2	Test Cases for Req. ID FR16	84
Table 2	Test Cases for Req. ID FR17	85
Table 2	Test Cases for Req. ID FR18	85
Table 2	Test Cases for Req. ID FR19	85
Table 2	Test Cases for Req. ID FR20	85
Table 2	Test Cases for Req. ID FR21	86
Table 2	Test Cases for Req. ID FR22	86

Table 2	Test Cases for Req. ID FR23	86
Table 2	Test Cases for Req. ID FR24	86
Table 2	Test Cases for Req. ID FR25	87
Table 2	Test Cases for Req. ID FR26	87
Table 2	Test Cases for Req. ID FR27	87
Table 2	Test Cases for Req. ID FR28	87
Table 2	Test Cases for Req. ID FR29	88
Table 2	Test Cases for Req. ID FR30	88
Table 2	Test Cases for Req. ID FR31	88
Table 2	Test Cases for Req. ID FR32	88
Table 2	Test Cases for Req. ID FR33	89
Table 2	Test Cases for Req. ID FR34	89
Table 2	Test Cases for Req. ID FR35	89
Table 2	Test Cases for Req. ID FR36	89
Table 2	Test Cases for Req. ID FR37	90
Table 2	Test Cases for Req. ID FR38	90
Table 2	Test Cases for Req. ID FR39	90

12 GLOSSARY

Glossary

4G 4th Generation (4G) is a mobile communications standard intended to replace 3G, allowing wireless Internet access at a much higher speed.. 4

Algorithm a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.. 71

API In computer programming, an application programming interface (API) is a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.. 7, 8, 17, 34, 50, 52

asymmetric cipher *See* public-key cryptography. 71

Authentication the process or action of verifying the identity of a user or process.. 12

B2C Business-to-consumer (B2C) is an Internet and electronic commerce (e-commerce) model that denotes a financial transaction or online sale between a business and consumer.. 1

Broadband a high-capacity transmission technique using a wide range of frequencies, which enables a large number of messages to be communicated simultaneously.. 4

CA Certificate Authorities (CA) issue Digital Certificates. A certificate authority (CA) is a trusted entity that issues electronic documents that verify a digital entity's identity on the Internet. The electronic documents, which are called digital certificates, are an essential part of secure communication and play an important part in the public key infrastructure (PKI).. 71

Client (in a network) a desktop computer or workstation that is capable of obtaining information and applications from a server.. 71

Closures In computer science, a closure is a function that has an environment of its own. Inside this environment, there is at least one bound variable.. 28

CMM The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process. The model describes a five-level evolutionary path of increasingly organized and systematically more mature processes.. 25

Compiler a program that converts instructions into a machine-code or lower-level form so that they can be read and executed by a computer. 28

Composition In computer science, function composition is an act or mechanism to combine simple functions to build more complicated ones. Object composition is a way to combine simple objects or data types into more complex ones.. 30

Computer Network a set of computers connected together for the purpose of sharing resources.. 1, 68

Concurrency In computer science, concurrency is the execution of several instruction sequences at the same time. In an operating system, this happens when there are several process threads running in parallel. These threads may communicate with each other through either shared memory or message passing.. 30

Correlation A correlation is a single number that describes the degree of relationship between two variables. 73

Cryptographic protocols A security protocol (cryptographic protocol or encryption protocol) is an abstract or concrete protocol that performs a security-related function and applies cryptographic methods, often as sequences of cryptographic primitives.. 68

Decryption Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. This term could be used to describe a method of un-encrypting the data manually or with un-encrypting the data using the proper codes or keys.. 72

Diffie-Hellman key exchange Diffie–Hellman key exchange (D–H) is a specific method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as originally conceptualized by Ralph Merkle and named after Whitfield Diffie and Martin Hellman.. 72

Digital Certificate In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic document used to prove the ownership of a public key.. 71

e-commerce *See* Electronic Commerce (also e-commerce). 4

EDI Electronic Data Interchange (EDI) is the electronic interchange of business information using a standardized format; a process which allows one company to send information to another company electronically rather than with paper.. 1

Electronic Commerce (also **e-commerce**) commercial transactions conducted electronically on the Internet.. 1

Email messages distributed by electronic means from one computer user to one or more recipients via a network.. 1, 34, 40

Encryption the process of converting information or data into a code, especially to prevent unauthorized access.. 71

Error handling Error handling refers to the anticipation, detection, and resolution of programming, application, and communications errors.. 30

First-class functions Specifically, this means the language supports passing functions as arguments to other functions, returning them as the values from other functions, and assigning them to variables or storing them in data structures. It means that functions are objects, with a type and a behaviour. They can be dynamically built, passed around as any other object, and the fact that they can be called is part of their interface.. 28

Forward Secrecy In cryptography, forward secrecy (FS; also known as perfect forward secrecy) is a property of secure communication protocols in which compromise of long-term keys does not compromise past session keys. Forward secrecy protects past sessions against future compromises of secret keys or passwords.. 71, 72

Garbage collection the automatic process of making space in a computer's memory by removing data that is no longer required or in use.. 27

Go-live (of a computer system) become operational.. 4

Handshake The handshaking process usually takes place in order to establish rules for communication when a computer sets about communicating with a foreign device. When a computer communicates with another device like a modem, printer, or network server, it needs to handshake with it to establish a connection.. 71

Hash functions A hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes.. 71

Hosting store (a website or other data) on a server or other computer so that it can be accessed over the Internet.. 8

HTTPS HTTPS (HTTP over SSL or HTTP Secure) is the use of Secure Socket Layer (SSL) or Transport Layer Security (TLS) as a sublayer under regular HTTP application layering. HTTPS encrypts and decrypts user page requests as well as the pages that are returned by the Web server.. 12, 71

i686 i686 is widely used to describe 32-bit P6 processor architecture which is compatible with Pentium Pro/II and has it's instruction set.. 5

Identifier An identifier is the user-defined name of a program element. It can be a namespace, class, method, variable or interface. Identifiers are symbols used to uniquely identify a program element in the code. They are also used to refer to types, constants, macros and parameters.. 28

- Inheritance** In object-oriented programming, inheritance enables new objects to take on the properties of existing objects.. 30
- Internet** a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.. 1
- Internet Marketing** Internet marketing, or online marketing, refers to advertising and marketing efforts that use the Web and email to drive direct sales via electronic commerce, in addition to sales leads from Web sites or emails.. 1
- Inventory Management** Inventory management is the supervision of non-capitalized assets (inventory) and stock items. A component of supply chain management, inventory management supervises the flow of goods from manufacturers to warehouses and from these facilities to point of sale.. 1
- Linear regression** In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X .. 73
- Linker** a program used with a compiler or assembler to provide links to the libraries needed for an executable program.. 28
- Linux** an open-source operating system modelled on UNIX.. 5
- LOC** Source lines of code (SLOC), also known as lines of code (LOC), is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code.. 73
- Marketplace** the arena of commercial dealings. An online marketplace (or on-line e-commerce marketplace) is a type of e-commerce site where product or service information is provided by multiple third parties, whereas transactions are processed by the marketplace operator.. 2
- Message Authentication Code** In cryptography, a message authentication code (MAC) is a short piece of information used to authenticate a message—in other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed in transit (its integrity).. 71
- Mobile Commerce** M-commerce (mobile commerce) is the buying and selling of goods and services through wireless handheld devices such as cellular telephone and personal digital assistants (PDAs).. 1
- Namespaces** A namespace in computer science (sometimes also called a name scope), is an abstract container or environment created to hold a logical grouping of unique identifiers or symbols.. 28
- Network security** Network security is protection of the access to files and directories in a computer network against hacking, misuse and unauthorized changes to the system.. 3

Object file An object file is a file containing object code, meaning relocatable format machine code that is usually not directly executable.. 28

Online Transaction Processing Online transaction processing, or OLTP, is a class of information systems that facilitate and manage transaction-oriented applications, typically for data entry and retrieval transaction processing.. 1

OOP Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.. 3

Open source denoting software for which the original source code is made freely available and may be redistributed and modified.. 27

OpenSSL OpenSSL is a software library to be used in applications that need to secure communications over computer networks against eavesdropping or need to ascertain the identity of the party at the other end.. 68

Operating System the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.. 7

Payment Gateway A payment gateway is a merchant service provided by an e-commerce application service provider that authorizes credit card or direct payments processing for e-businesses, online retailers, bricks and clicks, or traditional brick and mortar.. 7, 8, 17, 34, 50, 52

Paypal Paypal is an electronic commerce (e-commerce) company that facilitates payments between parties through online funds transfers.. 1

Persistent Persistent shopping carts save a customer's cart contents across sessions through "persistent cookies." A cookie is a small text file stored on a user's computer. In computer science, persistence refers to the characteristic of state that outlives the process that created it. This is achieved in practice by storing the state as data in computer data storage. Programs have to transfer data to and from storage devices and have to provide mappings from the native programming-language data structures to the storage device data structures. 13

Private Key A private key is a tiny bit of code that is paired with a public key to set off algorithms for text encryption and decryption. It is created as part of public key cryptography during asymmetric-key encryption and used to decrypt and transform a message to a readable format.. 72

PROBE Proxy-Based Estimating (PROBE) is an estimating process used in the Personal Software Process (PSP) to estimate size and effort.. 73

PSP The Personal Software Process (PSP) is a structured software development process that is intended (planned) to help software engineers better understand and improve their performance by tracking their predicted and actual development of code.. 8, 25, 73

RDBMS Relational Database management System (RDBMS) is a database management system based on relational model defined by E.F.Codd. Data is stored in the form of rows and columns. The relations among tables are also stored in the form of the table.. 3

Real-time transaction processing In a real-time processing system, transactions are processed immediately as they occur without any delay to accumulate transactions.. 1

Reflection In computer science, reflection is the ability of a computer program to examine, introspect, and modify its own structure and behavior at run-time.. 29

Search Engine a program that searches for and identifies items in a database that correspond to keywords or characters specified by the user, used especially for finding particular sites on the World Wide Web.. 1

Session the period of activity between a user logging in and logging out of a (multi-user) system.. 12

Session key A session key is an encryption and decryption key that is randomly generated to ensure the security of a communications session between a user and another computer or between two computers. Session keys are sometimes called symmetric keys, because the same key is used for both encryption and decryption.. 71

Social Media websites and applications that enable users to create and share content or to participate in social networking.. 4

Social Networks a dedicated website or other application which enables users to communicate with each other by posting information, comments, messages, images, etc.. 1

Software Requirement Specifications store (A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.. 10

SSL Secure Sockets Layer (SSL) is the standard security technology for establishing an encrypted link between a web server and a browser.. 1

Stateful Connection A stateful connection is one in which some information about a connection between two systems is retained for future use. 71

Statically typed Enforcement of type rules at compile time rather than at run time. Static typing catches more errors at compile time than dynamic typing.. 27

Supply Chain Management Supply chain management (SCM) is the oversight of materials, information, and finances as they move in a process from supplier to manufacturer to wholesaler to retailer to consumer.. 1

Symbol table In computer science, a symbol table is a data structure used by a language translator such as a compiler or interpreter, where each identifier (a.k.a. symbol) in a program's source code is associated with information relating to its declaration or appearance in the source.. 28

Symmetric cipher Symmetric ciphers are the oldest and most used cryptographic ciphers. In a symmetric cipher, the key that deciphers the ciphertext is the same as (or can be easily derived from) the key enciphers the clear text. This key is often referred to as the secret key.. 71

TLS Transport layer security (TLS) is a protocol that provides communication security between client/server applications that communicate with each other over the Internet.. 3, 68

TLS handshake protocol The Transport Layer Security (TLS) Handshake Protocol is responsible for the authentication and key exchange necessary to establish or resume secure sessions.. 71

TLS record protocol The Transport Layer Security (TLS) Record protocol secures application data using the keys created during the Handshake.. 71

TSP In combination with the personal software process (PSP), the team software process (TSP) provides a defined operational process framework that is designed to help teams of managers and engineers organize projects and produce software products that range in size from small projects of several thousand lines of code (KLOC) to very large projects greater than half a million lines of code. 25

Type information In computer programming, run-time type information refers to a mechanism that exposes information about an object's data type at runtime. Run-time type information can apply to simple data types, such as integers and characters, or to generic types.. 28

Type-safe Type-Safe is code that accesses only the memory locations it is authorized to access, and only in well-defined, allowable ways.. 28

URL A URL (Uniform Resource Locator), as the name suggests, provides a way to locate a resource on the web, the hypertext system that operates over the internet.. 28

UUID A UUID (Universal Unique Identifier) is a 128-bit number used to uniquely identify some object or entity on the Internet.. 24

Variadic functions In computer science, an operator or function is variadic if it can take a varying number of arguments; that is, if its arity is not fixed.. 28

Web Browser a computer program with a graphical user interface for displaying HTML files, used to navigate the World Wide Web.. 1

Web design Web design is the process of creating websites. It encompasses several different aspects, including webpage layout, content production, and graphic design. While the terms web design and web development are often used interchangeably, web design is technically a subset of the broader category of web development.. 3

Web security Web application security is a branch of Information Security that deals specifically with security of websites, web applications and web services. At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems.. 3

Web server A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients.. 8

World Wide Web an information system on the Internet which allows documents to be connected to other documents by hypertext links, enabling the user to search for information by moving from one document to another.. 1

X.509 certificates In cryptography, X.509 is an important standard for a public key infrastructure (PKI) to manage digital certificates and public-key encryption and a key part of the Transport Layer Security protocol used to secure web and email communication.. 72