CO3015 Computer Science Interim Report

Opule Accessories E-commerce Website

**Paidamoyo Mangi**

**Tuesday 24th of November 2020**

Department of Informatics

University of Leicester

Table of Contents

1. Aims & Objectives of the project
   1. Aims
   2. Customer User Requirements
   3. Admin User Requirements
   4. Objectives
2. Software Architecture, Algorithms, Data Structures
   1. Client/Server Diagram
   2. DAO/Model Layer Architecture diagram
   3. User Management Implementation Class Diagram
3. Prototype
   1. Features completed so far
   2. Tested – 4 features
   3. Prototype Diagrams
4. Survey of Literature
   1. Introduction
   2. Introduction of context
   3. Context Aware Applications and examples
   4. E-commerce websites
5. Planning & Timescales
6. Bibliography & Citations
7. Aims and Objectives
   1. Aims

I currently own a small business that sells faux eyelashes, jewellery and other accessories. For the past few months, I’ve been taking orders through social media or in person, however, I’ve now decided to invest my time into developing my own business website, whilst utilizing the technical skills and knowledge that I’ve acquired from my computer science degree and obtaining new skills through the development. Additionally, we’re living in times where a lot of things are being digitalized, due to the popularity of the internet, social media and progressing customer’s shopping habits.

The name of the web application is Opule Accessories. It will be comprised of 2 parts: the website front-end (customer shopping) and the backend (administration portal). This web application will be integrated with a social media platform (Facebook/Instagram) in a way that the users can view the product reviews left by customers on the social media. Additionally, this web application will have a ‘spin the wheel’ game on the home page, which includes a number of prizes (different types of discount codes ; 50% OFF, 30% OFF etc) and a user can partake by entering their email address and clicking on the spin button. The captured email address will be stored in the mail-list and used for email marketing purposes.

1.2 Customer User Requirements

* A user must be able to browse categories, products, product information + product image
* A user must be able to create an account, sign in and subscribe to a mail list.
* A User should be able to see most recently added products, best-selling and most favoured.
* A user should be able to participate in a ‘spin the wheel’ game, which captures the user’s email address for the mail list and gives a discount code in return.
* A user should be able to add a product to cart and edit the quantity and must be able to view the cart.
* The information in the shopping cart should be maintained during the customer’s session, therefore a user should be able to continue navigating the site before placing an order.
* A user should be able to add & view product reviews.
* User must be able to add products to a wish list and edit the wish list
* User must be able to search for products
* A user must be able to place an order using their preferred method of payment – (PayPal, credit card).
* A customer must receive a confirmation email after placing an order followed by a ‘dispatched order’ email’

1.3 Admin User Requirements (backend)

* The administrator must be able to perform user management activities such as creating a new user and manipulating user details.
* The administrator must be able to add and manipulate product categories details.
* The administrator must be able to add new products, manipulate the details as well as delete the product.
* The admin must be able to see new orders and reviews
* The admin must be able to see order and customer details
* The admin should be able update the status of an order and send automated order emails (order confirmation, order dispatched email).

1.4 Objectives

* Learn & Implement a “Spin the wheel” game/ mini application that captures the user email and gives the user a discount code in return.
* Learn & Implement a PayPal and credit card payment system at the checkout page.
* Integrate JUnit, UI Testing and Automation testing.
* Setup a SQL database to store user & admin data as well as product related information (reviews etc)
* Setup a Tomcat Apache server to host the system which will publish the web application.
* Implement Simple Mail Transfer Protocol client service to automatically send email notifications regarding order and user information.
* Build an application using MVC framework for building a web application.
* Sync the website with a social media platform (Instagram, Facebook), so that reviews that are added on these platforms can also appear on the review section of a product.

1. Software Architecture

This section will illustrate the underlying software/system architecture of the web application. Polytechnique defines systems Architecture as “a response to the conceptual and practical difficulties of the description and the design of complex systems” [1].

2.1 Client/Server Architecture Diagram

Diagram

Description automatically generated

This diagram illustrates the system architecture, it shows how the system is deployed and operates in the production environment. Customers and the administrator use a browser to access the opule website via the internet. The web application is hosted by tomcat server, which runs on Java Virtual machine and can be installed on any operating system. My web application interacts with the RDBMS/database via Java persistence JPA/Hibernate. The application server and database server can run on different physical machines, but they must both be inside the opule corporate network, which is protected by a firewall. The admin(manager) can access the website’s backend from within the opule network or from outside, via the internet.

2.2. DAO/model layer. Application Architecture Diagram

Diagram

Description automatically generated

The Model-View-Controller pattern is well known in software engineering and web application development. I’ve integrated this framework into my work to improve the development process.

View

The view layer, also known as the presentation layer, is used to create the user interface. This layer can generate dynamic content, based on the user’s request. This contains HTML and java code as well as JSTL(Java Server Pages Standard Tag Library). JSTL is a collection of useful JSP tags which encapsulates the core functionality common to many JSP applications.

Explanation

In my application, the Login.jsp is a view page that displays a login form. When a user enters login information, the adminLoginServlet is called, which invokes the userService class, which encapsulates all the business related to the user management, and in turn calls userDAO and passes the user object to authenticate the user information. The userDAO then uses JPA/Hibernate to query the database. If the login information is correct, a value is returned to the userService, value= True is successful. The value is returned to the servlet and the servlet decided which request will be sent to the user – login success page or incorrect.

Controller

The diagram illustrates that the userDAO class belongs to the controller layer. The view layer communicates with the controller layer, which is responsible for processing user requests. This layer is also accountable for rendering a response with the aid of both the model.

Business Service

Instead of implementing business logic directly in java servlet (controller), java servlet communicates with business/service layer. I’ve organised java classes that rule the business logic in the service layer.

Data Access Object/Model

In turn, the business/service layer communicates with the Data Access Object/model layer. The model layer is the central component of the pattern, it is the application’s dynamic data structure, independent of the user interface. Additionally, it manages the data, logic and rules of application. The DAO is responsible for updating and retrieving data from the database.

JPA/Hibernate

The Java Persistence API/Hibernate layer is a setup interface that defines how to model retrieving and updating the data for the enterprise of my application. JPA is a specification that defines what to do and hibernate is an implementation that takes care of the how to do. Hibernate framework simplifies data manipulation. Hibernate communicates with the database via JBDC (Java Database Connectivity), an API that allows the java program to communicate with the database.

* 1. Class diagram for the User Management Implementation

Diagram

Description automatically generated

This is a class diagram for the User management implementation, which I created using a website called lucid.app. Visual-paradigm defines a class diagram as “a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (methods, and the relationships among objects” [2].

The class diagram exhibits the com.opule.dao package, which is comprised of a GenericDAO interface and the JpaDAO and UserDAO classes. An interface in java is an abstract used to specify the behaviour of a class [3]. In this scenario, the GenericDAO interface is implemented by the UserDAO class, which in turn is a subclass of jpaDAO. The GenericDAO interface defines operations (create, update, get, delete, listAll and count) that are common to the UserDAO class. UserDAO inherits common operations provided by its superclass jpaDAO and also implements other specific functionalities (such as check user login/find user by email). JpaDAO uses JPA to implement some persistent operations that are common to userDAO. The main Purpose of the userDAO class is for encapsulating database functionalities, which can be used by the userService class in the business/service layer. UserService, is used by the java servlets classes in the controller layer. The java servlet classes handle requests from the clients.

1. **Prototype**

So far, I’ve managed to install and configure the Apache tomcat server to host my web application. I chose this server because it’s a lightweight application and provides relatively quick load and redeploy times. Additionally, I’ve installed and created my database using MySQL command line client together with writing the scripts for my data tables (category, customer, order\_detail, product, product\_order, review and users). I’ve mapped these model classes to the tables in the database using Hibernate and JPA(java persistence API) application programming interfaces. Java Persistence API is a specification for relational data management in applications using Java SE and Java EE, and it defines a way for simplifying database programming. I’ve used the 2.1 JPA version for my application. Additionally, JPA defines the JPQL (java persistence query language), which is an object-oriented language that is used to make queries against entities stored in a relational database. I’ve made use of the JPQL language in my model classes.

3.1 Features Implemented

I’ve started with developing the website’s backend, which is the administration page. This includes features of the user management, which are as follows :

* Create User – this feature allows me to create a new user in the admin portal and requires details such as the email, full name and password. This information will be automatically saved in the database. However, the email for new user has to be unique in the database for the user creation to be successful.
* List User – This feature allows me to display all the users that are saved in the database, in the admin portal. It also includes a table with columns such as index, ID, full name and email.
* Edit User - This feature enables me to edit any user’s details such as the email, full name and password.
* Delete User – This feature allows me to delete a user completely from the database and the admin portal.

3.2 Features Tested

I’ve utilised the JUnit Framework for the testing of these features and they have all passed.

Create User

Graphical user interface, application

Description automatically generated

Edit/Update User

Graphical user interface, text, application, email

Description automatically generated

List Users

Graphical user interface, table

Description automatically generated

Delete User

A picture containing graphical user interface

Description automatically generated

* 1. Prototype diagrams

3.31 Home Page – this is a prototype diagram for the home page.

Diagram

Description automatically generated

3.32 Product detail Page.

Diagram

Description automatically generated

1. Survey of literature

This literature review aims to give me an understanding and basis of which to develop my project on.

Oracle has been my main source of information for the development of my web application. The website provides well-documented articles about the java technologies that are mainly used for web application development such as Java Servlet API, JavaServer Pages Technology, JDBC API and Java Persistence API [4].

Additionally, I acquired my understanding of the Hibernate object/relational mapping (ORM) framework from hibernate.org website, where they state that “Hibernate ORM enables developers to more easily write applications whose data outlives the application process and hibernate is concerned with data persistence as it applies to relational database(via JDBC)” [5].

The module that I undertook in second year (Software engineering) introduced me to the MVC architecture, however I used upGrad blog [6] to refresh my understanding and this has aided me in the development of the level features for the user management (create user, delete user, update user, list users)section in the admin portal.

Furthermore, the tomcat apache website aided me in the installation and configuration of the web host server. This website is very resourceful as it provides additional information about deploying an application using the tomcat manager and the client deployer package [7]

Additionally, I’ve been doing marketing research about how I can use social media to attract customers to my website, (attracting people on social media), how I can integrate social media platforms with my website. The book ‘eCommerce Marketing: how to drive traffic that buys your website’ is an extremely insightful book that has advanced my knowledge on vital marketing methods such as: Content Marketing, Social Media, Email Marketing.

Furthermore, I’ve been researching methods about increasing user/customer interactions. One of my findings include a live chatbox between the administrator and the customer, which enables the customer to ask any questions about the products.

Also, the salesforce website states that “41% of consumers state that live chat is their preferred channel of support” . Therefore, I believe that this would be a useful tool on the website. [8]

Having a spin the wheel game to capture user email addresses for marketing purposes is a multi-functional feature as it also increases the user interaction. I’m still researching the implementation of this feature.

According to lyfemakerting, Facebook is the #1 social media platform for marketing, then followed by Instagram. Lyfemarketing also mentions that ‘facebook has over 2 billion monthly active users’[9]. Instagram is a visual platform that enables a business to form a more personal connection with customers through the use of the app’s engagement features.

Shopify defines ‘e-commerce, also known as electronic commerce or internet commerce’ as ‘the buying and selling of goods/services using the internet, and the transfer of money and data to execute these transactions[10]. I’ve conducted research on a few e-commerce websites to explore ways in which I can improve my final product. Below is a table that compares my web application with other existing online stores.

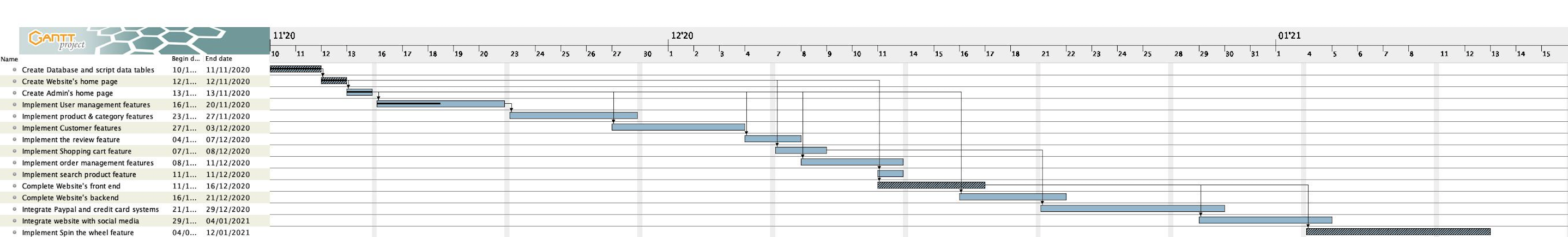
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PayPal & Credit Card Payment System | Social Media Integration | Product Wishlist | Product Categories | Spin the wheel/live chat/user->admin communication |
| Opule Accessories | YES | YES | YES | YES | YES |
| Bahamalashes[11] | YES | NO | NO | YES | NO |
| The Collection by NNT [12] | YES | NO | NO | YES | NO |
| Prya [13] | Credit Card | NO | NO | YES | NO |

As the table illustrates, the other e-commerce websites just provide a platform whereby users/customers can browse and purchase products. However, there isn’t much user interaction between the owner/admin and the customers.

1. Planning and Timescales

A Gantt chart is a planning tool for efficient project management, it offers features such as task creation as well as a graphical and global vision of the project. I’ve decided to use this tool/software for my project planning because it helps to manage the dependencies between tasks, which enables the project manager to establish the most efficient order in which the tasks should be executed.

Opule Accessories Gannt Chart



I created the above chart using a software called GanttProject. The chart shows the task names, start/end date on the left and the grouping and relationships/dependencies between the tasks on the right-hand side of the chart.

As I’ve already discussed in the prototype section of this document, I’ve completed a few of the tasks for this project ( such as create database and script data tables, create website’s home page and create admin’s home page) and this is illustrated by a straight horizontal line in the middle of the task bar.

The chart also exhibits the critical path, which is the sequence of tasks that could delay the project end date if any of the tasks are delayed themselves [REF]. This path is outlined by the shaded bars in the chart : 1. Create database and script data tables, 2. Create website’s homepage, 3. Complete website’s front end, 4. Implement spin the wheel feature.

My aim is to Complete the website front-end (customer/user portal) and the back-end (administrator portal) by the end of the first semester. I’ll be working on implementing the payment system and the spin the wheel feature and integrating social media in the second semester. My next task for this semester will be implementing the product and category features.

1. Bibliography

|  |  |
| --- | --- |
| [1] | “What is systems architecture ?,” *Polytechnique.fr*. [Online]. Available: https://www.lix.polytechnique.fr/~golden/systems\_architecture.html. [Accessed: 24-Nov-2020]. |
| [2] | “What is Class Diagram?,” *Visual-paradigm.com*. [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/. [Accessed: 24-Nov-2020]. |
| [3] | K. Rungta, “What is Interface in Java with Example,” *Guru99.com*, 01-Jan-2020. [Online]. Available: https://www.guru99.com/java-interface.html. [Accessed: 24-Nov-2020]. |
| [4] | “Java Technologies for Web Applications,” *Oracle.com*. [Online]. Available: https://www.oracle.com/technical-resources/articles/java/webapps.html. [Accessed: 24-Nov-2020]. |
| [5] | “Hibernate ORM,” *Hibernate.org*. [Online]. Available: https://hibernate.org/orm/. [Accessed: 24-Nov-2020]. |
| [6] | “A beginner’s guide to MVC architecture in java,” *Upgrad.com*, 09-Apr-2018. [Online]. Available: https://www.upgrad.com/blog/mvc-architecture-in-java. [Accessed: 24-Nov-2020]. |
| [7] | A. Crossley, “Apache Tomcat 8,” *Apache.org*. [Online]. Available: https://tomcat.apache.org/tomcat-8.5-doc/deployer-howto.html. [Accessed: 24-Nov-2020]. |
| [8] | “Live chat software - Thinking outside the text box,” *Salesforce.com*. [Online]. Available: https://www.salesforce.com/uk/learning-centre/customer-service/live-chat-software/. [Accessed: 24-Nov-2020]. |
| [9] | Sherman, “The 6 best social media platforms for your business in 2021,” *Lyfemarketing.com*, 10-Nov-2020. [Online]. Available: https://www.lyfemarketing.com/blog/best-social-media-platforms/. [Accessed: 24-Nov-2020]. |
| [10] | “Ecommerce,” *Shopify.com*. [Online]. Available: https://www.shopify.com/encyclopedia/what-is-ecommerce. [Accessed: 24-Nov-2020]. |
| [11] | “bahamalashes,” *Bahamalashes.co.uk*. [Online]. Available: https://bahamalashes.co.uk. [Accessed: 24-Nov-2020]. |
| [12] | “The Collection NNT,” *Thecollectionnnt.com*. [Online]. Available: https://thecollectionnnt.com. [Accessed: 24-Nov-2020]. |
| [13] | The, “PRYA,” *Prya.co.uk*. [Online]. Available: https://prya.co.uk. [Accessed: 24-Nov-2020]. |