International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 12 | Dec - 2021 ISSN: 2582-3930

Development of an e-commerce application for online shopping

Saleh Umar
Sharda University India

Tokpe Kossi
Sharda University India

Department of Computer Science And Information Technology

ABSTRACT:

Electronic commerce, commonly known as ecommerce, is a sort of business in which products are bought and sold through electronic networks such as the internet. The goal of this application is to learn concepts about ecommerce and how to create an interactive ecommerce platform from the ground up utilising client-side languages like JavaScript and HTML, as well as the server-side Java language via Java Server Faces. All of the implementation related to setting up the database, establishing session models for joining different user-interface (UI) pages, calculating shipping costs and sales tax, and so on is done on the server side, which is primarily Java. It's in charge of extracting data from the database and making it available to the user interface by mapping the category or item ID to the database's IDs. The client is in charge of displaying the full user interface, which includes CSS, HTML, and JavaScript.

Keynote: *E-commerce, Internet, database, JavaScript, Html and css.*

INTRODUCTION:

It is well acknowledged that, in today's market, starting a new small scale firm and maintaining it in the face of competition from well-established and established/brand owners is exceedingly challenging. Even if the product is of excellent quality, due to a lack of advertising or small-scale business, it frequently becomes just another face in the crowd, and the product does not reach a bigger audience. "In today's fast-paced world, where everyone is pressed for time, most individuals are picky when it comes to physical purchasing. In terms of logistics, a consumer finds a product more intriguing and appealing when they find it directly on a retailer's website and can read the item's details online"[4][5] Customers today are drawn to internet shopping not only because it is easy, but also because it offers a wider range of products, lower costs, more information about the product (including user reviews), and relatively simple navigation for product searching. Furthermore, because the overhead costs of starting and operating a physical store are higher, business owners frequently provide low-cost online buying options. Furthermore, with online purchasing, their products have access to a global market, increasing the number of clients from various ethnic groups, increasing customer value, and ensuring long-term marketing sustainability[8] Online web businesses, such as Amazon and eBay, have grown in popularity over the years as a result of their ability to sell nearly everything. These web stores also allow a lot of small businesses and manufacturers to reach the worldwide market and sell their items directly to customers without having to go through multiple corporations or intermediaries before reaching the shelves of a physical store. Furthermore, rather than using existing platforms, manufacturers can create their own web store to sell their items directly to the public.

Background:

The major goal of this project is to create an E-commerce online shopping application that allows customers to purchase from their homes or offices utilizing smart phones, PCs, and online credit cards.

- Online shopping is a type of electronic commerce that allows customers to buy goods or services directly from a seller over the Internet via a web browser.
- The Internet makes life easier and more innovative.
- The Internet has opened up new avenues for business promotion.

Aim of the Software

This software was created to assist computer science students like us in learning about application creation using JavaScript and HTML as a starting point. This application enables the student to grasp the fundamentals of how a first web page should look and how a whole working application can be constructed from the ground up. It enables students to grasp the concept of user-integrated graphics as well as how JavaScript may be incorporated into HTML. It also explains how the client-side language interacts with Java, the server-side language, and eventually the database. This shopping cart application is primarily intended to help computer science

students learn and grasp the notion of application development, but it may also be used to teach ecommerce and web-application concepts. Students can view the source code for all of the different parts shown on the UI to visually understand how a particular piece of code works. The application can be downloaded and installed on multiple machines, and students can view the source code for all of the different parts shown on the UI to visually understand how a particular piece of code works. This shopping-cart programmed is extremely versatile, and it may be expanded for commercial usage by adding more functionality and changing the aesthetics.

Literature Review

The history of ecommerce shopping carts began shortly after the World Wide Web, or WWW, became a key channel for global communication of information. Consumers can buy goods or services directly over the internet utilizing a web browser with ecommerce shopping cart software. This type of online shopping resembles the business-to-consumer (B2C) approach, in which a customer purchases directly from a company. A business-to-business (B2B) transaction occurs when one company purchases from another. eBay and Amazon, both of which were founded in 1995, are the best examples of shopping cart apps that use the B2B procedure. Currently, the majority of people who use these online shopping cart applications have a greater level of education, have been exposed to technical improvements, and are in a higher income bracket. These customers have a favorable opinion toward these time-saving purchasing methods[7] According to a study in December 2011, In a survey of 1,500 online customers, Equation Research discovered that 87 percent of tablet owners completed online purchases during the early Christmas shopping season[6] because of the high level of competition in the market, creating a new successful shopping cart is straightforward, and the designer of a shopping-cart application must consider the information load, co mplexity, and uniqueness[9].The quantity of elements offered on the shopping cart and the levels of marketing are referred to as build complexity, whereas novelty refers to the unexpected or new components of the site. A designer must also consider the demands and expectations of the customers.

Because, unlike actual stores, online shoppers come in various ages, genders, and cultures, a user-friendly design is vital to the success of any shopping cart application. [10] Logistics plainly states that in order to have a successful and lucrative online shopping application, firms must devote significant time and resources to its design, development, testing, and maintenance. In addition to a high-quality design and user experience, strong customer service practice must be followed[11]

A typical shopping cart should have functionality like adding products to the cart and checking out using the various payment methods. The majority of shopping-cart programs use HTTP cookies or query strings, and an HTML configuration is required to install the shopping cart on the servers that ultimately host the website on the internet. Because users can add or delete one or more things from the cart, most of these server-based applications require data relating to the goods entered in the shopping cart to be saved in a session object that can be accessed later and altered dynamically. Most simple shopping cart applications do not allow you to check out before you've added any goods to your cart. Data is frequently saved in an external database or application-based databases, which the application administrator can access in real time[12] There are numerous instances of online shopping program that have been created in various languages. The development platform and language to use are

determined by the policies of the firm for whom the app is being created. It also depends on a number of other aspects that are crucial when choosing a platform on which to build an application, such as how portable the application will be when it is constructed or whether it is open-sourced. For a variety of reasons, Java was chosen for this application. "It's a straightforward, cross-platform language. Because Java programs are compiled into platform-independent byte codes, they may be carried and run on any platform, including MAC and Linux"[11] Java is a very safe language because to its robustness, as it has exception handling and a layer technique for communicating with the database, which protects the system from crashing. Another major feature in terms of development is that the Java language is object-oriented, which means that everything is regarded as an object and class methods are used instead of functions or procedures, making the code very easy to grasp. Several Java shopping-cart apps were investigated, and implementation details were compared to the suggested design for this application in order to construct an even simpler architecture that is very easy to understand from a learning standpoint. Some applications for online purchasing are as follows:

- SoftSlate Commerce[12]
- Commerce4j[13]
- Cs.Cart[14]
- Apache Ofbiz[15]

These applications are created for industrial use in order to make cash by selling them to customers that want to start a website for their enterprises. [17] The application described in this work focuses on creating a simple, yet full, application for computer science students to master the fundamentals of application design and development. This programmed performs all of the above-mentioned applications' basic operations, such as picking an item and putting it to the shopping cart. Logging in or enrolling as a user, checking out an item, and so on. Other functionalities that could be added to this application are proposed for future work, and they would be required in a more comprehensive and complicated design. As described in the Future Work, the finished application should bring us knowledges to create more complicated functionalities.

Paper Organization

The rest of the text is broken down into three sections: goals, implementation, and testing. The Need for Building the System is listed in the Objectives chapter. It includes use cases to aid commercial and technical users in comprehending the material. It also includes a full description for each use case to aid in design and implementation, as well as a list of software restrictions. The complete architecture of the system is discussed in the Implementation chapter, which includes the Class Diagram, Activity Diagram, and Component Diagram. This chapter also offers a full description for each component, as well as how the class and its components interact with one another when doing specific tasks, as well as mock screen pictures of software.

Motivation:

The reasons for the motivation are as follows:

- Interest in a web-based application with some functionalities is growing how Alibaba and Amazon work.
- To develop a user-friendly application that allows customers to place agricultural goods orders.
- To help us comprehend how Alibaba and Amazon, for example, manage their web applications,

- Designing an online shopping application came about as a result of preferring online shopping to spending a lot of time in physical
markets. The motivation for designing this Development of an e-commerce application for online shopping prefer to shopping online rather than spend time in physical
markets. Furthermore, because custom-designed platforms are expensive, there is also the option of designing one's own customized shopping-cart application from scratch using the available stores to sell the products. In addition, I value my current knowledge of the Java and JavaScript programming languages, as well as understanding how strong and dynamic they are when it comes to web design and application development. Aside from assisting computer science students in comprehending the fundamentals of web application design, it would be extremely simple to combine the idea of employing programming approaches from the accessible visuals to comprehend how a piece of code appears on a user interface. JavaScript, HTML, and Java were chosen to create this application since I found them to be highly beneficial while working on the technologies at my business Thomson Reuters is a news and information company.

Objectives:

The following sections detail all of the phases involved in the software-analysis process for this project (product function, user characteristics, functional and nonfunctional requirements, constraints, assumptions, and dependencies for the online shopping application).

Requirements Analysis

Any software engineering project's success hinges on the requirements analysis and collection methods. In software engineering, requirements analysis is a process that defines the tasks needed to establish the needs and conditions for designing a new product or making changes to an existing product/application. This process examines the system's documentation and validation, as well as the competing requirements of all stakeholders. The requirements should be actionable, quantifiable, and testable, as well as relevant to the system design's declared demands. Requirements analysis is a three-step process in terms of software engineering.

1. Requirements Elicitation: Elicitation of requirements, also known as requirements gathering, includes the task of identifying various requirement types from stakeholders or from project documentation.

2. Requirements Analysis: The purpose of requirement analysis is to see if the requirements acquired are clear, full, and consistent. The analysis also deals with any confusing requirements that don't clearly specify what needs to be done, which could result in a waste of resources and time if discovered later in the development or
testing process. Identification of stakeholders and consideration of their demands are required for requirement analysis to help them understand the implications of developing
the new system, as well as which modules are worth installing and which are more cost effective. After that, you'll need to write a software-requirement specification paper. Different procedures, such as establishing a scenario or user stories, and determining the use case that is being used for the project, can be used to explicitly elicit

the stakeholders' requirements. Stakeholder analysis states that in order to clearly gather project requirements, analysts must first identify the stakeholders. Stakeholders are individuals or organizations with a legitimate interest in or use of the system. The steps to identify the stakeholders are as follows:

- ✓ Anyone who is in charge of the system.
- ✓ Anyone who reaps the benefits of the system
- ✓ Anyone who is directly or indirectly involved in the system's purchase
- ✓ Opponents of the system (individuals or organizations)
- ✓ Organizations in charge of the system's design
- ✓ Organizations in charge of the system's financial and safety aspects

After the stakeholders have been identified, interviews are done using various methods, the system's needs and requirements are established, and a requirements specification paper is created. The paper is then discussed with the key stakeholders to find any discrepancies in the system's requirements and understanding.

3. Documenting the requirements in various ways, such as summary lists, natural language documents, visual documents, use cases, user stories, or process specifications, is part of this step. A requirement specification document can be classified in a variety of ways depending on the needs of the stakeholders, assisting in the creation of a clear contract between development and business. The many types of requirements specification documents that are required for creating this application are listed in the sections below: system requirements, restrictions, and functional requirements, etc.

Product Perspective

The online shopping-cart application is a web-based system. It can be accessed using Internet Explorer 8.0 and above, Mozilla Firefox 2.0, and Google Chrome.

User Interface

The two interface types found in the online shopping-cart application are as follows:

1.Users can view the shopping-cart application's home page, browse the various categories, browse and add any number of items from any category to the shopping cart, look for product information, delete items from the shopping cart, save the cart for later viewing, check out or continue shopping after adding an item to the cart, and check out the items by filling out the required information in the order form.

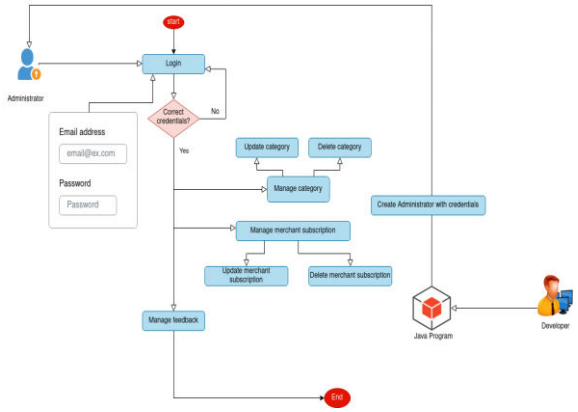
2. Admin Interface: The administrator can access the database to view the information supplied by users during checkout, edit the information, pricing, and shipping charges of items, and add or remove things from the main display. The External Interface Requirement section goes over both interfaces in great depth.

FLOW CHART OF THE PROJECT

Three types of users will be considered in this project. They are Administrator, merchant and customer. All these types of person need to login the system to do some actions specific to them. And the login page needs that a person provides a username and password. If one or the two credentials are wrong the user can't login but if he provides good credentials he can access his dashboard. Below we represent flow charts of the different types of users through the use of the tool flow.io

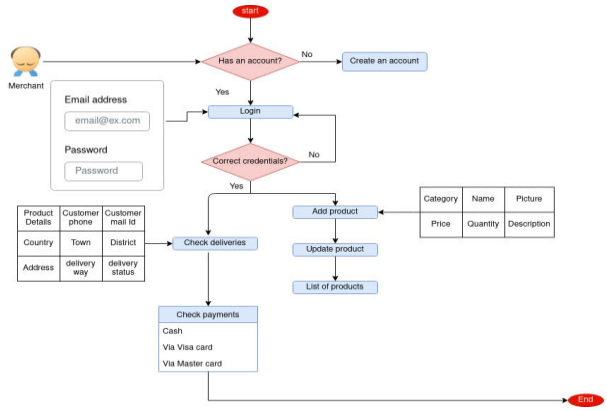
Case 1: Administrator

Administrator is the person created by the developer and whose credentials are provided by the developer. If he logs in the system he can add categories of product, he can update a category and can also delete a category. He can add merchant subscription, he can update merchant subscription and can also delete merchant subscription. This is represented by the bellow flow chart.



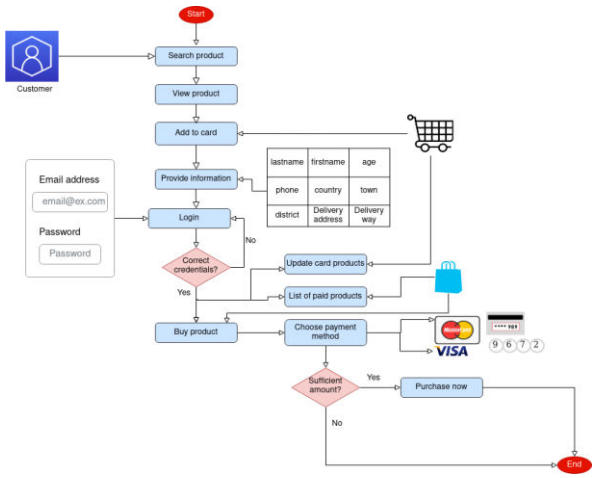
Case 2: Merchant

A merchant is a person who wants to sell articles. If he logs in the system he can add an article, he can update an article and he can also delete an article. He can check deliveries of customer, he can check payments of customer which can be by cash, master card or visa card. This is represented by the below flow chart.



Case 3: Customer

A customer is a final user who can buy an article on our application. He doesn't need to log on the system but he needs to provide his information and provide the good address where the articles can be delivered. To pay an article he can pay directly on the website via master card or visa card or he can pay when the article is delivered. When he adds a product to a cart his information are saved and then after he can log on the system to check his deliveries and his payments. This scenario is represented by the below figure.

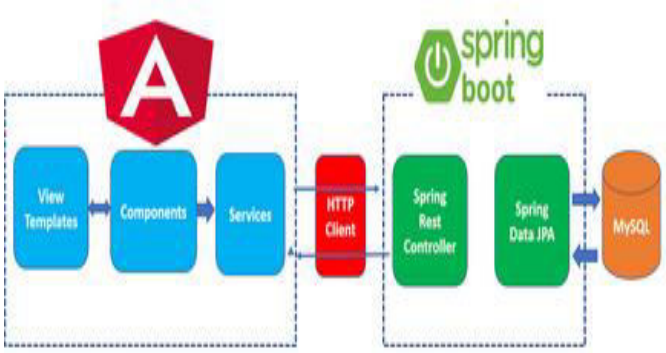


TECHNICAL ARCHITECTURE

As this project is an ecommerce project we plan to make the mobile part later. The real architecture needed in this king of project is web-service

architecture. About that the backend of the project will be make in Spring Boot which is a java framework and the frontend will be in angular which is javascript framework.

In Spring Boot we produce an endpoint which will be consumed by the frontend Angular. The two parts communicate via HTTP(Hypertext Transfer Protocol) Client. Database used is MySQL for development testing reasons. But once in production firebase database will be used because it is a realtime database. Below is the representation of the technical architecture of the project.



Result and Challenges of ecommerce Online Shopping

The overall idea of doing this project is to get a real time experience. Learn new technologies .

Result: Because it leverages the GUI supplied in the user dialogue, the application will be simple to use. Users will be able to interact with the screens, which will make it a fun activity for them.

- **Challenges:**
- The estimated time for this the web version will last for three months.
- Compatibility with browsers like Mozilla Firefox, Internet explorer etc
- Developing the mobile version of the project.
- Learning new technologies with little direction, such as employing JavaScript for drag-and-drop activities and Ajax toolkit controls with Vanilla Js.

Futures:

- Users can establish accounts and save things to wish lists using the existing system, which can be expanded.
- Payment Methods: Different payment options, such as Visa, MasterCard, PayPal, and others, should be included, with the ability to preserve card information for future checkouts..
- Shopping Options: Different sorts of shipping options should be added, such as ordinary shipping, expedited shipping, international shipping, and so on. etc.
- Recent History: The user's recently visited items are displayed in the recent-history tab.

Hardware Interface

The online shopping-cart application shall provide minimum hardware requirements. The following hardware configurations are required for a PC using the online shopping-cart application:

Pentium processor

- ✓ 32 MB of free hard-drive space
- ✓ 128 MB of RAM
- ✓ Software Interface

This section outlines the requirements that must be met in order for the system to function properly. The operating system required for the
1. Operating System: Windows (Vista/Windows 7) or MAC OS

2. Web Brower: Internet Explorer (8.0 and above), Mozilla Firefox (3.0 and above), or Google

3. Drivers: Java Runtime Environment

4. Integrated Development Environment: Eclipse Juno or Apache Tomcat

5. Third-Party Tool: Microsoft Word

Product Function

The online shopping-cart application would have the following basic functions:
1. On the system's main page, display all of the shopping categories.

2. Show all the things associated with each of the categories provided on the main page.

3. Allow the administrator to add new items to the available items list.

4. Allow users/administrator to remove items.

5. Allow the administrator to modify the price of each item.

6. Allow the administrator to make changes to each item's description.

7. Allow the administrator to see and change information about each user who uses the system to check out items.

system to function properly, the application's interface, the driver for running Java web applications, the application's integrated development environment, and the third-party tool used for editing purposes are as follows:

User Characteristics

Customers and administrators are the two types of users of the online shopping cart application (owner). These users are determined based on their technical knowledge and experience.

1 **Admin:** This online shopping cart application's owner is the administrator. A fundamental awareness of computers and the internet is required, as well as prior experience with the eclipse and Java programming languages. The administrator is in charge of keeping track of all of the system's training documents. The administrator can perform the following functions:

- ✓ Assign or change the price of the items, update the items in the list, and delete the items.
- ✓ Assign sales tax for different states at the time of checkout.
- ✓ View the history of the customers who purchased the items.

2. **Users:** Customers who would shop to test the programmed are the users of this online shopping cart application. Anyone with shopping experience and the ability to navigate a shopping cart application qualifies as one of these users. They must have a working knowledge of computers and the internet. Users should be able to complete tasks the following functions using this system:

- ✓ View, browse, and select a category on the home page.
- ✓ View, add, and update items in the cart.
- ✓ Delete items from the cart.
- ✓ Check out the items from the application or continue shopping.
- ✓ Sign-on/login using a username and password.
- ✓ Place the order by completing the order form.

Constraints

1. Hardware Requirements: The system requires 128 MB of RAM and a 32 MB hard disc as a minimum.
2. Accessibility: The software should first be made available as a desktop application for a small group of users to test.
3. Other requirements: The application should be written in Java with JavaScript embedded in HTML, and it should be first available through the Eclipse IDE before being deployed on a server.

Assumptions and Dependencies

The assumptions and dependencies are as follows:

1.Users and administrators are used to the paper-based method, therefore learning to utilize the online shopping cart application would be necessary.

2. To run, the system requires the presence of an Apache Tomcat Server.

3. Users are assumed to follow the system's minimum software and hardware requirements.

4. This system will make use of third-party software, and it is believed that users will be familiar with it.

Specific Requirements

This section offers information about all of the software needed for designers to create a system that meets the needs of users and testers to test the requirements. This section offers a description of each GUI's interface for the various system users. These sections include provide descriptions of all system inputs, functions performed by the system, and system outputs (responses).

Functional Requirements

The prerequisites for the online shopping cart application are listed in this section. To support the types of user interactions that the system will have, the functional requirements received from the users have been grouped as follows.

1. The primary goal of this online shopping application is to educate computer science students the fundamentals of the Java, JavaScript, and HTML programming languages, as well as web-application design techniques.

FR01: The customers shall be able to view the source code for the entire application.

FR02: The customers shall be able to, individually, view and understand the code for all pieces on the UI.

FR03: The customers shall be able to debug the application's source code using Firebug, which is an online tool to inspect, edit, and monitor HTML, CSS, and JavaScript requests directly on the web page.

2. User: View Categories and Items: The users shall be able to see the home page of the online shopping application when they first run the program. The users shall be able to view the different categories, select categories, browse through the items in each category, and add items to the shopping cart. The users shall be able to view the shopping cart and more information about each item.

FR04: On the application's home page, users will be able to see the categories.

FR05: The users shall be able to view items in different categories.

FR06: The users shall be able to add items to the cart.

FR07 Before adding an item to the cart, users should be able to view more information about it**.**

FR08: The users shall be able to view the shopping cart.

FR09: Users will be able to look through the things that are accessible.

2.User: Users will be able to see their preferred home page after the application has been run for the first time. Users should be able to view the products in their shopping cart after browsing through them and adding them to their cart.

3. Users will have the option to check out or continue shopping. Users will be able to remove things from their shopping carts.

FR010: The users shall be able to view the items added to the cart.

FR011: Users will be able to check out with their current cart contents..

FR012: Users will be able to check out with the items of their shopping cart.

FR013: Users will be able to remove things from their shopping carts.

4. **User: Item Checkout**

FR014: Only when there are products in the shopping basket will users be allowed to check out.

5. **Login/ User**

FR015: The users shall login or register using the user authentication form.

FR016: If the information is incomplete or incorrect, the users will not be able to login or register..

6. **User: Place Order**

FR017: Users must complete the information on the order form in order to place an order.

FR018: If the information in the order form is incorrect or partial, the users will be unable to place an order.

7. **Admin: View User Information**

FR019: The administrator will have access to all of the information that the users enter on the order form and during the checkout process.

8. **Admin: Add/Update/Delete Shopping** Items

FR020: The administrator will have the ability to add new products to the shopping list.

FR021: The administrator shall be able to modify/update an item's price and description.

FR022: The administrator should be allowed to delete things from the shopping-cart application's main page.

Additional Functional Requirements

FR023: The administrator will have access to the shopping-cart application's main page and will be able to delete items.

FR024: The administrator will have access to the whole history of users who have completed the checkout procedure successfully.

Performance Requirements

This section lists the performance requirements expected from the online shopping-cart application.

1. **PR01:** In less than 5 seconds, consumers should be able to add an item to their cart.

2. **PR02:** The users shall be able to view information about an item in fewer than 5 seconds.

3. **PR03:** After completing the order form, users should be able to check out the items in their shopping cart within 10 seconds.

4. **PR04:** The navigation between pages shall take fewer than 5 seconds.

5. **PR05:** To avoid erroneous or incomplete information, the application must be able to perform a validation check on the information provided in the user-authentication and place-order forms.

Design Constraint

This section lists the design requirements for the online shopping-cart application.

DR01: Fonts and font sizes must be specified in the user interface (UI). The system must use the same typefaces and font sizes across the programmed.

Software System Quality Attribute

1. Integrity

QA01: The online shopping cart application can only be accessed by the authorized user.

QA02: The online shopping cart programmed must provide a user-specific interface based on the user type.

2. Correctness

QA03: The assigned task should be received by the specified user.

3. Availability

QA04: The user/administrator will have access to the system all the time.

4. Robustness

QA05: The system shall be able to save items to the shopping cart.

Implementation

The detailed design utilized to create the online shopping-cart application is covered in this chapter. The system's design is used to develop screen layouts, business rules, process diagrams, and other documents that define the functions and operations of the gathered requirements. The new system, which is defined as a collection of modules and subsystems, is described in this chapter's output. The initial input requirements stated in the approved requirements specification document are used in this design stage. For each need, a collection of one or more design elements is created utilizing several prototypes. These design pieces, which include functional hierarchy diagrams, screen layouts, activity diagrams, and class diagrams, detail the desired software functionality. These diagrams are intended to provide a detailed description of the software so that the system may construct the application with little further design input. Mock screen pictures of the system are shown later in this chapter.

Detailed Scope

This project is meant to be completed in three parts, with each phase adding to the project's usability and acceptability.

1. In the first distribution, the application must be able to add an item to the shopping cart and case.

- ✓ On the home page, look through the categories.
- ✓ Choice a category and browse through the items
- ✓ More information on an item can be found by clicking on it.
- ✓ Add an item to the shopping cart.
- ✓ Continue shopping or proceed to the checkout to purchase the item.

2. The app needs to be able to check out the products from the pushcart.

- ✓ Check out the items.
- ✓ Continue shopping.
- ✓ Delete the items to update the shopping pushcart.

3. Before checking out, the programmed requires user authentication.

- ✓ Add items to the cart.
- ✓ Check out the items
- ✓ Log in with a valid username and password.

4. The application must bring up the order form for the check out.

- ✓ Complete the information on the order form.
- ✓ Place the order.

Static Decomposition and Dependency Description

This section includes a system use-case diagram for the online shopping-cart application, as well as a full description for each of the system's use cases.

High-Level Use Case Diagram

The system's use case shows the user a detailed view of the system and how the actors would interact with each other and with the system. The explanation for each use case is then provided below the system use case for the administrator (Figure 1) and the merchant (Figure 2), helping the user to understand who are the actors areas as well as giving the description for each use case along with its pre- and post-conditions that should be satisfied once the use case is implemented in the software.

Figure 1: Demonstrates an administrator's use case in which he or she has access to the programmed. The administrator can navigate to the home page, choose a category, and add/remove products from the shopping cart.

1. **User Authentication:** This class is used to retrieve user information from the database and to verify the users' identities. The methods utilized in this class are depicted in Figure 3's customer flow diagram, and each class's description is mentioned below. User Authentication: This message is used to verify the identity of a user who has provided the system with their login credentials and wishes to log in. This method checks the database credentials. Username to check: This method determines whether the provided username is already in the database. If a user with the same name already exists, the user is prompted to choose a different username while creating an account.

Register User: By submitting a valid username and password, a new user can register for an online shopping cart account. If the username is already in the database, the user will be asked to choose a new one.

Login User: Existing users can use this approach to log into the database using the credentials they used to signup for the application.

2. **DB Controller:** This class is responsible for retrieving users and product information from the database, as well as updating the database with new-user registration, product checkout, and user details. Initialize DB: This function allows the application to create a database on the first run.

Retrieve Items List: This function saves all things from the code/workspace to the database and allows the administrator to view their details. Get a list of categories: This function returns a list of all accessible categories for the shopping-cart application.

1**.Retrieve User's Details:** This role locates all of the database's registered users, as well as any new users who register by filling out the user-authentication form. Get Order Details: This role adds the details of the user who successfully checks out the items and places an order to a database table. Get things in order. Product Data: After the items have been checked out and the order has been placed successfully, this role enters the order's details. This role modifies the row in the database to reflect the specifics of checked-out products.

2**. Place Order:** This class is used to handle all information related to product orders. Get Checkout Information: When the checkout method is called, this method returns information about the goods in the shopping cart. The user-authentication procedure will be called as a result of this. Get User Authentication: This function retrieves information about the user's authentication.

The place-order class is invoked once all three methods are successfully called. If the user authentication fails, the place-order class is not executed.

3**.Retrieve Order Form Information:** This method is executed once user authentication is a success. This method evaluates an order form on the UI for dynamic input from the user. If any of the information is incomplete or is invalid, then an error message is depicted, and the place-order class is not called.

4. **Cart:** The shopping cart is triggered by this class. When no items are added to the cart, when items are already in the cart, or when the user adds an item to the cart, this class can be called. Get the following items: This function obtains information about all products added to the shopping cart by the user.

5. **Checkout:** The checkout button is triggered by this class. When there are products in the shopping cart, the checkout class is called. The checkout class is not called in the case of an empty shopping cart, hence no checkout option is displayed on the user-interface screen. Retrieve Information about your shopping cart: If there are products in the cart, this method obtains the information from the shopping cart and then calls the checkout class. Get the following items: This function retrieves the entire contents of the shopping cart. If the number of products in the shopping cart exceeds zero, this function updates the card information, indicating that the cart is not empty, and calls the checkout class.

Testing

This chapter includes the methods that were used for testing, validating, and evaluating the system.

Methodology

With this testing approach, all of the specs were ready for a prototype, and a plan had previously been created to be demonstrated; the tester began creating code and saw if he or she could get the

same results as the specs indicated. The specifications were tested on each prototype in this manner, and continuous testing was used. This also helped to reduce the amount of testing that had to be done at the end of the product lifecycle. Throughout the process, every part of the software was examined. Steps to take when putting the methodology into practice are as follows:

1. Start with a base function that you want to implement.

2. Create a document that includes a complete need specification, an activity diagram with a flow description, database tables to be used, a component diagram, and a description of each component, including the preconditions and tables that the component will influence.

3.Give the paper to the tester, and work with him or her while he or she builds the code to see if the steps in the document can be implemented, as well as the results of each use case.

4. Move to step 2 if the tester finds a step difficult to implement or believes he or she needs more knowledge to accomplish the capability; otherwise, go to step 3.

5. Ask the tester to log on all the errors and complications he or she encountered while working on the prototype implementation.

6. Work on the other need and expand the prototype to final software once the prototype is complete and the results of the developer's and tester's prototypes match.

7. The following advantages and disadvantages of the testing approach were discovered when it was applied.

Pros of using the methodology

- ✓ Helps give a better understanding about the requirements.
- ✓ Better design at the end of the cycle.
- ✓ Reduced testing to be performed at the end of the cycle
- ✓ Documents produced would be of higher quality.

Cons of using the methodology

- ✓ The person working on the document should be experienced.
- ✓ There are increased time and money involved with testing.
- ✓ Different viewpoints for the same problem can lead to varying results.

Interface Testing

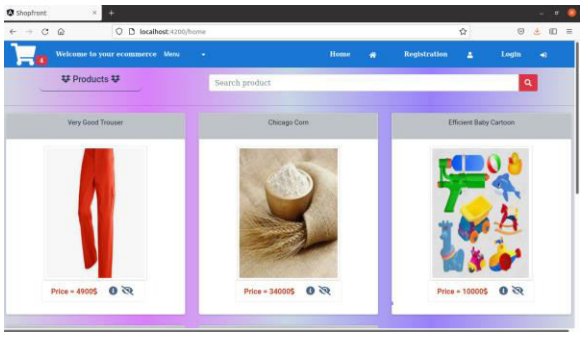
The functional requirements used to create the test-case table, the test cases used to verify the interface table, and the test-cases table results are all listed in this section.

Graphical interfaces of Implementation:

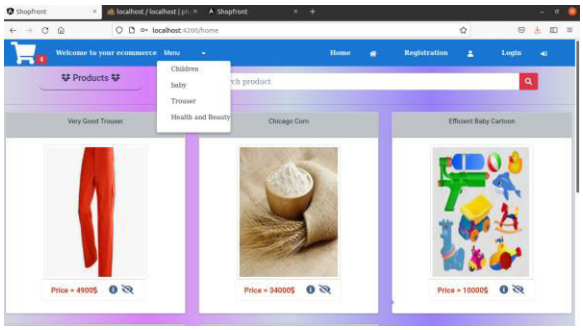
In the implementation, we can mention some important parts shown as below:

Home page

The home page of the application is made up of the list of all products registered in the application.

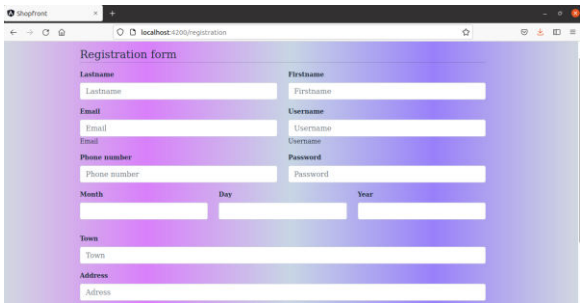


It is made up also with the list of all categories shown by a menu.

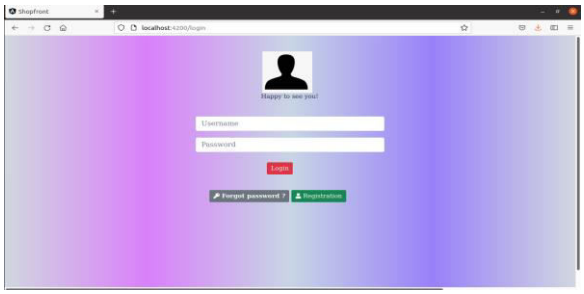


Registration

To register in the application a user must provide these below information:



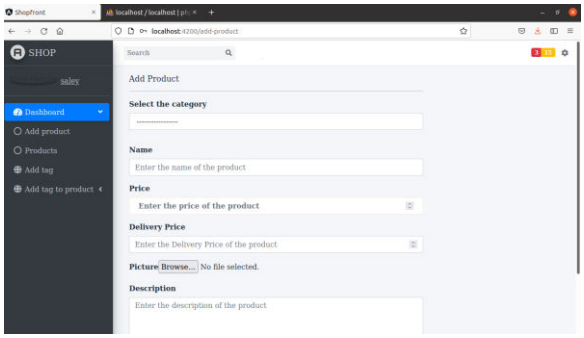
Login



To log in the application the user must provide his username and a password.

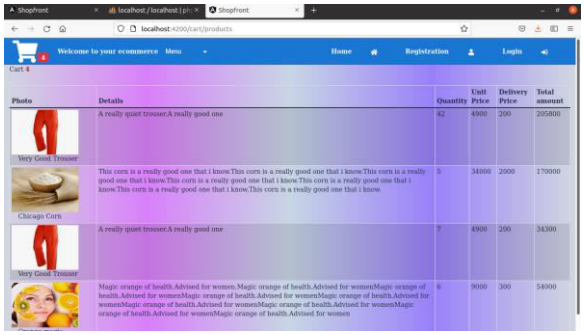
Add Product

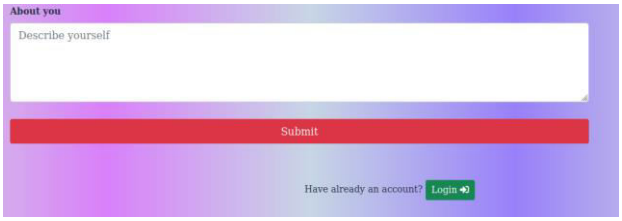
When a merchant is connected to his dashboard he can add a product by filling this form



Cart

A customer can see all products ordered in his card.





Conclusion/future work

The conclusion reached after constructing the current version of the programmed to meet the system objectives is included in this chapter. The system that was constructed is

compared to the initial specifications that were developed at the start of the project. It also outlines the Future Work that will be carried out with later versions of the software. The application's main goal is to assist users in learning the fundamentals of Java, JavaScript, and HTML. Researchers should be able to quickly grasp the implementation by browsing through the programmed and looking at the code for each graphical interpretation. The following results have been achieved after the completing the system and relate back to the system's objective.

1. Users i.e. computer science students, should be able to browse through the code and application: this is accomplished when users, i.e. computer science students, may execute and install the application. When they start the application, they may look at how different objects are implemented.

2. Users should be able to browse via several product categories: this is accomplished when the user initially launches the application and is taken to a home page with categories for all of the different types of items that may be purchased using this online shopping cart application. The user can browse through the categories and click on any one to see the things that belong to that category.

3. Users should be able to add products to their shopping carts as well as read full information about each item: By clicking the cart symbol in the right-hand corner of each item, customers can add any number of goods from any of the listed categories to their shopping cart. By clicking the more icon next to the cart symbol, users may see a thorough description of the item as well as the price.

4. Users should be able to check out their purchases: This is accomplished by clicking the checkout button in the shopping cart. When there are no goods in the shopping basket, the checkout button vanishes. This means that users can only complete the checkout process if they have items in their shopping basket.

5. Users of the system should be able to place orders by filling out the following order form: when a user selects the checkout button, an order form opens on the same page, revealing the detailed order total with shipping charges and sales tax, and requiring the user to fill out all of the information in order to place an order successfully. If any information on the form is invalid or empty, the user will be unable to place an order.

6. After placing an order, the user should receive the following message: when a user successfully places an order by completely filling out all of the rows and then clicking the place-order button, this is accomplished. The user is notified that the order has been placed successfully.

Future Work

The following section discusses the work that will be implemented with future releases of the software.

1. Detailed categories: Future development could include the creation of more detailed categories with additional objects.

2. Work can add a watch list or wish list so that people can add an item to a list and keep an eye on it to see if the price drops or if there is a sale on any of those goods.

3. Improved User Interface: work on improving the user interface by including more interactive features for the user.

4. Add a bar that shows the most-recommended things, based on the number of times each item has been purchased by any user.

5. Payment Methods: include a variety of payment options, such as Visa, MasterCard, PayPal, and others, and allow users to save their credit card information for future purchases.

6. Add many types of shipping options, such as ordinary shipping, expedited shipping, overseas shipping, and so on.

7. Recent History: display the user's recently browsed items in the recent-history tab.

References:

[1] Howe, A. von Mayrhauser, and Mraz, R. T. Test case generation as an AI planning problem. Automated Software Engineering, 4:77-106, 1997.

[2] Koehler, J., Nebel, B., Hoffman, J., and Dimopoulos, Y. Extending planning graphs to an ADL subset. Lecture Notes in Computer Science, 1348:273, 1997.

[3] Treutner, M. F., and Ostermann, H. Evolution of Standard Web Shop Software Systems: A Review and Analysis of Literature and Market Surveys.

[4] Jarvenpaa, S. L., and Todd, P. A. (1997). Consumer reactions to electronic shopping on the World Wide Web. International Journal of Electronic Commerce, 1:59–88.

[5] Peterson, R. A., Balasubramanian, S., and Bronnenberg, B. J. (1997). Exploring the implications of the internet for consumer marketing. Journal of the Academy of Marketing Science, 25:329–346.

[6] "More Consumers Using Tablets to Holiday Shop [STUDY]." Retrieved on Mashable. December 8, 2011. " http://mashable.com/2011/12/08/tablets-to-holiday-shop/"

[7] Bigne, Enrique. (2005). The Impact of internet user shopping patterns and demographics on consumer mobile buying behavior. Journal of Electronic Commerce Research, 6(3).

[8] King, Stephen F. en Juhn-ShiuanLiou (2004), A framework for internet channel evaluation. International Journal of Information & Management 24:473–488.

[9] Falk, Louis K., Sockel, Hy, and Chen, Kuanchin. (2005). E-Commerce and consumer's expectations: what makes a website work. Journal of Website Promotion, 1:65–75.

[10] Elliot, Steve, and Fowell, Sue. (2000). Expectations versus reality: a snapshot of consumer experiences with Internet retailing", International Journal of Information Management 20: 323–336.

[11] Java vs.C#, Retrieved on 2013. "http://www.cs.yale.edu/homes/hudak/CS112F06/java.html"

[12] Java Shopping Cart Powered by SoftSlate., R etrieved on 2013. "http://www.softslate.com/featureList.html"

[13] Commerce4j, Java Based e-Commerce Application and Online Catalog Management System. , Retrieved on May 18, 2010. "https://code.google.com/p/commerce4j/"

[14] CS-Cart.com (Simbirsk Technologies Ltd), © 2004-2013.http://www.cs-cart.com/

[15] Ofbiz, The Apache Open for Business Project ., Retrieved on 2013. "http://ofbiz.apache.org/index.html"

[16] Comparison of shopping cart software., Retrieved on June 28, 2013. http://en.wikipedia.org/wiki/Comparison_of_shopping_cart_software