

Lamar University
Department of Computer Science

A Project
Report On

CPSC 5360_48F_1
Software Engineering

Entitled
**“The Simulation of an online purchasing of a
product “**

Instructor:
Dr. Stefan Andrei

Submitted By:
Monica Pandey(L20400582)
Palak Shah (L20400586)
Bhavya Dalwadi(L20393359)
Priteshkumar Patel (L20399769)

Submitted On: 05/09/2016

ACKNOWLEDGMENT

I would like to express my gratitude to every person who helped me and supported for completion of this project in all possible ways. First, I would like thank to Dr. Stefan Andrei for helping and guiding the team throughout master's degree. I would also like to convey my thanks to Department of Computer science for providing the courses and environment that assisted our team in completion of this project and report. Lastly, I would like to thank our friends for their continuous support, without that this project would not have been possible at all. I am obliged to all those individuals whose guidance lead to the completion of this report.

Finally, I would like to thank all my friends for their help, suggestion or comments and encouragement that directly or indirectly helped me to complete the report.

ABSTRACT

This report is based on the e-commerce application that has been developed for the fulfillment of course requirement of Software Engineering. The project is developed using use-case driven object-oriented approach to develop a system to support the simulation of the implementation of the bank transactions using credit card to purchase an item online. This project is developed in order to make students knowledgeable in the field of ecommerce and help them to figure out how an interactive ecommerce application is developed from base level using client-side languages such as JavaScript and HTML. “The simulation of an online purchasing of a product” is semester project for the semester Spring 2016 as a part of CPSC 5360 “Software Engineering” course. In this system we will be simulating how a user will be purchasing a product from online store.

The major aspects of application development have been discussed for the project being considered. Asp dot net and C# has been used to implement classes and the functionality of the system has been shown through a menu-driven interface. In this report, all the aspects of software development have been presented in brief. The main purpose of this project is to describe simulation of the implementation of the bank transactions using a credit card to purchase an item online.

Contents

1 Introduction

| | |
|--|----|
| 1.1 Motivation..... | 5 |
| 1.2 Objective | 5 |
| 2.1 User Class and Characteristics:..... | 6 |
| 2.2 Product Function..... | 6 |
| 2.3 External Interface Requirements..... | 6 |
| 2.3.1 User Interface..... | 6 |
| 2.3.2 Hardware Interface..... | 7 |
| 2.3.3 Software Interface | 7 |
| 2.3.4 Communication Interface..... | 7 |
| 2.4 Constraints | 7 |
| 2.5 Assumptions and Dependencies..... | 7 |
| 2.6 Specific Requirements | 7 |
| 2.6.1 Functional Requirements | 7 |
| 2.6.2 Non-Functional Requirements | 8 |
| 2.6.3 Performance Requirements | 8 |
| 2.6.4 Quality Requirements | 9 |
| 3.1 Use Case Diagram..... | 10 |
| 3.2 Sequence Diagram | 13 |
| 3.3 Domain Model | 14 |
| 3.4 Activity Diagram | 15 |
| 3.5 State Chart Diagram..... | 16 |
| 3.6 Class Diagram..... | 18 |
| 4 Testing | |
| 4.1 Testing: | 19 |
| 5 Conclusions | |
| 5.1 Conclusion | 21 |

1. Introduction

The use of internet has increased rapidly in past few decades. With the increment in internet usage, the use of application for online purchasing has also been increased. In today's fast paced life everyone is engrossed towards online shopping. Online shopping not only has benefitted customers but also the business owners. It has really made easier for customers who are engulfed in the flow of time by providing wide range of products with reasonable price, comprehensive information of the products and simplified navigation for searching the related product. Similarly, business owners get profited from online shopping as they don't have to spend as more as they are required to as opening and running a store in physical location. On the other hand, their variety of products gets exposed to large number of customers. Online shopping has gained vast popularity in past decades as people find it more convenient compared to physical shopping. Not only that, it has been a platform for a lot of small-scale manufacturers to sell their product in global market directly without involving middlemen before their products can be available at the physical store. Manufacturers and business owners can simply make their own website and sell their products easily.

1.1 Motivation

Internet has been growing vastly in past few decades. The use of internet has also been changed from a source of information to a complete transaction enabler. Anyone browsing through internet today is able to purchase variety of goods online. However, the rapid growth of this business is being hindered by the perception of customers regarding poor internet security. Most payment methods revolve around the credit card, and customers are quite diffident to reveal their card information on the Internet. Also, not surprisingly, credit card frauds on the Internet have registered a dramatic increase. The purpose of this project is to describe a simulation of the implementation of the bank transactions using a credit card to online purchase an item.

1.2 Objective

"The simulation of an online purchasing of a product" is semester project for the semester Spring 2016 as a part of CPSC 5360 "Software Engineering" course. In this system we will be simulating how a user will be purchasing a product from online store. The main objective of this project is to help students of computer science to help them understand and learn about the ecommerce applications as well as application development. The application helps students to learn about object oriented approach, develop complete application and give vision about how client side scripting language interacts with server side scripting language. The main purpose of the project is to describe a simulation of the implementation of bank transactions using a credit card to purchase an item online.

2. System Requirement Specification

The developed application is a web based application which can be accessed through all the browsers. All the functional and non-functional requirements along with constraints, assumptions and dependencies are described in following section.

2.1 User Class and Characteristics:

Based on the roles, there are two types of user of the developed system:

- **Customers (Registered/Unregistered):** These users able to view home page and browse different categories of the product displayed in the webpage. They can add items they want to buy in the shopping cart and check out the order. They can also add the products in the cart and save them for later viewing. Users can also delete the items added in the cart or continue shopping for other items after adding to the cart.
- **Admin:** They can add, edit and delete the product and update the description and provide services to the customer. Admin is responsible for all the training documents required for the system.

2.2 Product Function

The developed application has must provide following functionalities:

1. The system should display all the products available for the system and display all the information related to the selected product.
2. The system should allow admin to add new products, remove or update the product description or modify the price of each product.
3. The system should allow users/admin to remove the items.
4. The system should allow user to create and edit their profile and set their credentials.
5. The system should provide shopping cart during online shopping.
6. The system should allow user to add or delete items to the cart and check out the order.
7. The system should allow user to confirm purchase.
8. The system should calculate taxes and perform payment.
9. The system should enable user to enter the card information and shipping details
10. The system should allow user to cancel the order.

2.3 External Interface Requirements

This section discusses other requirements of the system.

2.3.1 User Interface

User of the system will be provided with the Graphical user interface and shall provide constant look and feel throughout the webpages. The user interface will be compatible to any browsers such as Internet Explorer, Mozilla Firefox or Google Chrome through which user can access the system. The user interface has been implemented using Asp .net and C#.

2.3.2 Hardware Interface

Since the developed system must run over internet, all the hardware that are required to connect to the internet will be the hardware interface for the system. For example: Modem, WAN, LAN, Ethernet cable. Also, hardware configuration required for a PC using developed system is;

- Pentium Processor
- 32 MB of free hard-drive space
- 128 MB of RAM

2.3.3 Software Interface

All the minimum requirements such as operating system, drivers, IDEs that are required to run the system efficiently are listed below:

1. **Operating System:** Windows (Vista/Windows 7) or MAC OS
2. **Web Browser:** Internet Explorer (8.0 and above), Mozilla Firefox (3.0 and above), or Google Chrome.
3. **Integrated Development Environment:** Java EE
4. **Development Language:** Java, Java Server Pages

2.3.4 Communication Interface

The two parties should be connected through either by LAN or WAN for communication. The protocol used shall be HTTP and port number used will be 80. Also, there should be logical address of the system in IPv4 format.

2.4 Constraints

1. The system should always be available.
2. The application should be built using Asp.net, and it should, initially, be accessible through the eclipse IDE and later published on a server
3. The minimum hardware requirement should be as listed in section above.

2.5 Assumptions and Dependencies

1. We assume that users and the administrator are trained to use the developed system.
2. We assume that system users follow to the system's minimum software and hardware requirements.
3. This system will use third-party software, and it is assumed that system users are familiar with the software.
4. The system must be able to respond to database software within reasonable time.

2.6 Specific Requirements

This section describes about all the functional and non-functional requirements, system inputs, functions performed by the system, system output.

2.6.1 Functional Requirements

User: Customer

| S.N | Use Case | Description | Dependencies |
|------|--------------|---|--------------|
| FR01 | Registration | Customer gets registered when they provide valid information of their | None |

| | | | |
|------|--------------------|--|---------------------------------|
| | | details. | |
| FR02 | Login | Customer gets access to the system if the login credentials are correct | FR01 |
| FR03 | View Items | System displays all available items and allows filtering based on certain details items, price, When customer selects the item then it gets added to the shopping cart | FR02 |
| FR04 | Add Items | Customer adds items to the cart | FR02, FR03 |
| FR05 | Delete Items | Customer deletes items on the cart | FR02,FR03,FR04 |
| FR06 | View Shopping Cart | Display the shopping cart with the selected items | FR02,FR03,FR04,FR05 |
| FR07 | Checkout | Customer clicks checkout button, system performs calculate taxes, System performs Payment | FR02,FR03,FR04,FR05, FR06 |
| FR08 | Payment | System displays the total amount to the customer, Customer enters the card information , Customer enters shipping details, Customer clicks pay option | FR02,FR03,FR04,FR05, FR06, FR07 |

User: Admin

| FR | Use Case | Description | Dependencies |
|------|-----------------------|--|--------------|
| FR09 | View user Information | Admin views the user's information. | FR01 |
| FR10 | Add Products | Admin adds products | None |
| FR11 | Update Products | Admin modifies or edits the price and description of product | FR10 |
| FR12 | Delete Products | Admin deletes the products | FR10 |

2.6.2 Non-Functional Requirements

Following are the non-functional requirements for the developed system:

1. Secure access and security of customer's confidential data.
2. The website should always be available.
3. The system should perform better even at the peak time.
4. System should be reliable, secure and maintainable.
5. It should also be portable and has proper resource allocation.
6. The system should run on every environment.
7. The system should be secure in terms of data transfer and data storage.

2.6.3 Performance Requirements

| Performance Requirement | Description |
|-------------------------|--|
| PR01 | User is able to add items to the cart in lesser than 10 seconds. |

| | |
|------|--|
| PR02 | User is able to view items in the cart in lesser than 10 seconds. |
| PR03 | The navigation between pages should not take longer than 3 seconds |
| PR04 | The application will check for validation of information entered. |

2.6.4 Quality Requirements

| Quality Requirements | Description |
|----------------------|--|
| Integrity | |
| QR01 | The authenticated user should be allowed to access the system |
| QR02 | Application shall provide specific user interface based on user (Registered/ Unregistered) |
| Correctness | |
| QR03 | The system shall display correct results every time. |
| Availability | |
| QR04 | The system shall be available every time. |
| Robustness | |
| QR05 | The system shall be able to save items to the cart every time. |

3. Implementation

This section provides the comprehensive design used to build the system. This section provides diagrams with the intension to describe the developed system in detail. The included diagrams are **Use-case Diagram, Class Diagram, Activity Diagram, Interaction Diagram and Domain Model** with detailed description.

3.1 Use Case Diagram

Use Cases Identification

A preliminary list of use cases are:

1. Customer login to the system
2. Register new customer
3. Customer Logs out of the system
4. View Items
5. Make Purchase
6. Checkout
7. Payment
8. View/Update Shopping Cart

And for the credit card processing system some use cases identified are:

1. Authorize
2. Capture
3. Void
4. Verify

Actors:

1. Primary Actors: online users – (registered/unregistered)
2. Secondary Actors:
 - Merchant's Credit Card Processing System (e.g. Verisign)
 - Merchant's bank
 - Customer's credit card bank

Now we will be presenting the facts in a use case diagram that will show interaction among these use cases and possible actors of the system.

The use case diagram in Figure 1 below is that of online shopping with a focus on credit card payment option. The online customer actor surfs the web page to purchase some items online. The actor can be either registered to the online shopping site or an unregistered user. In this model the unregistered client, if any, is made to register first before they can perform any checkout. But the diagram can be modified to accommodate a guest user checkout as well.

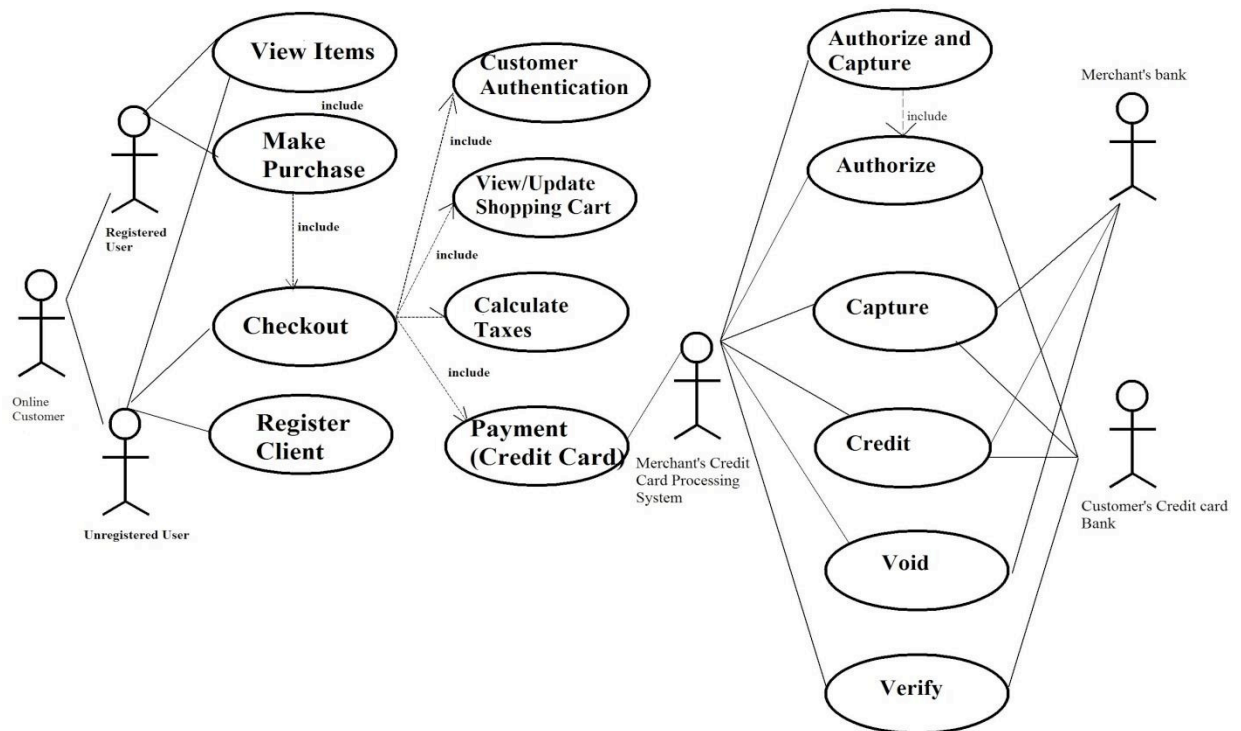


Figure 1 . Use Case Diagram

View Items: The view items use case alone can be made a top level use case for just “Window shopping”. In this case the web customer just views the items and signs off. The view items can also be made a part of Make Purchase Use case. The view items can, although not necessary on all occasions, extend to other use cases such as search items, browse items, wish list etc.

Make Purchase: The make purchase use case cannot be a single entity, that is, it must include view items and proceed to checkout use cases. In other words, the customer must view the items in order to make a purchase and must proceed to checkout to complete the purchase.

Register Client: The register client use case is for the unregistered customer who wants to make a purchase. In this case, the unregistered client can view the items but needs to register in case he wants to make a purchase.

Use Case: Customer Login; Basic Flow

Actor: Customer

- Customer enters username and password
- Customer gets access to the system.

Use Case: Customer Login- invalid credentials; Alternative Flow

Actor: Customer

- Customer enters invalid username or password
- System displays error and terminates

Use Case: Register Customer; Basic Flow

Actor: Customer

- Customer provides personal information telephone number, name, and address.

- System validates the information.
- System stores the information and displays approval message.
- System performs Customer Login

Use Case: Register Customer- invalid information; Alternative Flow

Actor: Customer

- Customer provides personal information telephone number, name, and address.
- System validates the information and cannot match the record.
- System displays “record not found” message and terminates.

Use Case: View Items; Basic Flow

Actor: Customer

- System displays all available items and allows filtering based on certain details : items, price
- When customer selects the item then it gets added to the shopping cart

Use Case: View Shopping Cart; Basic Flow

Actor: Customer

- Display the shopping cart with the selected
- System allows customer to edit the quantity of the items or remove the items

Use Case: Delete Customer Record; Basic Flow

Actor: System Admin

- Performs View Record
- Admin selects the record to delete
- System deletes the record, logs it and display success message.

Use Case: Checkout; Basic Flow

Actor: Customer

- Customer clicks checkout button
- System performs calculate taxes
- System performs Payment

Use Case: Payment; Basic Flow

Actor: Customer

- System displays the total amount to the customer
- Customer enters the card information
- Customer enters shipping details
- Customer clicks pay option
- Payment is successful and system displays the receipt

Use Case: Payment - user cancels; Alternative Flow

Actor: Customer

- System displays the total amount to the customer
- Customer enters the card information
- Customer selects the cancel option
- Perform View Shopping Cart.

Use Case: Payment- payment not successful; Exceptional Flow

Actor: Customer

- System displays the total amount to the customer
- Customer enters the card information
- Customer enters shipping details
- Customer clicks pay option
- Payment is not successful
- System displays the error message

After the client views items and the registered user decides to purchase a product, he is moved to a checkout page. The checkout can be of different form but here we use only the credit card service of payment.

The primary actor for the processing system is the merchant's bank. The use cases for this process are shown. The authorize and capture is the most common form of transactions. Here the payment is first authorized by Customer's Bank and approved for settlement which, if approved, gets deposited to the Merchant's bank. The credit use case takes care of refunds, voids for cancels and verify to check for user's data

3.2 Sequence Diagram

In this section, we have presented sequence diagrams for our project. Here we discuss few sequence diagrams including Login, placing an order.

Login

Figure 2 shows sequence diagram for selecting the product, adding it to cart and placing the order.

Customer provides username and password („usr" and „pwd") to the user interface, it checks the session. If the user is already stored on session, it returns the user. Otherwise, user is created and stored on session. The user is verified with the database object and status is sent back to the customer.

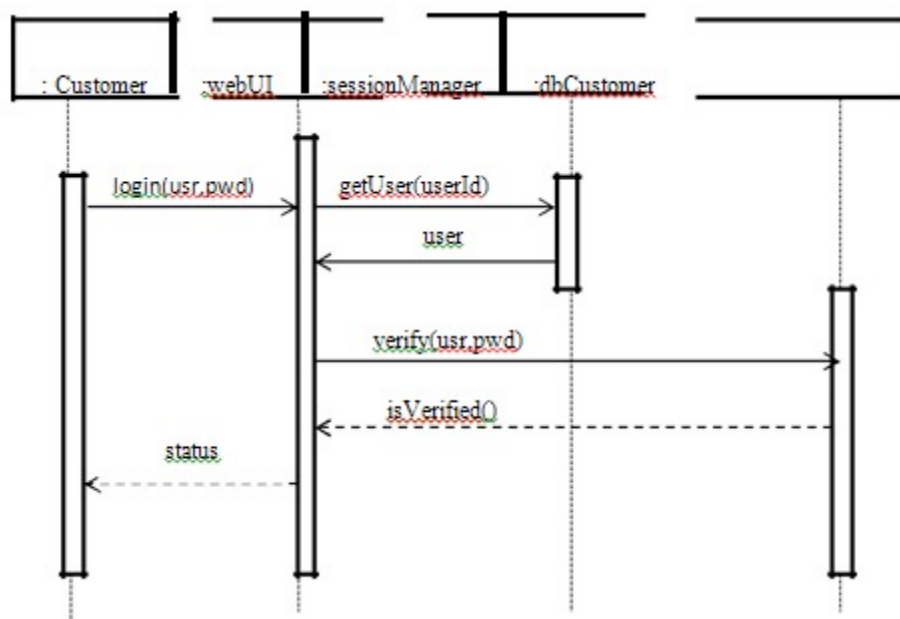


Figure 2 Sequence diagram for Logging in to the System

Placing an Order

Figure 3 shows the sequence diagram for selecting the product, adding it to cart and placing the order.

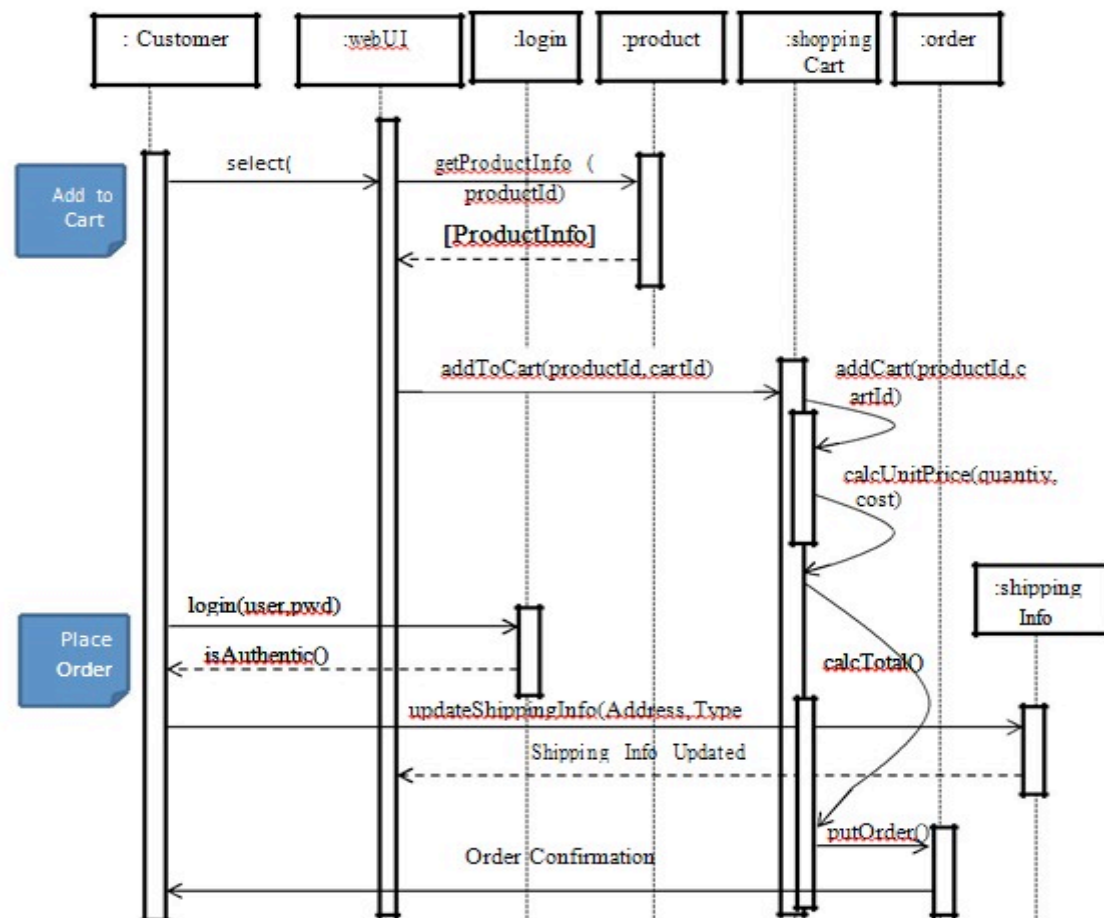


Figure 3: Sequence diagram for selecting product and placing and order

Above sequence diagram shows how a customer can select an item and place an order. Customer can select items from the web user interface and get the full product information using `getProductInfo()` method. `addToCart(productId, cartId)` adds product with „productId“ to the cart with „cartId“. `calcUnitPrice(quantity, cost)` calculates total unit price based on quantity. When user provides the shipping information then the `updateShippingInfo()` method updates the shipping details and acknowledges back to the user. Once the user provides shipping details and is confirmed, order is placed. User is acknowledged back with the confirmation message.

3.3 Domain Model

Let us describe the basic behavior and incorporated data for our system using domain modeling.

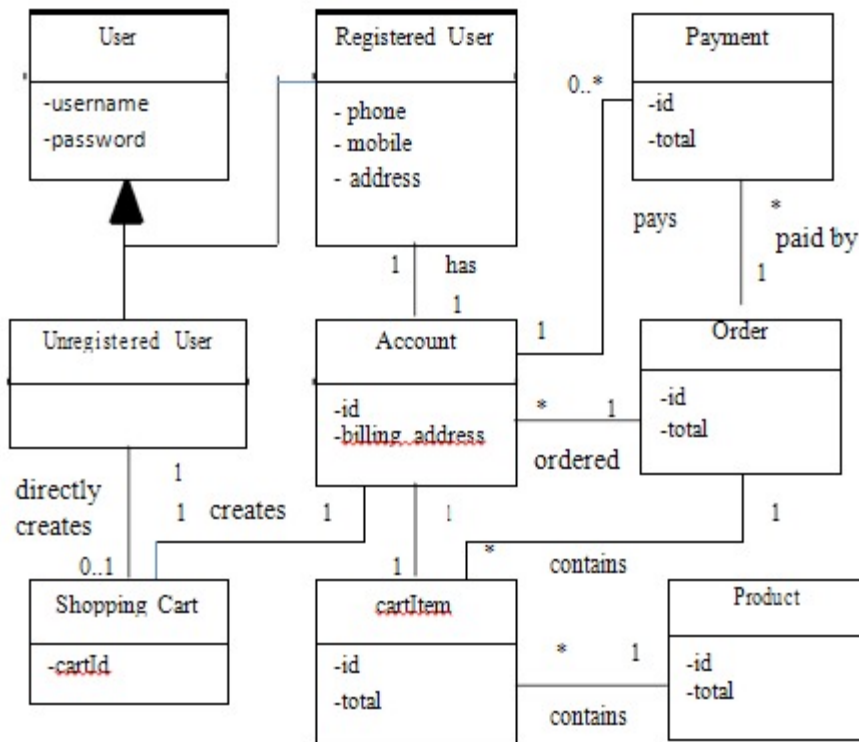


Figure 4: Domain model diagram for “The Simulation of an online purchasing of a product”

Domain model for our system presented above describes the basic outline on how our system functions. Our system has two types of users: unregistered user and registered user. Unregistered user is allowed to shop directly as a guest or is allowed to register to the system. Registered user has an account associated with that user. Shopping cart consists of the product item/s. Once the user selects the checkout option, an order is created. Payment is made for the order through the account. Above domain model presents a preliminary overview of the system.

3.4 Activity Diagram

Figure 5 below establishes the activity flow for the system developed. The flow of activity is comparable for both the user and administrator. User at first opens the application in the web browser. User browses through the displayed list of categories and can choose either to select a category or to directly view the cart. User can view detailed description of the selected product before adding it to the cart. The user can then either continue shopping or can check out by clicking on the checkout option. If there are no items in the cart, then user will not have option for checkout. Checkout can be done by registered user only. Without logging in the system user cannot checkout.

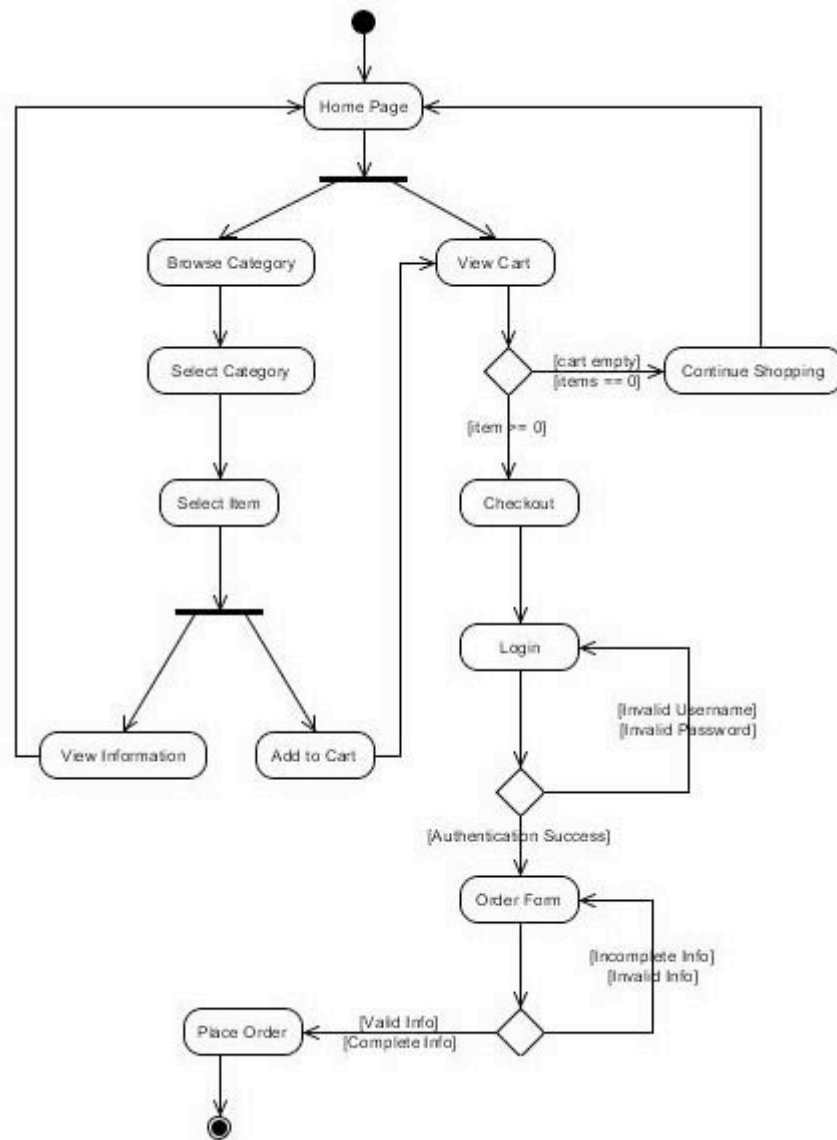


Figure 5: Activity diagram for “The Simulation of an online purchasing of a product”

3.5 State Chart Diagram

State chart represents the state logic of the components of the system. It describes the different states of a component in a system. State chart can be defined as a machine containing different states which are controlled by the external or internal events. The state chart to represent the simulation of the purchasing of a product by a customer based on the sequence diagrams we accomplished above can be represented as below.

State Diagram for logging in to System:

According to our sequence diagram, customer is allowed to login to the system and necessary validations are performed. Let us consider one for thing that only up to 3 invalid logins are allowed otherwise the account will be suspended.

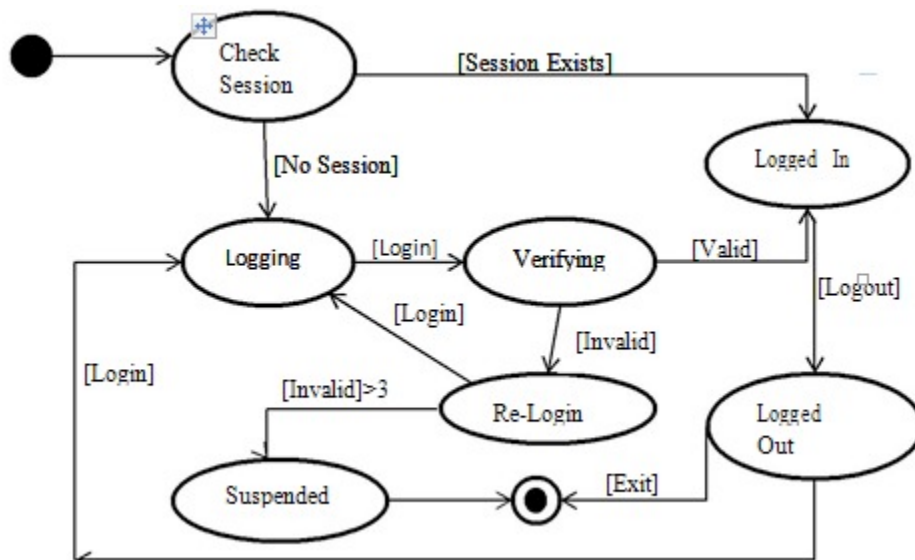


Figure 6: State chart for Login to the System

Complete State Diagram for placing an order:

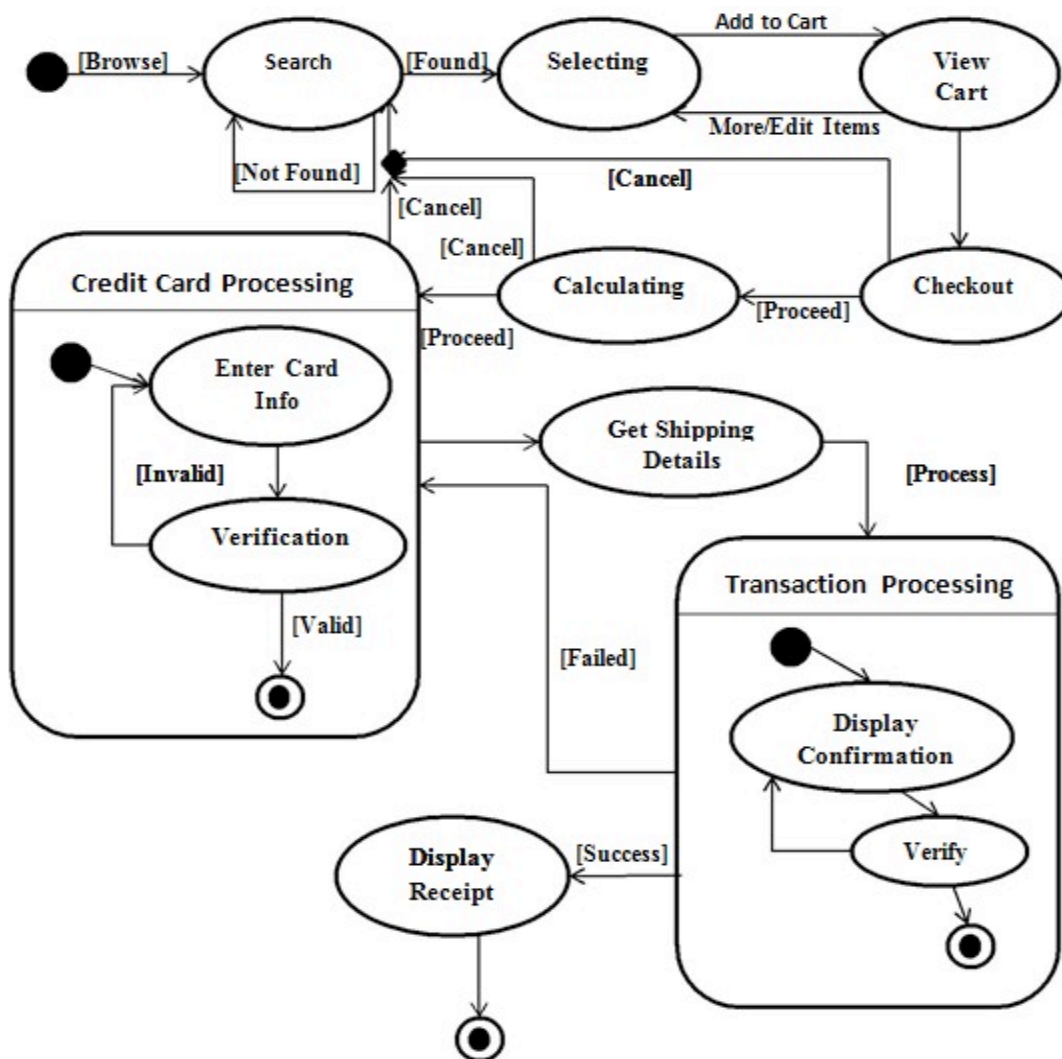


Figure 6: State chart for placing an order

We have to draw a state diagram for the customer placing an order and purchasing an item from the online site. Customer must be able to search for any item in the site, view the details, add an item to the cart, and edit the item as required. S/he is also allowed to add multiple items. Then after confirming the order in the cart, customer is given an option to checkout. Customer provides the credit card information along with the shipping details and the confirmation is displayed in the screen. Customer is allowed to cancel at any time. Necessary validation of the card information is performed.

3.6 Class Diagram

Class diagram is a refinement over the domain model designed above. We add all the possible properties and behaviour (methods) for the classes mentioned above.

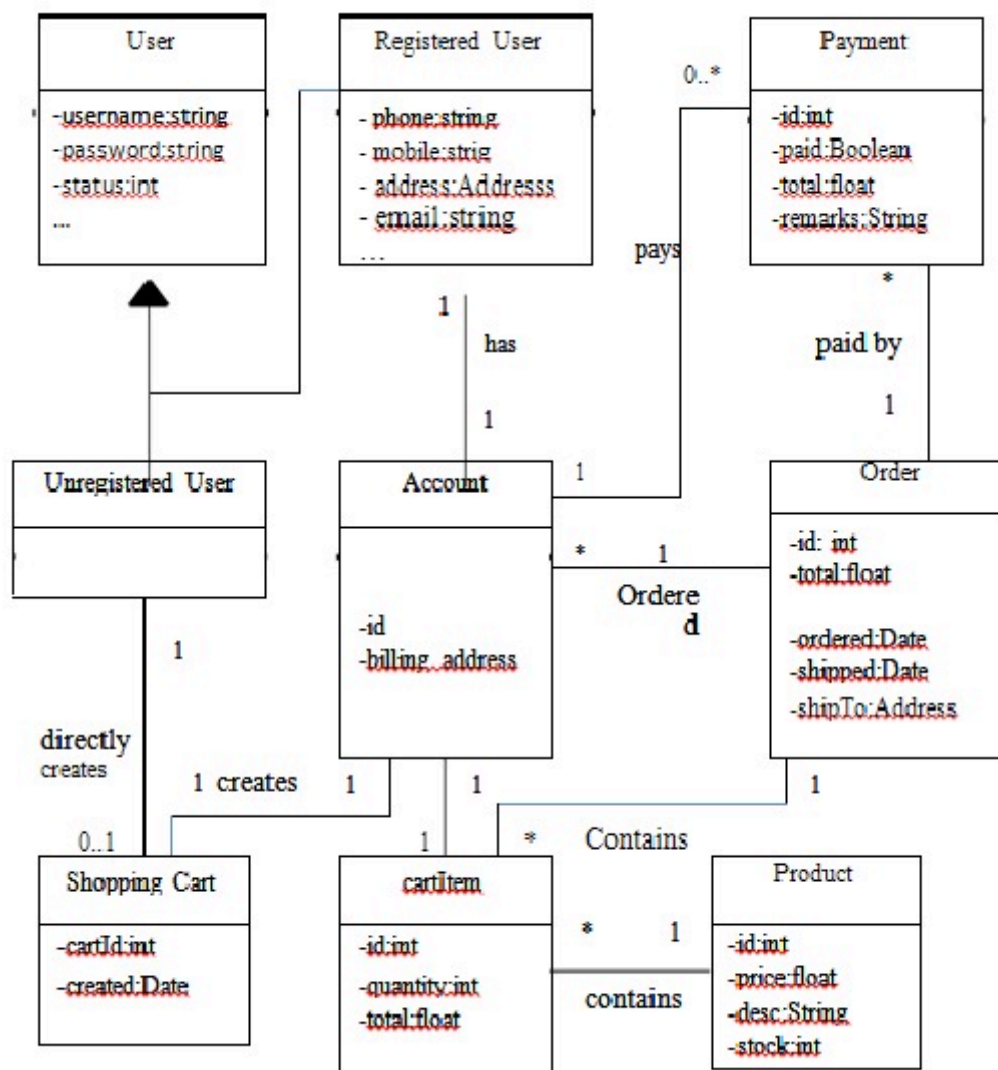


Figure 8: Class diagram for “The Simulation of an online purchasing of a product”

4. Testing

In this section we discuss the methods that were used for testing the system. We use the list of functional requirements mentioned in section 2 to write down the test cases for the purpose of testing. The test case developed is shown in table below:

4.1 Testing:

| FR Number | Test Case No | Test Case Description |
|-----------|--------------|---|
| FR02 | TC01 | To test the Login interface for the users. |
| FR10 | | To test the Login interface for the Admin. |
| FR06 | TC02 | To test whether users can view the items they added in the shopping cart. |
| FR10 | TC03 | To test whether Admin can add new products |
| FR09 | TC04 | To test whether Admin can view all the users registered in the system |
| FR07 | TC05 | To test whether user can check out with an empty shopping cart. |

Now we discuss the steps that should be taken by the user, the conditions for successful execution of the test case, and the end result for the test case to pass.

1) TC01

- **Input:** Login credentials
- **Output:** Valid Destination Page
 - i. If (User == Valid User), an order form appears to complete the checkout process.
 - ii. If (User! = Valid User), an error message is displayed on the Login interface.

2) TC02

- **Input:** Add an item to shopping.
- **Output:** The shopping-cart page should be displayed showing the item that is added by the user.
 - i. If (Selection == Item and document == exists), the user is able to add that item to the cart, and the item shows up in the shopping cart, prompting user to delete the item, to continue shopping, or to check out the item.
 - ii. If (Selection = Item and Selection = View Cart), an empty shopping cart pops up with buttons to check out or to continue shopping.

3) TC03

- **Input :** User=Admin and Selection=Items
- **Output:** New or modified items or categories in the shopping cart.

- i. If (User type = “Admin” & Selection = (Items || Category)&& Item/Category = existing), then display the modified items or categories in the shopping cart.
- ii. If (User type == “Admin” & Selection == (Items || Category) & Item/Category=existing), then display newly added items or categories in the shopping cart.

4) **TC04**

- **Input:** Login credentials ; User = Admin **and** Selection = View Database
 - **Output:** User List
- i. If (login type == “Admin” & Database.clicked = „true“ and list.clicked=true and userlist.exists=true), then display users.
 - ii. If (login type == “Admin” & Database.clicked = „true“ and list.clicked=true and userlist.exists=false), then display the empty database.

5) **TC05**

- **Input:** Login Credentials **and** User = Users
 - **Output:** Disabled checkout button.
- i. If (login type== “Users” & Items.AddToCart = „false“ & ViewCart.clicked=“true”), then display the empty shopping cart with no items and a disabled checkout button.
 - ii. If (login type == “Users” & Items.AddToCart = „true“ & ViewCart.clicked=“true” & Checkout.clicked==“true”), then display items in the shopping cart with the checkout button enabled so that users can check out.

Now we will display the results that we obtained from performing the above mentioned test cases. The following table shows whether the resulted output matched the expected output.

| Test Case Number | Expected Output | Actual Output |
|------------------|-----------------|---------------|
| TC01 | Pass | Pass |
| TC02 | Pass | Pass |
| TC03 | Pass | Pass |
| TC04 | Pass | Pass |
| TC05 | Pass | Pass |

5. Conclusions

5.1 Conclusion

“The simulation of an online purchasing of a product” is semester project for the semester Spring 2016 as a part of CPSC 5360 “Software Engineering” course. Through this system we learnt simulation of how a user will be purchasing a product from online store. The main objective of this project was to help students of computer science to help them understand and learn about the ecommerce applications as well as application development. The application helped us to learn about object oriented approach, develop complete application and give vision about how client side scripting language interacts with server side scripting language. The main purpose of the project was to describe a simulation of the implementation of bank transactions using a credit card to purchase an item online and thus has been achieved.