

E-COMMERCE WEB APPLICATION DEVELOPMENT FOR RANSI'S ARCADE, RAJAGIRIYA

N L C M GUNATHILAKA

April 2020



E-COMMERCE WEB APPLICATION DEVELOPMENT FOR RANSI'S ARCADE, RAJAGIRIYA

N L C M GUNATHILAKA

R140516

1405162

Name of the supervisor:

Mr. P. Asanka Senarathna

April 2020

BIT



This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology (external) of the
University of Colombo School of Computing

Declaration

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate: 

Date: 02 / 12 / 2019

Name of Candidate: N. L. C. M. Gunathilaka

Countersigned by:

Signature of Supervisor: 

Date: 02 / 12 / 2019

Name of Supervisor: P. Asanka Senarathna

Abstract

Ransi's Arcade is a growing saree shop located in Rajagiriya which imports exclusive sarees from India and Bangladesh. The most valuable part of any kind of trading business is its customers. Currently, there is no even a basic website for Ransi's Arcade to attract more customers boosting profits. This was the main issue they faced when dealing with customers.

Difficulties in the manual system were time-consuming, lots of paper-work, no proper records of sales, no proper records of inventory and difficulty to track slow-moving products, and inaccurate data due to less integrity.

To overcome these difficulties, the best solution is an e-commerce web application. The application will lead to simplify the process to help achieve its organizational goals and objectives. This application covers almost all the operations and processes related to e-commerce including customer and admin account management, product management, shopping cart, wishlist, online payments, cash on delivery, order management, dashboard facility, report generations, etc.

The system was developed as a RESTful web application using MongoDB, Express, React and Node (MERN stack) . Node for server-side scripting along with Node.js web server, Express used as module framework for Node, React is a JavaScript library for building user interfaces and MongoDB as the database. Further, Bootstrap for user interface development and CSS for styling.

Through achieving the objectives of the project, the target is to increase sales by 100%, decrease their overhead cost by 30% and all of these are achieved by providing an efficient and user-friendly e-commerce system, increasing customer satisfaction and drop in complains and negative reviews.

The developed system was completely tested through the testing approach and evaluated to determine whether user requirements and specifications of the system were accomplished. Therefore, the developed system confirms to its specification by providing the intended functionality and expected outcome.

This dissertation documents all the work carried out at each phase of the project.

Acknowledgment

First of all, I would like to express my gratitude to the BIT panel of the University of Colombo School of Computing for offering such a valuable degree. The resources and support that they provided were a large part of what made this project successful.

A special word of thanks goes to my supervisor Mr. Asanka Senarathne for the guidance, encouragement, ceaseless cooperation and all the useful techniques provided throughout the project. His continuous supervision and mentoring despite his busy schedule are greatly appreciated.

I would also like to offer my special heartfelt thanks to the management and staff of Ransi's Arcade and especially to the owner of Mrs. Ransi Mala Riza and sale manager, Mrs. Kumari Kulathunga for their valuable and constructive suggestions during this project. I am particularly grateful for the sales assistant for their support and being responsive in providing the necessary information.

A special word of thanks goes to Miss. Dewduni De Alwis, my friend, who helped me in checking grammar errors and fixing them and other colleagues and friends who helped me in completing the project with their ideas and suggestions and for the motivation they given.

Finally, I would like to express my greatest gratitude and deepest regards to all those who helped me throughout the project to complete successfully.

Table of Contents

Declaration.....	ii
Abstract.....	iii
Acknowledgment.....	iv
Table of Contents	v
List of Figures.....	ix
List of Tables	xi
List of Acronyms	xiii
Chapter 1 – Introduction.....	1
1.1 Introduction of the client and the project	1
1.2 Motivation of the project.....	1
1.3 Objectives of the project	2
1.4 Scope of the proposed project.....	3
1.5 Structure of The Dissertation	3
1.5.1 Chapter 2 - Analysis.....	3
1.5.2 Chapter 3 – Design.....	4
1.5.3 Chapter 4 – Implementation.....	4
1.5.4 Chapter 5 – Evaluation.....	4
1.5.5 Chapter 6 - Conclusion	4
Chapter 2 - Analysis.....	5
2.1 Introduction	5
2.2 Fact Finding Techniques	5
2.3 Analyzing the Current System	5
2.4 Drawbacks in the Current Manual System	7
2.5 Review of Similar Systems with References	7
2.5.1 Laveena.lk	7
2.5.2 nithijayasaree.com.....	8
2.6 Functional Requirements of the System	9
2.6.1 Customer Management	10
2.6.2 User Access Management.....	12
2.6.3 Products Management.....	12
2.6.4 Category Management	12
2.6.5 Fabric Management.....	13
2.6.6 Delivery Charges Management.....	13

2.6.7 Shopping Cart Management.....	13
2.6.8 Order Management	13
2.6.9 Report Management.....	13
2.6.10 Email notification Management.....	14
2.6.11 Feedback Management	14
2.6.12 Advertisement Management	14
2.6.13 Testimonials Management	14
2.6.14 Career Management	14
2.6.15 Responsive Web Design	15
2.7 Non-Functional Requirements of the System	15
2.7.1 Browser Compatibility	15
2.7.2 Security and Privacy	15
2.7.3 User-Friendliness	15
2.7.4 Availability.....	15
2.7.5 Response time	16
2.7.6 Accuracy	16
2.7.7 Maintainability	16
2.8 Software Process Models	16
2.8.1 Waterfall Model	16
2.8.2 Rapid Application Development.....	16
2.8.3 Rational Unified Process.....	16
2.8.4 Selected Process Model and Justification	17
Chapter 3 – Design.....	18
3.1 Introduction	18
3.2 Alternative Design Strategies.....	18
3.3.1 Use Existing E-Commerce Platforms	18
3.3.2 Use Third-Party Content Management System (CMS)	18
3.3 Justification for The Selected Design Strategy	18
3.4 Architectural Design of The System.....	19
3.5 Module Design.....	20
3.5.1 Use Case Diagram for Customer Management Module	20
3.5.2 Use Case Diagram for User Access Management Module.....	22
3.5.3 Use Case Diagram for Product Management Module	23
3.5.4 Use Case Diagram for Shopping Cart Module	25
3.5.5 class diagram for the e-commerce system	26
3.5.6 Sequence Diagram for Customer Registration.....	28

3.5.7 Sequence Diagram for User Login.....	28
3.6 User Interface Design.....	28
3.6.1 Navbar	30
3.6.2 Footer	30
3.6.3 Product Search Option	30
3.6.4 Basic Filtering Option (Single value filtering)	31
3.6.5 Advanced Filtering Option (Multivalue filtering)	31
3.6.6 Product cards.....	31
3.6.7 Comment on Products.....	32
3.6.8 Review and Rate on products.....	33
3.6.9 Shopping Cart	33
3.6.10 Card Payment Interface.....	34
3.6.11 Responsive Web Design	34
3.6.12 Product Detail Page.....	35
Chapter 4 – Implementation	36
4.1 Introduction.....	36
4.2 Implementation Environment	36
4.2.1 Hardware Requirement	36
4.2.2 Software Requirements for development.....	36
4.3 Development Technologies.....	37
4.4 Existing Reusable Packages	38
4.5 Implementation Architecture	39
4.6 Justification for Selecting RESTful Architecture	40
4.7 Module and Folder Structure	40
4.8 Major Code Segments.....	41
4.8.1 Server Configuration.....	41
4.8.2 Connect Database.....	42
4.8.3 User Registration.....	42
4.8.4 User Login.....	43
4.8.5 CRUD Operations	44
4.8.6 Add to Cart Method	47
4.8.7 Remove from Cart Method	48
4.8.8 Payment Gateway	48
4.8.9 Security	49
4.8.10 Protection of User Routes	49
Chapter 5 - Evaluation	51

5.1 Introduction.....	51
5.2 Testing Approach.....	51
5.3 Test Plan and Test Cases.....	51
5.3.1 Customer Registration.....	52
5.3.2 User Login (Customer/Admin/Owner)	52
5.3.3 Customer Account Management.....	52
5.4 Test Data and Test Results.....	53
5.5 User Acceptance Testing	57
Chapter 6 – Conclusion	59
6.1 Critical Assessment.....	59
6.2 Lessons Learnt	60
6.3 Future Enhancements	60
References	61
Appendix A – System Documentation.....	64
Appendix B - Design Documentation	66
Appendix C - User documentation	73
Appendix D - Management Reports.....	80
Appendix E – Test Cases and Test Results	87
Appendix F - Client Certificate.....	99
Glossary.....	100
Index.....	101

List of Figures

Figure 2.1: The use case diagram for the existing system	6
Figure 2.2: The official website of Laveena sarees	8
Figure 2.3: Tthe official website of Nithijaya.....	9
Figure 2.4: The RUP model	17
Figure 3.1: The proposed architectural design.....	19
Figure 3.2: The use case diagram for the entire customer management module.....	20
Figure 3.3: The use case diagram for user access management module.....	22
Figure 3.4: The use case diagram for the product management module.	24
Figure 3.5: The use case diagram for the shopping cart module	25
Figure 3.6: The class diagram for the e-commerce web application.	27
Figure 3.7: The sequence diagram for customer registration.....	28
Figure 3.8: The sequence diagram for customer registration.....	28
Figure 3.9: The color theme of entire web design.	29
Figure 3.10: Different kinds of alert messages	29
Figure 3.11: Navbar of the website	30
Figure 3.12: Footer of the website	30
Figure 3.13: Product search option	30
Figure 3.14: Basic filtering options.....	31
Figure 3.15: Advanced filtering option.....	31
Figure 3.16: Product cards	32
Figure 3.17: The comment model	32
Figure 3.18: The review and rate on products	33
Figure 3.19: Shopping cart interface.....	33
Figure 3.20: Card payment interface.....	34
Figure 3.21: Responsive web design for RAEWA	34
Figure 3.22: The product detail page	35
Figure 4.1: module and folder structure.....	41
Figure 5.1: Customer evaluation results	57
Figure 5.2: The client evaluation form filled by the sales manager.....	58
Figure B.1: The use case diagram for the delivery charges module	66
Figure B.2: The use case diagram for the customer order module	67
Figure B.3: The use case diagram for the advertainment module	68

Figure B.4: The use case diagram for the testimonial module.....	69
Figure B.5: The use case diagram for the career module.....	70
Figure B.6: The use case diagram for the report module.....	71
Figure C.1: Navbar for navigating to the login screen of the system	73
Figure C.2: Login screen of the system	73
Figure C.3: Admin dashboard.....	74
Figure C.4: The navbar of the dashboard with its elements.....	74
Figure C.5: Side menu bar of the dashboard with its elements.....	75
Figure C.6: View testimonials page with edit and delete buttons.....	75
Figure C.7: User profile	76
Figure C.8: How to add an admin by clicking the setting dropdown	76
Figure C.9: Add admin page	77
Figure C.10: View all admins.....	77
Figure C.11 common buttons.....	78
Figure C.12: Add product page.....	79
Figure C.13: View products page	79
Figure D.1: Category wise inventory report	80
Figure D.2: Fabric wise inventory report.....	81
Figure D.3: Price-wise inventory report	82
Figure D.4: The inventory report	83
Figure D.5: Sales/Cost/Profit report.....	84
Figure D.6: Sold products report.....	84
Figure D.7: The customer report.....	85
Figure D.8: The liked product report	85
Figure D.9: The commented products report	86
Figure D.10: The reviewed and rated products report	86

List of Tables

Table 3.1: The use case description for customer registration	21
Table 3.2: The use case description for the customer account management panel	22
Table 3.3: The use case description for adding a new admin account	23
Table 3.4: The use case description for adding a new product	24
Table 3.5: The use case description checkout process	26
Table 4.1: Development technologies	38
Table 5.1: The test cases for the customer registration process	52
Table 5.2: The test cases for the user login process	52
Table 5.3: The test cases for the customer account management process	53
Table 5.4: The test results for the customer registration process	54
Table 5.5: The test results for the user logins	55
Table 5.6: The test cases for the customer account management process	57
Table A.1: Hardware requirements	64
Table A.2: Software requirements	64
Table B.1: The use case description of adding a new delivery charge	67
Table B.2: The use case description of updating the customer order status	67
Table B.3: The use case description of adding a new advertisement	68
Table B.4: The use case description of adding a new testimonial	69
Table B.5: The use case description of adding a new career	70
Table B.6: The use case description of profit report generation	72
Table E.1: The test cases for the searching products	87
Table E.2: The test cases for the basic filtering products	87
Table E.3: The test cases for the advanced filtering products	87
Table E.4: The test cases for the wishlist management	88
Table E.5: The test cases for the shopping cart management	88
Table E.6: The test cases for the online payment checkout	89
Table E.7: The test cases for the cash on delivery checkout	90
Table E.8: The test cases for the admin registration	90
Table E.9: The test cases for the owner and admin profile management	90
Table E.10: The test cases for the product management	91
Table E.11: The test cases for the category management	91
Table E.12: The test cases for the fabric management	92

Table E.13: The test cases for the delivery charges management	92
Table E.14: The test cases for the customer order management	93
Table E.15: The test cases for the advertisement management	93
Table E.16: The test cases for the testimonial management	93
Table E.17: The test cases for career management	94
Table E.18: The test cases for report management	94
Table E.19: The test cases for common functions.	95
Table E.20: The test results for online payment checkout	97
Table E.21: The test results for product management module	98

List of Acronyms

CD – Compact Disc

CD-ROM – Compact Disc Read Only Memory

CMS - Content Management Systems

CRUD – Create Read Update Delete

CSS – Cascading Style Sheet

CVV – Card Verification Value

DBMS - Database Management System

DOM – Document Object Model

FAQ - Frequently Asked Questions

HTTP – HyperText Transfer Protocol

JSON - JavaScript Object Notation

JWT – JSON Web Token

MVC – Model View Controller

NPM – Node Package Manager

ODM - Object Data Modeling

PDF – Portable Document Format

RAD – Rapid Application Development

RAEWA – Ransi's Arcade E-commerce Web Application

RAM – Random Access Memory

REST - REpresentational State Transfer

RUP – Rational Unified Process

SATA - Serial AT Attachment

SEO - Search Engine Optimizations

UI – User Interface

UML – Unified Modelling Language

URL – Uniform Resource Locator

VGA – Video Graphics Array

Chapter 1 – Introduction

1.1 Introduction of the client and the project

Ransi's Arcade was registered in the Department of the Provincial Register of Business Names for Western Province under certificate No. W/PP/8569 on the 30th of March 2015. It is a growing saree shop located in Rajagiriya which imports exclusive sarees from India and Bangladesh. The most valuable part of any kind of trading business is its customers. Currently, there is not even a basic website for Ransi's Arcade to attract more customers boosting profits.

The main objective of Ransi's Arcade is to increase its profit. Ransi's Arcade E-commerce Web Application (RAEWA) is proposed as the best solution to achieve this main objective while attracting more customers and satisfying them, increasing sales, reducing operational costs, reducing promotional and advertising costs, etc. The following sections will be elaborated further on this project.

1.2 Motivation of the project

With the aid of an e-commerce web application, Ransi's Arcade gets an opportunity to have their products and services available to their customers for 24 hours. It gives good exposure to their business and helps them to reach out to most of their potential customers. Nowadays, people are busy with their tight schedules. Therefore, they do not have enough time for shopping. Since most of the people prefer to shopping online, Ransi's Arcade can easily make more sales with a website. With the aid of an e-commerce web application, people have the opportunity to select and buy desired products at any time. They can easily pay through cards or PayPal.

When it comes to buying gifts, an e-commerce shopping cart is what people love nowadays. Almost everything works online. Just need to visit the website, select a product, add it to the shopping cart and pay. The gifts will be delivered on time.

Another feature of the e-commerce website is that can save an item to ‘wishlist’ and buy it later. Therefore, do not need to go through the product search process as well, since it is already saved in the wishlist. This is why e-commerce websites are very user-friendly.

This e-commerce web application will immensely reduce issues of the existing manual system where there are no proper records of sales and expenses to track their profits, no proper way to maintain their inventories, no proper way to look at daily, monthly reports and difficulty to track slowly moving products, etc.

Finally, RAEWA allows their customers to post likes, comments, and reviews about products. Therefore, the website owner receives feedback and can use feedback to improve services and make them better. Keeping in touch with customers can also help in long-term business survival. Therefore, the growth of e-commerce is very important in today's competitive world to gain a competitive advantage. [1]

1.3 Objectives of the project

The main objective of RAEWA is to increase the ultimate profit of the business. The following are going to be implemented to achieve the main objective.

- To attract new customers through the RAEWA.
- To extend beyond physical location.
- To increase customer satisfaction by allowing like on products, comment on products, review on products, spend less time for searching and giving more product details, etc.
- To run the business over 24*7.
- To increase monthly income as well as to reduce operational and other costs.
- To give a more flexible service.
- To retain existing customers and attract potential customers and keep them coming back.
- To increase customer satisfaction by providing a good-looking user-friendly interface where every customer can easily access the site to purchase online.
- To reduce advertising and promotional expenses etc.

1.4 Scope of the proposed project

When deciding to develop a system, the scope of the project is the most important fact which we need to consider. Ecommerce system will cover the following functional aspects;

- Customer management
- User access management
- Product management
- Category management
- Fabric management
- Delivery charges management
- Shopping cart management
- Order management
- Report management
- Email notification management
- Feedback Management
- Advertisement management
- Testimonial management
- Career management
- Responsive web design (These points are described further under “Functional Requirements of the System”)

1.5 Structure of The Dissertation

This dissertation contains 6 (six) main chapters including the introductory chapter which provides an overall understanding of the project.

1.5.1 Chapter 2 - Analysis

The analysis chapter describes the requirement gathering techniques, reviews of similar systems with its references, current system explained using diagrams, functional and non-functional requirement identification, and justifies choice of the development life cycle.

1.5.2 Chapter 3 – Design

The design chapter describes alternative design strategies, justifications for the selected design strategy, the architectural design of the system, data modeling diagrams and user interface design.

1.5.3 Chapter 4 – Implementation

The implementation chapter describes the architectural design, implementation environment, major code segments, reused existing codes or APIs, software tools and technologies used in software development.

1.5.4 Chapter 5 – Evaluation

This chapter describes the testing approach, test plan, proof of testing of work with the result of work and its end-user evaluation.

1.5.5 Chapter 6 - Conclusion

The final chapter explains the critical assessment of the project. It also describes the lessons learned from the project and suggested future implementations.

Chapter 2 - Analysis

2.1 Introduction

The analysis chapter describes requirement gathering techniques, explains the current system using diagrams, reviews of similar systems with references, identifies the functional and non-functional requirements, and provides justification for the choice of the development life cycle.

2.2 Fact Finding Techniques

Several fact-finding techniques were used to collect data and information for the identification of system requirements and specifications. The success of any project is deeply dependent on the accuracy of the collected data and information.

Usually, a combination of fact-finding techniques was used to gather requirements and specifications. The sampling of documentation, questionnaires, interviews, researches, site visits and observations are different fact-finding techniques. In this project, interviews, document reviews and observations were a few of the several techniques that were used to gather data and information. [2]

The owner, manager, staff and customers of the business were interviewed to gather information regarding the existing system and new requirements of the system. The observation and document reviews were done to analyze the existing users, processes, and performance. Furthermore, the limitations of the current system were analyzed to identify the requirement for the new system.

2.3 Analyzing the Current System

The current system was analyzed to develop an efficient and effective system. The major activities of the current system are as follows.

- Import sarees from India and Bangladesh
- There are over two thousand sarees at any given time
- All sarees are unique
- Record sold sarees only on a book

- There's no record for inventory (Therefore, cannot say exactly how many sarees are left in stocks now)
- Current system cannot say the sales, costs and profit for a week, month or annually.
- Each saree has a tag with the price
- Customers should order the products by visiting the physical shop
- Customers select sarees by observing.
- Payments can be made on cash and/or card payments
- There is not even a basic website
- A Facebook account called “Ransis Arcade” is available

The use case diagram for the existing system is presented in Figure 2.1.

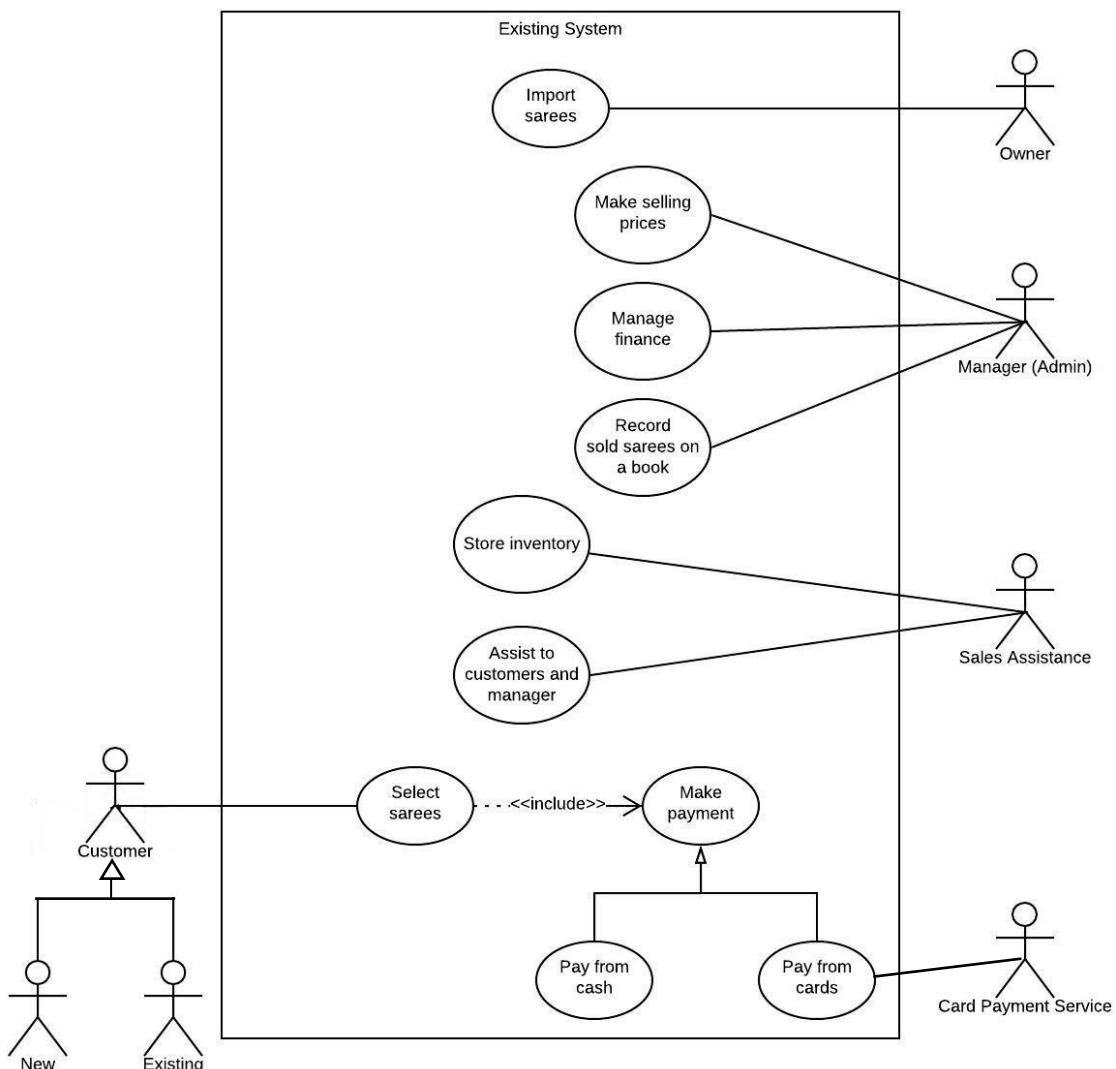


Figure 2.1: The use case diagram for the existing system

2.4 Drawbacks in the Current Manual System

Some major drawbacks of the current system were identified while gathering requirements. The identified drawbacks are listed below.

- No basic website to attract more customers
- No proper inventory control system
- Poor customer handling.
- All the records are kept on the books and it leads to data redundancy
- Bookkeeping is improper and does not use even spreadsheets to record data.
- Poor handling of payments
- No proper records of sales, costs, and profits
- Unorganized documentation
- Lack of efficiency and effectiveness
- Time consuming
- Lack of accurate and summarized reports

2.5 Review of Similar Systems with References

The following web sites were studied to get an overview understanding of the features and functions of e-commerce web applications.

2.5.1 Laveena.lk

Laveena sarees is a local business located in the Colombo area. This website was analyzed to identify features and functions they used. [3]

It consisted of a basic home page with new arrivals, shop page, about us page and contact us page. The product search options are limited to product names only therefore this website does not allow products to be searched by price, color, etc. Furthermore, the basic filter options are also limited to price and category (no color, occasion, etc.) but consist of sort options by price, latest arrivals, ratings, and popularity.

The website had a detailed product page, review option with ratings, social media share options, related product section, wishlist, shopping cart, cash on delivery system and the ability to perform online by card and direct bank transfer.

After further analysis, it was listed that their website lacked customer registration and login. The add-ons like the customer control panel to view previous order items and order status, and update profile were also missing. An advanced filtering option (multi-value filtering by category, price, color, occasion, etc.) was one important feature that has not been incorporated into the website. Though not essential, it has been observed that new web pages like FAQ, career, testimonial section (what customers think about the service) and space for business advertisements were not included. Features of like and comment on products option, contact form to send emails and no location map were noted missing.

The official website of Laveena sarees is presented in Figure 2.2.

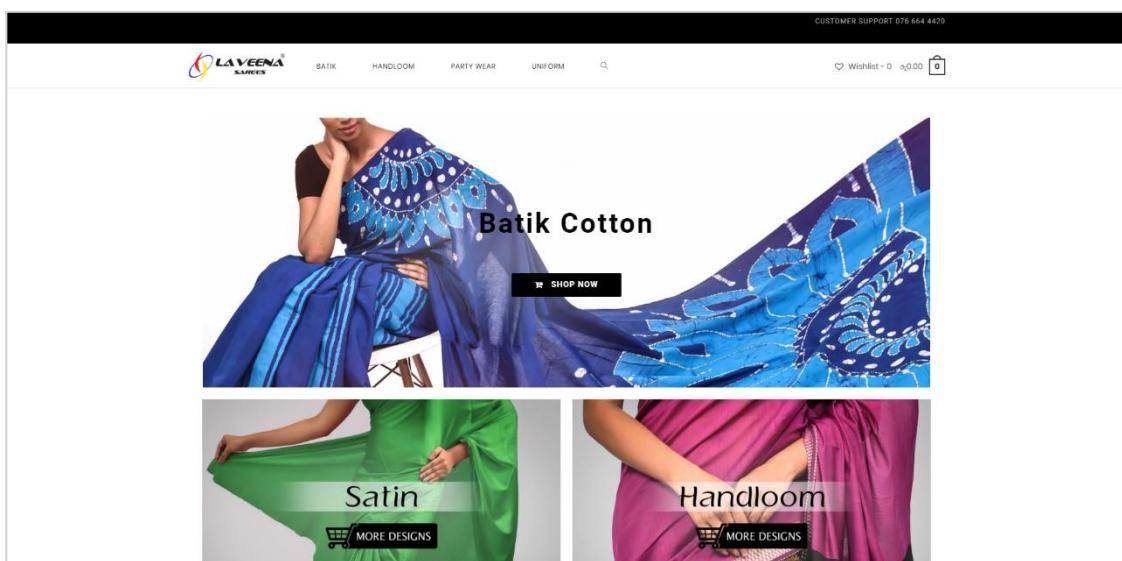


Figure 2.2: The official website of Laveena sarees

2.5.2 nithijayasaree.com

Nithijaya saree (Pvt) Ltd is a local business located in Nugegoda. This website was analyzed to identify features and functions they used. [4]

It consisted of a customer registration and login, customer control panel to view previous order items and order status, update profile, a home page with new arrivals and bestsellers, category pages, about us page and contact us page, FAQ page, etc. On analysis, it was noted that the product search option does not function. To list down other features that are available from this particular website were sort options by price, latest arrivals, ratings, and popularity, review option with ratings, social media share options,

related product section, wishlist, shopping cart, cash on delivery system, online payment system by card and direct bank transfer.

After analysis, it was listed that their website lacked both of basic filter options such as filtering by price, category, color, occasion, etc. and advanced filtering options such as multi-value filtering by category, price, color, occasion, etc. Further, testimonial section and space for business advertisements and features of like and comment on products option were not included.

The official website of Nithijaya is presented in Figure 2.3.

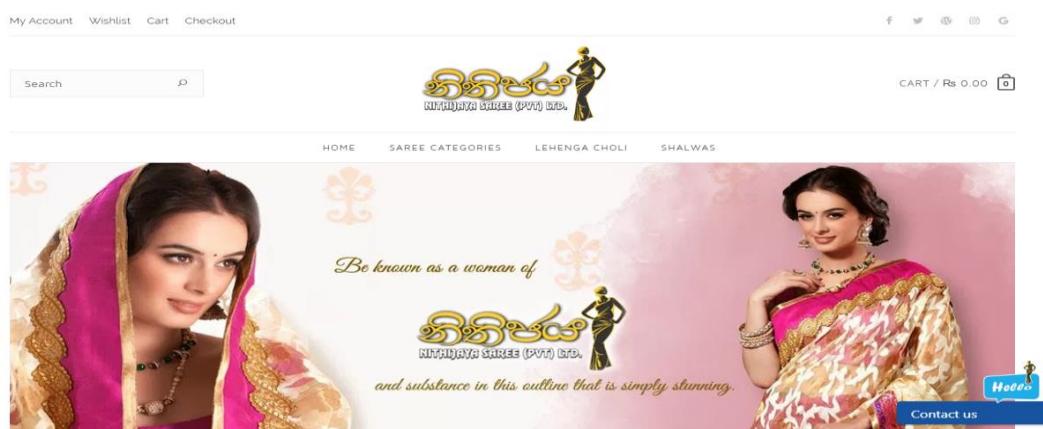


Figure 2.3: Tthe official website of Nithijaya.

2.6 Functional Requirements of the System

Following are the key functionalities modules of the project.

- Customer management
- User access management
- Product management
- Category management
- Fabric management
- Delivery charges management
- Shopping cart management
- Order management
- Report management
- Email notification management
- Feedback Management
- Advertisement management

- Testimonial management
- Career management
- Responsive web design

2.6.1 Customer Management

2.6.1.1 Unregistered Customers

Unregistered customers will be able to:

- Register into the website.

Once a customer shows interest and wants to get an account then he will be taken to a page where he will be asked to submit a form that would have various fields for the customer to enter their personal details creating a profile to their own. An automatically generated mail will be sent by the system after successful customer registration and this customer will automatically be assigned into the account manager role for this particular account.

2.6.1.2 Registered Customers

Registered customers will be able to:

- Manage their accounts:
 - Login into their accounts.
 - View his/her account.
 - Update his/her account (first name, last name, mobile number, and date of birth (DOB)).
 - View his/her orders and their status.
 - Change the password.
 - Reset the forgot password (Details will be mailed to the customer).
 - Logout.
- Order products through online payments or cash on delivery.
- Make payments for their orders by using cards or PayPal (Mail will be sent automatically after successful order submission with including order information).
- Like on products.
- Comment on products.
- Review and rate on products.

2.6.1.3 Both Unregistered and Registered Customers

Both unregistered and registered customers will be able to:

- View home page with sections of latest arrival products, most Liked products, most commented products, website advertisements, and customer testimonials.
- View about us page to know business history, products, business vision, and mission, etc.
- View the FAQ page to know frequently asked questions by customers.
- View Contact us page to know the contact details of the business with the location map.
- Contact the business staff through an e-mail by filling the contact form in the contact us page.
- View the career page to know what the current opening positions would be at the moment.
- View shopping page which has basic and advanced filtering options, product search option, etc.
- Easily search for products by using different keywords like name, category, fabric, price color, etc.
- Do basic filtering by category or fabric or occasion or price or color or likes or comments or ratings.
- Do advanced filtering by category and/or fabric and/or occasion and/or price and/or color.
- View the product cards with the main image, clickable thumbnail images and other details such as ratings, category, product name, discounted price, old price, likes, and comments, etc.
- View the product detail page with all product details and image zoom effect to enlarge images.
- View detail product page with all details such as product name, category, fabric, color, description, discounted price, old price, occasions, wash & care, likes and comments, ratings, etc.
- View related products after selecting a particular product.
- Add/update items in shopping cart in real-time.
- Add/view/delete wishlist items and add to cart them
- Share products/items on social media like Facebook, Twitter, etc.

2.6.2 User Access Management

Owner will be able to:

- Access his/her owner account.
- Add admin accounts by providing information of first name, last name, mobile, date of birth (DOB), and password.
- View the list of all admins.
- Delete admin accounts.
- Do admin tasks.

Admin will be able to:

- Access his/her admin account.
- Update his/her profile.
- Do admin tasks.

Customer will be able to:

- Access his/her customer account.
- Update his/her profile.
- Do customer tasks.

2.6.3 Products Management

Admin will be able to:

- Add a product by providing information of product name, category, fabric, color, price, quantity, cost, description, images, occasions, wash and care recommendations and searching keywords, etc.
- View the list of all products.
- Update products.
- Activate/deactivate products.
- Delete products.

2.6.4 Category Management

Admin will be able to:

- Add a category by providing information of category name.
- View the list of all categories.
- Update categories.
- Delete categories.

2.6.5 Fabric Management

Admin will be able to:

- Add a fabric by providing information of fabric name.
- View the list of all fabrics.
- Update fabrics.
- Delete fabrics.

2.6.6 Delivery Charges Management

Admin will be able to:

- Add delivery charge by providing information of city and delivery charges.
- View the list of all delivery charges with cities.
- Update delivery charges.
- Delete delivery charges.

2.6.7 Shopping Cart Management

Customer will be able to:

- View the shopping cart.
- Update the quantities of cart items before check out (The system will be automatically updated the amounts of the cart).
- Delete cart items.
- Select the checkout option either online payment or cash on delivery.
- Go back to shop more by clicking “continuous shopping” link.

2.6.8 Order Management

Admin will be able to:

- View list of all customer orders.
- View the delivery details.
- Update order status as unprocessed, processing, delivered, etc.

2.6.9 Report Management

Admin will be able to generate followings reports:

- List of customers.
- Sales, costs and profit report.
- Stock availability reports etc.

Admin is able to generate reports periodically (daily, weekly, monthly and annually).

2.6.10 Email notification Management

Customer will receive a notification after:

- Registrations.
- confirm of “Cash on Delivery” orders.
- online payments.

2.6.11 Feedback Management

Customer will be able to:

- Add likes on products.
- Add comments on products.
- Add review and rating on products.

Admin will be able to:

- View customers who add likes on products.
- View customers who add comments on products.
- View customers who add review and rating on products.

2.6.12 Advertisement Management

Admin will be able to:

- Add advertisement by providing information of advertisement purpose and advertisement image.
- View list of all advertisements.
- Delete advertisements.

2.6.13 Testimonials Management

Admin will be able to:

- Add testimonial by providing information of customer name and comment.
- View list of all testimonials.
- Delete testimonials.

2.6.14 Career Management

Admin will be able to:

- Post advertisements by providing information of position name, position type (Full time or part-time), closing date, and advertisement image.
- View list of all career opportunities.
- Delete career opportunities.

2.6.15 Responsive Web Design

The RAEWA should be automatically resizing, hiding, shrinking, or enlarging to make it look good on all devices such as desktops, laptops, tablets, and mobile phones.

2.7 Non-Functional Requirements of the System

Non-functional requirements describe important characteristics of the system. It specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The non-functional requirements defined for the newly proposed system are given below.

2.7.1 Browser Compatibility

The system should be compatible to run with different web browsers such as Chrome, Firefox, Safari, Edge, and Opera, etc.

2.7.2 Security and Privacy

Security is a key non-functional requirement in this system because of multiple users use this website. Therefore, appropriate security measures should be taken to keep the data secure and confidential. Adequate technical measures will be implemented to prevent unauthorized access to data.

2.7.3 User-Friendliness

The system should be easy to understand and easy to use while being user-friendly through friendly interfaces, simple English words (avoiding Jargon) and notifications, etc.

2.7.4 Availability

The users should be able to access the system 24/7. Since it is a web-based system, the system should be made available to be accessed from any location.

2.7.5 Response time

The response time of the system should be fast and cause no delay.

2.7.6 Accuracy

System reports and the information provided by the system should be accurate.

2.7.7 Maintainability

The system should be easy to maintain for administration purposes.

2.8 Software Process Models

A simplified representation of a software process is known as a software process model. It describes the structure of phases for the life of a software product. There are many process models for software development, some of them are described below.

2.8.1 Waterfall Model

The waterfall model is a sequential design process. Therefore, a new step cannot be started before the completion of the previous stage. This is the oldest model and a very structured process that does not suitable for systems with dynamic requirements or when requirements are not clear. [2]

2.8.2 Rapid Application Development

RAD is a type of incremental process model that starts with an initial prototype model and active user involvement is the focus of this module. RAD is a team-oriented approach. Therefore, this process model is not suitable for individual projects. [2]

2.8.3 Rational Unified Process

RUP is an iterative software development process model that is short in the development cycle, typically 60-90 days. But it is a flexible process framework that should be designed by development companies and software project teams. Select the components of the process that best satisfy their needs.

The RUP development process model has a determined project life-cycle consisting of four stages as inception, elaboration, construction, and transition. [5]

Inception, elaboration, construction, and transition are the four phases of the RUP development process model.

- Inception - The business context and the scope for the project are set and the feasibility is evaluated initially.
- Elaboration - The project plan and the architecture of the system is further evaluated in this stage.
- Construction - The components of the project are developed based on project architecture.
- Transition - The system is released to its end users for operation. End-user training and beta testing are done.

The RUP model is presented in Figure 2.6.

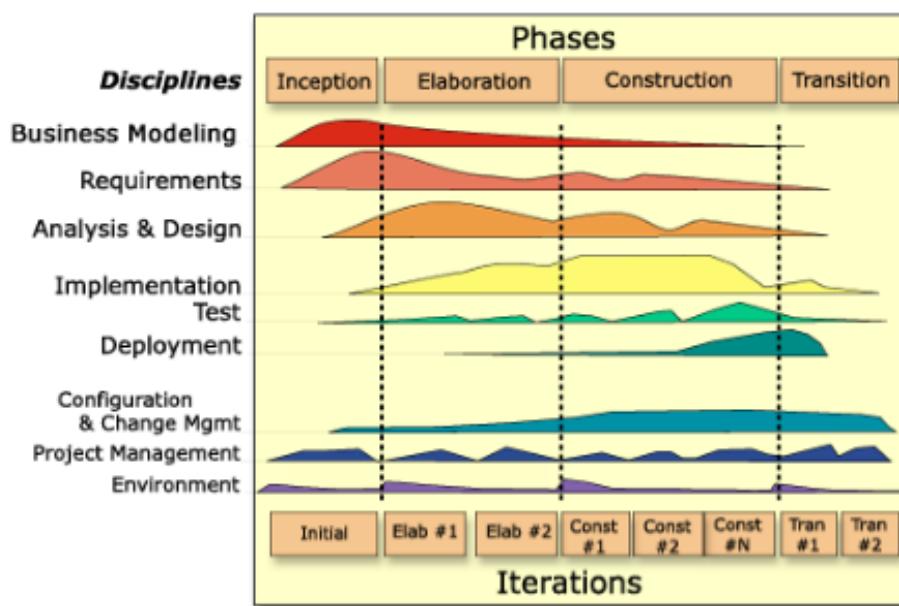


Figure 2.4: The RUP model

2.8.4 Selected Process Model and Justification

RUP was selected for the project because it is an iterative process with object-orientation support, flexibility to adapt new requirements or changes in the business objectives. The decision to select the RUP model depended after considering different process models and evaluating their benefits and limitations in project environments, RUP proved to be the most suitable model for this project.

Chapter 3 – Design

3.1 Introduction

This design chapter describes alternative design strategies, justifications for the selected design strategy, the architectural design of the system, data modeling diagrams and user interface design.

3.2 Alternative Design Strategies

The alternative design strategies for the e-commerce web application were evaluated during the ongoing project to find an appropriate solution that would be able to fulfill the client's requirements in the most effective manner. Below are the appraised alternative strategies.

3.3.1 Use Existing E-Commerce Platforms

Existing e-commerce platforms can be used as an alternative. The e-commerce platform is a software application that allows online businesses to manage their website, marketing, sales, and operations. [6] BigCommerce, Wix, Magento, WooCommerce, Shopify, 3dcart, Volusion, Prestashop, Weebly, and Squarespace are examples of e-commerce platforms.

3.3.2 Use Third-Party Content Management System (CMS)

Third-party CMS are systems that allow you to control and manage the content of a website without any technical training. With this simple setup, one can easily add, delete and edit text on your website when you fly. Examples of popular CMS systems are WordPress, Drupal, and Joomla. All these CMS have third-party plugins that are able to convert a website into an e-commerce platform. [7]

3.3 Justification for The Selected Design Strategy

The selected strategy was to develop an e-commerce application from scratch. Listed are the reasons are as follows:

- There are client specific requirements that cannot be fulfilled from CMS systems. (E-commerce platforms and CMS are limited client's requirements, features, functions and interface designs, etc.)

- This approach makes the development process a flexible and scalable project.
- Allows having full control over the system.
- It is easy to maintain and manage coding, designs, etc.
- It does not need to depend on third-party software solutions.

3.4 Architectural Design of The System

The MVC design pattern has been selected for the proposed system. MVC is a software design pattern used to develop web applications. The advantage of the MVC is, that it facilitates in differentiating the layers of a project in Model, View, and Controller to reuse and maintain codes. [8] The proposed architectural design is presented in Figure 3.1.

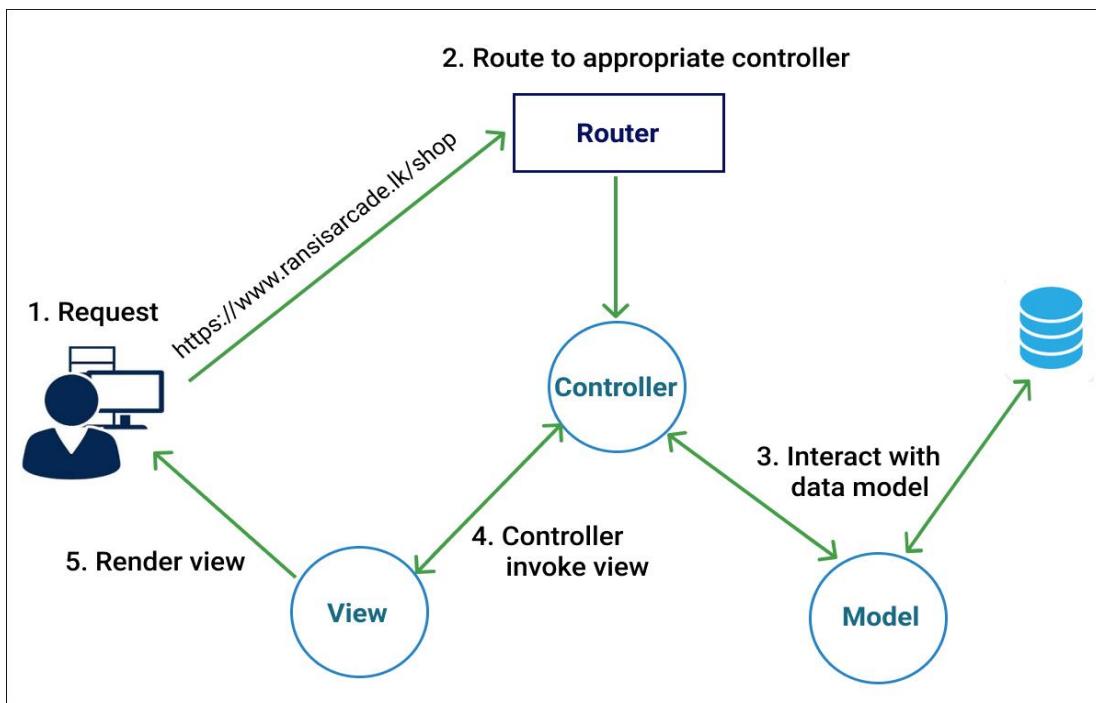


Figure 3.1: The proposed architectural design

- **Model** - Represents the interactions while underlying data with the database are done through here. It serves as a data service layer where data is fetched.
- **View** - Users are presented with an interface and these files are handled here.
- **Controller** - Acts as middle-ware or a bridge between the model and view. [7]

Further, RESTful is used as implementation architecture that is elaborated widely under “Implementation Architecture”.

3.5 Module Design

The overall system is divided into modules. A few of them are graphically represented through use-case diagrams that provide a clear understanding of the system.

3.5.1 Use Case Diagram for Customer Management Module

The customer management module is a module that handles both unregistered and registered customer events. This was elaborated widely under the “Functional requirements of the System”. The use case diagram for the entire system of the customer management module is presented in Figure 3.2.

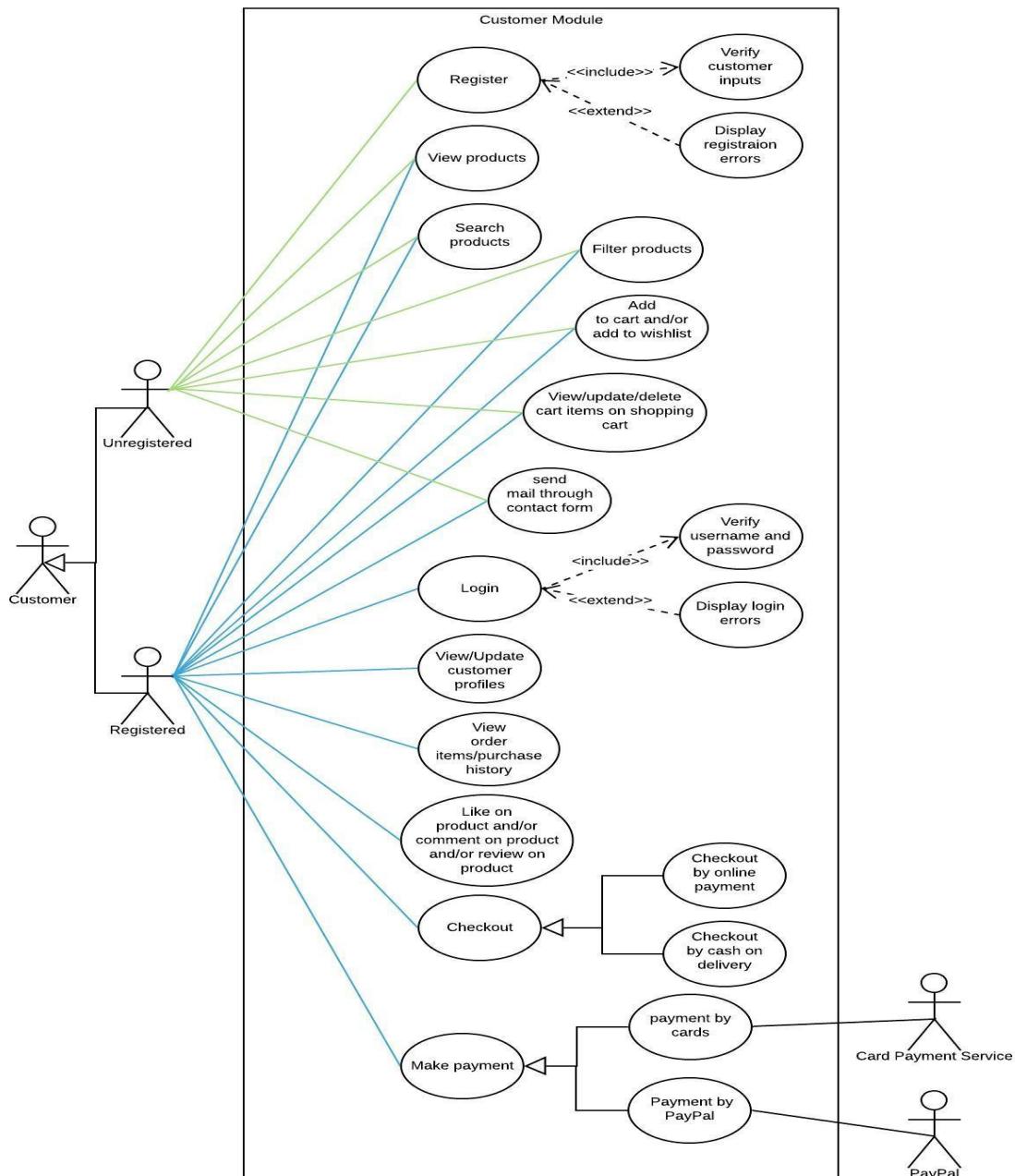


Figure 3.2: The use case diagram for the entire customer management module.

The use case description for customer registration is given in Table 3.1.

Use case	Customer registration
Actor(s)	Customer (unregistered)
Description	
Unregistered customers can be able to register an account.	
Pre-conditions	
The customer can be able to visit the e-commerce web application.	
Flow of events	
<ol style="list-style-type: none"> 1. Visit the customer registration page 2. Enter all required fields in the registration form 3. Submit the form 	
Post-conditions	
<ul style="list-style-type: none"> • If the entered form data are not valid, then display error message(s). Otherwise, data are saved in the database and display a successful registration message and send an email for thanking for registration. 	

Table 3.1: The use case description for customer registration

The use case description for customer account management is given in Table 3.2

Use case	Customer Account Management
Actor(s)	Customer (registered)
Description	
Registered customers should be able to view their accounts, update their accounts, change their account passwords, view their orders and status, and view their ‘wishlist’.	
Pre-conditions	
Registered customers should be able to login to their accounts by submitting a username and password.	
Flow of events	
<ol style="list-style-type: none"> 1. Login into his/her account by submitting username and password 2. System Automatically directs to his/her account 3. The customer should be able to view his/her account and orders and status, 4. The customer should be able to update his/her account 5. The customer should be able to update his/her account password 	
Post-conditions	

- Successful message should be displayed after successful account updating and password changing. Otherwise, an error message should be displayed.

Table 3.2: The use case description for the customer account management panel

3.5.2 Use Case Diagram for User Access Management Module

This section is where the system controls user access by blocking unauthorized accesses, directing relevant user interfaces after logins, creating/deleting admin accounts, etc. The use case diagram for the user access management module is presented in Figure 3.3.

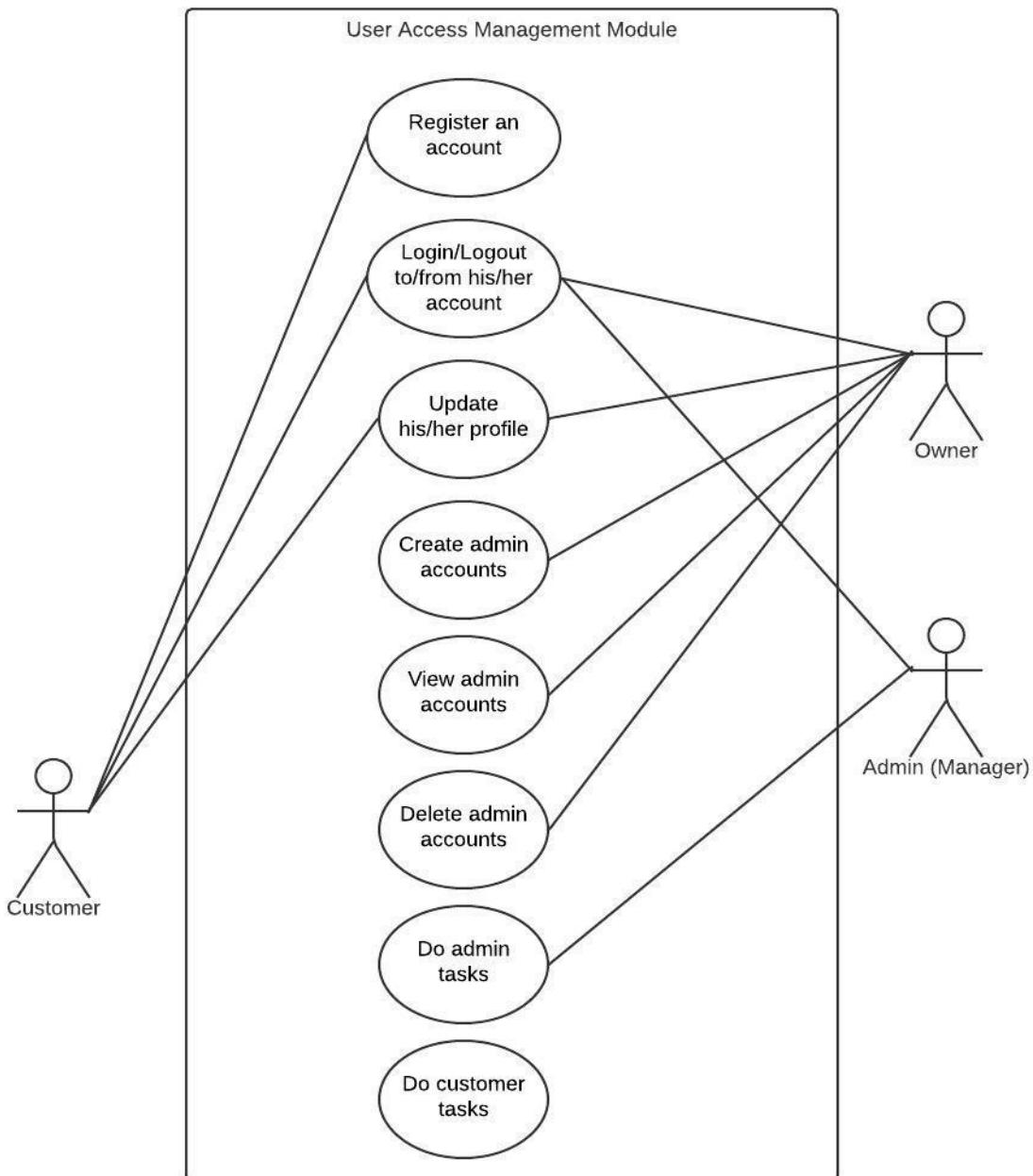


Figure 3.3: The use case diagram for user access management module

The use case description for adding a new admin account is given in Table 3.3.

Use case	Add a new admin account
Actor(s)	Owner
Description	
The only owner should be able to add a new admin account	
Pre-conditions	
The owner should be logged into the system panel.	
Flow of events	
1. Visit the admin registration page 2. Enter all required fields in the registration form 3. Submit form	
Post-conditions	
<ul style="list-style-type: none">• If the entered form data are not valid, then display error message(s). Otherwise, entered data is saved in the database and displays a successful registration message.	

Table 3.3: The use case description for adding a new admin account

3.5.3 Use Case Diagram for Product Management Module

Usually, management of the system is done by the admin but the owner also has equal rights to do anything. Admin should be able to add, edit, view, and delete products. The use case diagram for the product module is presented in Figure 3.4.

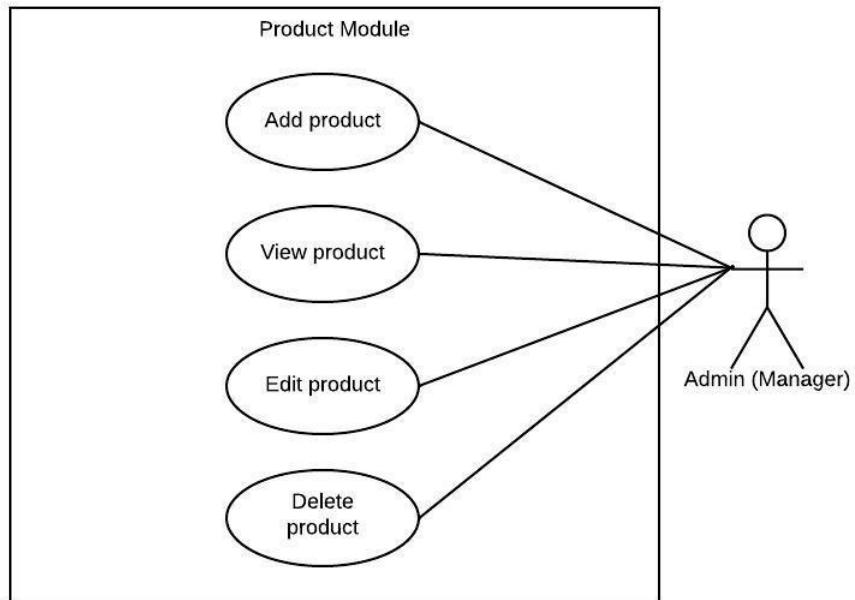


Figure 3.4: The use case diagram for the product management module.

The use case description for adding a new product is given in Table 3.4.

Use case	Add a new product
Actor(s)	Admin
Description	
Admin should be able to add a new product	
Pre-conditions	
Admin should be logged into the system panel.	
Flow of events	
<ol style="list-style-type: none"> 1. Visit the ‘add product’ page 2. Enter all required fields in the add product form 3. Submit form 	
Post-conditions	
<ul style="list-style-type: none"> • If the entered form data is not valid, then display error message(s). Otherwise, entered data is saved in the database and displays a success message. 	

Table 3.4: The use case description for adding a new product

3.5.4 Use Case Diagram for Shopping Cart Module

Customers/clients should be able to add products to the shopping cart, view the shopping cart, delete the cart items, and select the payment option before the checkout process. The use case diagram for the shopping cart module is presented in Figure 3.5.

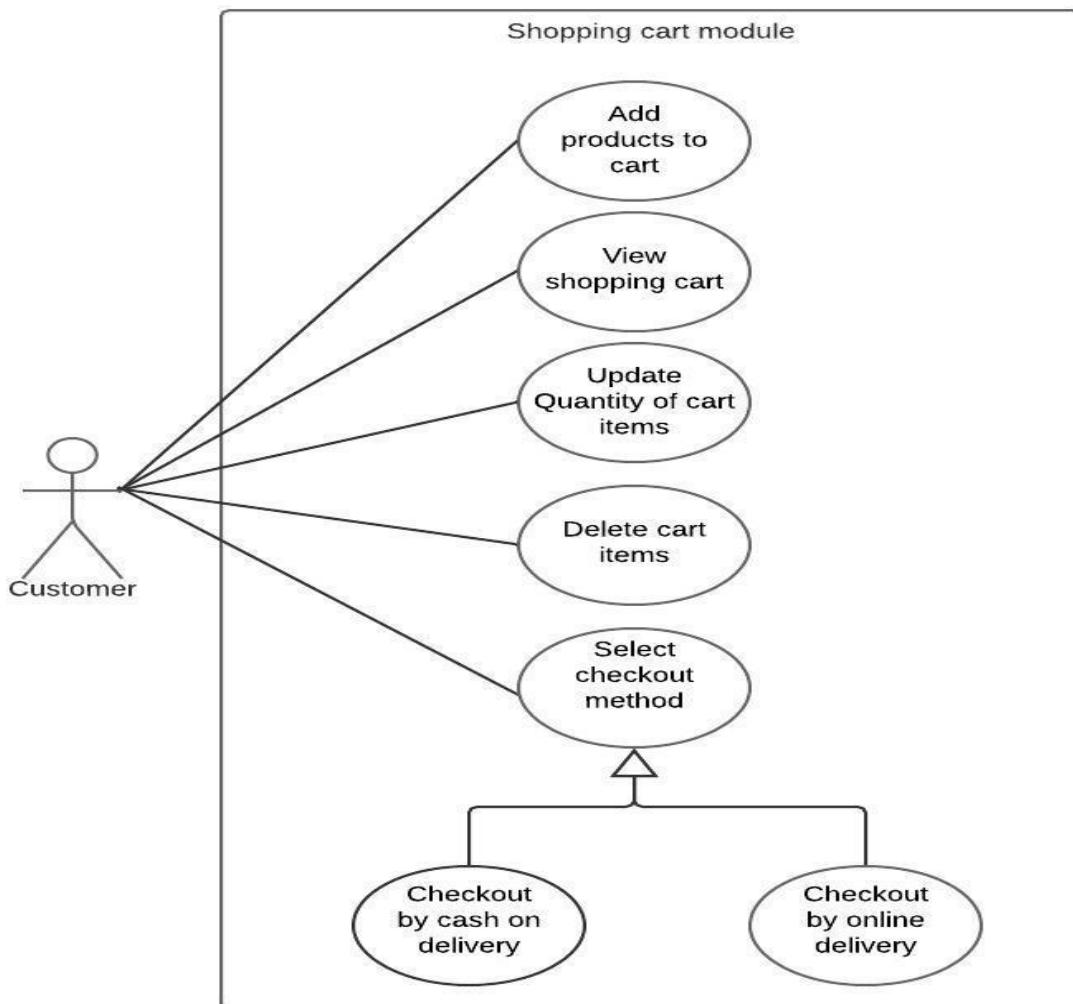


Figure 3.5: The use case diagram for the shopping cart module

The use case description checkout process is given in Table 3.5.

Use case	Checkout process
Actor(s)	Customer
Description	
Customers can add products to the shopping cart, view the shopping cart, update the number of cart items, delete the cart items, and select the payment option before the checkout process.	
Pre-conditions	
The customer should be able to visit the e-commerce web application.	

Flow of events
<ol style="list-style-type: none"> 1. Add products to the shopping cart by clicking ‘add to cart’ button 2. Visit the shopping cart page 3. Increase and decrease buying quantities of cart items 4. Delete unwanted items 5. Select checkout option either ‘cash on delivery checkout’ or ‘online payment checkout’ 6. Click checkout button
Post-conditions
<ul style="list-style-type: none"> • If the customer is already logged, direct to the appropriate checkout page. Otherwise, direct to the login page.

Table 3.5: The use case description checkout process

3.5.5 class diagram for the e-commerce system

The class diagram is the main building block of object-oriented modeling and static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram for the e-commerce web application is presented in Figure 3.6.

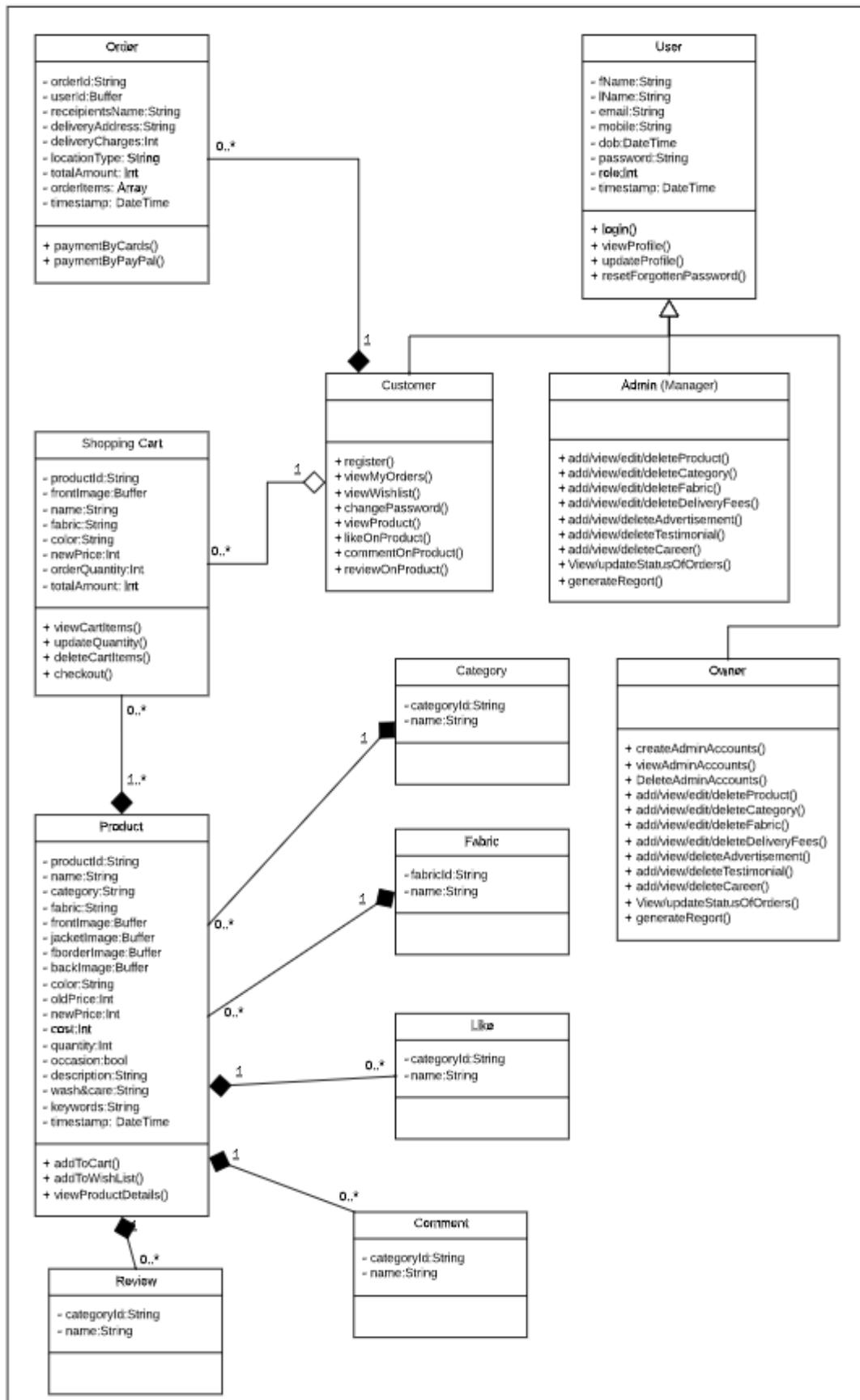


Figure 3.6: The class diagram for the e-commerce web application.

3.5.6 Sequence Diagram for Customer Registration

The sequence diagram simply represents the interaction between objects in sequential order. The sequence diagram for customer registration is presented in Figure 3.7.

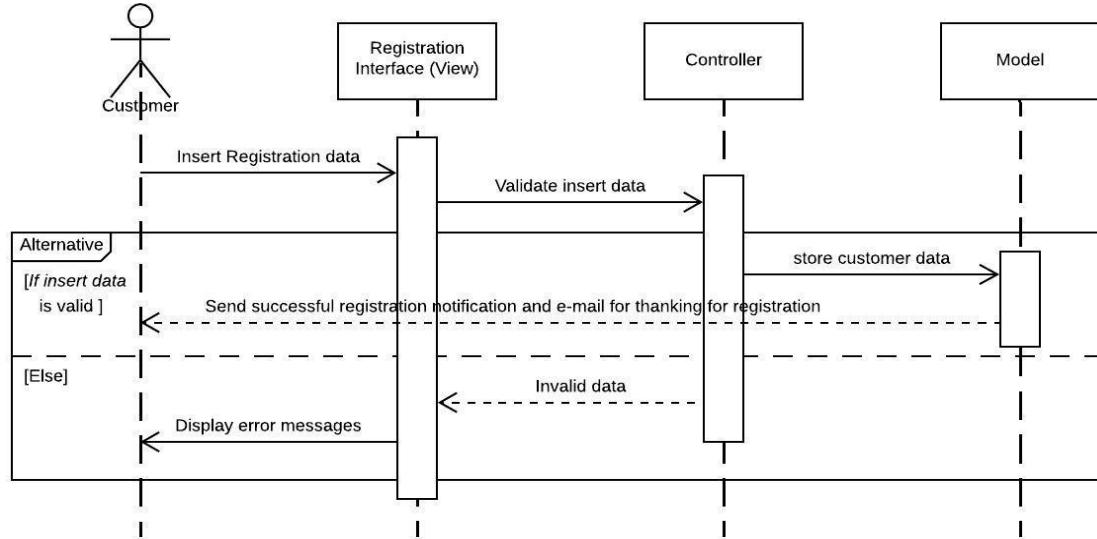


Figure 3.7: The sequence diagram for customer registration

3.5.7 Sequence Diagram for User Login

The sequential order of user login is presented in Figure 3.8.

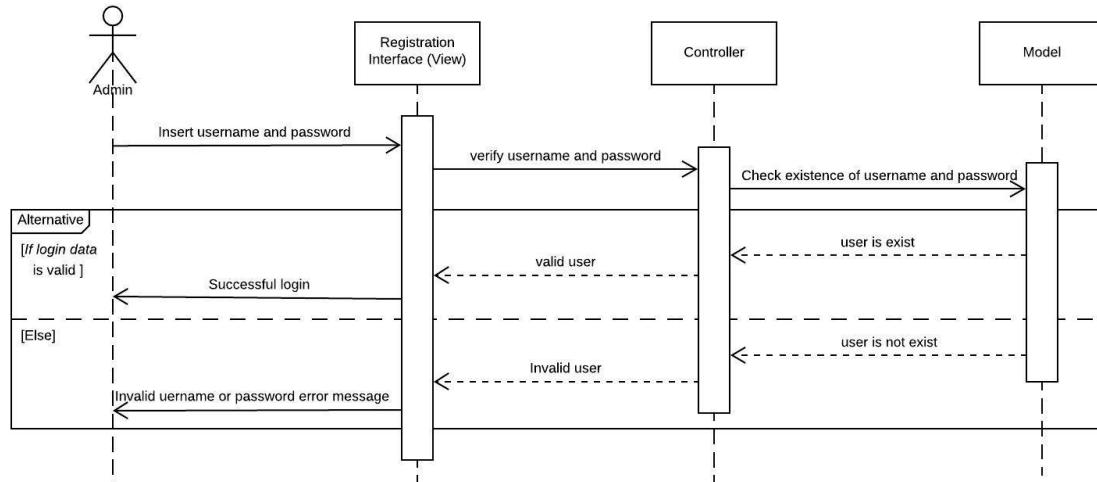


Figure 3.8: The sequence diagram for customer registration.

3.6 User Interface Design

User interface design is one of the most significant parts of designing. A good user interface could lead to better user experience, efficiency, and productivity. To ensure

that good interfaces are designed and well organized, the following design principals were followed,

- **Unique web design** – No templates were used for this project. Each and every component either small or big was designed and developed by me. Therefore, this is a unique design.
- **Consistent color theme** – The color theme was decided according to the business logo. [9] The color theme of the entire web design is given in Figure 3.9.

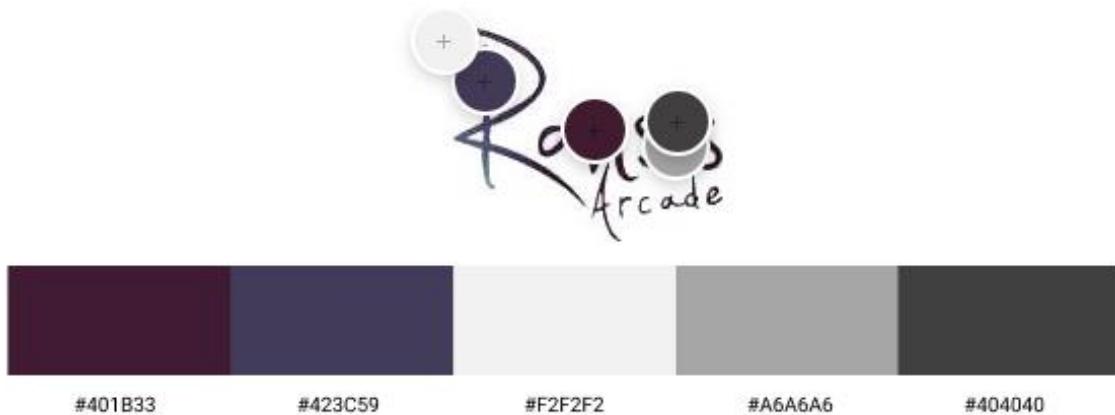


Figure 3.9: The color theme of entire web design.

- **Alert messages** - Displaying red color alert box for error messages, green color alert box for success messages, and orange color alert box for warning messages in all pages. Different kinds of alert messages are presented in Figure 3.10.

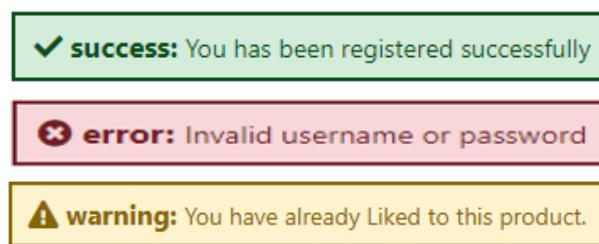


Figure 3.10: Different kinds of alert messages

- **Validation** - Form validations were used to validate the user input data.
- **Graphical icons** - Different types of icons were used to make the system familiar.
- **Charts/Tables** - Used for better data presentation.

Some of the user design interfaces are as follows;

3.6.1 Navbar

Navbar of the RAEWA will be navigated to the main pages of the system such as a home, shop, customer registration, and login, etc. The navbar of the system is presented in Figure 3.11.

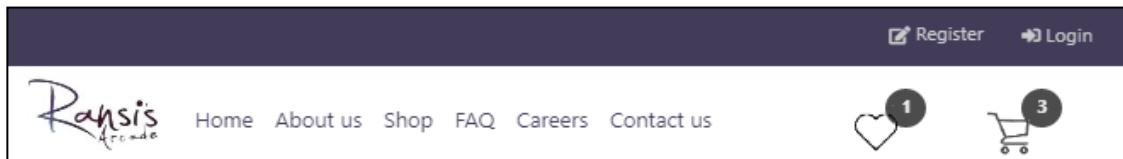


Figure 3.11: Navbar of the website

3.6.2 Footer

The footer of the RAEWA is presented in Figure 3.12 which navigates to quick links, support links, legal and privacy links, social media links and contact details.

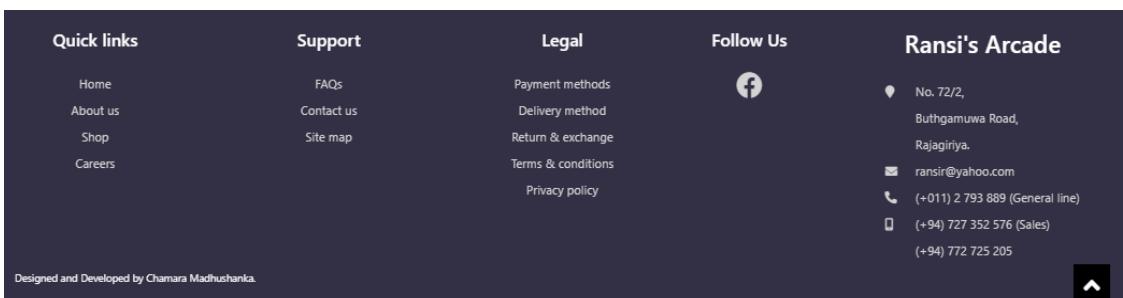


Figure 3.12: Footer of the website

3.6.3 Product Search Option

Products can be searched by its name, category, fabric, color, etc. the product search option is presented in Figure 3.13.



Figure 3.13: Product search option

3.6.4 Basic Filtering Option (Single value filtering)

Basic filtering can be done by category, fabric, occasion, price, color, likes, ratings, etc. The basic filtering options of the shop page are presented in Figure 3.14.

OCCASION	PRICE	CATEGORY	COLOR	LIKES
Casual	Rs. 0 - Rs. 2,500	Urban Drape	■ - Black	 - Like
Party	Rs. 2,501 - Rs. 5,000	Mendes	□ - White	
Office	Rs. 5,001 - Rs. 7,500		■ - Red	
Cocktail	Rs. 7,501 - Rs. 10,000		■ - Blue	
Wedding & Engagement	Rs. 10,001 - Rs. 15,000	Pure Cotton	■ - Yellow	
	Rs. 15,001 - Rs. 20,000	Pure Silk	■ - Green	
	Rs. 20,001 - Rs. 25,000	Cotton with Metallic Threading	■ - Orange	
	Rs. 25,001 - Rs. 35,000		■ - Pink	
	Rs. 35,001 - Rs. 50,000		■ - Grey	
	Rs. 50,001 - Rs. 100,000		■ - Purple	
			■ - Brown	

COMMENTS

Comment

RATINGS



Figure 3.14: Basic filtering options

3.6.5 Advanced Filtering Option (Multivalue filtering)

Advanced filtering can be done by category, fabric, occasion, price, color, etc.

The advanced filtering options of the shop page are presented in Figure 3.15.

Category	Fabric	Occasion	Price	Color
<input style="width: 100%; height: 100%;" type="button" value="Any"/>	<input style="width: 100%; height: 100%;" type="button" value="Any"/>	<input style="width: 100%; height: 100%;" type="button" value="Any"/>	<input style="width: 100%; height: 100%;" type="button" value="Any"/>	<input style="width: 100%; height: 100%;" type="button" value="Any"/>
<input style="width: 100%; background-color: #555; color: white; font-weight: bold; padding: 5px; border: none;" type="button" value="Filter Now"/>				

Figure 3.15: Advanced filtering option

3.6.6 Product cards

Product cards consist of the main image, four clickable thumb images and other details such as ratings, category, product name, new price, old price, likes, and comments with add to cart, add to wishlist, and view details. Logged customers are able to like and comment on products. Product cards used in the entire system are presented in Figure 3.16.

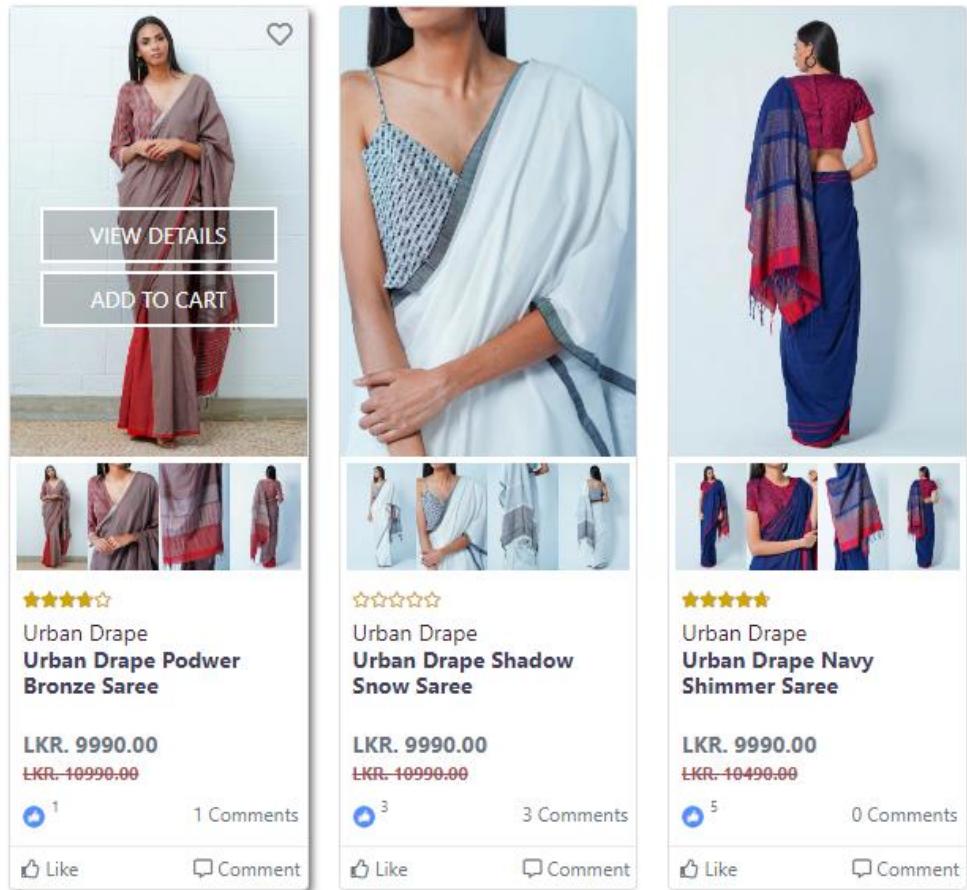


Figure 3.16: Product cards

3.6.7 Comment on Products

Customers are able to comment on products and view comments. The comment model is presented in Figure 3.17.

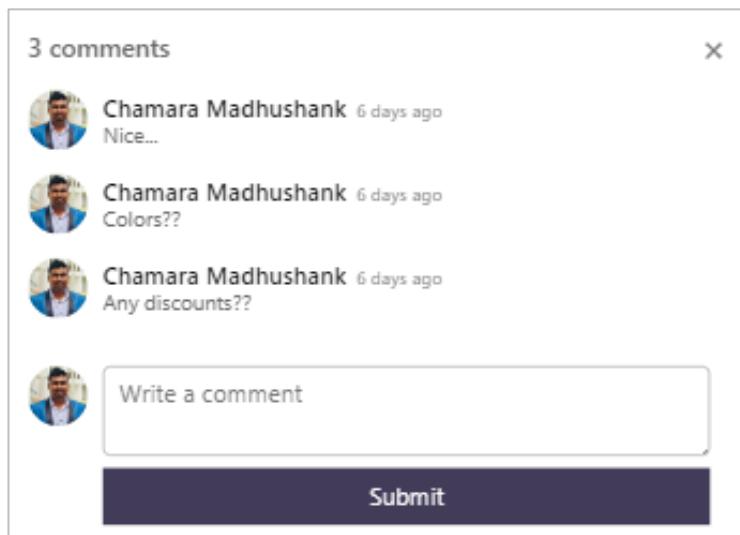


Figure 3.17: The comment model

3.6.8 Review and Rate on products

The review and rate on products are presented in Figure 3.18.

Reviews  3.8



Chamara Madhushank 6 days ago



wow



Chamara Madhushank 6 days ago



nice



Submit

Figure 3.18: The review and rate on products

3.6.9 Shopping Cart

Customers are able to navigate to shopping cart from any page and view shopping without login and can update the cart before checkout. The shopping cart is presented in Figure 3.19.

Shopping Cart

Product	Name	Color	Price	Qty	Subtotal		
	Urban Drape Podwer Bronze Saree	mixed with red and gray	LKR. 9990.00	1	LKR. 9990.00		
	Urban Drape Navy Shimmer Saree	blue saree is paired with a red hand woven blouse	LKR. 9990.00	1	LKR. 9990.00		
	Urban Drape Shadow Snow Saree	white saree is mixed with light gray and black	LKR. 9990.00	1	LKR. 9990.00		
Total				LKR. 29970.00			
<input checked="" type="radio"/> Online payment <input type="radio"/> Cash on delivery							
« Continue Shopping			Login to Checkout »				

Figure 3.19: Shopping cart interface

3.6.10 Card Payment Interface

Customers can pay through cards or PayPal. The card payment interface is presented in Figure 3.20.

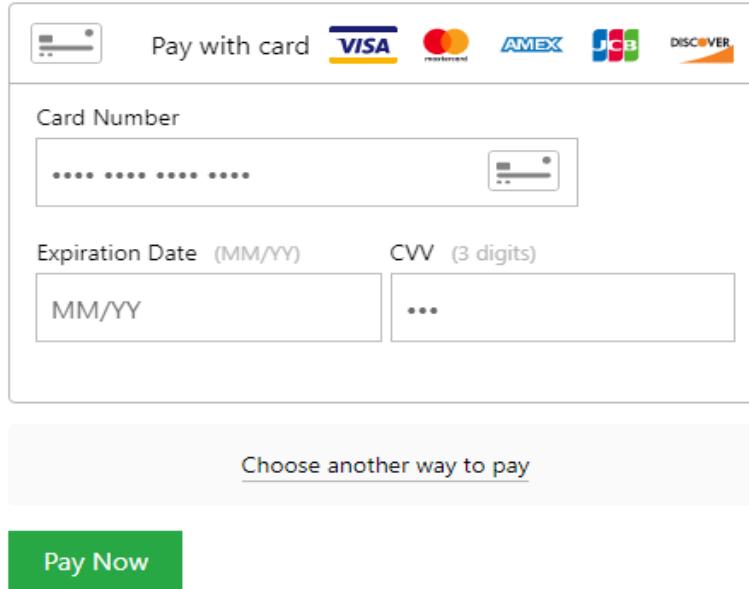


Figure 3.20: Card payment interface

3.6.11 Responsive Web Design

The e-commerce web application is responsive for all devices such as desktops, laptops, tablets, and mobile phones. The responsive web design for RAEWA is presented in Figure 3.21.

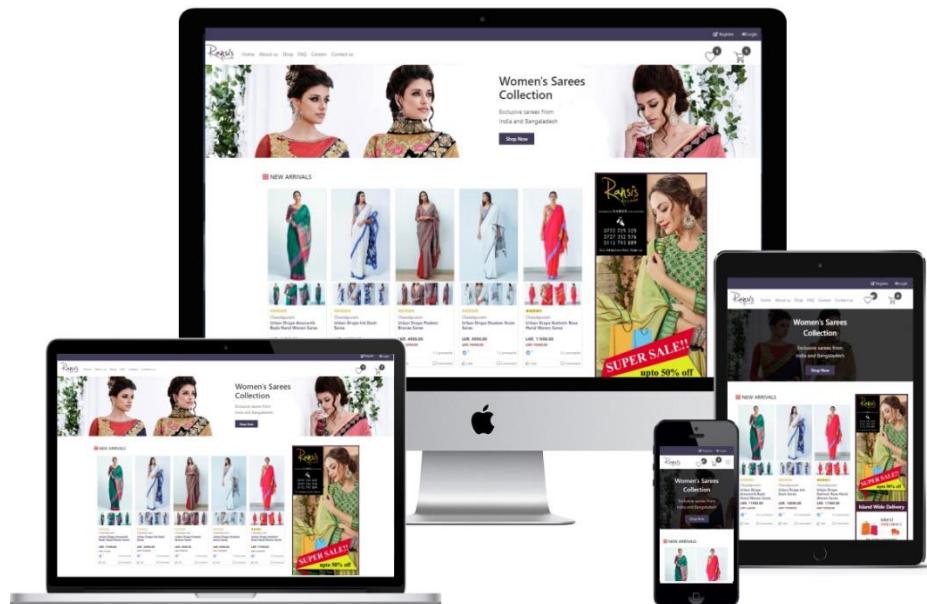


Figure 3.21: Responsive web design for RAEWA

3.6.12 Product Detail Page

The product detail page for a selected product is presented in Figure 3.22.



Urban Drape Navy Shimmer Saree

★★★★★

Rs. 9990.00

This navy blue saree is paired with a red hand woven blouse to give a modern look. This 100% cotton saree is soft and draped beautifully on you. The saree is suitable to wear for a special occasion and best worn with minimal jewellery to complete the look.

 Add to Cart  Add to Wishlist

Delivery within 5 days for online payments
Cash on delivery eligible

Figure 3.22: The product detail page

Chapter 4 – Implementation

4.1 Introduction

Implementation is the phase where the system design is converted into an executable program while fulfilling the requirements. The ultimate goal is to deliver a complete working system.

E-commerce web application is developed using latest technologies and various tools upon considering future maintenance. Coding standards and best practices were followed when engaging in the coding process.

4.2 Implementation Environment

The implementation environment describes the hardware and software that were required to develop and operate the system.

4.2.1 Hardware Requirement

Developing Environment

Laptop computer including following core features;

- Intel Core i3 2348M @ 2.30GHz processor
- 4.00GB Dual-Channel DDR3 RAM
- 2GB Intel HD Graphics 3000 (HP) VGA
- 500GB (SATA) Hard Disk
- Internet connection

Client Environment

- Desktop, Laptop, tablet or mobile phone with Internet facility

4.2.2 Software Requirements for development

Operating System

- Windows 10 Pro 64-bit

Documentation

- Microsoft Word for creating the dissertation
- Lucid Chart for creating professional UML models

- Adobe Reader for reading pdf files

4.2.2.3 Design and Photo Editing Tools

- Figma online tool for UI designs
- Photopea online photoshop tool for photo editing

4.2.2.4 Code Editors

- Visual Studio Code
- Sublime Text

4.2.2.5 Web Browsers

Test web site on all popular web browsers such as;

- Google chrome
- Mozilla Firefox
- Edge
- Safari etc.

4.3 Development Technologies

The development technologies used for the system is given Table 4.1.

Technology	Description
Frontend Development	
React	React (also knowns as React.js or ReactJS) is the fastest-growing JavaScript library for building user interfaces. React can be used as a single-page or mobile applications. It also has an awesome virtual DOM which provides invaluable performance to the environment. [10]
CSS	CSS for styling
Bootstrap	Bootstrap for building responsive components. [11]
Backend Development	
Node	Node (also knowns as Node.js or NodeJS) is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node is asynchronous programming which best suitable for e-commerce web applications. [12]

Express	Express (also known as Express.js) is a web application framework for Node and used for developing web applications and APIs. [13]
Both Frontend and Backend Development	
JavaScript	JavaScript is a scripting language which used for both client-side and server-side scripting in this project
Database	
MongoDB	MongoDB is an open-source database management system (DBMS) that uses a document-oriented database model that supports various forms of data. MongoDB architecture is made up of collections and documents instead of using tables and rows as in relational databases. [14]
Mongoose	Mongoose is an object data modeling (ODM) library that provides a precise modeling environment for data and connects the model and the database. [15]
Data Exchange between Client and Server	
JSON	JSON is a lightweight data exchange format between a browser and a server. It is easy for humans to read and understand. [16]

Table 4.1: Development technologies

4.4 Existing Reusable Packages

The npm is the world's largest Software library. It contains over 800,000 open-source code packages. The following npm packages were used for development.

- **Body parser** - body-parser for extracting the entire body portion of an incoming request stream and exposes it on req.body. JSON, buffer, string, and URL encoded are parsed by body-parser. [17]
- **Jsonwebtoken** – JSON web token for user authentication. [18]
- **Bcrypt** - The bcrypt hashing function for building hash passwords. [19]
- **Gravatar** - Gravatar stands for Globally Recognized Avatar for getting globally recognized avatar from gravatar servers. [20]
- **Axios** - Axios for performing HTTP requests [201]
- **Validator** – Validator for validating user inputs. [22]

- **Braintree and Braintree Web Drop-in React** - The Braintree Node library provides integration access to the Braintree Gateway to process online payments through credit cards, debit cards, digital wallets, PayPal, etc. [23]
- **Nodemailer** - Nodemailer for sending emails. [24]
- **Moment** - A lightweight JavaScript date library for parsing, validating, manipulating, and formatting dates [25]
- **Formidable** - Formidable is a Node.js module for parsing form data, including multipart/form-data file upload. [26]
- **Font Awesome Icons** - An icon library for displaying graphical icons. [27]
- **React input mask** - Allows the validation of specific data formats such as phone numbers, and Dates, etc. [28]
- **React rating** - React Rating for rating products. [29]
- **React share** - React share for social media sharing. [30]
- **React helmet** - React helmet for setting titles in the head section. [31]
- **CanvasJSReact** - CanvasJSReact for creating interactive graphical charts [32]
- **Classnames** - Classnames for conditionally joining class names together. [33]

4.5 Implementation Architecture

REST (REpresentational State Transfer) is used as implementation architecture for this project that is maintainable, lightweight, and scalable. A service that is built on the REST architecture is known a RESTful service. The underlying protocol for REST is HTTP web layer and it uses the below key verbs to work with resources on the server;

- POST - To create resources
- GET - To retrieve resources
- PUT - To update the state of resources
- DELETE - To delete resources

Any RESTful web service has to comply with the RESTful Client-Server, stateless, cacheable, and layered system characteristics completely for it to be called RESTful. [36]

In addition, as mentioned in the previous chapters, MVC architecture is used to develop and maintain applications easily.

4.6 Justification for Selecting RESTful Architecture

Below listed are the reasons for selecting the RESTful Architecture:

1. Heterogeneous languages and environments

- Communication between web applications on distinct programming languages are made simpler
- Web applications are allowed to stay in different environments like where some could be on Windows, and others could be on Linux with the aid of services.

Social media like Facebook, Twitter, and search engines like Google show their functionality in the form of Restful web services. Therefore, allowing any client application to call these web services via REST.

2. The event of Devices

Everything has to work on mobile devices. Let it be mobile devices, notebooks, tablet or even car systems. It's even a challenge for normal web applications, but RESTful APIs make this job less complex because as mentioned earlier, it doesn't require to know what the underlying layer for the device is.

3. The event of the Cloud

The trend is the cloud. Amazon and Azure have moved to cloud already. Therefore, it has become a necessity to be made compatible with the Cloud. Also, because all Cloud-based architectures work on the REST principle, it is the better choice for web services to be programmed on the REST-based architecture so that it could be used to make the maximum use of Cloud-based services. [34]

4.7 Module and Folder Structure

As mentioned in the design chapter, the RAEWA was built on MVC design pattern (to enhance the efficiency of the system and overcome of performance related issues). Given diagram (folder structure) shows the way the model, view and controller are being organized. Moreover, using this MVC design pattern facilitates code reusability, easy maintenance and high flexibility. Figure 4.1 illustrates the module and folder structure

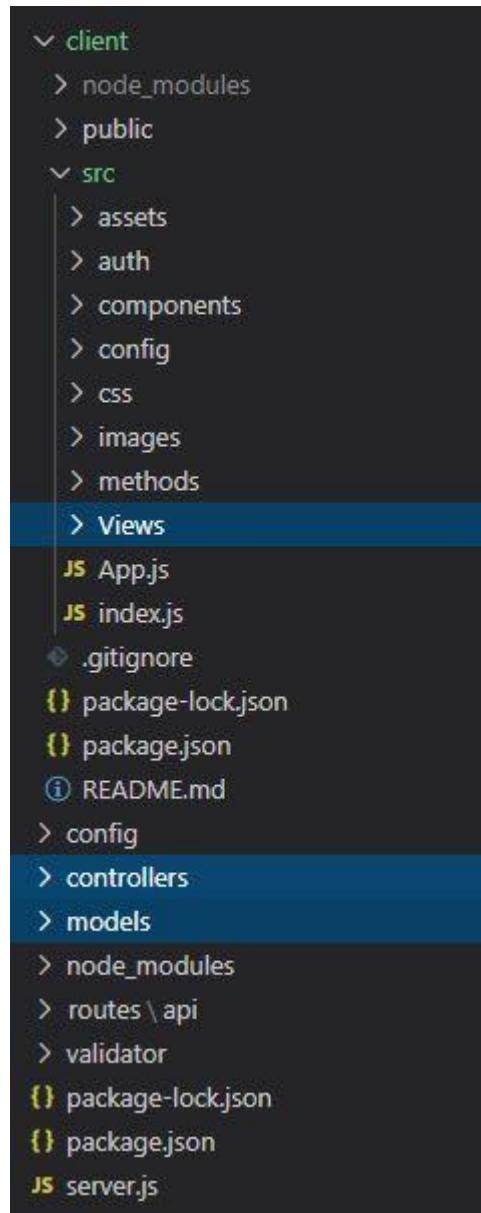


Figure 4.1: module and folder structure

4.8 Major Code Segments

Some of the major code segments are given below and the rest can be found in Appendix F.

4.8.1 Server Configuration

Express is used as a framework for Node for building web applications and APIs. Server configuration is the very first part of backend development as follows.

```

// import express package
const express = require("express");

// app
const app = express();

// port
const PORT = process.env.PORT || 5000;

// run the server
app.listen(PORT, () => console.log(`server is running in Port ${PORT}`));

```

4.8.2 Connect Database

Mongoose is an object data modeling (ODM) library that is used to communicate with MongoDB. The below code illustrates how MongoDB is connected using mongoose.

```

// import mongoose package
const mongoose = require("mongoose");

// connect database
mongoose
  .connect("mongodb://localhost:27017/ecommerce", {
    useNewUrlParser: true,
    useCreateIndex: true
  })
  .then(result => console.log("db is running"))
  .catch(err => console.log(err));

```

4.8.3 User Registration

Register the user after validating data, checking email and mobile existence. Bcrypt function is used for password hashing and gravatar used for pulling globally recognized avatar from gravatar servers according to given email. [19,20] The below code illustrates how to register a user.

```

// customer registration controller
exports.customerRegistration = (req, res, next) => {
  const { errors, isValid } = userRegisterValidator(req.body);

  // check data validity
  if (!isValid) {
    res.status(400).json(errors);
  } else {
    User.findOne({ email: req.body.email })
      .exec()
      .then(user => {
        // check email is already exist or not
        if (user) {
          errors.email = "Email already exists";
          return res.status(400).json(errors);
        } else {
          User.findOne({ mobile: req.body.mobile })
            .exec()
            .then(userMobile => {
              // check mobile is already exist or not
              if (userMobile) {
                errors.mobile = "Mobile already exists";
                return res.status(400).json(errors);
              } else {
                // hash the password
                bcrypt.hash(req.body.password, 10, (err, hash) => {
                  if (err) {
                    return res.status(400).json({ error: "Password error" });
                  } else {
                    // get profile picture from gravatar servers
                    const avatar = gravatar.url(req.body.email, {
                      s: "200", // size
                      r: "pg", // rating
                      d: "mm" // default image
                    });
                    // create new user object
                    const user = new User({
                      fName: req.body.fName,
                      lName: req.body.lName,
                      image: avatar,
                      email: req.body.email,
                      dob: req.body.dob,
                      mobile: req.body.mobile,
                      role: req.body.role,
                      password: hash
                    });
                    // save to the database
                    user
                      .save()
                      .then(result => {
                        res.status(200).json(result);
                      })
                      .catch(err => {
                        res.status(400).json({ error: "user is not added" });
                      });
                  }
                });
              }
            });
        }
      });
  }
};


```

4.8.4 User Login

User login is done after verifying username and password. JWT (JSON Web Token) is a token-based authentication used in this project. JWT is digitally signed since

information can be trusted and verified. JWT can be signed using a secret with the HMAC algorithm or a public/private key pair using RSA or ECDSA. The below code illustrates how users are logged in.

```
// user login
exports.userLogin = (req, res, next) => {
  const { errors, isValid } = userLoginValidator(req.body);

  // check data validity
  if (!isValid) {
    res.status(400).json(errors);
  } else {
    User.findOne({ email: req.body.email })
      .exec()
      .then(user => {
        // verify email as username
        if (!user) {
          errors.message = "Invalid username or password";
          res.status(400).json(errors);
        } else {
          // verify password
          bcrypt.compare(req.body.password, user.password, (err, result) => {
            if (result) {
              // create jwt token after password verification is sucessful
              const token = jwt.sign(
                {
                  email: user.email,
                  _id: user._id,
                  image: user.image
                },
                secretOfKey
              );

              //set token to cookie
              res.cookie("t", token, { expire: new Date() + 3600 });
              res.status(200).json({ token: token, user: user });
            } else {
              errors.message = "Invalid username or password";
              res.status(400).json(errors);
            }
          });
        }
      })
      .catch(err => {
        res.status(400).json({ error: err });
      });
  }
};
```

4.8.5 CRUD Operations

4.7.5.1 Read Products

The following code illustrates how to read all products from the database and send them to the frontend.

```

/**
 * sold - /products?sortBy=sold&order=desc&limit=6
 * latest arrival - /product?sortBy=createdAt&order=asc&limit=6
 * if there's no any query params, all products are returned
 */

// read products
exports.readProducts = (req, res, next) => {
  const sortBy = req.query.sortBy ? req.query.sortBy : "_id";
  const order = req.query.order ? req.query.order : "asc";
  const limit = req.query.limit ? parseInt(req.query.limit) : "";

  Product.find()
    .select("-imageFront -imageJacket -imageBorder -imageBack")
    .populate("category", "name")
    .populate("fabric", "name")
    .sort([[sortBy, order]])
    .limit(limit)
    .then(result => {
      if (result.length > 0) {
        res.status(200).json(result);
      } else {
        res.status(200).json({ msg: "No Products" });
      }
    })
    .catch(err => {
      res.status(500).json({ error: err });
    });
};

```

4.8.5.2 Create Product

The formidable module is used for parsing form data, including multipart/form-data file upload. The following code illustrates how to create a product and save in the database and send feedback to the frontend.

```

// create product
exports.createProduct = (req, res, next) => {
  const form = new formidable.IncomingForm();
  form.keepExtensions = true;
  form.parse(req, (err, fields, files) => {
    // validate data through validator
    const { errors, isValid } = productValidator(fields, files);

    // validate data
    if (!isValid) {
      res.status(400).json(errors);
    } else {
      // create new product object
      const product = new Product(fields);
      // save images in database
      if (files.imageFront) {
        product.imageFront.data = fs.readFileSync(files.imageFront.path);
        product.imageFront.contentType = files.imageFront.type;
      }
    }
  });
};

```

```

        if (files.imageJacket) {
            product.imageJacket.data = fs.readFileSync(files.imageJacket.path);
            product.imageJacket.contentType = files.imageJacket.type;
        }

        if (files.imageBorder) {
            product.imageBorder.data = fs.readFileSync(files.imageBorder.path);
            product.imageBorder.contentType = files.imageBorder.type;
        }

        if (files.imageBack) {
            product.imageBack.data = fs.readFileSync(files.imageBack.path);
            product.imageBack.contentType = files.imageBack.type;
        }

        // save to database
        product
            .save()
            .then(result => {
                res.status(200).json(result);
            })
            .catch(err => {
                errors.fail = "Product is not added successfully";
                res.status(400).json(errors);
            });
    });
}

```

4.8.5.3 Update Product

The following code illustrates how to update a product from the database.

```

// update product
exports.updateProduct = (req, res, next) => {
    Product.findOne({ _id: req.params.id })
        .exec()
        .then(product => {
            const form = new formidable.IncomingForm();
            form.keepExtensions = true;
            form.parse(req, (err, fields, files) => {
                if (err) {
                    res.status(400).json({ error: "image was not uploaded" });
                }

                product = _.extend(product, fields);

                if (files.frontImage) {
                    product.frontImage.data = fs.readFileSync(files.frontImage.path);
                    product.frontImage.contentType = files.frontImage.type;
                }

                if (files.jacketImage) {
                    product.jacketImage.data = fs.readFileSync(files.jacketImage.path);
                    product.jacketImage.contentType = files.jacketImage.type;
                }
            });
        });
}

```

```

    // save product on database
    product
      .save()
      .then(result => {
        res.status(200).json(result);
      })
      .catch(err => {
        res.status(400).json({ error: err });
      });
    });
  )
  .catch(err => {
    res.status(500).json({ error: err });
  });
};


```

4.8.5.4 Delete Product

The following code illustrates how to delete a product from the database.

```

// delete product
exports.deleteProduct = (req, res, next) => {
  // find relevant product using id and delete
  Product.deleteOne({ _id: req.params.id })
    .exec()
    .then(result => {
      res.status(200).json(result);
    });
};


```

4.8.6 Add to Cart Method

The cart is not stored in the database. Instead, the local storage mechanism was used to store items in the cart. The following code illustrates items are stored in the cart by clicking add to cart button.

```

// add to cart
export const addToCart = (cartItem, next) => {
  // create empty array variable called cart
  let cart = [];

  // check window is not undefined
  if (typeof window !== "undefined") {
    if (localStorage.getItem("cart")) {
      // get all cart items from local storage
      cart = JSON.parse(localStorage.getItem("cart"));
    }

    // add count into the cart
    cart.push({
      ...cartItem,
      count: 1
    });
  }
};


```

```

    // remove duplicated cart items
    cart = Array.from(new Set(cart.map(item => item._id))).map(id => {
      return cart.find(item => item._id === id);
    });

    // store cart on local storage
    localStorage.setItem("cart", JSON.stringify(cart));
    next();
  }
);

```

4.8.7 Remove from Cart Method

The following code illustrates how to items are removed from the cart by clicking the remove button.

```

// remove a cart item
export const removeCartItem = id => {
  // create empty array variable called cart
  let cart = [];

  // check window is not undefined
  if (typeof window !== "undefined") {
    if (localStorage.getItem("cart")) {
      // get all cart items from local storage
      cart = JSON.parse(localStorage.getItem("cart"));
    }

    // remove from cart using splice method
    cart.map((item, i) => {
      if (item._id === id) {
        cart.splice(i, 1);
      }

      // store cart on local storage
      localStorage.setItem("cart", JSON.stringify(cart));
    });
  }
};

```

4.8.8 Payment Gateway

Braintree is used as the payment gateway which is an international payment gateway that serves over forty-six counties and accepts Visa, Mastercard, and American Express as well as Apple Pay, Android Pay, PayPal, etc. The following codes illustrate how to connect Braintree API and generate Braintree tokens.

```

// connect braintree to the API
const gateway = braintree.connect({
  environment: braintree.Environment.Sandbox,
  merchantId: braintreeMerchantId,
  publicKey: braintreePublicKey,
  privateKey: braintreePrivateKey
});

// generate token
router.get("/braintree/getToken/:userId", requireLogin, isAuthenticated, (req, res) => {
  // generate client token
  gateway.clientToken.generate({}, (err, response) => {
    if (err) {
      res.status(500).json(err);
    } else {
      res.status(200).json(response);
    }
  });
});

```

4.8.9 Security

Since the system contains sensitive data and information related to the users and business processes, it is important to implement good security procedures in order to limit access to sensitive information by unauthorized users.

Passwords are encrypted and salted using BCRYPT hash algorithm. Below is the code snippet for hashing password when registering users and verifying the password when login.

```

// hash the password
let salt = bcrypt.genSaltSync(10);
const hash = bcrypt.hashSync(req.body.password, salt);

// verify password
let compare = bcrypt.compareSync(req.body.password, user.password);

```

4.8.10 Protection of User Routes

The three types of users in RAEWA are customers, administrators, and the owner. Therefore, routes are protected to control user access as a security purpose. The following code snippets illustrate how to protect customer routes.

```
// create a private route for customers
const PrivateRoute = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
    render={props =>
      isAuthenticated() && isAuthenticated().user.role === 0 ? (
        <Component {...props} />
      ) : (
        <Redirect
          to={{ pathname: "/login", state: { from: props.location } }}
        />
      )
    }
  />
);
```

Chapter 5 - Evaluation

5.1 Introduction

Testing is the process of evaluating a system or its components to find whether it satisfies the system requirements or not. Further testing is evaluating a system to identify any gaps, errors or missing requirements.

This chapter describes the testing approach, test plan, test cases, proof of testing of work with the result of work and its end-user evaluation.

5.2 Testing Approach

RAEWA was tested in parallel since parallel testing is a component of the RUP development methodology as an iterative increment. The following testing methods were carried out with the given order.

1. Unit testing - unit testing was done on each block of code during development.
2. Integration testing - this testing is performed before, during and after integration of a new module into the web application. This involves testing of each code module. It is essential to test each module's effect on the entire program model.
3. System testing - the entire system was tested as a whole before it is introduced to the market.
4. Acceptance testing - beta testing was done by the end-users.

Further, regression testing is done to make sure none of the changes made throughout the development process have created new bugs. It also makes sure no old bugs arise from the addition of new software modules over time. [35]

5.3 Test Plan and Test Cases

A software testing plan is a methodological approach to testing a software system. An appropriate test plan is designed before introducing the system with test objectives, schedules, testing methods, and test cases.

As mentioned above, different testing methods were used to fully test the system under different conditions and situations. Further, test cases are the most important part of a

test plan. Therefore, test cases are designed separately to reduce the complexity of the testing process. Some of the test cases are given below and the rest of the test results can be found in Appendix E.

5.3.1 Customer Registration

The test cases for the customer registration module is given Table 5.1.

ID	Test Description	Testing Procedures	Expected Results
CR1	Register a new customer	Fill in all the necessary details at appropriate fields and click register button	Customer registered along with success message and email will be sent
CR2	Email address uniqueness	Enter an existing email address and click register button	Display ‘email already exists’ error message
CR3	Mobile number uniqueness	Enter an existing mobile number and click register button	Display ‘Mobile number already exists’ error message
CR4	Data validation	Enter invalid data or leave the mandatory fields blank	Display appropriate error messages

Table 5.1: The test cases for the customer registration process

5.3.2 User Login (Customer/Admin/Owner)

The test cases for the user login process is given Table 5.2.

ID	Test Description	Testing Procedures	Expected Results
UL1	Valid login	Enter correct email and password	Login to customer/admin/Owner account
UL2	Invalid login	Enter incorrect email or password	Display ‘invalid login’ error message
UL3	Data validation	Enter invalid data or leave fields blank	Display appropriate error messages

Table 5.2: The test cases for the user login process

5.3.3 Customer Account Management

The test cases for the customer account management process is given Table 5.3.

ID	Test Description	Testing Procedures	Expected Results
CAM1	View an existing customer account detail	Login and select the relevant account	Display related customer details
CAM2	Update customer account	Enter valid data and click update button	Customer updated along with success message
CAM2	Change account password	Enter valid data and click change password button	Changed password along with success message
CAM3	Data validation	Enter invalid data or leave fields blank	Display appropriate error messages
CAM4	View customer orders	Click 'My order' link and see	Display orders were done by customers
CAM5	Logout	Click 'logout' button	Logged out and redirect to login page

Table 5.3: The test cases for the customer account management process

5.4 Test Data and Test Results

Both dummy and actual data were used to ensure the accuracy of each form field values. A common set of error messages were designed and displayed in appropriate places to maintain consistency across the entire system.

Some of the test results are given below and the rest of the test results can be found in Appendix E.

Module: Customer Registration

Following table illustrates test data and test results of customer registration process that includes checking email and mobile number uniqueness, data validation, success and fail alert message, etc.

Test ID	Test Case	Actual Output	Status
CR1	Register a new customer	<div style="background-color: #e0f2e0; padding: 5px; border: 1px solid #80c080;"><p>✓ success: You has been registered successfully</p></div>	Pass
CR2	Email address uniqueness	<p>Email*</p> <input type="text" value="nlc.madhushanka@gmail.com"/> <p>Email already exists</p>	Pass

CR3	Mobile number uniqueness	<p>Mobile number*</p> <input type="text" value="(+94) 71 390 7528"/> <p>Eg: Mobile Format (+94) 7X XXX XXXX</p> <p>Mobile already exists</p>	Pass
CR4	Data validation	<p>First Name*</p> <input type="text"/> <p>First name is required</p> <p>Last Name*</p> <input type="text"/> <p>Last name is required</p> <p>Email*</p> <input type="text"/> <p>Email is required</p> <p>Mobile number*</p> <input type="text"/> <p>Eg: Mobile Format (+94) 7X XXX XXXX</p> <p>Mobile is required</p> <p>Date of Birth*</p> <input type="text" value="DD/MM/YYYY"/> <p>Date of birth is required</p> <p>Password*</p> <input type="text"/> <p>Password is required</p> <p>Confirm Password*</p> <input type="text"/> <p>Confirm password is required</p> <p>Register</p>	Pass

Table 5.4: The test results for the customer registration process

Module: User Login (Customer/Admin/Owner)

Following table illustrates test data and test results of user login process.

Test ID	Test Case	Actual Output	Status

UL1	Valid login (Customer)	 <p>Chamara</p> <p>Update Profile My orders Change Password</p> <table border="1"> <thead> <tr> <th colspan="2">PERSONAL DETAILS</th> </tr> </thead> <tbody> <tr> <td>First Name</td> <td>Chamara</td> </tr> <tr> <td>Last Name</td> <td>Madhushanka</td> </tr> <tr> <td>Date of Birth</td> <td>06/11/1993</td> </tr> <tr> <td>Mobile No</td> <td>(+94) 713907524</td> </tr> <tr> <td>E-mail</td> <td>nlcm.gunathilaka@gmail.com</td> </tr> </tbody> </table>	PERSONAL DETAILS		First Name	Chamara	Last Name	Madhushanka	Date of Birth	06/11/1993	Mobile No	(+94) 713907524	E-mail	nlcm.gunathilaka@gmail.com	Pass
PERSONAL DETAILS															
First Name	Chamara														
Last Name	Madhushanka														
Date of Birth	06/11/1993														
Mobile No	(+94) 713907524														
E-mail	nlcm.gunathilaka@gmail.com														
UL2	Invalid login	<p>✗ error: Invalid username or password</p>	Pass												
UL3	Data validation	<p>Email</p> <input type="text"/> <p>Email is required</p> <p>Password</p> <input type="password"/> <p>Password is required</p> <p>Forgot your Password?</p> <p>Login</p> <p>Please click here to register now</p>	Pass												

Table 5.5: The test results for the user logins

Module: Customer Account Management

Following table illustrates test data and test results of customer account management.

Test ID	Test Case	Actual Output	Status												
CAM1	View an existing customer account detail	 <p>Chamara</p> <p>Update Profile My orders Change Password</p> <table border="1"> <thead> <tr> <th colspan="2">PERSONAL DETAILS</th> </tr> </thead> <tbody> <tr> <td>First Name</td> <td>Chamara</td> </tr> <tr> <td>Last Name</td> <td>Madhushanka</td> </tr> <tr> <td>Date of Birth</td> <td>06/11/1993</td> </tr> <tr> <td>Mobile No</td> <td>(+94) 713907524</td> </tr> <tr> <td>E-mail</td> <td>nlcm.gunathilaka@gmail.com</td> </tr> </tbody> </table>	PERSONAL DETAILS		First Name	Chamara	Last Name	Madhushanka	Date of Birth	06/11/1993	Mobile No	(+94) 713907524	E-mail	nlcm.gunathilaka@gmail.com	Pass
PERSONAL DETAILS															
First Name	Chamara														
Last Name	Madhushanka														
Date of Birth	06/11/1993														
Mobile No	(+94) 713907524														
E-mail	nlcm.gunathilaka@gmail.com														
CAM2	Update customer account	<p>✓ success: Your account has been updated successfully</p>	Pass												

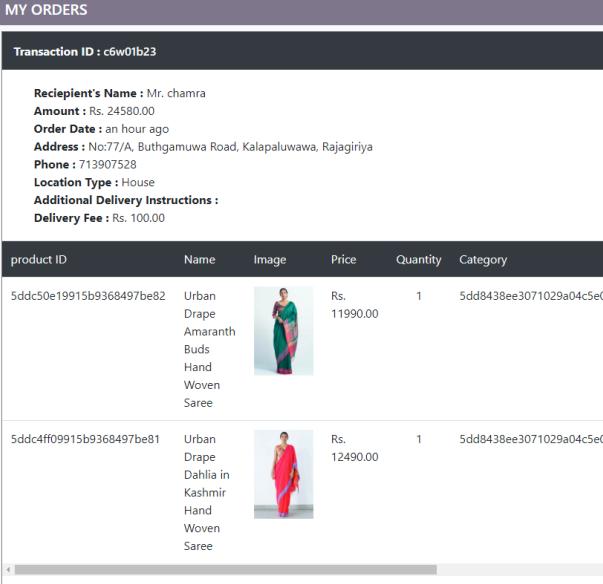
CAM2	Change account password		Pass																		
CAM3	Data validation	<p>UPDATE YOUR PROFILE</p> <p>First Name <input type="text"/> <small>First name is required</small></p> <p>Last Name <input type="text"/> <small>Last name is required</small></p> <p>Mobile No <input type="text"/> <small>(+94) 71 390 7524</small></p> <p>Date of Birth <input type="text"/> <small>DD/MM/YYYY</small> <small>Date of birth is required</small></p> <p>E-mail <input type="text"/> <small>nlc.m.gunathilaka@gmail.com</small></p> <p>CHANGE YOUR PASSWORD</p> <p>Old Password <input type="text"/> <small>Please enter old password</small> <small>Old password is required</small></p> <p>New Password <input type="text"/> <small>Please enter new password</small> <small>New password is required</small></p> <p>Confirm Password <input type="text"/> <small>Please Confirm your password</small> <small>Confirm password is required</small></p> <p>MY ORDERS</p> <p>Transaction ID : c6w01b23</p> <p>Recipient's Name : Mr. chamra Amount : Rs. 24580.00 Order Date : an hour ago Address : No:77/A, Buthgamuwa Road, Kalapaluwawa, Rajagiriya Phone : 713907528 Location Type : House Additional Delivery Instructions : Delivery Fee : Rs. 100.00</p> <table border="1"> <thead> <tr> <th>product ID</th> <th>Name</th> <th>Image</th> <th>Price</th> <th>Quantity</th> <th>Category</th> </tr> </thead> <tbody> <tr> <td>5ddc50e19915b9368497be82</td> <td>Urban Drape Amaranth Buds Hand Woven Saree</td> <td></td> <td>Rs. 11990.00</td> <td>1</td> <td>5dd8438ee3071029a04c5e0f</td> </tr> <tr> <td>5ddc4ff09915b9368497be81</td> <td>Urban Drape Dahlia in Kashmir Hand Woven Saree</td> <td></td> <td>Rs. 12490.00</td> <td>1</td> <td>5dd8438ee3071029a04c5e0f</td> </tr> </tbody> </table>	product ID	Name	Image	Price	Quantity	Category	5ddc50e19915b9368497be82	Urban Drape Amaranth Buds Hand Woven Saree		Rs. 11990.00	1	5dd8438ee3071029a04c5e0f	5ddc4ff09915b9368497be81	Urban Drape Dahlia in Kashmir Hand Woven Saree		Rs. 12490.00	1	5dd8438ee3071029a04c5e0f	Pass
product ID	Name	Image	Price	Quantity	Category																
5ddc50e19915b9368497be82	Urban Drape Amaranth Buds Hand Woven Saree		Rs. 11990.00	1	5dd8438ee3071029a04c5e0f																
5ddc4ff09915b9368497be81	Urban Drape Dahlia in Kashmir Hand Woven Saree		Rs. 12490.00	1	5dd8438ee3071029a04c5e0f																
CAM4	View customer orders		Pass																		
CAM5	Logout	<p>Email <input type="text"/></p> <p>Password <input type="password"/></p> <p>Forgot your Password?</p> <p>Login</p> <p>Please click here to register now</p>	Pass																		

Table 5.6: The test cases for the customer account management process

5.5 User Acceptance Testing

Customer evaluation was done with five randomly selected customers using a customer evaluation form. Customers who had the opportunity to test the RAEWA were attracted to the system and all of the feedback that were received was positive. Customer evaluation results are presented in Figure 5.1.

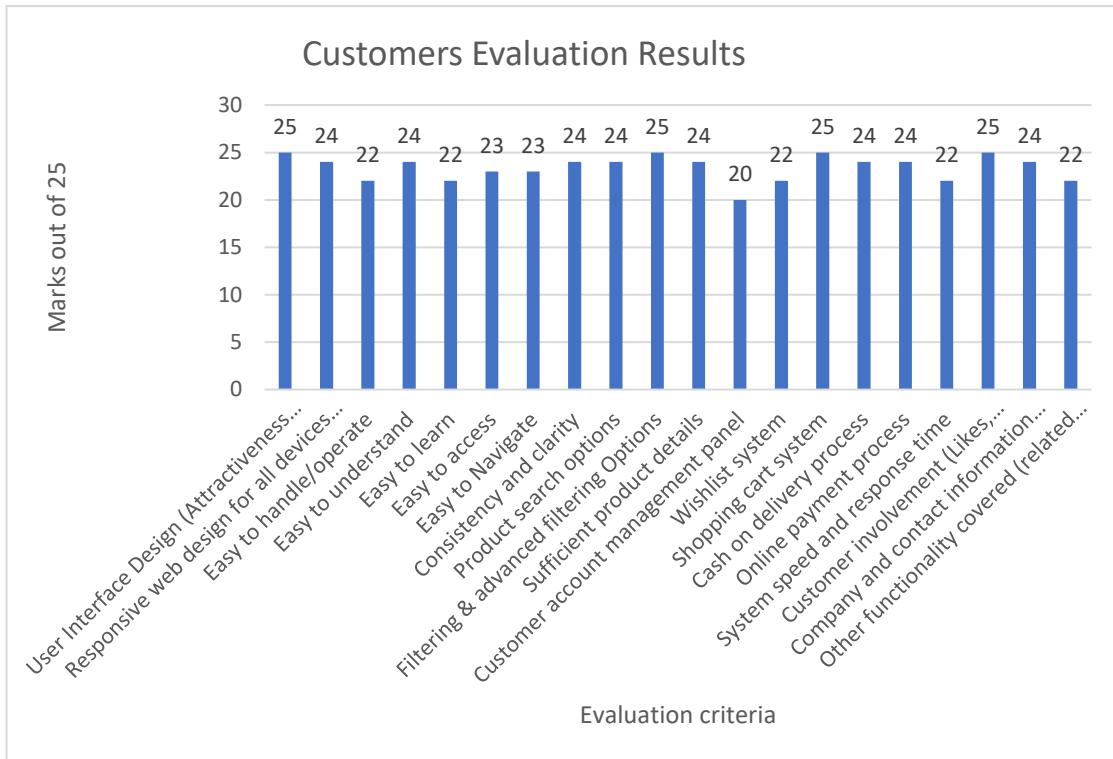


Figure 5.1: Customer evaluation results

Client evaluation was done with the sales manager (Admin) using a client evaluation form. The manager stated that the application was in a better state than expected, especially mentioning the user-friendly and attractive interface was brilliant. The client evaluation form filled by the sales manager is presented in Figure 5.2.

Client Evaluation Form for Ransi's Arcade E-commerce Web Application

Name of Client: S.R.V.P.K. tulatunga

Role of Client: Sales Manager

Please 'tick' in particular cell.

No	Evaluating Item	5 Very Good	4 Good	3 Average	2 Poor	1 Very Poor
1	User Interface Design (Attractiveness of the website)	✓				
2	Responsive web design for all devices (desktop, laptop, tab and mobile)		✓			
3	Easy to handle/operate	✓				
4	Easy to understand	✓				
5	Easy to learn	✓				
6	Easy to access	✓				
7	Easy to Navigate		✓			
8	Consistency and clarity	✓				
9	Product search options	✓				
10	Filtering & advanced filtering Options	✓				
11	Sufficient product details	✓				
12	Customer account management panel		✓			
13	Wishlist system	✓				
14	Shopping cart system	✓				
15	Cash on delivery process	✓				
16	Online payment process	✓				
17	System speed and response time	✓				
18	Customer involvement (Likes, comments, reviews and ratings)	✓				
19	Company and contact information and call to action section	✓				
20	Other functionality covered (related products, social media sharing, FAQ, Career section, advertisement section, testimonials section, latest products, top liked products, top rated products, etc.)	✓				
21	Admin Dashboard with Statistics	✓				
22	Admin Functionality Covered	✓				
23	Management Reports with Charts	✓				
Column tick count		20	3	-	-	-
Multiply by		5	4	3	2	1
Marks		100	12	-	-	-
Total marks out of 115		112				

Comment:

Very successful better than expected.

Ransi's Arcade

S.R.V.P.K. tulatunga

Signature

30.11.2019

Date

Proprietor

Figure 5.2: The client evaluation form filled by the sales manager

Chapter 6 – Conclusion

6.1 Critical Assessment

There was not even a basic website for Ransi's Arcade to attract customers boosting profits. This was the main issue they faced when dealing with new and potential customers. Their target was to increase their sales by 100% and overcome the difficulties in the manual system which was time-consuming, had lots of paper-work, no proper records of sales, no proper records of inventory and difficulty to track frequently moving products and inaccurate data caused by the lack of integrity. Therefore, this e-commerce web application was recommended as the solution to overcome all the above issues and achieve the sales target.

Based on the user's feedback, the developed RAEWA has made a great impact and revolutionized the way the operations were performed before the introduction of RAEWA. In addition, to comply with the client requirements and achieving the objectives, it has also simplified the operations to a significant extent.

The developed system supports all the operations related to the e-commerce process such as shopping cart, customer management, payment gateway, etc. The system allows maintaining information related to products, users, customer orders, etc. In addition, the system generates well documented meaningful reports to assist the top management in the decision-making process.

All the effort has been made to ensure that a simple and attractive user interface was presented with minimal effort and ease on the mind to perform tasks that are related to the system and this was proved with the feedback provided by the users in the evaluation phase. The system guarantees validation of field values to ensure accuracy where before the introduction of RAEWA the chances of making errors in those areas were very high. Special consideration has been given to ensure the security and integrity of the system because of the availability of online transactions and sensitive user information, etc.

Most importantly, this application also allows customers to post their likes, comments, and reviews about products and services. Therefore, the website owner receives feedback and has the ability to use these feedbacks in order to improve services. Keeping in touch with customers helps to survive as a long-term business. Therefore, the

development of the e-commerce web application was a very important ransi's Arcade to gather competitive advantages over other competitors.

6.2 Lessons Learnt

The followings were learned throughout the BIT project that how to;

- Application of the theories learned throughout the BIT degree program into the real-world business environment.
- Gather and analyze the requirements
- Draw UML diagrams such as class, use-case, sequence, and activity
- Design and develop user interfaces using React, Bootstrap, and CSS.
- Implement a RESTful web application through MVC structure.
- Use Node as the backend language.
- Use JSON as data interchange between client-side and server-side.
- Connect Mongodb to the application
- Token based user authentication using JWT (Json Web Tokens)
- Form validation from server-side
- Implement the test process
- Fix bugs found throughout the testing
- Manage the project according to time plan
- Communicate with the supervisor and the client
- Write a dissertation according to given guidelines

6.3 Future Enhancements

The following are recommended as future enhancements;

- Develop an inventory control system and connect to the e-commerce web application to control inventories easily.
- Improve product 'like' system with different reactions such as love, wow, etc.
- Improve product comment system with like and reply option.
- Improve the e-commerce system as a dealer site too where other sellers can sell their product through this website.
- Getting registered with local banks to accept installment payments through credit cards such as HSBC, Sampath Bank, Commercial, etc.

References

- [1] Consultpr.net, Importance of E-commerce Website, October 2010 [Online]. Available: <https://www.consultpr.net/importance-of-e-commerce-website/>. [Accessed: 2-July-2019].
- [2] B. Shah, System Analysis & Design: Fact Finding Techniques, 2008.
- [3] Laveena, “Sarees Worth Wearing”, laveena.lk, 2019. [Online]. Available: <http://www.laveena.lk/>. [Accessed: 10-Sep-2019].
- [4] Nithijaya saree, “Ultimate Shopping Experience in Buying Sarees & Lehngas Online”, nithijayasaree.com, 2019. [Online]. Available: <http://nithijayasaree.com/>. [Accessed: 10- Sep - 2019].
- [5] Wikipedia, Rational Unified Process. [Online]. Available: https://en.wikipedia.org/wiki/rational_unified_process/. [Accessed: 12-Jun-2019].
- [6] Websitesetup.org, 9 Best “Easy” eCommerce Platforms & Sites, 2019. [Online]. Available: <https://websitesetup.org/best-ecommerce-platform/>. [Accessed: 25-May-2019].
- [7] Wikipedia, Content Management System. [Online]. Available: https://en.wikipedia.org/wiki/Content_management_system/. [Accessed: 27-May - 2019].
- [8] Wikipedia, Model View Controller. [Online]. Available: <https://en.wikipedia.org/wiki/Model%20view%20controller/>. [Accessed: 27-May-2019].
- [9] Color wheel, a color palette generator - Adobe Color. [Online]. Available: <https://color.adobe.com/create/>. [Accessed: 5-Jul-2019].
- [10] React - A JavaScript library for building user interfaces. [Online]. Available: <https://reactjs.org/>. [Accessed: 6-Jun-2019].
- [11] Bootstrap. [Online]. Available: <https://getbootstrap.com> [Accessed: 01-May-2019].

- [12] Node.js. [Online]. Available: <https://nodejs.org/en/>. [Accessed: 04-Jul-2019].
- [13] Express. [Online]. Available: <https://www.npmjs.com/package/express/>. [Accessed: 04-Jul-2019]
- [14] MongoDB. [Online]. Available: <https://www.mongodb.com/>. [Accessed: 5-Jul-2019].
- [15] Mongoose. [Online]. Available: <https://www.npmjs.com/package/mongoose/>. [Accessed: 05-Jul-2019].
- [16] JSON. [Online]. Available: <http://www.json.org/>. [Accessed: 06-Jul-2019].
- [17] Body parser. [Online]. Available: <https://www.npmjs.com/package/body-parser/>. [Accessed: 06-Jul-2019].
- [18] Jsonwebtoken. [Online]. Available: <https://www.npmjs.com/package/jsonwebtoken/>. [Accessed: 15-Jul-2019].
- [19] Bcrypt. [Online]. Available: <https://www.npmjs.com/package/bcrypt/>. [Accessed: 15-Jul-2019].
- [20] Gravatar. [Online]. Available: <https://www.npmjs.com/package/gravatar/>. [Accessed: 15-Jul-2019].
- [21] Axios. [Online]. Available: <https://www.npmjs.com/package/axios/>. [Accessed: 2-Aug-2019].
- [22] Validator. [Online]. Available: <https://www.npmjs.com/package/validator/>. [Accessed: 25-Aug-2019].
- [23] Braintree. [Online]. Available: <https://www.npmjs.com/package/braintree/>. [Accessed: 15-Sep-2019].
- [24] Nodemailer. [Online]. Available: <https://www.npmjs.com/package/nodemailer/>. [Accessed: 05-Oct-2019].
- [25] Moment. [Online]. Available: <https://www.npmjs.com/package/moment/>. [Accessed: 20-Oct-2019].

- [26] Formidable. [Online]. Available: <https://www.npmjs.com/package/formidable/>. [Accessed: 23-Aug-2019].
- [27] Font Awesome Icons. [Online]. Available: <https://fontawesome.com> [Accessed: 16-May-2018].
- [28] React input mask. [Online]. Available: <https://www.npmjs.com/package/react-input-mask/>. [Accessed: 18-Oct-2019].
- [29] React rating. [Online]. Available: <https://www.npmjs.com/package/react-rating/>. [Accessed: 28-Oct-2019].
- [30] React share. [Online]. Available: <https://www.npmjs.com/package/react-share/>. [Accessed: 30-Oct-2019].
- [31] React helmet. [Online]. Available: <https://www.npmjs.com/package/react-helmet/>. [Accessed: 02-Nov-2019].
- [32] CanvasJSReact. [Online]. Available: <https://canvasjs.com/react-charts/pie-chart/>. [Accessed: 5-Nov-2019].
- [33] Classnames. [Online]. Available: <https://www.npmjs.com/package/classnames/>. [Accessed: 07-Aug-2019].
- [34] Guru99, RESTful Web Services. [Online]. Available: <https://www.guru99.com/restful-web-services.html/>. [Accessed: 05-Sep-2019].
- [35] Guru99, what is system testing. [Online]. Available: <https://www.guru99.com/system-testing.html/>. [Accessed: 28-Sep-2019].

Appendix A – System Documentation

This document consists of necessary guidelines to be followed when deploying the e-commerce web application. This also includes the minimum hardware and software requirements to operate the system.

Hardware Requirements

Table A.1 illustrates the hardware requirements of the system

Hardware	Recommended Minimum Requirement
Processor	Intel Core i3 or similar
RAM	2GB RAM
Hard disk	250 GB or higher
Display	higher Display 1366 x 786 resolution
A constant internet connection	

Table A.1: Hardware requirements

Software Requirements

Table A.2 illustrates the hardware requirements of the system

Software	Recommended Minimum Requirement
Operating system	Windows 10/ CentOS 7 or any other compatible operating system
Code editor	Visual studio code/sublime text/any other code editor
Web browser	Mozilla Firefox/ Google Chrome/ Safari/ Microsoft Edge

Table A.2: Software requirements

Installation of the E-Commerce System

Step 1 - Install the necessary relevant software:

- Download and install the node to your operating system from <https://nodejs.org/en/download/>
- Download and install a suitable web browser. (Google Chrome is recommended as it is the most popular web browser)
- Download and install the MongoDB community version from <https://www.mongodb.com/download-center/community>

Step 2 - Setting up RAEWA:

- Browse through the given supplementary CD and copy the “Ecommerce” folder and paste it as you wish. As an example, paste on desktop.
- Install node modules for server-side as follows:
 - ✓ Open the command prompt and navigate to “C:\Users\Chamara\Desktop\Ecommerce”.
 - ✓ Type “npm install” and press enter.
 - ✓ All node modules will be installed.
- Install node modules for client-side as follows:
 - ✓ Open the command prompt and navigate to “C:\Users\Chamara\Desktop\Ecommerce\client”.
 - ✓ Type “npm install” and press enter.
 - ✓ All node modules will be installed.

Step 3 - Database Setup:

- Go to “<https://www.youtube.com/watch?v=f395EhMoPKA>” link and watch this video to configure MongoDB after installation.
- Create a database called “ecommerce”.
- Browse through the supplementary CD and select the “ecommerce” folder and restore the database (go to <https://www.youtube.com/watch?v=CHNB38MAvKY> link and watch this video to restore/import).

Step 4 - Launching the System:

- Development server should be started as follows:
 - ✓ Open the command prompt and navigate to “C:\Users\Chamara\Desktop\Ecommerce\client”.
 - ✓ Type “npm start” and press enter.
 - ✓ Development server will start and open the e-commerce web application on default browser as <http://localhost:3000/>.
- After finishing the start of development server, node server should be started as follows:
 - ✓ Again, open another command prompt and navigate to “C:\Users\Chamara\Desktop\Ecommerce”.
 - ✓ Type “npm run server” and press enter.
 - ✓ Node server will be started with the database.

Appendix B - Design Documentation

Use-Case Diagrams and Descriptions

Some of the main use-case diagrams and descriptions are mentioned below which provides an overall overview and understanding about the design of the system.

B.1 Delivery Charges Module

The use case diagram for the delivery charges module is presented in Figure B.1.

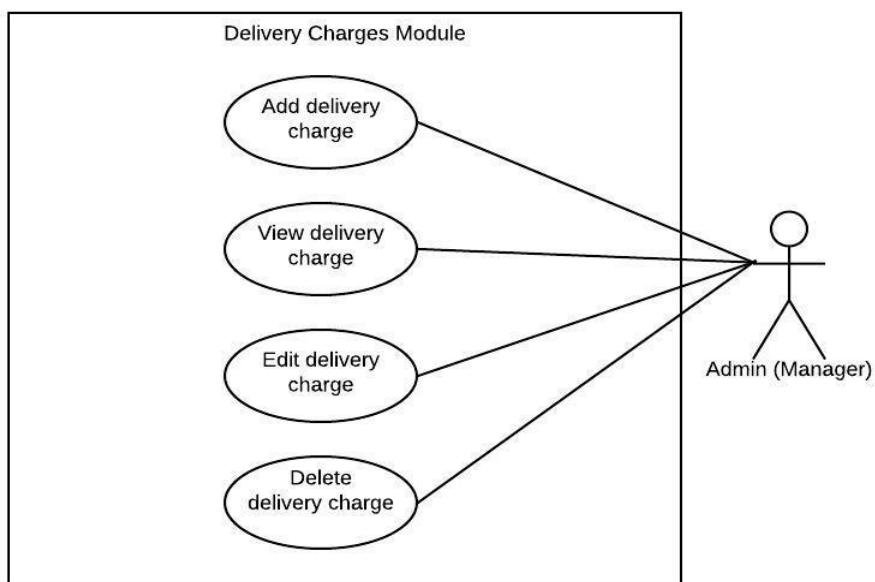


Figure B.1: The use case diagram for the delivery charges module

The use case description of adding a new delivery charge is given in Table B.1.

Use case	Add a new delivery charge
Actor(s)	Owner/admin
Description	
Owner/admin should be able to add a new delivery charge	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
1. Visit the 'add delivery charge' page 2. Enter all required fields in the form 3. Submit form	
Post-conditions	

- If form data is not valid, then display error message(s). Otherwise, the data is saved in the database and displays a success message.

Table B.1: The use case description of adding a new delivery charge

B.2 Customer Order Module

The use case diagram for the customer order module is presented in Figure B.2.

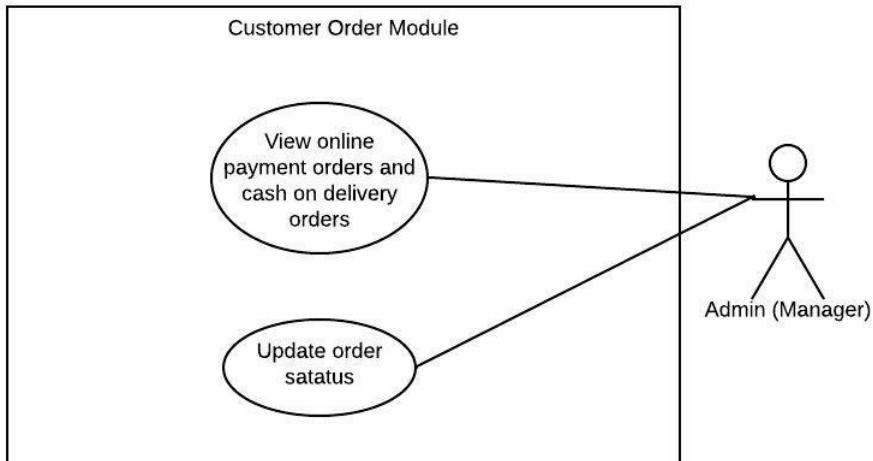


Figure B.2: The use case diagram for the customer order module

The use case description of updating customer order status is given in Table B.2.

Use case	Update customer order status
Actor(s)	Owner/admin
Description	
Owner/admin should be able to update the customer order status	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
<ol style="list-style-type: none"> 1. Visit the ‘customer order’ page 2. Select order status from the status dropdown 3. Automatically update 	
Post-conditions	
<ul style="list-style-type: none"> • If form data is not valid, then display error message(s). Otherwise, the data is saved in the database and displays a success message. 	

Table B.2: The use case description of updating the customer order status

B.3 Advertisement Module

The use case diagram for the advertisement module is presented in Figure B.3.

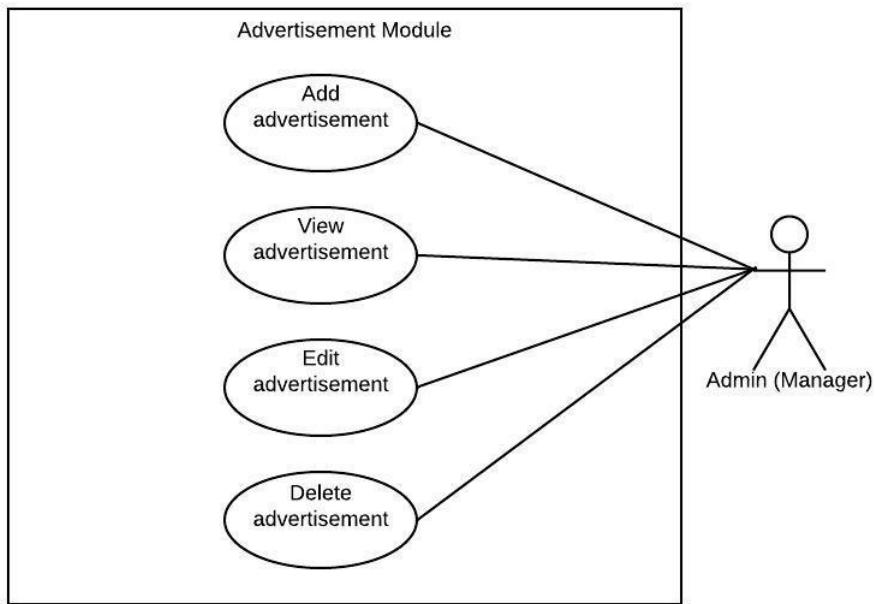


Figure B.3: The use case diagram for the advertisement module

The use case description of adding a new advertisement is given in Table B.3.

Use case	Add a new advertisement
Actor(s)	Owner/admin
Description	
Owner/admin should be able to add a new advertisement	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
<ol style="list-style-type: none"> 1. Visit the ‘add advertisement’ page 2. Enter all required fields in the form 3. Submit form 	
Post-conditions	
<ul style="list-style-type: none"> • If form data is not valid, then display error message(s). Otherwise, the data is saved in the database and displays a success message. 	

Table B.3: The use case description of adding a new advertisement

B.4 Testimonial Module

The use case diagram for the testimonial module is presented in Figure B.4.

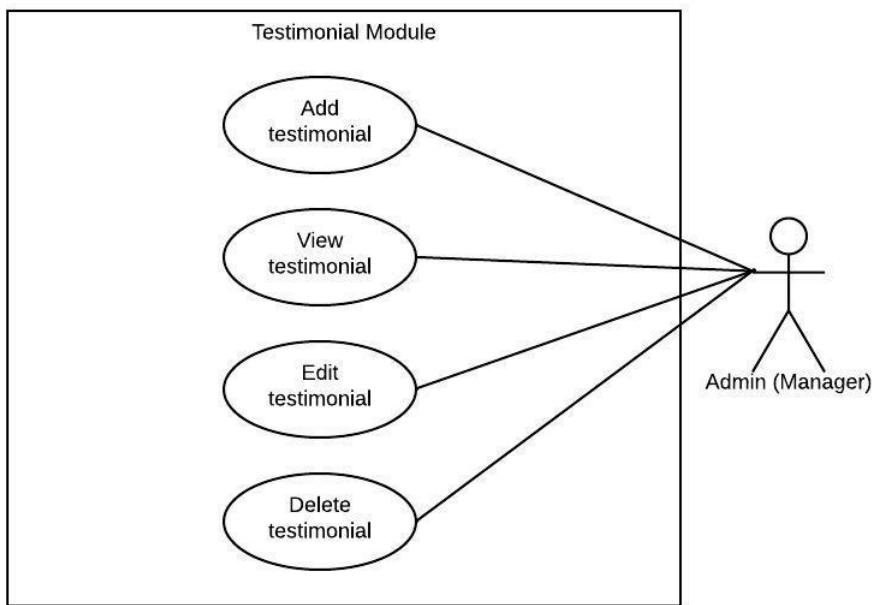


Figure B.4: The use case diagram for the testimonial module

The use case description of adding a new testimonial is given in Table B.4.

Use case	Add a new testimonial
Actor(s)	Owner/admin
Description	
Owner/admin should be able to add a new testimonial	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
<ol style="list-style-type: none"> 1. Visit the ‘add testimonial’ page 2. Enter all required fields in the form 3. Submit form 	
Post-conditions	
<ul style="list-style-type: none"> • If form data is not valid, then display error message(s). Otherwise, the data is saved in the database and displays a success message. 	

Table B.4: The use case description of adding a new testimonial

B.5 Career Module

The use case diagram for the career module is presented in Figure B.5.

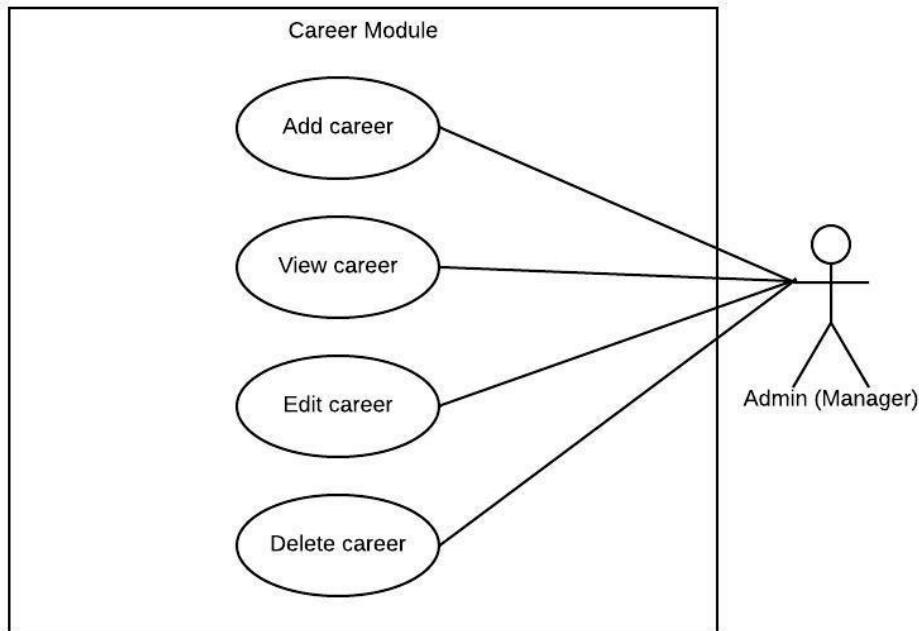


Figure B.5: The use case diagram for the career module

The use case description of adding a new career is given in Table B.5.

Use case	Add a new career
Actor(s)	Owner/admin
Description	
Owner/admin should be able to add a new career	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
1. Visit the 'add career' page 2. Enter all required fields in the form 3. Submit form	
Post-conditions	
<ul style="list-style-type: none">If form data is not valid, then display error message(s). Otherwise, the data is saved in the database and displays a success message.	

Table B.5: The use case description of adding a new career

B.6 Report Module

The use case diagram for the report module is presented in Figure B.6.

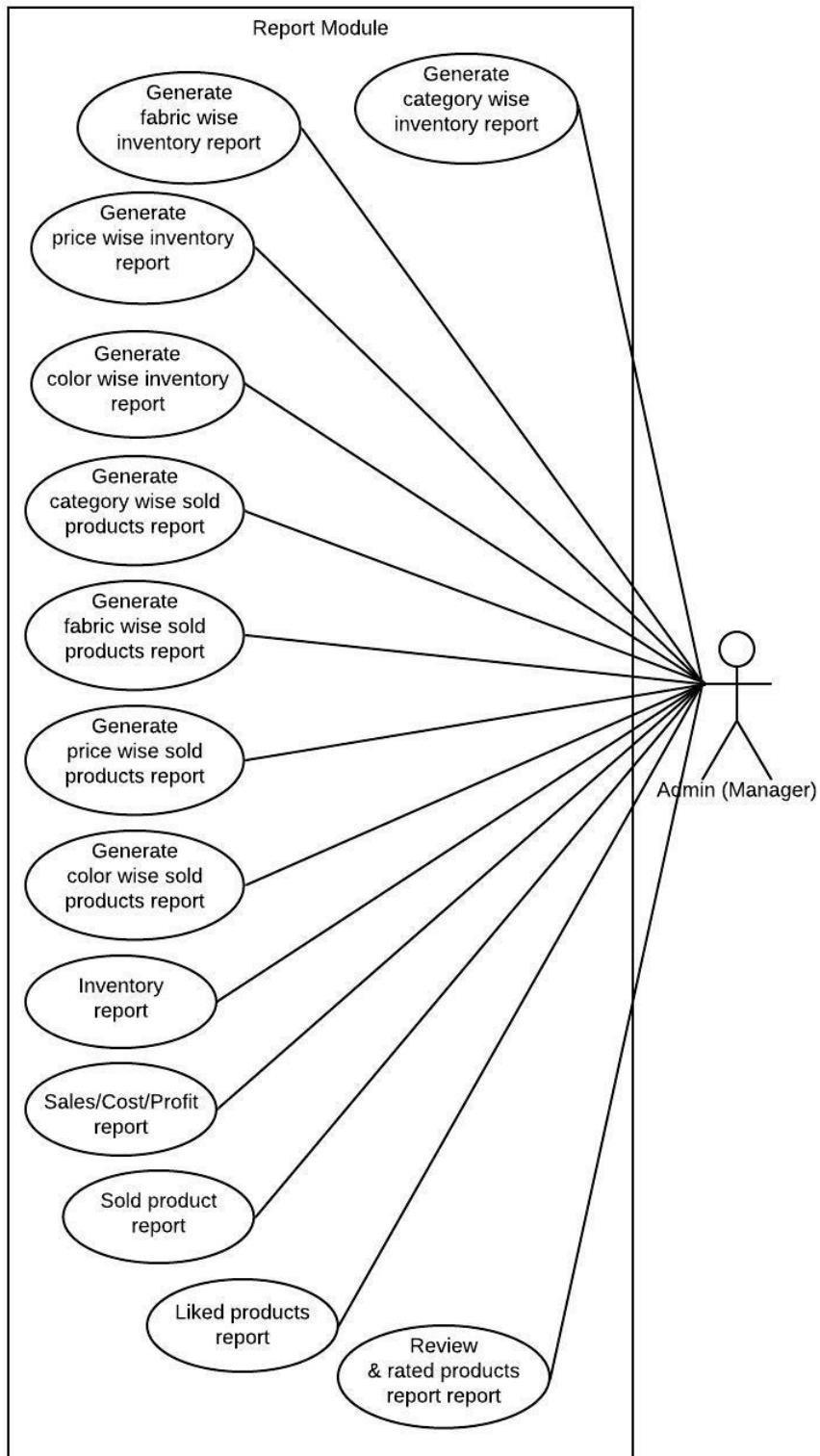


Figure B.6: The use case diagram for the report module

The use case description of profit report generation is given in Table B.6.

Use case	Generation of profit report
Actor(s)	Owner/admin
Description	
Owner/admin should be able to generate the profit report	
Pre-conditions	
The owner/admin should be logged into the system panel.	
Flow of events	
1. Visit the ‘report generation accordion’ page 2. Select profit report 3. Select time period the from dropdowns 4. Click generate report 5. View profit report	
Post-conditions	
<ul style="list-style-type: none">• If there are items on the selected period, the profit report will be resulted.	

Table B.6: The use case description of profit report generation

Appendix C - User documentation

This user documentation contains all the necessary information and guidelines on how to use the e-commerce web application. The main purpose of this kind of document is to guide the users to manage the RAEWA as an alternative for the training program. Some of the guidelines are given below and the rest of the parts can be found under the manual folder in supplementary CD.

Login Page

Visit the home page of the Ransi's Arcade e-commerce web application and click login link for both external (customers) and internal (owner and admin) users to login.

The navbar for navigating to the login screen of the system is presented in Figure C.1.

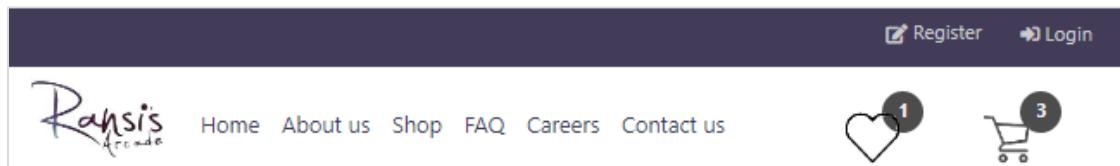


Figure C.1: Navbar for navigating to the login screen of the system

The login screen of the system is presented in Figure C.2. Users who have been authorized can access the system by providing a valid username and password.

A screenshot of the Ransi's Arcade e-commerce website's login screen. The screen has a light gray background. It features two input fields: 'Email' and 'Password', each with a corresponding text input box. Below the password field is a blue link 'Forgot your Password?'. At the bottom of the screen is a dark blue 'Login' button. Below the button, there is a blue link 'Please click here to register now'.

Figure C.2: Login screen of the system

Internal User Dashboard

Once the internal user is authenticated, the user will be redirected to the relevant dashboard either admin or owner. The admin dashboard is presented in Figure C.3.

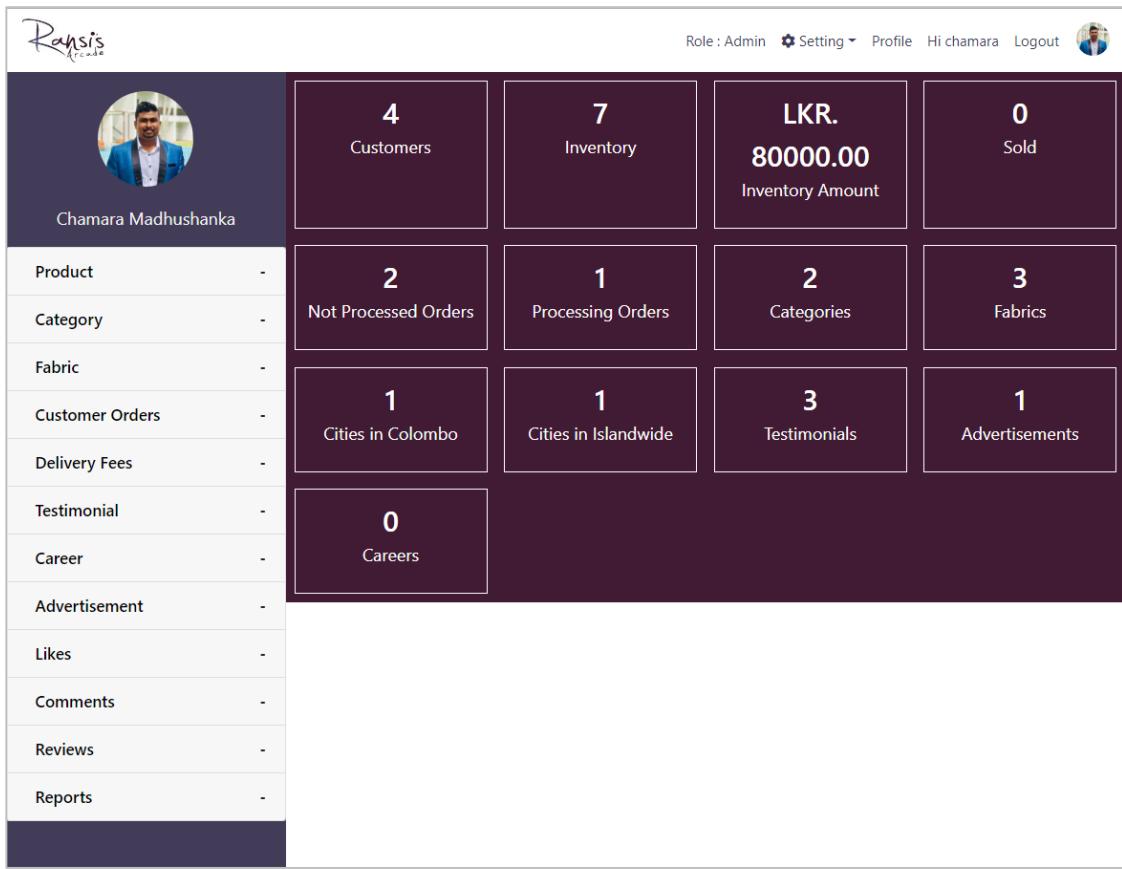


Figure C.3: Admin dashboard

The following sections explain each component in the internal user dashboard.

Dashboard Navbar

Navbar is visible throughout the system. Users are able to logout from the system by clicking on the logout link at the right-hand corner of the navbar. The navbar of the dashboard with its elements is presented in Figure C.4.



Figure C.4: The navbar of the dashboard with its elements

Dashboard Side Menu Bar

Modules can be accessed by clicking the relevant sidebar tabs. The side menu bar of the dashboard with its elements is presented in Figure C.5.

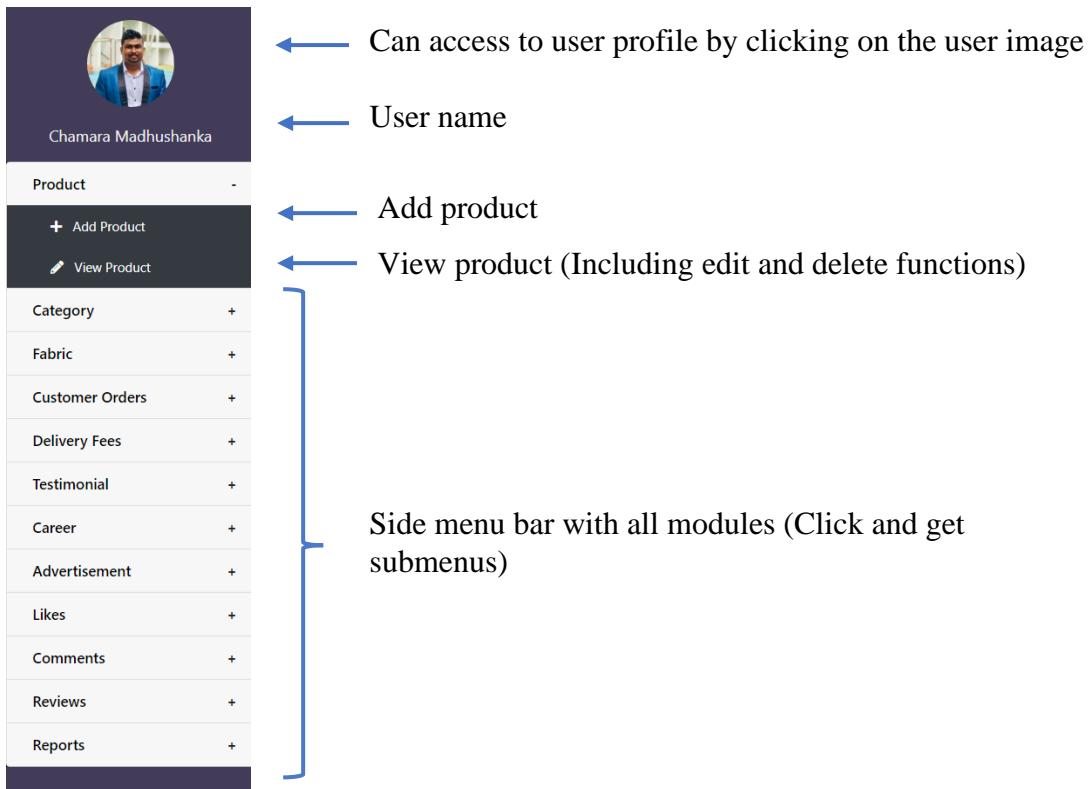


Figure C.5: Side menu bar of the dashboard with its elements

Content

Content will be displayed with the relevant module and their features. The view of the testimonials page with edit and delete buttons on each of them is presented in Figure C.6.

VIEW TESTIMONIALS			
Customer Name	Customer Image	Customer Opinion	Action
Punsara Hewapathirana		The best place to buy sarees, and they have a wonderful collection for the best price! Love the new pearl design sarees at Ransi's.	
Sarani Suchira Dissanayake		Unique collection and extremely reasonable prices. Superb customer care too. Good luck and will be shopping more in future ❤	
Sharani Kodithuwakku		Best sarees ever.... Nice collection with reasonable prices...	

Figure C.6: View testimonials page with edit and delete buttons.

User Profile

By clicking on the profile, the user is able to view his/her profile, update the profile, and change the password. The user profile is presented in Figure C.7.

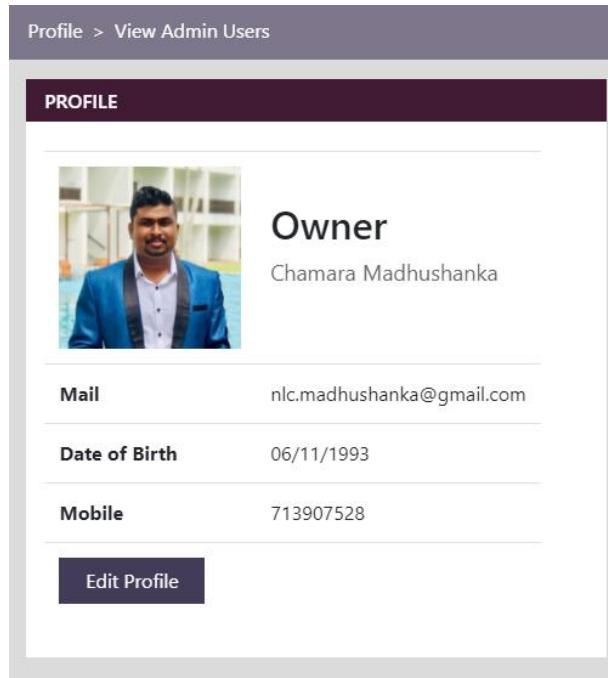


Figure C.7: User profile

Add Admin Account

A new admin account can be created only by the owner. Go to the ‘settings’ dropdown menu and select the ‘add admin’ tab to create a new admin. How to add an admin by clicking setting dropdown only by the owner is presented in Figure C.8.

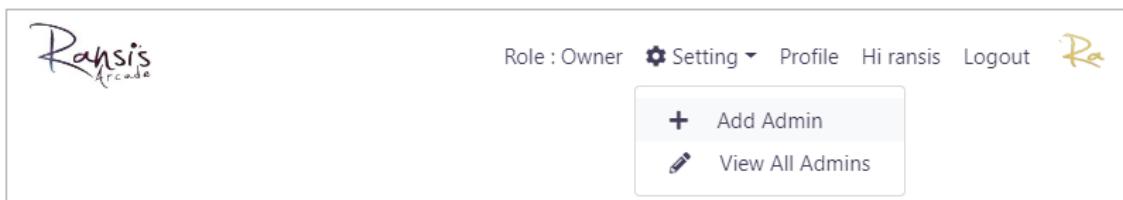


Figure C.8: How to add an admin by clicking the setting dropdown

The add admin page is presented in Figure C.9.

First Name*

Last Name*

Email*

Mobile number*

Eg: Mobile Format (+94) 7X XXX XXXX

Date of Birth*

DD/MM/YYYY

Password*

Confirm Password*

Register

Figure C.9: Add admin page

View Admin Accounts

Viewing of the admin accounts can only be done by the owner. Further, the owner is able to delete admin accounts. Go to ‘settings’ dropdown and select ‘view all admins’ tab to view all admins. The ‘view all admins’ page is presented in Figure C.10.

Image	Admin Name	Email	Mobile Number	Date of Birth	Action
	chamara Madhushanka	nlc.madhushanka@gmail.com	(+94) 713907528	06/11/1993	
	Kumari Mala	kumari@gmail.com	(+94) 772589648	05/06/1956	

Figure C.10: View all admins.

Common Buttons

The common buttons; blue color for viewing more details, grey color for updating details, and red color for deleting are presented in Figure C.11.



Figure C.11 common buttons

Product Images

Following recommendation should be followed while adding images and updating images.

- Use ‘jpg’ images since they support any kind of web browser
- Name images in a meaningful manner for Search Engine Optimizations (SEO)
- Images should be optimized and resized to increase the speed of page loading

The followings should be followed before uploading product images.

- Image width and height should be 400px and 600px
- Name as “<product name> <category> <fabric> <color> .jpg” to increase SEO

Product Management Module

Managing products such as adding products, editing products and deleting products can be performed in the product management module. The add product page is presented in Figure C.12.

Product > Add Product

ADD PRODUCT

Product name *

Category *

Product Image - Front * No file chosen

Product Image - Jacket * No file chosen

Product Image - Border * No file chosen

Product Image - Back * No file chosen

Fabric *

Color *

Old Price (LKR)

New Price (LKR) *

Cost (LKR) *

Occasion * Casual
 Party
 office
 Cocktail
 Wedding & Engagement

Wash and Care *

Description *

Keywords *

Activate

* Denotes required.

Figure C.12: Add product page

The ‘View Products’ page with edit, delete, and view more details functionalities is presented in Figure C.13.

Product > View Products

VIEW PRODUCTS

Product ID	Product Name	Category	Fabric	Image	Price	Status	Action
5ddc4e299915b9368497be7f	Urban Drape Navy Shimmer Saree	Urban Drape	Pure Cotton		Rs. 9990.00	Active	 
5ddc4ff09915b9368497be81	Urban Drape Dahlia in Kashmir Hand Woven Saree	Urban Drape	Pure Cotton		Rs. 12490.00	Active	 
5ddc50e19915b9368497be82	Urban Drape Amaranth Buds Hand Woven Saree	Urban Drape	Cotton with Metallic Threading		Rs. 11990.00	Active	 

Figure C.13: View products page

Appendix D - Management Reports

Management reports are important for the management of the business to make decisions. Multiple reports can be generated through RAEWA that extremely helpful to enhance the decision-making process. Some of them are as follows.

Category Wise Inventory Report

Category wise inventory will be generated by this management report. This report is represented in the form of a table including category name, inventory, and percentage of inventory with pie chart. Category wise inventory report is presented in Figure D.1.

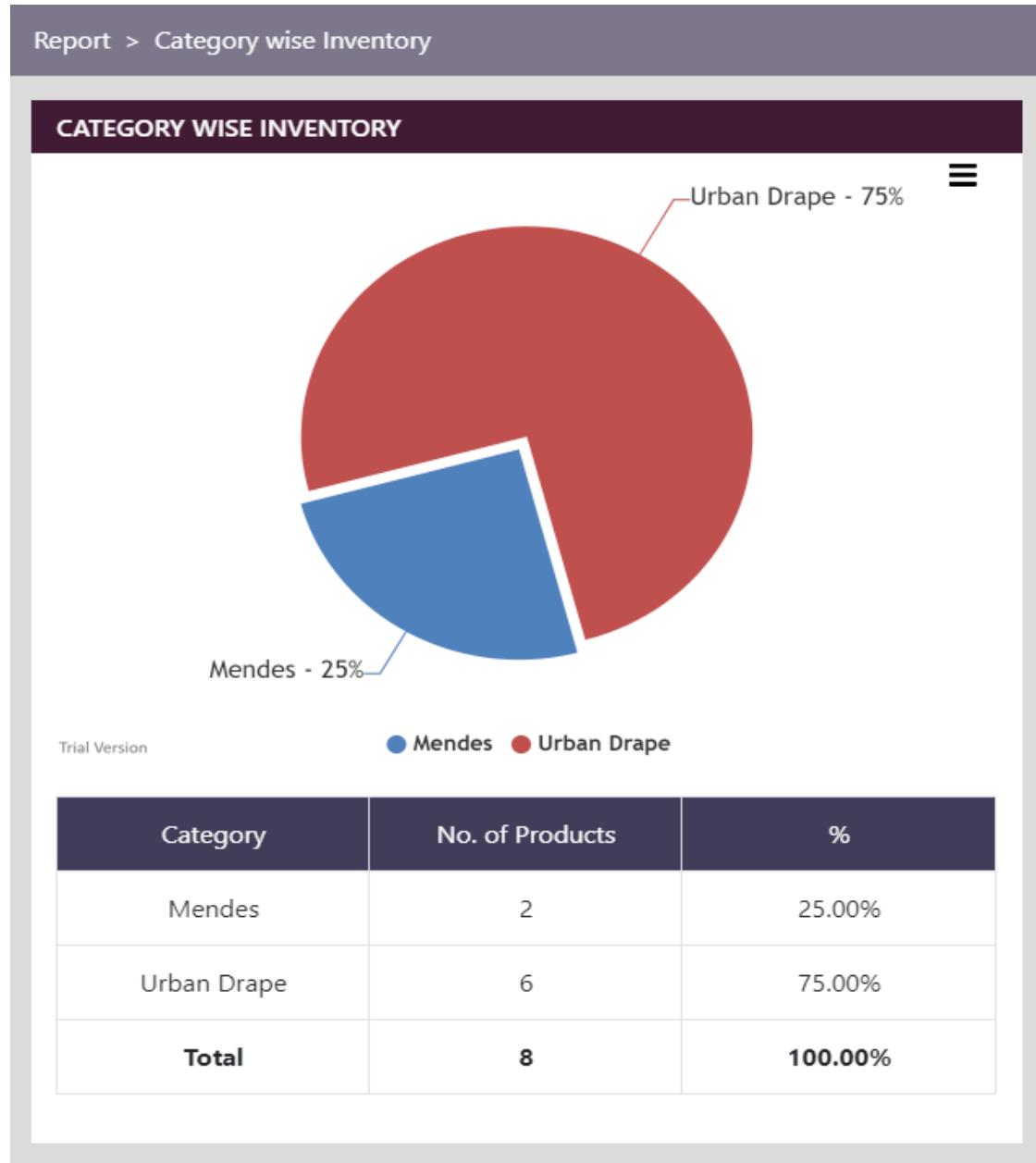


Figure D.1: Category wise inventory report

Fabric Wise Inventory Report

Fabric wise inventory will be generated by this management report. This report is represented in the form of a table including fabric, inventory, and percentage of inventory with pie chart. Fabric wise inventory report is presented in Figure D.2.

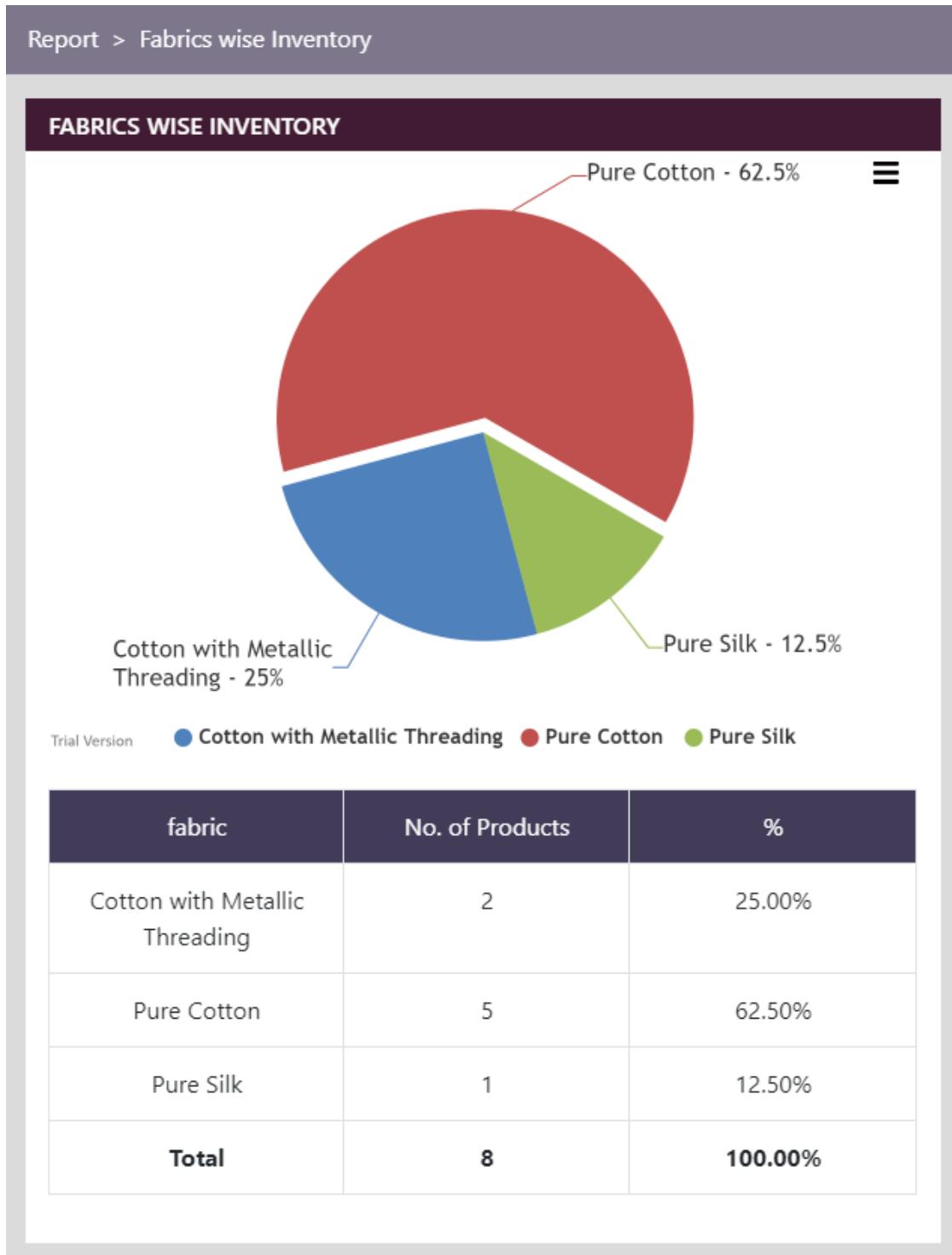


Figure D.2: Fabric wise inventory report

Price Wise Inventory Report

Price range wise inventory will be generated by this management report. This report is represented in the form of a table including price range, inventory, and percentage of inventory with pie chart. Price-wise inventory report is presented in Figure D.3.

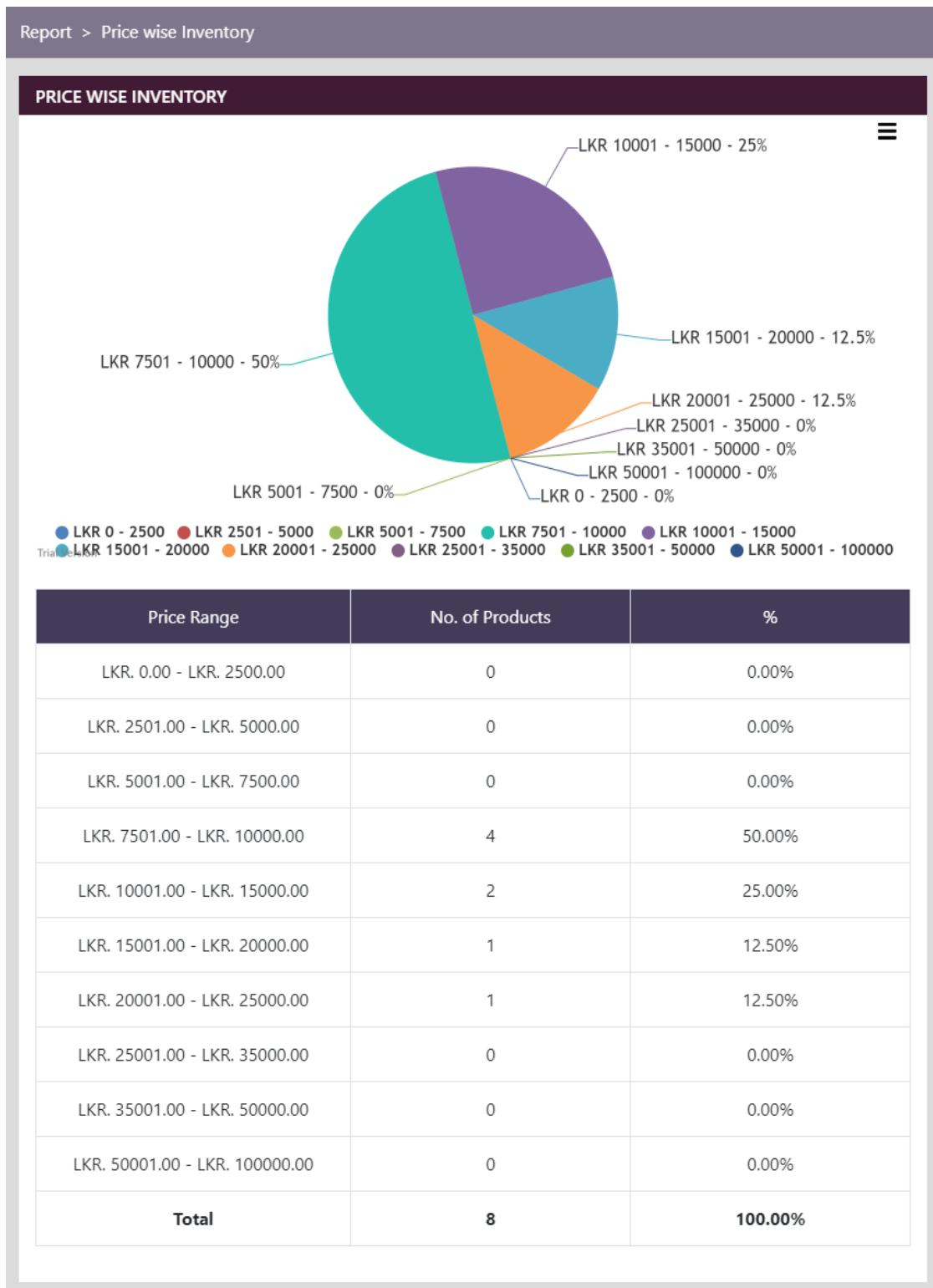


Figure D.3: Price-wise inventory report

Inventory Report

The total inventory will be generated by this management report. This report represents total inventory count and inventory value including the date of creation, product ID, product name, quantity, cost of product and unit selling price. The inventory report is presented in Figure D.4.

INVENTORY					
Created At	Product ID	Name	Quantity	Unit Cost (LKR.)	Unit Selling Price (LKR.)
4 days ago	5ddc4c969915b9368497be7d	Urban Drape Podwer Bronze Saree	1	9000.00	9990.00
4 days ago	5ddc4d5f9915b9368497be7e	Urban Drape Shadow Snow Saree	1	9000.00	9990.00
4 days ago	5ddc4e299915b9368497be7f	Urban Drape Navy Shimmer Saree	1	9000.00	9990.00
4 days ago	5ddc4eed9915b9368497be80	Urban Drape Ink Dash Saree	1	16000.00	16990.00
4 days ago	5ddc51d29915b9368497be83	Urban Drape Float Saree	1	14000.00	14990.00
2 days ago	5ddecbc61e6a4b155820804b	Mendes Grey Angeles	1	10500.00	11490.00
2 days ago	5ddecd601e6a4b155820804c	Mendes Orange Crush Saree	1	7750.00	8500.00
2 days ago	5decf6f1e6a4b155820804d	Urban Drape Tulip Breeze Batik Saree	1	18000.00	20990.00
			8	93250.00	102930.00

Figure D.4: The inventory report

Sales/Cost/Profit Report

The total profit will be generated for any given period by this management report. This report represents total sales, total cost and total profit including product ID, product name, sold quantity, unit price, unit cost and percentage of profit. Sales/Cost/Profit report is presented in Figure D.5.

Sales Report																	
From:		To:															
11/27/2019		11/27/2019															
Generate																	
2019-11-27 to 2019-11-27																	
ProductId	Product	Quantity	Unit Price (LKR.)	Unit Cost (LKR.)	Total Revenue (LKR.)	Total Cost (LKR.)	Total Profit (LKR.)	%									
November 28, 2019 4:06 AM																	
5ddc50e19915b9368497be82	Urban Drape Amaranth Buds Hand Woven Saree	1	11990.00	Rs. 11004.00	11990.00	11004.00	986.00	8.22%									
5ddc4ff09915b9368497be81	Urban Drape Dahlia in Kashmir Hand Woven Saree	1	12490.00	Rs. 11996.00	12490.00	11996.00	494.00	3.96%									
					24480.00	23000.00	1480.00	6.05%									

Figure D.5: Sales/Cost/Profit report

Sold Products Report

The total sold products will be generated for any given period by this management report. This report represents total sales value including date of creation, product ID, product name, quantity, cost of product and unit selling price. Sold products report is presented in Figure D.6.

SOLD ITEMS						
Created At	Product ID	Name	Quantity	Unit Cost (LKR.)	Unit Selling Price (LKR.)	
4 days ago	5ddc4ff09915b9368497be81	Urban Drape Dahlia in Kashmir Hand Woven Saree	1	11996.00	12490.00	
4 days ago	5ddc50e19915b9368497be82	Urban Drape Amaranth Buds Hand Woven Saree	1	11004.00	11990.00	
			2	23000.00	24480.00	

Figure D.6: Sold products report

Customer Report

The total customers will be generated by this management report. This report represents total customers including date of creation, customer name, customer profile image, and email. The customer report is presented in Figure D.7.

CUSTOMERS						
Created At	Customer Image	Customer Name	Email	Mobile	DOB	
25 days ago		Chamara Madhushank	nlcm.gunathilaka@gmail.com	713907524	06/11/1993	
4 days ago		dewduni Alwis	dewduni.d@gmail.com	710193314	19/06/1996	
4 days ago		Ransi Mala	ransi@gmail.com	772568974	05/05/1969	

Figure D.7: The customer report

Liked Products Report

This report represents the customers who liked products. Date of creation, product name, product image, customer name, and customer profile image are given in a table. Therefore, the management has the ability to recognize customers who are interested in a particular product. The liked product report is presented in Figure D.8.

VIEW LIKES						
Created At	Product ID	Product	Name	Customer	Customer Name	
15 hours ago	5ddecbbc61e6a4b155820804b		Mendes Grey Angeles		dewduni Alwis	
15 hours ago	5ddc51d29915b9368497be83		Urban Drape Float Saree		dewduni Alwis	
15 hours ago	5ddc4ff09915b9368497be81		Urban Drape Dahlia in Kashmir Hand Woven Saree		dewduni Alwis	
16 hours ago	5ddc4eed9915b9368497be80		Urban Drape Ink Dash Saree		Chamara Madhushank	
2 days ago	5ddc4e299915b9368497be7f		Urban Drape Navy Shimmer Saree		Chamara Madhushank	

Figure D.8: The liked product report

Commented Products Report

This report represents the customers who have commented on products. Date of creation, product name, product image, customer name, customer profile image, and comment are given in a table. Therefore, the management has the ability to recognize customers who

have commented recently and reply to them if required. The commented products report is presented in Figure D.9.

VIEW COMMENTS						
Created At	Product ID	Product	Name	Customer	Customer Name	Comment
4 days ago	5ddc4d5f9915b9368497be7e		Urban Drape Shadow Snow Saree		Chamara Madhushank	Any discounts??
4 days ago	5ddc4d5f9915b9368497be7e		Urban Drape Shadow Snow Saree		Chamara Madhushank	Colors??
4 days ago	5ddc4d5f9915b9368497be7e		Urban Drape Shadow Snow Saree		Chamara Madhushank	Nice...
4 days ago	5ddc4c969915b9368497be7d		Urban Drape Podwer Bronze Saree		Chamara Madhushank	Woow..

Figure D.9: The commented products report

Reviewed and Rated Products Report

This report represents customers who have reviewed and rated on products. Date of creation, product name, product image, customer name, customer profile image, review, and rating are given in a table. Therefore, the management has the ability to recognize customers who have reviewed recently and those who are interested in a particular product. The reviewed and rated products report is presented in Figure D.10.

VIEW REVIEWS							
Created At	Product ID	Product	Name	Customer	Customer Name	Ratings	Reviews
4 days ago	5ddc4e299915b9368497be7f		Urban Drape Navy Shimmer Saree		Chamara Madhushank	5	nice
4 days ago	5ddc4e299915b9368497be7f		Urban Drape Navy Shimmer Saree		Chamara Madhushank	5	woow
4 days ago	5ddc4c969915b9368497be7d		Urban Drape Podwer Bronze Saree		Chamara Madhushank	5	nice
4 days ago	5ddc4c969915b9368497be7d		Urban Drape Podwer Bronze Saree		Chamara Madhushank	5	woow

Figure D.10: The reviewed and rated products report

Appendix E – Test Cases and Test Results

The following test cases and the resulting screenshots are given below in order to give an overview understanding of how the testing process was carried out in RAEWA.

Test Cases:

Test cases for searching products are given Table E.1

ID	Test Description	Testing Procedures	Expected Results
SP1	Search products by existing keywords	Enter name/color/fabric/price keywords and click ‘search’ button	Display products related to keywords
SP2	Search products by in existing keywords	Enter in existing name/color/fabric/price keywords and click ‘search’ button	Display ‘no products’ message

Table E.1: The test cases for the searching products

Test cases for the basic filtering products are given Table E.2.

ID	Test Description	Testing Procedures	Expected Results
BF1	Filter products by category, fabric, occasion, price, color, etc.	Click a category/fabric/occasion/price/color etc.	Display related products or ‘no products’ message

Table E.2: The test cases for the basic filtering products

Test cases for the advanced filtering products are given in Table E.3.

ID	Test Description	Testing Procedures	Expected Results
AF1	Filter products by multiple inputs.	Select multiple filtering options by dropdowns and click ‘advanced filtering’ button	Display related products or ‘no products’ message

Table E.3: The test cases for the advanced filtering products

Test cases for the wishlist management are given in Table E.4

ID	Test Description	Testing Procedures	Expected Results
WLM1	Add to wishlist	Select a product and click ‘heart’ icon	Item added to wishlist and display on wishlist with No. of wishlist items

WLM2	Add existing item to wishlist	Select existing product in a wishlist and click ‘heart’ icon	No duplications and quantity increasing
WLM3	Deleting an item from wishlist	Click ‘dustbin icon’ to delete	Item removed from wishlist
WLM4	Add to cart item in wishlist	Select a wishlist item and click ‘add to cart’ button	Item added to cart and display on cart with No. of cart items

Table E.4: The test cases for the wishlist management

Test cases for the shopping cart management are given in Table E.5.

ID	Test Description	Testing Procedures	Expected Results
SCM1	Add to cart	Select a product and click ‘add to cart’ button	Item added to cart and display on cart with No. of cart items
SCM2	Add existing item to cart	Select existing product in a cart and click ‘add to cart’ button	No duplications and quantity increasing
SCM3	Deleting an item from cart	Click ‘dustbin icon’ to delete	Item is removed from cart
SCM4	Increase buying quantity of an item in cart	Click ‘+’ sign to increase buying quantity	Buying quantity should be increased up to available quantity and subtotal and total should be changed with correct amounts.
SCM5	Decrease buying quantity of an item in cart	Click ‘-’ sign to decrease buying quantity	Buying quantity should be decreased up to 1 and subtotal and total should be changed with correct amounts.

Table E.5: The test cases for the shopping cart management

Test cases for the online payment checkout are given in Table E.6

ID	Test Description	Testing Procedures	Expected Results
OPC1	Access to online payment checkout page by already logged customer	Select online payment checkout option and click ‘checkout’ button	Go to online payment checkout page
OPC2	Attempting to checkout without login	Click ‘Login to checkout’ button without login	Redirect to login page

OPC3	Submit delivery information	Enter all required data and click ‘continue’ button	Continue to the 2 nd section
OPC4	Data validation	Enter invalid data or leave the mandatory fields blank	Display appropriate error messages.
OPC5	View full order details with delivery changers	Complete 1 st section of online payment checkout page and click ‘continue’ button	Display full order details with delivery changers
OPC6	Select a payment method	Continue to 3 rd section to select a payment method. Then, select card or PayPal option	Display payment methods. After selection, display relevant user interface for payment method (card or PayPal)
OPC7	Pay by valid card data	Enter valid card number, CVV and expiry date and click ‘pay’ button	Coming successful payment message, Cart will be empty, Email will be sent including order details
OPC8	Pay by invalid card data	Enter invalid card number, CVV and expiry date and click ‘pay’ button	Display relevant error messages.

Table E.6: The test cases for the online payment checkout

Test cases for the cash on delivery checkout are given in Table E.7.

ID	Test Description	Testing Procedures	Expected Results
CDC1	Access to cash on delivery checkout page by already logged customer	Select cash on delivery checkout option and click ‘checkout’ button	Go to cash on delivery checkout page
CDC2	Attempting to checkout without login	Click ‘Login to checkout’ button without login	Redirect to login page
CDC3	Submit delivery information	Enter all required data and click ‘continue’ button	Continue to the 2 nd section
CDC4	Data validation	Enter invalid data or leave the mandatory fields blank	Display appropriate error messages.
CDC5	View full order details with delivery changers	Complete 1 st section of cash on delivery checkout page and click ‘continue’ button	Display full order details with delivery changers
CDC6	Order by cash on delivery	Click ‘order now’ button	Coming successful ordering message, Cart will be empty, Email will be received including order details

Table E.7: The test cases for the cash on delivery checkout

Test cases for the admin registration are given in Table E.8

ID	Test Description	Testing Procedures	Expected Results
AR1	Owner Login	Enter correct email and password	Login to Owner account
AR2	Register a new admin	Fill in all the necessary details at appropriate fields and click ‘register’ button	Admin added along with success message
AR3	Email address uniqueness	Enter an existing email address	Display ‘email already exist’ error message
AR4	View all admin accounts	Visit ‘view all admins’ page	Display all admins
AR5	Delete an admin	Click on delete button	Admin is removed and display ‘admin has been deleted successfully’ message
AR6	Data validation	Enter invalid data or leave the mandatory fields blank	Display appropriate error messages

Table E.8: The test cases for the admin registration

Test cases for the owner/admin profile management is given in Table E.9.

ID	Test Description	Testing Procedures	Expected Results
OAPM1	View profile	Click ‘profile’ link on dashboard	Display related profile
OAPM2	Update profile	Enter valid data and click update button	Profile updated along with success message
OAPM3	Change password	Enter valid data and click change password button	Changed password along with success message
OAPM4	Data validation	Enter invalid data or leave fields blank	Display appropriate error messages
OAPM5	Logout	Click on delete button	Logged out and redirect to login page

Table E.9: The test cases for the owner and admin profile management

Test cases for the product management are given in Table E.10.

ID	Test Description	Testing Procedures	Expected Results
PM1	Add a product	Enter and upload all required data and images and save	Display ‘Product has been added successfully’ message and added product is in view product page

PM2	View all products	Visit ‘view product’ page	Display all products
PM3	Update a product	Enter valid data and save changes	Update product along with success message and redirect to ‘view product’ page
PM4	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
PM5	Delete a product	Click on delete button	Product is removed and display ‘Product has been deleted successfully’ message

Table E.10: The test cases for the product management

Test cases for the category management are given in Table E.11.

ID	Test Description	Testing Procedures	Expected Results
CM1	Add a category	Enter required data and save	Display ‘category has been added successfully’ message and added category is in ‘view category’ page
CM2	View all categories	Visit ‘view category’ page	Display all categories
CM3	Update a category	Enter valid data and save changes	Update category along with success message and redirect to ‘view category’ page
CM4	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
CM5	Delete a category	Click on delete button	Category is removed and display ‘category has been deleted successfully’ message

Table E.11: The test cases for the category management

Test cases for the fabric management are given in Table E.12.

ID	Test Description	Testing Procedures	Expected Results
FM1	Add a fabric	Enter required data and save	Display ‘fabric has been added successfully’ message and added fabric is in ‘view fabric’ page
FM2	View all fabrics	Visit ‘view fabric’ page	Display all fabrics

FM3	Update a fabric	Enter valid data and save changes	Update fabric along with success message and redirect to ‘view fabric’ page
FM4	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
FM5	Delete a fabric	Click on delete button	Fabric is removed and display ‘fabric has been deleted successfully’ message

Table E.12: The test cases for the fabric management

Test cases for the delivery charges management are given in Table E.13.

ID	Test Description	Testing Procedures	Expected Results
DCM1	Add a delivery charge	Enter required data and save	Display ‘delivery charge has been added successfully’ message and added delivery charge is in ‘view delivery charge’ page
DCM2	View all delivery charges	Visit ‘view delivery charges’ page	Display all delivery charges
DCM3	Update a fabric	Enter valid data and save changes	Update fabric along with success message and redirect to ‘view fabric’ page
DCM4	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
DCM5	Delete a delivery charge	Click on delete button	Delivery charge is removed and display ‘delivery charge has been deleted successfully’ message

Table E.13: The test cases for the delivery charges management

Test cases for the customer order management are given in Table E.14.

ID	Test Description	Testing Procedures	Expected Results
CO1	View customer orders	Visit ‘view customer orders’ page	Display all orders done by customers in descending order
CO2	Change order status	Select appropriate order status within dropdown list	Order status is changed

Table E.14: The test cases for the customer order management

Test cases for the advertisement management are given in Table E.15.

ID	Test Description	Testing Procedures	Expected Results
AM1	Add an advertisement	Enter required data and save	Display ‘advertisement has been added successfully’ message and added advertisement is in ‘view advertisement’ page
AM2	View all advertisements	Visit ‘view advertisements’ page	Display all advertisements
AM3	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
AM4	Delete an advertisement	Click ‘dustbin’ icon button	Advertisement is removed and display ‘advertisement has been deleted successfully’ message

Table E.15: The test cases for the advertisement management

Test cases for the testimonial management are given in Table E.16.

ID	Test Description	Testing Procedures	Expected Results
TM1	Add a testimonial	Enter required data and save	Display ‘testimonial has been added successfully’ message and added testimonial is in ‘view testimonial’ page
TM2	View all testimonials	Visit ‘view testimonial’ page	Display all testimonials
TM3	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
TM4	Delete a testimonial	Click on delete button	Testimonial is removed and display ‘testimonial has been deleted successfully’ message

Table E.16: The test cases for the testimonial management

Test cases for career management are given in Table E.17

ID	Test Description	Testing Procedures	Expected Results
COM1	Add a career	Enter required data and save	Display ‘career has been added successfully’

			message and added career is in ‘view career’ page
COM2	View all careers	Visit ‘view career’ page	Display all careers
COM3	Data validation	Enter invalid data or leave required fields blank and save	Display appropriate error messages
COM4	Delete a career	Click on delete button	Career is removed and display ‘career has been deleted successfully’ message

Table E.17: The test cases for career management

Test cases for report management are given in Table E.18.

ID	Test Description	Testing Procedures	Expected Results
RM1	Generate category wise inventory	Click on ‘category wise inventory’ in report generation	Display category wise inventory with pie chart
RM2	Generate fabric wise inventory	Click on ‘fabric wise inventory’ in report generation	Display fabric wise inventory with pie chart
RM3	Generate price wise inventory	Click on ‘price wise inventory’ in report generation	Display price wise inventory with pie chart
RM4	Generate profit report	Click on ‘profit report’ in report generation and select period and generate	Display profit report for relevant period
RM5	Generate sold products report	Click on ‘sold product report’ in report generation	Display sold products report
RM6	Generate inventory report	Click on ‘inventory report’ in report generation tab	Display inventory report
RM7	Generate customer report	Click on ‘customer report’ in report generation tab	Display customer report

Table E.18: The test cases for report management

Test cases for common functions are given in Table E.19.

ID	Test Description	Testing Procedures	Expected Results
CF1	Visit pages	Click links and see	Display the relevant page (E.g. home, about us, etc.)
CF2	404 Page not Found error	Type invalid URL path and check	Display 404 page
CF3	Browser compatibility	Check site using Firefox, Chrome, Safari, edge, etc.	Display site with appropriate functionalities

CF4	Responsive web design	Check site using mobile, tablet, laptop, and desktop	Display site with appropriate functionalities
-----	-----------------------	--	---

Table E.19: The test cases for common functions.

Test Results:

Test results for online payment checkout are given Table E.20

Module: Online Payment Checkout			
Test ID	Test Case	Actual Output	Status
OPC1	Access to online payment checkout page by already logged customer	<p>① Delivery Information</p> <p>Title Recipient's Name <input type="text"/> <input type="text"/></p> <p>Recipient's Phone <input type="text" value="713907528"/> <small>(Format 7xx xxx xxxx)</small></p> <p>Delivery Address <input type="text"/> <small>(Apartment No, Street Address, City, State)</small></p> <p>Delivery City <input type="text"/></p> <p>Delivery charges will be added based on the city.</p> <p>Location Type <input type="text"/></p> <p>Special delivery instructions: (Optional) <input type="text"/> <small>Please make your special instructions short and clear. Example: 'Call the recipient before the delivery.' Specific delivery times are not guaranteed.</small></p> <p style="text-align: right;">Continue »</p> <p>② Your Order with Full Amount</p> <p>③ Payment</p>	Pass

OPC2	Attempting to checkout without login	<p>Email <input type="text"/></p> <p>Password <input type="password"/></p> <p>Forgot your Password?</p> <p>Login</p> <p>Please click here to register now</p>	Pass
OPC3	Submit delivery information	✓ Delivery Information	Pass
OPC4	Data validation	<p>① Delivery Information</p> <p>Title <input type="text"/> Recipient's Name <input type="text"/> <small>Title is required Recipient's name is required</small></p> <p>Recipient's Phone <input type="text"/> <small>(Format 7xx xxx xxx) Phone number is required</small></p> <p>Delivery Address <input type="text"/> <small>(Apartment No, Street Address, City, State) Delivery address is required</small></p> <p>Delivery City <input type="text"/> <small>Delivery charges will be added based on the city. Delivery city is required</small></p> <p>Location Type <input type="text"/> <small>Location type is required</small></p> <p>Special delivery instructions: (Optional) <input type="text"/> <small>Please make your special instructions short and clear. Example: 'Call the recipient before the delivery.' Specific delivery times are not guaranteed.</small></p> <p>Continue »</p>	Pass

OPC5	View full order details with delivery changers	<p>④ Delivery Information</p> <p>② Your Order with Full Amount</p> <table border="1"> <thead> <tr> <th>Product</th><th>Price</th><th>Quantity</th><th>Subtotal</th></tr> </thead> <tbody> <tr> <td>Urban Drape Amaranth Buds Hand Woven Saree, Elegant emerald green saree color, Cotton with Metallic Threading saree</td><td>Rs.11990.00</td><td>1</td><td>Rs. 11990.00</td></tr> <tr> <td>Urban Drape Dahlia in Kashmir Hand Woven Saree, mutai pinks and powder blue color, Pure Cotton saree</td><td>Rs.12490.00</td><td>1</td><td>Rs. 12490.00</td></tr> <tr> <td>Total</td><td>Rs. 24480.00</td><td></td><td></td></tr> <tr> <td>Delivery Fee</td><td>+ Rs. 100.00</td><td></td><td></td></tr> <tr> <td>Grand Total</td><td>Rs. 24580.00</td><td></td><td></td></tr> </tbody> </table> <p>③ Payment</p>	Product	Price	Quantity	Subtotal	Urban Drape Amaranth Buds Hand Woven Saree, Elegant emerald green saree color, Cotton with Metallic Threading saree	Rs.11990.00	1	Rs. 11990.00	Urban Drape Dahlia in Kashmir Hand Woven Saree, mutai pinks and powder blue color, Pure Cotton saree	Rs.12490.00	1	Rs. 12490.00	Total	Rs. 24480.00			Delivery Fee	+ Rs. 100.00			Grand Total	Rs. 24580.00			Pass
Product	Price	Quantity	Subtotal																								
Urban Drape Amaranth Buds Hand Woven Saree, Elegant emerald green saree color, Cotton with Metallic Threading saree	Rs.11990.00	1	Rs. 11990.00																								
Urban Drape Dahlia in Kashmir Hand Woven Saree, mutai pinks and powder blue color, Pure Cotton saree	Rs.12490.00	1	Rs. 12490.00																								
Total	Rs. 24480.00																										
Delivery Fee	+ Rs. 100.00																										
Grand Total	Rs. 24580.00																										
OPC6	Select a payment method	<p>✓ Delivery Information</p> <p>✓ Your Order</p> <p>③ Payment</p> <p>Choose a way to pay</p> <div style="background-color: #f0f0f0; padding: 10px;"> <input checked="" type="radio"/> Card <input type="radio"/> PayPal Developer Error: Something went wrong. Check the console for details. </div> <p>Pay Now</p>	Pass																								
OPC7	Pay by valid card data	<p>✓ Delivery Information</p> <p>✓ Your Order</p> <p>✓ Payment</p>	Pass																								
OPC8	Pay by invalid card data	<p>Pay with card</p> <p>This card number is not valid.</p> <p>Expiration Date (MM/YY) CVV (3 digits)</p> <p>Please fill out an expiration date. Please fill out a CVV.</p> <p>A Please check your information and try again.</p> <p>Choose another way to pay</p>	Pass																								

Table E.20: The test results for online payment checkout

Test results for product management module are given Table E.21

Module: Product Management Module			
Test ID	Test Case	Actual Output	Status
PM1	Add a product	✓ success: Product is added successfully.	Pass

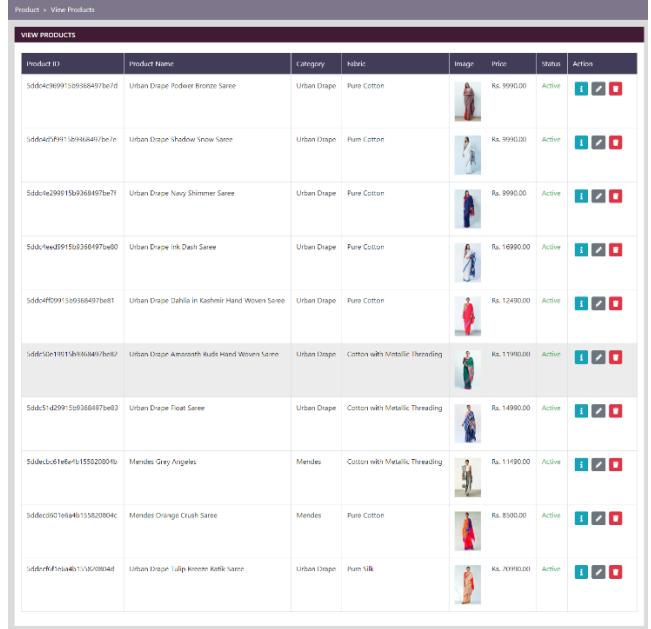
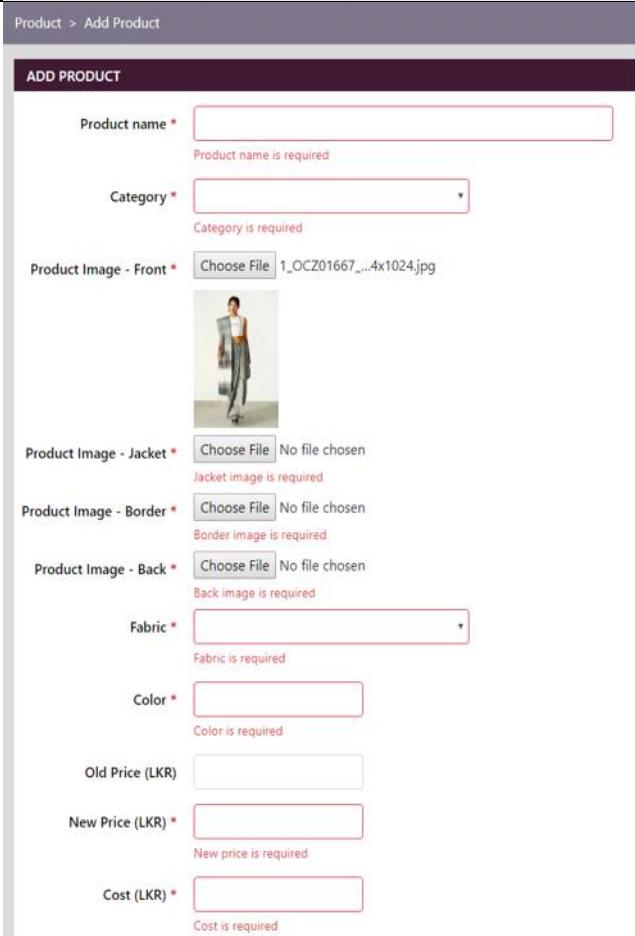
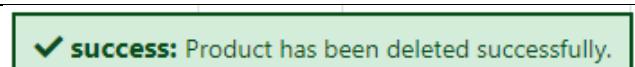
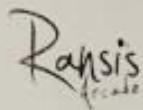
PM2	View all products		Pass
PM3	Update a product		Pass
PM4	Data validation		Pass
PM5	Delete a product		Pass

Table E.21: The test results for product management module

Appendix F - Client Certificate



25th of November 2019

Project Examination Board
Bachelor of Information Technology
University of Colombo School of Computing
Colombo 07

Dear Sir/Madam

Letter of Certification

Ransi's Arcade was registered in the Department of the Provincial Register of Business Names for Western Province under certificate No. W/PP/8569 on the 30th day of March 2015.

This is to certify that **Mr. Neluwe Liyanage Chamara Madhushanka Gunathilaka** who is an undergraduate at the University of Colombo School of Computing (UCSC) has successfully developed **e-commerce web application** for **Ransi's Arcade**. The project will be hosted in the near future as a live site. Therefore, the e-commerce web application will extremely support us to carry our business smoothly and more efficiently, while providing better customer service with fulfilling our targets.

We are pleased to certify that the developed **Ransi's Arcade E-commerce Web Application** has proven to be successful than we expected and has been accepted by the Ransi's Arcade.

This letter of certification is issued at the request of **Mr. Neluwe Liyanage Chamara Madhushanka Gunathilaka**.

We wish his success in all his future endeavors.

Thanking you

Very truly yours

Ransi's Arcade
S.R.Y.P.K. Kulatunga

for

Proprietor

Address: No. 72/2, Butugamuwa Road, Rajagiriya, Sri Lanka.

Gen: +94 11-2-793-88,

Sales: +94 77-2-725-205, +94 72-7-352-576

Glossary

Axios - Axios for performing HTTP requests.

Bcrypt - The bcrypt hashing function used for building hash passwords.

Body parser - Body-parser used to extract the entire body portion of an incoming request stream and exposes it on req.body.

Braintree and Braintree Web Drop-in React - The Braintree Node library provides integration access to the Braintree Gateway to process online payments through credit cards, debit cards, digital wallets, PayPal, etc.

CanvasJSReact - CanvasJSReact for creating interactive graphical charts

Express - Express is a web application framework for Node and for developing web applications and APIs.

Font Awesome Icons - An icon library for displaying graphical icons.

Formidable - Formidable is a Node.js module for parsing form data, including multipart/form-data file upload.

Gravatar - Gravatar stands for Globally Recognized Avatar for getting globally recognized avatar from gravatar servers.

JSON - JSON is a lightweight data exchange format between a browser and a server.

Jsonwebtoken – JSON web token used for user authentication.

Moment - A lightweight JavaScript date library for parsing, validating, manipulating, and formatting dates.

MongoDB - MongoDB is an open-source database management system (DBMS) that uses a document-oriented database model that supports various forms of data.

Mongoose - Mongoose is an object data modeling (ODM) library that provides a precise modeling environment for data and connects the model and the database.

Node - Node is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser.

Nodemailer - Nodemailer for sending emails.

React - React is the fastest-growing JavaScript library for building user interfaces.

React helmet - React helmet for setting titles in the head section.

React input mask - Allows the validation of specific data formats such as phone numbers, and Dates, etc.

React rating - React Rating for rating products.

Index

A		M	
Analysis	3, 5, 61	Moment	39, 62, 100
architecture	17, 38, 39, 40	MongoDB	iii, 38, 42, 62, 64, 65, 100
Axios	38, 62, 100	Mongoose	38, 42, 62, 100
		MVC	xiii, 19, 39, 60
B		N	
Bcrypt	38, 42, 62, 100	Node	iii, xiii, 37, 38, 39, 41, 60, 62, 65, 100
Body parser	38, 62, 100	Nodemailer	39, 62, 100
Braintree	39, 62, 100		
C		O	
CanvasJSReact	39, 63, 100	objectives	iii, 17, 51, 59
class diagram	26, 27		
CMS	xiii, 18		
Conclusion	4, 59		
CSS	iii, xiii, 37, 60		
D		R	
Design	4, 15, 18, 19, 20, 28, 34, 37, 60, 61, 66	RAD	xiii, 16
		RAEWA	xiii, 1, 2, 15, 30, 34, 51, 57, 59, 65, 73, 80, 87
Evaluation	4, 51	Ransi's Arcade	iii, iv, xiii, 1, 59, 73
Express	iii, 38, 41, 62, 100	React	iii, 37, 39, 60, 61, 63, 100
		RESTful	iii, 39, 40, 60, 63
		RUP	xiii, 16, 17, 51
E		S	
e-commerce	iii, 1, 2, 7, 18, 21, 25, 26, 27, 34, 37, 59, 60, 61, 64, 65, 73	sequence diagram	28
Evaluation	4, 51	system	iii, 2, 3, 4, 5, 6, 7, 10, 13, 15, 16, 17, 18, 19, 20, 22, 23, 24, 26, 29, 30, 31, 36, 37, 38, 39, 51, 53, 57, 59, 60, 61, 63, 64, 66, 67, 68, 69, 70, 72, 73, 74, 100
Express	iii, 38, 41, 62, 100		
F		T	
fact-finding techniques	5	testing	iii, 4, 17, 51, 60, 63, 87
Formidable	39, 63, 100		
G		U	
Gravatar	38, 62, 100	UML diagrams	60
		use case diagram	6, 20, 22, 23, 24, 25, 66, 67, 68, 69
I		V	
Implementation	4, 36, 39	Validator	38, 62
J		W	
JavaScript	iii, xiii, 37, 38, 39, 61, 100		
JSON	xiii, 38, 43, 60, 62, 100		
Jsonwebtoken	38, 62, 100	waterfall	16