

# ***Software Requirement and Design Specifications***

## **LIGHTHOUSE – ECOMMERCE WEBAPP**

Version: 1.0



Course Code	CS-2005 CL-2005
Instructor	Ms. Eman Shahid Ms. Mafaza Mohi
Project Team	Bilal Arshad (19K1381) Abdul Rafay (19K1316)
Submission Date	23-Dec-2021

## Table of Contents

<b>1. INTRODUCTION .....</b>	<b>3</b>
1.1. Purpose of Document .....	3
1.2. Intended Audience .....	3
1.2. Intended Audience .....	3
<b>2. OVERALL SYSTEM DESCRIPTION.....</b>	<b>4</b>
2.1. Project Background.....	4
2.2. Project Scope .....	4
2.3. Not In Scope .....	4
2.4. Project Objectives .....	4
2.5. Stakeholders .....	4
2.6. Operating Environment .....	4
2.7. System Constraints .....	5
2.8. Assumptions & Dependencies .....	5
<b>3. EXTERNAL INTERFACE REQUIREMENTS.....</b>	<b>6</b>
3.1. Hardware Interfaces .....	6
3.2. Software Interfaces .....	6
3.3. Communications Interfaces .....	6
<b>4. FUNCTIONAL REQUIREMENTS .....</b>	<b>7</b>
4.1. FUNCTIONAL HIERARCHY.....	7
4.2. Use Cases .....	7
4.2.1. Login/SignIn .....	8
4.2.2. Purchase Products .....	8
4.2.3. Review Purchased Product .....	9
<b>5. NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>10</b>
5.1. Performance Requirements .....	10
5.2. Safety Requirements .....	10
5.3. Security Requirements .....	10
<b>Software Development Specification .....</b>	<b>11</b>
<b>6. SYSTEM ARCHITECTURE .....</b>	<b>12</b>
6.1. SYSTEM LEVEL ARCHITECTURE .....	12
6.2. SOFTWARE ARCHITECTURE .....	13
<b>7. DESIGN STRATEGY .....</b>	<b>14</b>
<b>8. DETAILED SYSTEM DESIGN .....</b>	<b>15</b>
8.1. DATABASE DESIGN .....	15
8.1. DATABASE DESIGN .....	16
<b>9. APPLICATION DESIGN .....</b>	<b>18</b>
9.1.1. Sequence .....	19
9.1.2. State .....	21
9.1.3. Activity .....	22
9.1.4. Use Cases .....	23
<b>10. REFERENCES .....</b>	<b>23</b>

# 1. Introduction

## 1.1 Purpose of document

To show the development view, internal and external view of project.

## 1.2 Intended Audience

Targeted Audience is from 20-40 years old, but can be used by any age and group.

## 1.3 Document Convention

Font size for body: 10px

**Font size for Heading: 12px**

Font: Arial

## **2. Overall System Description:**

### **2.1 Project Background**

Idea for this project was inspired by the fact that the biggest trending in the world of 2021 is Crypto, then Ecommerce, and then Social Media Marketing. So, when the entire world is shifting towards a change; you can't ignore it, so we made a project in Ecommerce. LightHouse is created solely for selling lights on global level by utilizing inventory with Ali Express and delivery via FedEx. But, to start off, at the beginning we are focusing on Pakistan.

### **2.2 Project Scope**

LightHouse will serve as an ecommerce platform to the World for Lights.

- 2.2.1 Account Signin/Login
- 2.2.2 Customers can check their previous purchases
- 2.2.3 Review and rate previous purchases
- 2.2.4 Track their orders Dynamically
- 2.2.5 Make a return of a previous purchase
- 2.2.6 Search and filter products
- 2.2.7 Payment Processing
- 2.2.8 Promotion via News Letter
- 2.2.9 Seasonal Coupons for discounts
- 2.2.10 Add products to Cart
- 2.2.11 Edit Cart
- 2.2.12 Proceed to checkout with cart
- 2.2.13 Handle Abandoned Cart
- 2.2.14 Multiple delivery options

### **2.3 Project Objectives**

Projects is dedicated to provide an ecommerce platform to the Pakistani market that specializes in all sorts of lights.

### **2.4 Stakeholders**

Business Owner/Executives/Admin  
Software developers/engineers  
Industrial person with experience in Lights industry expert  
Ecommerce Mentor  
Marketing expert  
General People

### **2.5 Operating Environment**

System is hosted on the web-server and utilizes Chromium as its main OS.

## 2.6 System Constraints

### **Cultural Constraint**

Pakistani Market is familiar with traditional hardware stores and can get most of their lights from there, so we need to market our products well, and provide such products that don't exist in the traditional market.

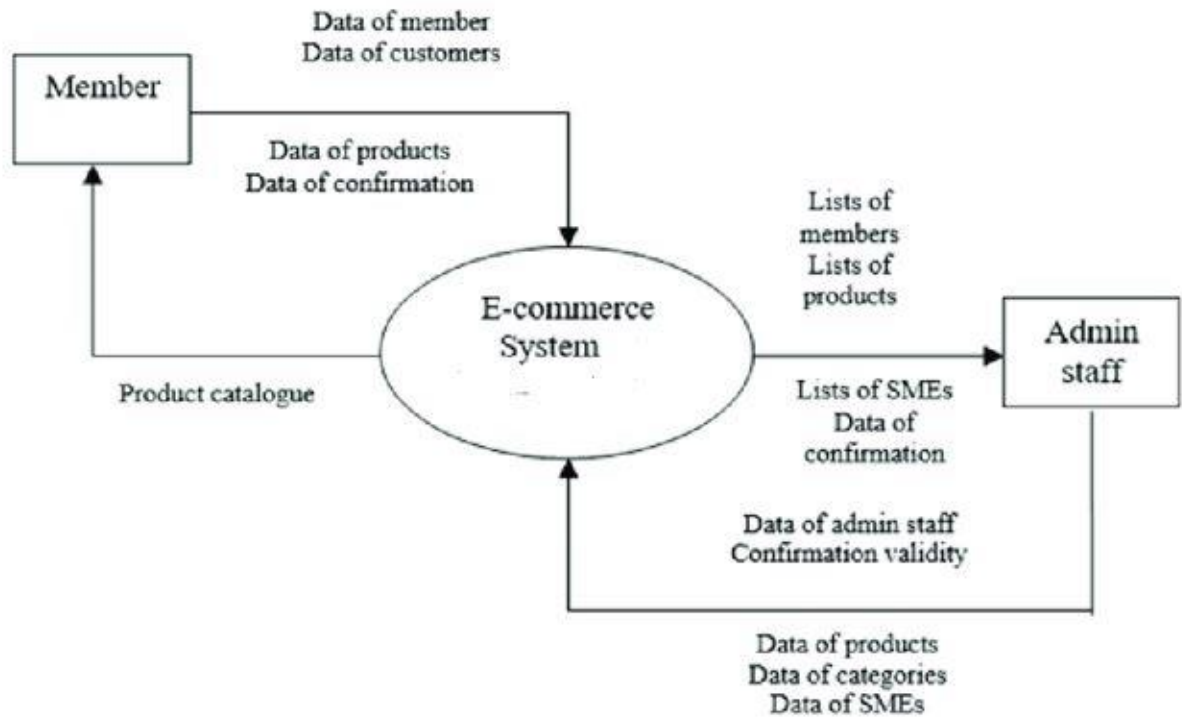
### **Software Constraints**

Internet is needed to access the online store, and updated versions of Chrome, Edge, Opera is required to access the latest and advance features of the software.

## 2.7 Assumptions and Dependencies

Dependent on internet, and the hosted server.

### 3. External Interface Requirements



#### 3.1 Hardware Interfaces

Project requires a computer of any sort from smart phone to laptops to desktop PCs, and an active internet connection.

#### 3.2 Software Interfaces

Software is built on React to manage component and plugin/plugin out system which is backed by NodeJS. For the UI/UX, HTML and CSS along with Bootstrap, Material UI and Animations via AOS were used. All the data is kept into a vault of MySQL, and the data is transferred via Restful APIs. Only previous purchases data is shared to the audience and that data is also restricted to their purchases. Products data is shared on the landing page, and Login/SignIn page utilizes User phone number and emails to identify a genuine user from a fake user.

#### 3.3 Communication Interfaces

Login/Sign in page uses data of previous/existing users to identify genuine users, and each user is prompted to sign in before making a purchase. Sign in form asks for email, phone number, first name, last name, billing address, password, and city to maintain records, and the passwords are encrypted for security.

Proceeding to checkout button asks for payment details if the details are not already set, but the user can also go with Cash on Delivery method. Payment details form requires for card number, expiry and security code which are all encrypted.

SSL certificates along with encryption are used to prevent hackers from stealing information.

## 4 Functional Requirements

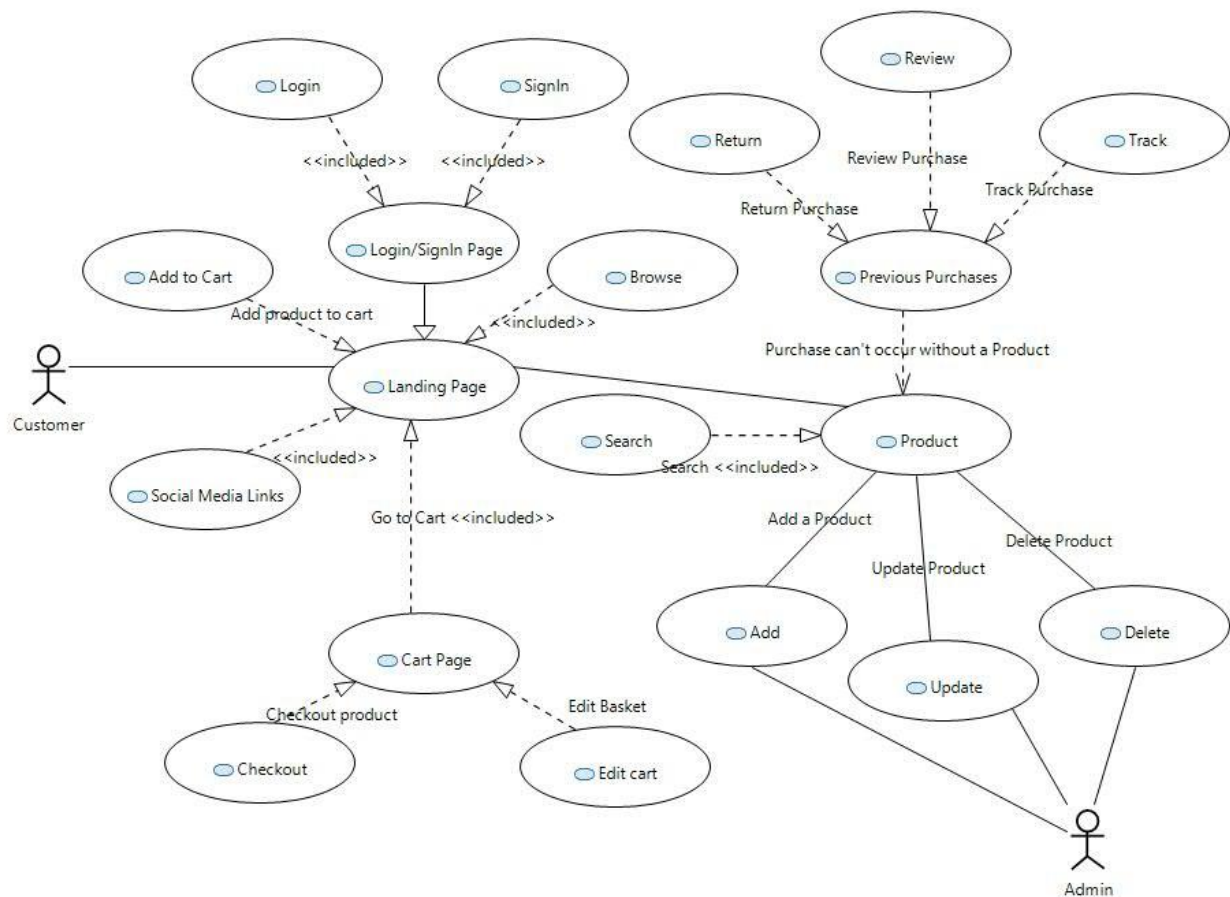
### 4.1 Functional hierarchy

There are three subsystems involved in this project, the first is the front end, Backend and then lastly Database. Each subsystem has its own components and functions.

Front-end has Header, Footer, Login page, Cart Page, Contact Us page, Your Orders page, Product as components and payments processing is a function used by cart page; main page also utilizes search function to sort/filter products.

Back-end has post, update, delete, get request functions for product, and SQL manages the records of customers and their purchases.

### 4.2 Use Cases



#### 4.2.1 Login/SignIn

Use Case Description	Customers can successfully sign up to become a registered User. After that they can log into their accounts to make purchases.
Use Case Name:	Login/Sign In
Primary Actor: Customer	Other Actors: None
Stake Holders:	Customer, admin
Relationships: <ul style="list-style-type: none"><li>• Includes: Login, Sign in</li><li>• Extends: Landing Page</li></ul>	
Pre-conditions: None	
Flow of Events: <ul style="list-style-type: none"><li>• Customer lands onto the main page</li><li>• Click on login page link</li><li>• Customer signs up</li><li>• Customer can now log in</li><li>• Customer returns to landing page by clicking on home</li></ul>	
Alternative and Exceptional flows: <ul style="list-style-type: none"><li>• Customer tries to login without signing up</li><li>• Customer uses already registered email to sign up</li><li>• Customer forgets password</li></ul>	
Post Conditions: None	

#### 4.2.2 Purchase Products

Use Case Description	Customer can purchase products enlisted on the website.
Use Case Name:	Landing, Product, Cart Page
Primary Actor: Customer	Other Actors: None
Stake Holders:	Customer, admin
Relationships: <ul style="list-style-type: none"><li>• Includes: Search, add to cart, edit cart, browse</li><li>• Extends: Landing Page, Product</li></ul>	
Pre-conditions: User must be logged in	



<p>Flow of Events:</p> <ul style="list-style-type: none"> <li>• Customer lands onto the main page after logging in</li> <li>• Click on add to cart button to add product</li> <li>• Customer clicks on cart icon to go to cart page</li> <li>• Customer can now edit cart</li> <li>• Customer proceeds to checkout</li> </ul>
<p>Alternative and Exceptional flows:</p> <ul style="list-style-type: none"> <li>• Customer tries to purchase without logging in</li> <li>• Customer tries to checkout empty cart</li> </ul>
<p>Post Conditions: None</p>

#### 4.2.3 Review Purchased Product

Use Case Description	Customers can successfully view their bought products, and review/rate them
Use Case Name:	Previous Purchases, Product, Landing Page
Primary Actor: Customer	Other Actors: None
Stake Holders:	Customer, admin
<p>Relationships:</p> <ul style="list-style-type: none"> <li>• Includes: Return, Review, Track</li> <li>• Extends: Product, Landing Page</li> </ul>	
<p>Pre-conditions:</p> <ul style="list-style-type: none"> <li>• User must be logged In</li> <li>• User must have bought atleast one product</li> </ul>	
<p>Flow of Events:</p> <ul style="list-style-type: none"> <li>• Customer lands onto the main page after logging in</li> <li>• Click on Your Orders</li> <li>• Customer clicks on review product</li> <li>• Customer can now choose to review product on their satisfaction</li> <li>• Customer returns to Your Orders page</li> </ul>	
<p>Alternative and Exceptional flows:</p> <ul style="list-style-type: none"> <li>• Customer tries to view Your Orders page without logging in</li> <li>• Customer hasn't bought a product</li> <li>• Customer has bought a product, but tries to review before delivery of product</li> </ul>	
<p>Post Conditions: None</p>	

## **5. Non-functional Requirements**

### **5.1. Performance Requirements**

The website must load within 3 seconds that's the standard for today's fast paced world, all the purchased orders must be shown on Your Orders page, and the Login page must always support a registered user. Webapp must be mobile friendly, for this a responsive design is must as most users are using mobiles to access websites. Website must be available to the customers 24/7 to provide seamless experience.

### **5.2 Safety Requirements**

The password and payments details must be kept encrypted from the initial front-end forms so that no one can listen to them at data transfer, not even admin. SSL certificates must be kept to ensure safety. Backups are required for the webapp to ensure little to none recovery time.

### **5.3. Security Requirements**

Users must be identified by their login details like email, and phone number to ensure proper authentication. SSL certificates must be used, and no sensitive info must be kept so as to avoid legal matters.

## **Software Development Specification**

## 6. System Architecture

System provides customers to sign in and log in, ability to add products to cart and purchase their cart via checking out, ability to return a purchased product or review a product, customers can also track their purchases status.

Front-end with React is responsible to create such UI/UX that user wants to stay on website for considerable time, making sure that every customer leaves with something either a product or an experience. Front-end will also implement forms and sensitive info will be encrypted to ensure no listeners gets any info from the initial stage. Each UI is implemented in the form of components, so header, footer, login, signup, product, etc every thing is a component so updates and scalability is at edge.

Back-end with NodeJS is responsible for the components functionality and communication with the database, every component has its various functions implemented in back-end file.

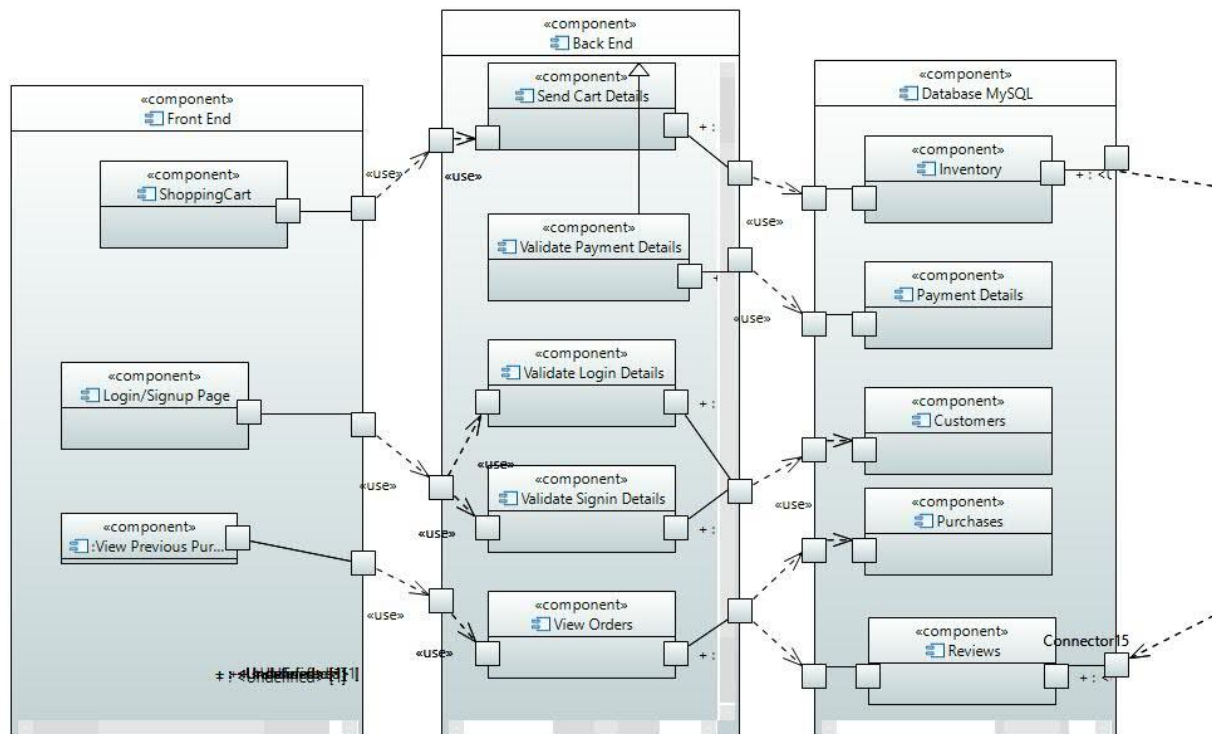
Database is implemented on MySQL, and every record of customer info is stored in customers info table, and their purchases in purchases table, similarly products, inventory, coupons all have their own tables to ensure 3NF and scalability.

### 6.1. System Level Architecture

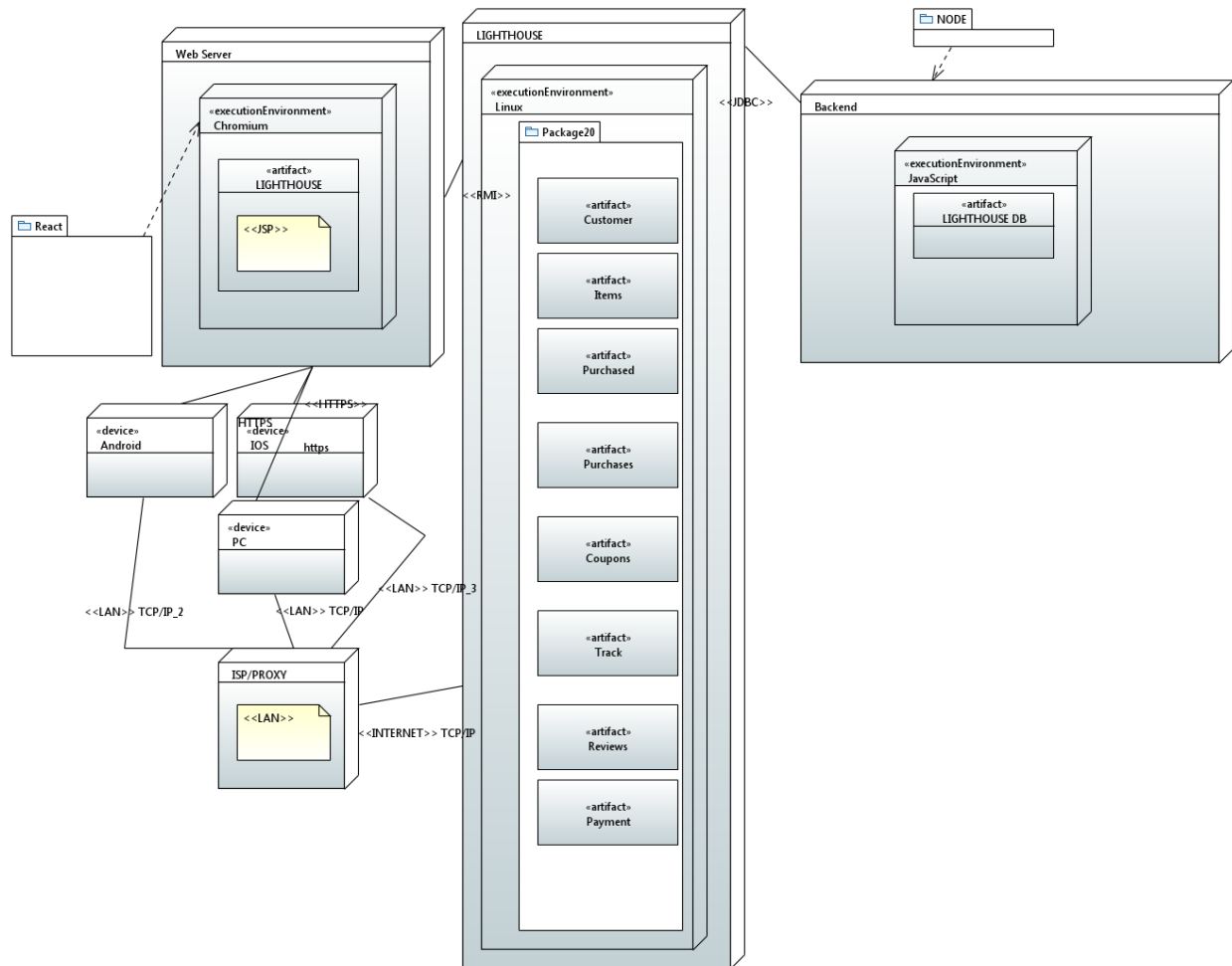
Every page has been decomposed into components to provide plug in and plug out functionality

- Landing Page has components Header, Footer, and Product; similarly Cart Page and Your Orders Page is also broken into multiple functions and components.
- Landing Page is related to Cart Page and Your Orders page.
- Webapp connects to webserver via web2 protocols
- The website has not been inspected by legal officers, and we do not offer our own delivery rather a third-party delivery method like FedEx.

### Component Diagram



## Deployment Diagram



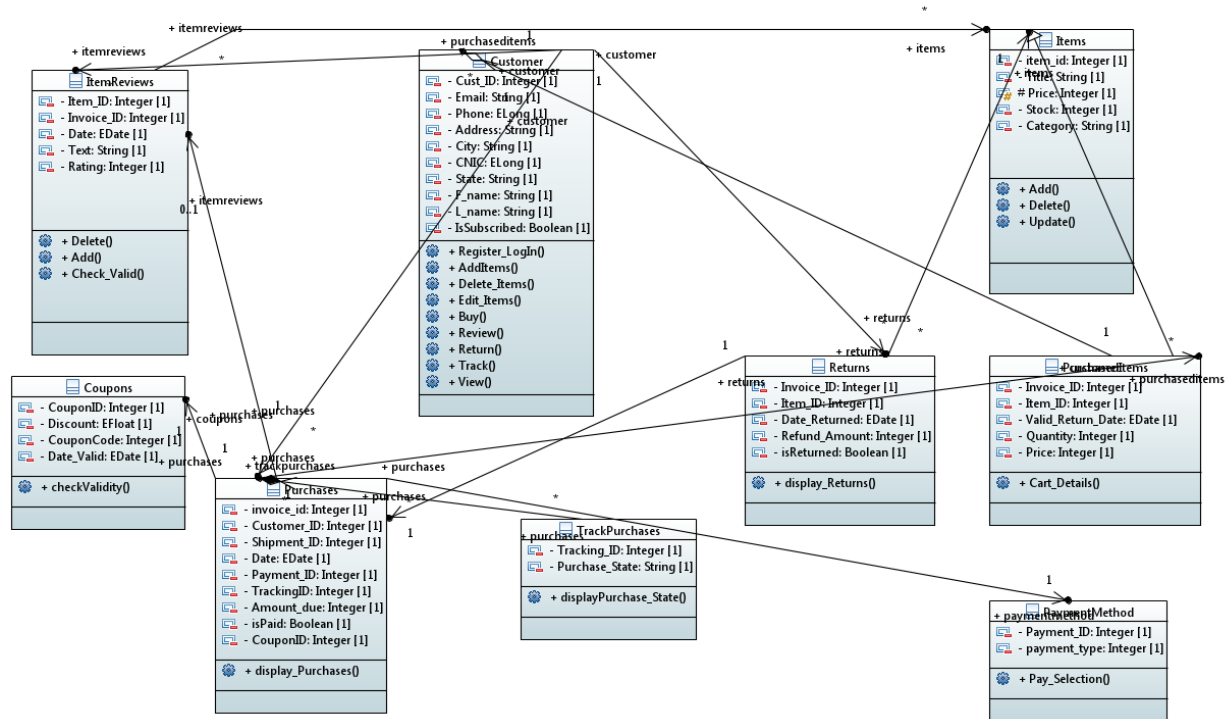
## ***7. Design Strategy***

LightHouse has a strategy that is all about the voice of the customer. "If we embrace where the consumer is going, our brands will thrive, and our system will continue to grow. This is Our Way Forward".

LightHouse has become focused entirely on consumers and what they want from ecommerce websites. As consumer tastes change, for example toward options with more elegance, LightHouse is moving with them. In recent years, LightHouse have rolled out new products in response to consumer demand, from side lamps to floor lamps, to fancy lights. Consumers want lights with benefits. Some call for smaller, more convenient packages than the classic bulb light while others want those big crystal bulbs with dark yellow LEDs. LightHouse's strategy is to continue to listen to the voice of the customer and to respond. This is not your everyday ecommerce website, it interacts on every movement, provides high end, and reliable products with easy to navigate pages.

- *Nation wide delivery*
- *Make a Return, no questions asked*
- *Want a product not on website, get in touch!*
- *Data management (storage, distribution, persistence)*
- *Use any currency for payment on website*

## 8. Detailed System Design



**Functions:**

### Database Design and Attributes of Class Diagram:

**Customers:**

1. cust\_id – every customer is given an id to index them.
2. email – used to identify each customer
3. phone – also used for verification of genuine user
4. address – to deliver products
5. city – to maintain good and structured records
6. state/province – to perform better analysis
7. cnic – for legal purposes
8. first\_name – to help interact with user
9. last\_name – for legal purposes
10. is\_subd – has the user opted for news letter

**Items:**

1. item\_id – index for every item
2. title – description of every item
3. price
4. stock
5. category – to structure each product in categories

**Purchase Items:**

1. invoice\_id – identification of every purchase
2. item\_id – which item was sold
3. valid\_return\_date – to ensure product is not invalid
4. quantity – to maintain how many products are bought
5. price

**Purchases:**

1. invoice\_id – identification of every purchase
2. cust\_id – to associate each purchase with a customer
3. shipment\_id – identifies delivery type
4. date
5. payment\_id – identifies payment type
6. tracking\_id – identifies tracking status
7. amount\_due
8. is\_paid – identifies if payment has been made
9. coupon\_id – identifies discount type

**Coupons:**

1. coupon\_id - index
2. discount – percentage of discount
3. coupon\_code
4. date\_valid – to ensure coupon validity

**TrackPurchases:**

1. tracking\_id index
2. purchase\_state (preparing, shipping, delivered) – statuses

**ItemReviews:**

1. item\_id – identifies which item is being reviewed
2. invoice\_id – which purchase is involved
3. date
4. text
5. rating

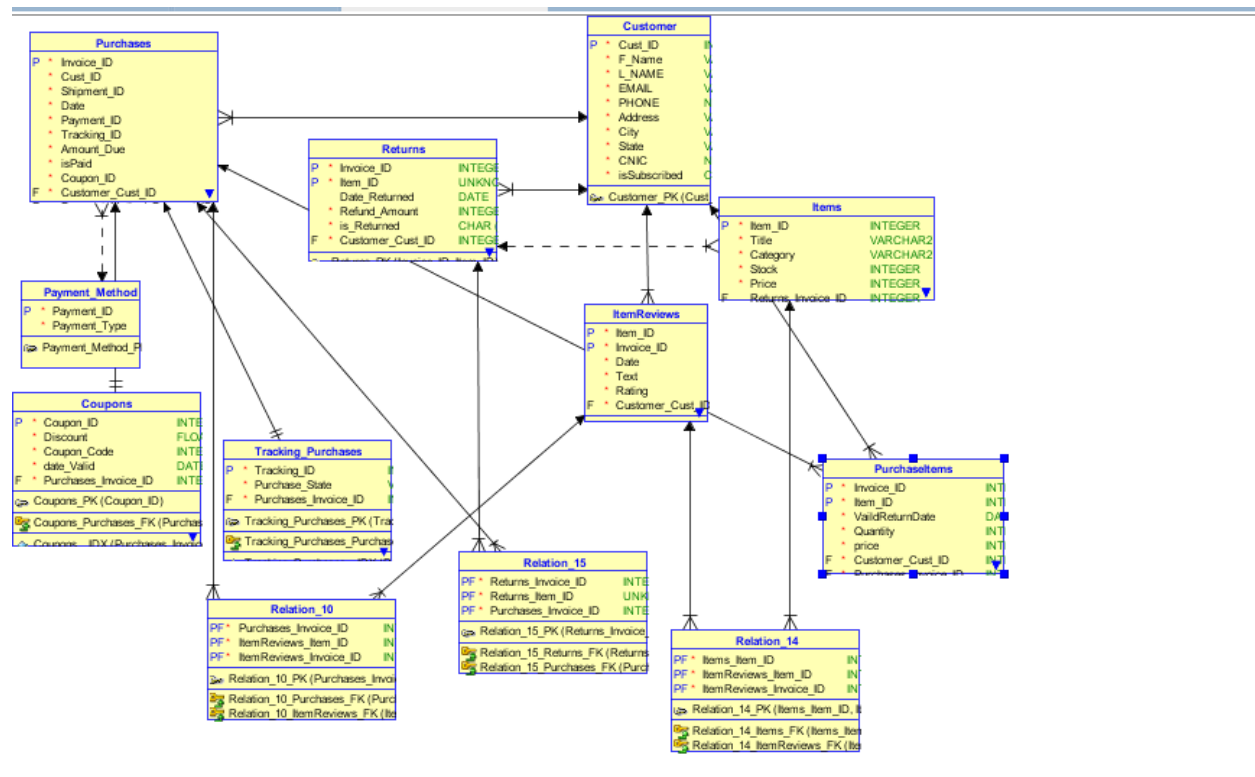


**PaymentMethod:**

1. payment\_id - index
2. payment\_type

**Returns:**

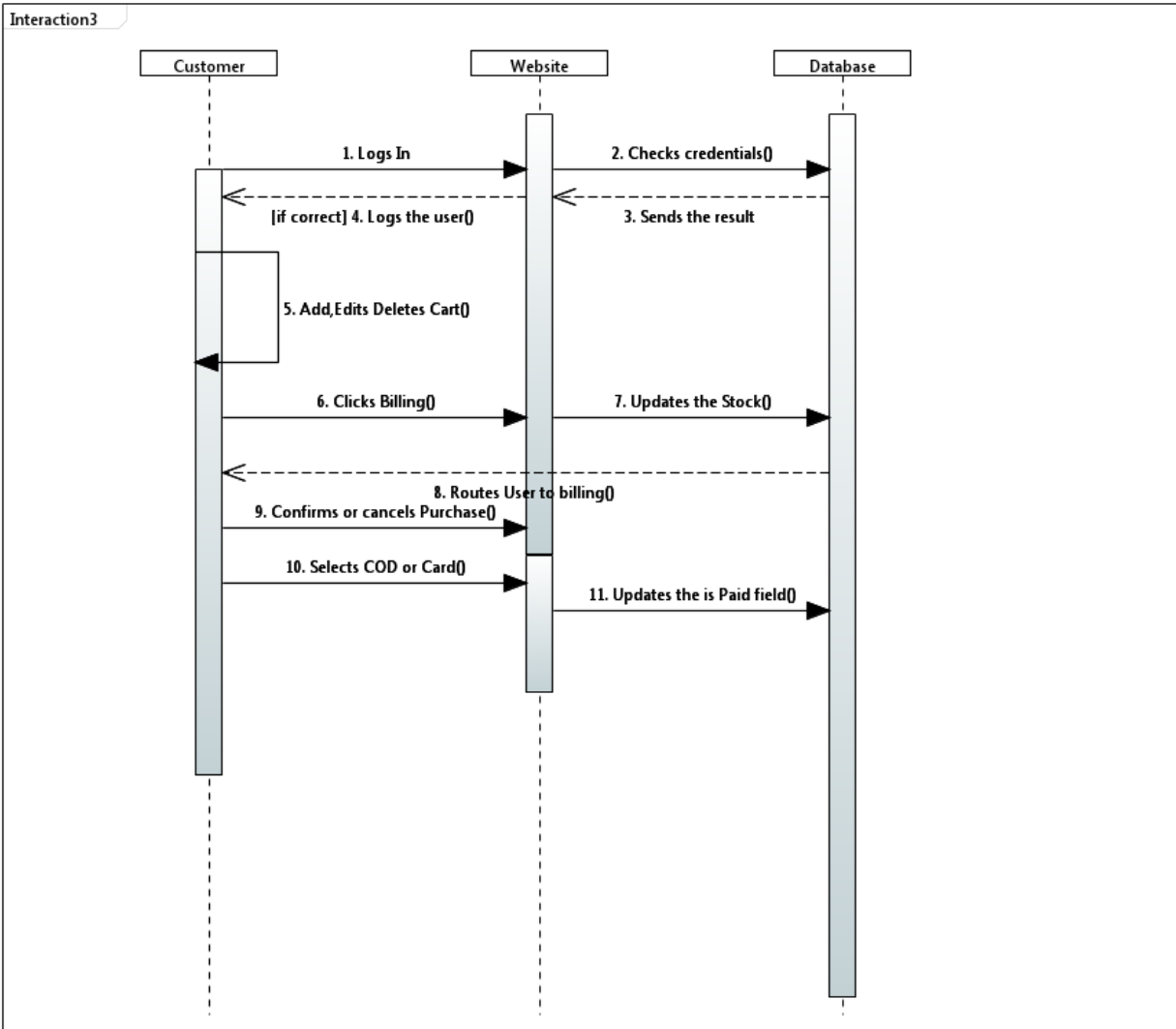
1. invoice\_id – to identify which purchase is involved
2. item\_id – what item is being returned
3. date\_returned – to check validity of return
4. refund\_amount
5. is\_returned – tells if item has been received

**8.2 ER Diagram:**

## 9. Application Design

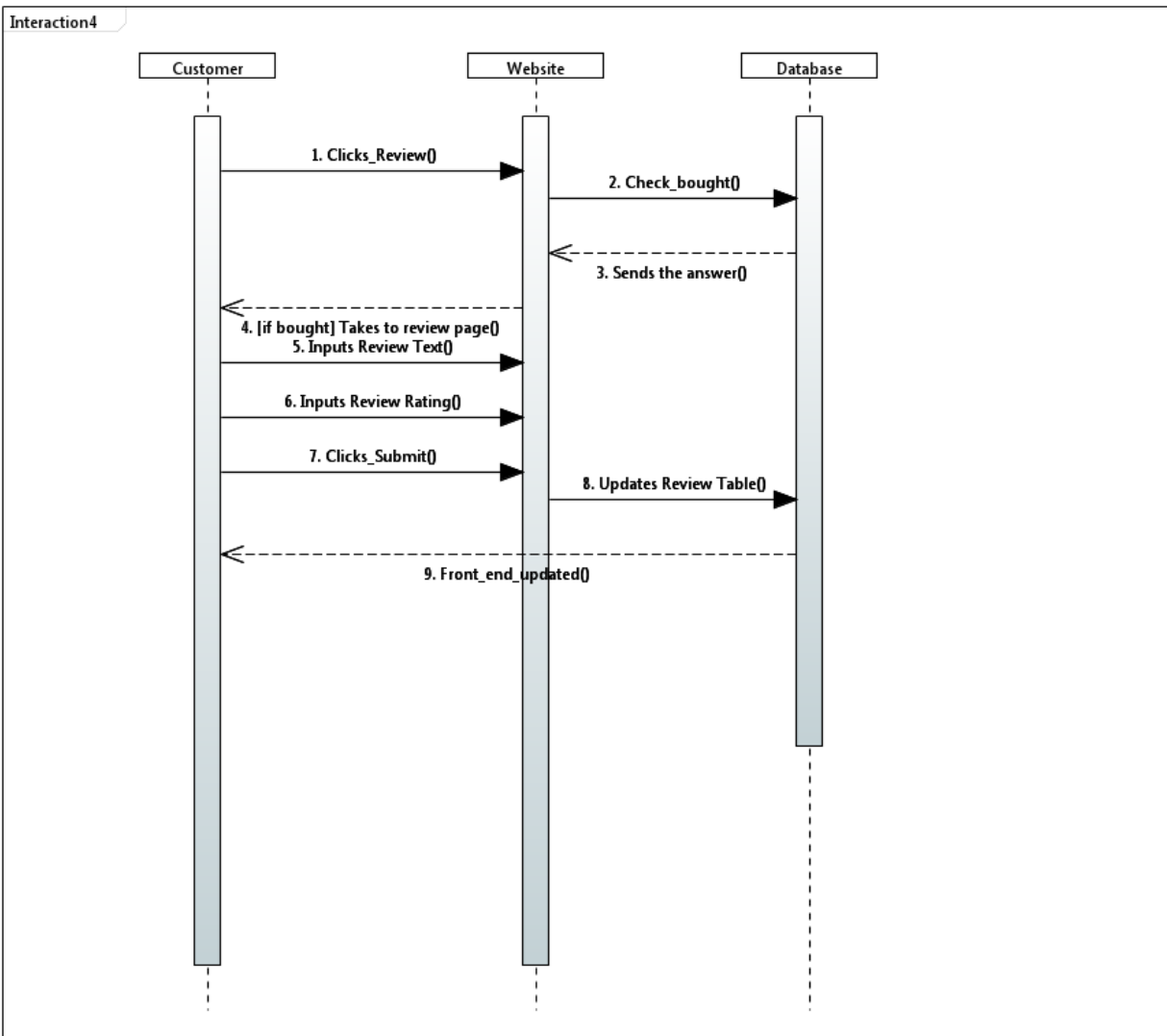
### 9.1.1 Sequence Diagram

#### 9.1.1.1 Sequence Diagram 1



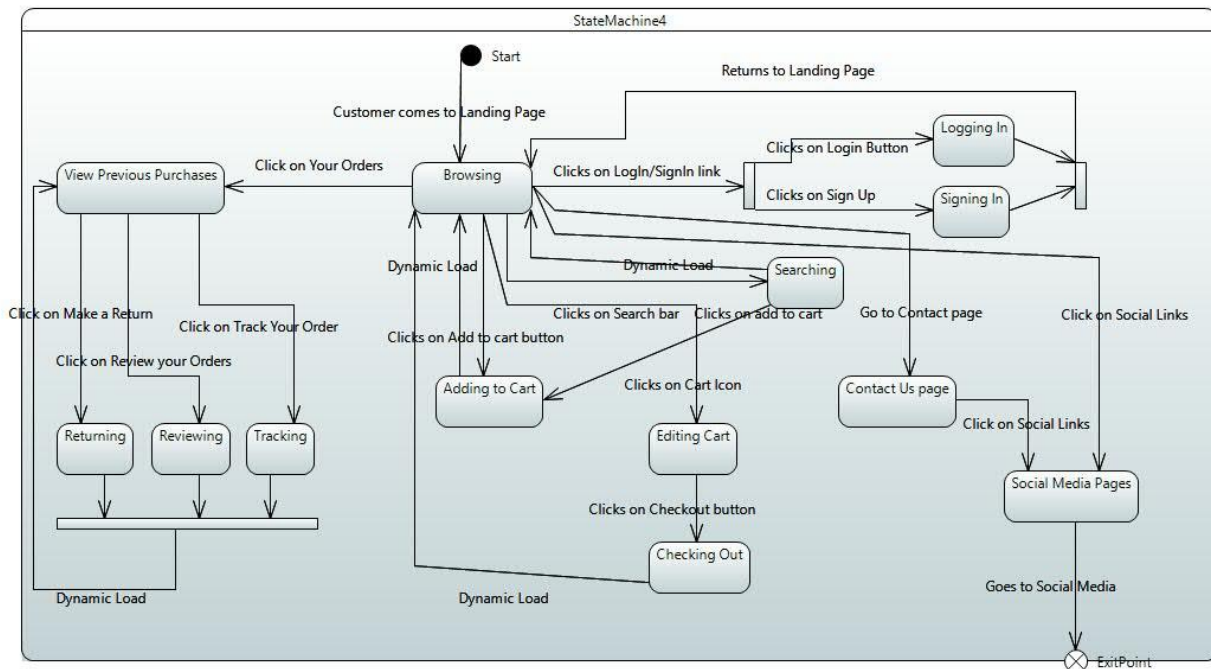
- Customer tries to login, the data is sent to the database if it returns true, customer is logged in.
- Customer then add products to cart, clicks on proceed to checkout.
- User is given last chance to cancel purchase.
- Now, customer selects COD or card payment processing, website updates the paid field in database, and a purchase is finally made.

### 9.1.1.2 Sequence Diagram 2



- Customer has come back, now he wants to review the purchased products, clicks on review, website makes sure that user has made a successful purchase.
- If he has, customer is taken to the review page. Customer then reviews and rates the products and submits the review.
- Website updates the review and rating of those products.
- Customer is taken back to the landing page.

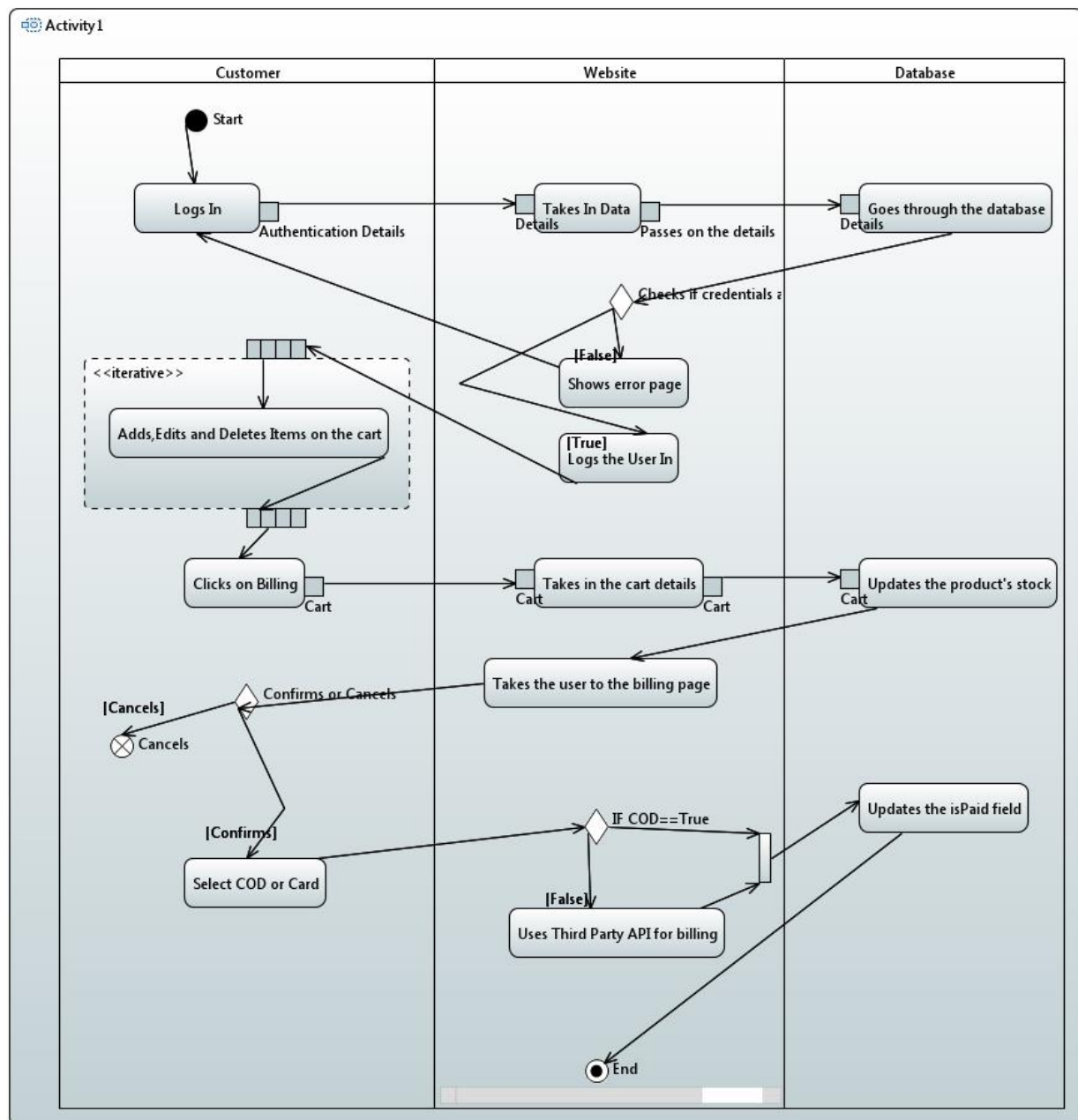
### 9.1.2 State Diagram



- Customer lands onto the main page
- Click on login page link
- Customer signs up
- Customer can now log in
- Customer returns to landing page by clicking on home
- Customer lands onto the main page after logging in
- Click on add to cart button to add product
- Customer clicks on cart icon to go to cart page
- Customer can now edit cart
- Customer proceeds to checkout
- Click on Your Orders
- Customer clicks on review product
- Customer can now choose to review product on their satisfaction
- Customer returns to Your Orders page

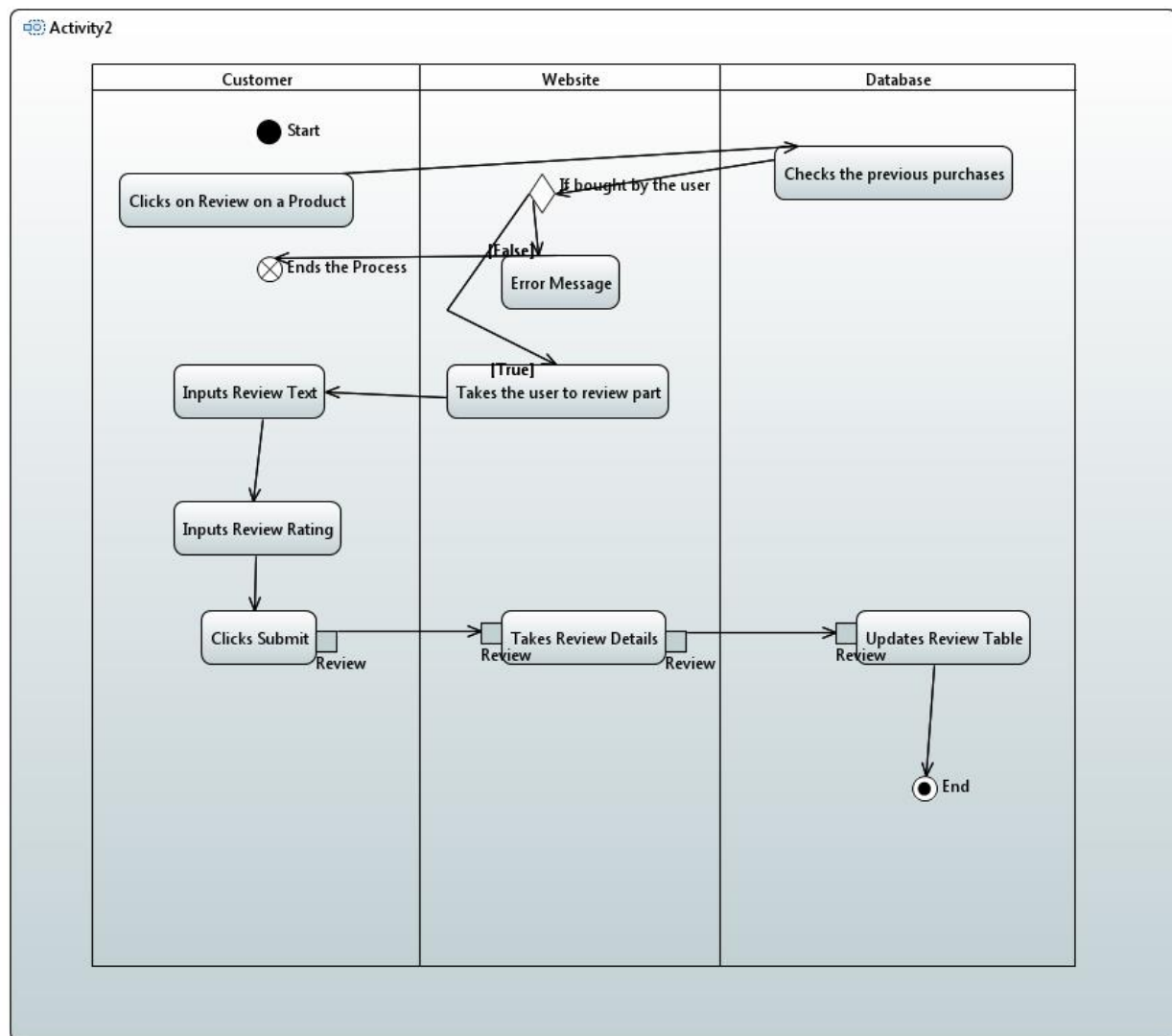
### 9.1.3 Activity Diagram

#### 9.1.3.1 Activity Diagram 1

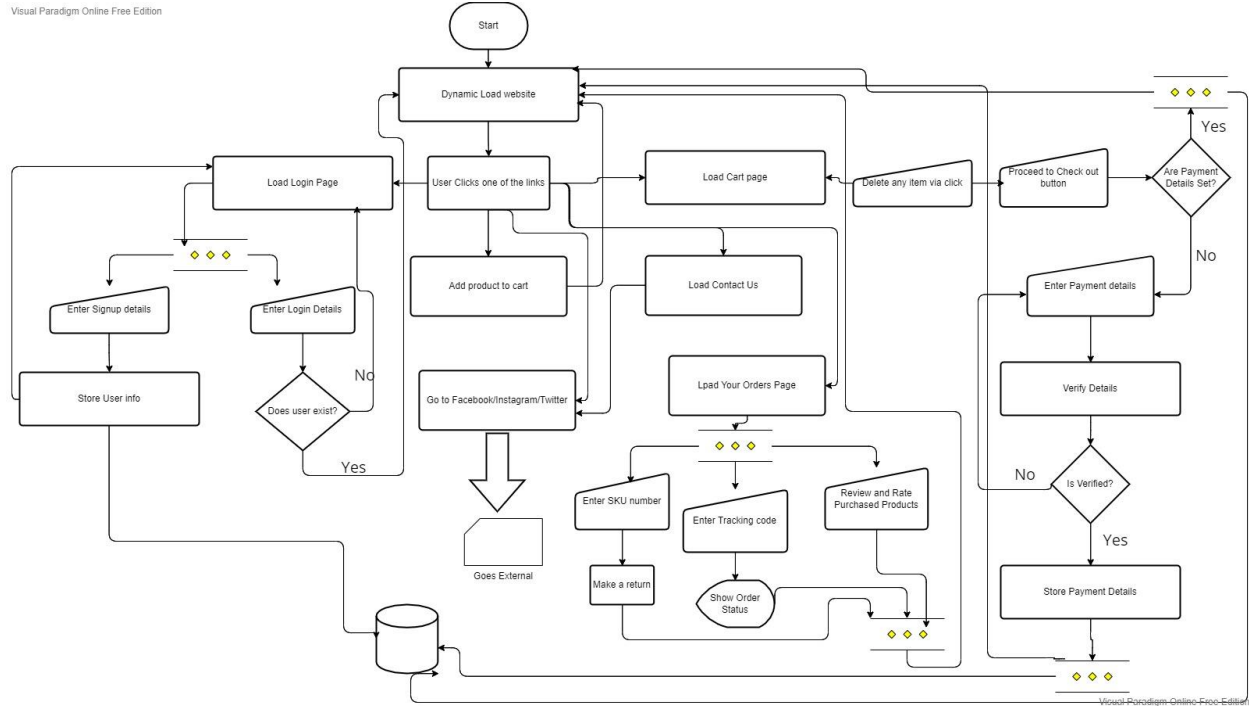


### 9.1.3 Activity Diagram

#### 9.1.3.2 Activity Diagram 2



### 9.1.4 Flowchart Diagram



## 10. References

Flowchart maker: <https://online.visual-paradigm.com/diagrams/solutions/free-flowchart-maker-online/>