A picture containing text, clipart

Description automatically generated

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (PERAK)

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS AND NETWORKING

24/4/2023

# **UCCD3243 SERVER-SIDE WEB APPLICATION DEVELOPMENT**

GROUP ASSIGNMENT

**Group 12**

**Online Order Management Monitoring System**

Goh Kwoon Boon – 1902262 – CN

Sam Jian Him 1903626 – CN

Ng Warren Cin Yen – 1903437 – CN

Yoong Yee Wei – 1903284 - CN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Actual Marks** | **Student 1**  **Goh Kwoon Boon** | **Student 2**  **Sam Jian Him** | **Student 3**  **Ng Warren Cin Yen** | **Student 4**  **Yoong Yee Wei** |
| **Report Section (Group)** | **20** |  |  |  |  |
| Analysis – Application Level – Functional charts | 15 |  |  |  |  |
| Analysis – Application Level – UML diagrams |
| Design – Functional Requirements |
| Strengths and Limitations | 5 |  |  |  |  |
| **Application Section (Individual)** | **60** | Order Module | Payment, Login, Profile Module | Product, Checkout module | Staff module |
| Application – Functional Requirements | 20 |  |  |  |  |
| Application – Design/Architecture | 20 |  |  |  |  |
| Application – Technologies | 20 |  |  |  |  |
| **Application Section (Group)** | **20** |  |  |  |  |
| Professional Outlook (Front End and Back End) | 5 |  |  |  |  |
| Overall Quantity and Quality | 15 |  |  |  |  |
| **Total** | **100** |  |  |  |  |

Checked by,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Dr Suthashini Subramaniam)

Chapter 1: Introduction

**1.1 Current Scenario**

Carmazon is an automotive selling company that sells and provides their customers with the best automotive such as cars, motorcycles, ships, planes, trucks, etc. Carmazon is based in Kampar, Perak and has multiple branches across the country to serve customers from different geographical locations.

Carmazon has a large call center with a team of sales representatives who receive requests from customers looking to purchase a car. The process is currently conducted conventionally, without the use of any computer software. When a customer calls in, the sales representative will record their information on paper, including their preferred car make and model, budget, and any other specific requirements. The sales representative will then manually search through the database of registered car dealers to find a match for the customer's needs. They will contact the dealer, negotiate a price, and arrange for a test drive and final sale.

**1.2 Problem Statement**

Due to the lack of implementation of an effective system, the manual recording of customer information can lead to errors and inconsistencies. Call centre agents may have to deal with a large volume of customer requests, which can make it difficult to record all the necessary details accurately. Additionally, the manual process can be time-consuming, which can lead to delays in connecting with potential customers. These errors could impact the customer experience and the reputation of the company, as customers may become frustrated with incorrect information.

Another problem that Carmazon is facing is the lack of a proper website for customers to browse and purchase vehicles. With the current trend of online shopping, customers expect a seamless and user-friendly experience when browsing for products, including vehicles. Without a proper website, customers may become frustrated with the limited options and information available to them, leading to missed sales opportunities. Moreover, not having a website can negatively impact the company's online presence and brand recognition. In today's digital age, having an online presence is crucial for any business to reach a wider audience and compete with other companies. Without a website, Carmazon may miss out on potential customers who are searching for automotive dealerships online.

**1.3 Objectives**

This project has the goal of addressing issues from both the customer and business perspectives. To achieve this, a web application will be developed that allows customers to interact with the business online. Additionally, an online order management monitoring system will be developed to help streamline business operations.

1. Develop a user-friendly website with a searchable database of vehicles.

* To increase customer engagement to the organization
* To allow customers to browse and view the product catalogue online.
* To allow customers to checkout online

1. Develop an online order management monitoring system for better management of business operations.

* To eliminate the use of paperwork
* To provide a database to store all of the information centrally.
* To allow better management of the records of the products, orders, customers, payments, and employees
* To provide CRUD functions for the records in the database
* To allow admin to keep track of the customers’ orders.

Chapter 2: Analysis

* 1. **Functional Hierarchy Chart**

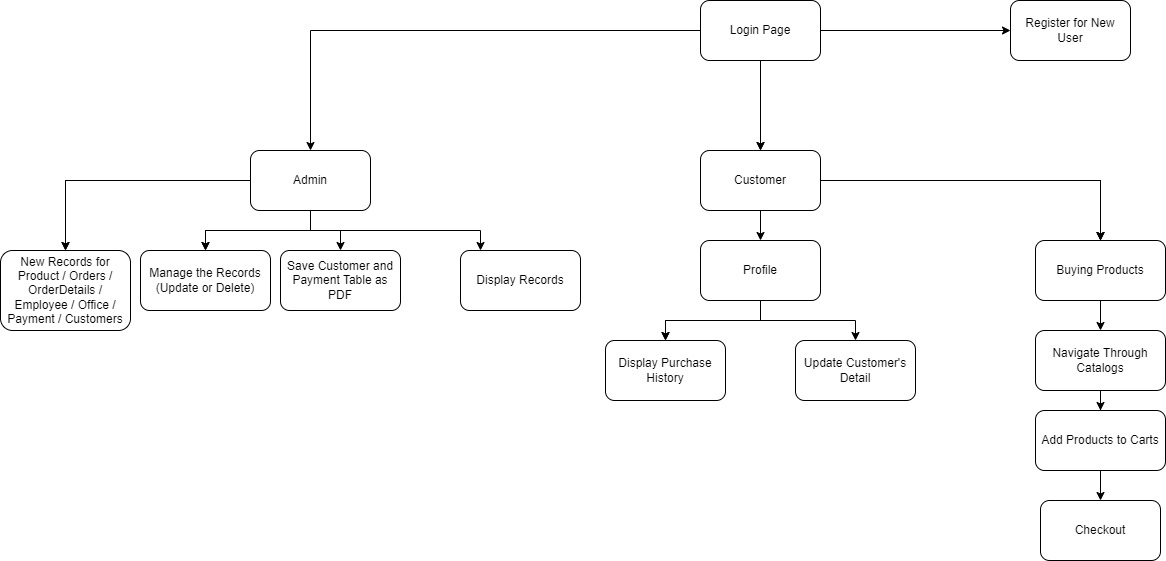
****

Figure 2.1 – Functional Hierarchy Chart

* 1. **UML Diagrams**
     1. **- Analysis Class Diagram**

**Table

Description automatically generated**

**2.2.2 - Use-Case Diagram**

**Diagram

Description automatically generated**

Figure 2.3 – Use Case Diagram

**2.2.3 – Design Class Diagram**

**Diagram, schematic

Description automatically generated**

Figure 2.4 – Design Class Diagram

**Table

Description automatically generated with low confidence**

Figure 2.5 – Design Class Diagram (II)

* 1. **- Other Diagram**

**2.3.1 – ERD Diagram**

**Timeline

Description automatically generated**

Figure 2.6 – ERD Diagram

* 1. **Application Architecture Diagram**

**Diagram

Description automatically generated**

Figure 2.7 – MVC Architecture

The architecture that the application is suing is the MVC (Model-View-Controller) or 3 tier architecture which represent into 3 categories whereby Model represent the data and business logic of the application; View represent the presentation layer which is user interface of applicationl Controller manages the user interactions and act as a middleman for View and Model. Such model allows the separation of concerns as it is modular and more maintainable. Since each component of the MVC is independent of each other, we are able to work on multiple modules in parallel without depending on one’s part which speeden up the development time and increase the productitivity on the development of the application.

JSP is used in the View layer as it enables the creation of dynamic pages accordingly to display interfaces and data that the user would want to see and also to separate the presentation from content. Servlet are used as Controller as it is able to handle user request and interact with the presentation layer and data layer. When the user sends a request, the servlet would receive the request and interact with other servlet or JSP to perform task and return a response dynamically on behalf of the user. EJB container is used as Model component manages the persistenece of the data and transaction when accessing the persistence which allows developers to focus on the business logic of the application without any worries.

* 1. **Functional Requirement**
     1. **– Login Module**

This module would allow users to key in their login credentials to determine whether they are user or admin as they would be redirected to different pages accordingly. The module would also perform authentication on the username and password that inputted by the users to ensure that users exist in the database. After logging in, the module would initialize a session for the respective user. Lastly, the module would allow the registration of new users by inputting their personal details and would link it to customer entity database.

* + 1. **– Checkout Module**

This module would allow the navigation of catalog by the customer through different product lines and allow them to store their products that they chosed in a cart to provide convenience to the user. The module would allow the customers to see the details of the product, quantity and total price in the cart. The module would allow customer to checkout with their product which would be recorded into several entity databases and would then be revealed to the customers in their own profile page.

* + 1. **– Profile Module**

This module allows the customers to view the transaction history based on their order and payment. Customer would be displayed with order details such as product,quantity and price of a specific order. This module would also allow customers to update their own details which would then be recorded and updated in the customer entity database.

* + 1. **– Admin Module**

This module provides an overview of the entire entity database, whereby admin could read and perform create, delete and update reocrds to products, order, employee, office, payment and customer database. This module would allow the download of database records in a pdf format for payment and customer database only.

**2.6 Operational Requirements**

There are 2 operational requirements that needed to be implemented before running our application which is the API library and changes to the standalone file.

We had implemented itextpdf library to the project to allow users to convert some of the databases table into pdf format and to be downloaded in their respective desktop for documentation and ease of display of data. The library is located inside the library of WEB.inf, developer would need to go to Project Structure under File, and add click on Libraries and add the libraries accordingly.

Text

Description automatically generated

Figure 2.8 – Project Structure

Other than that, developer should also add the line of codes below into their standalone file to correctly link our entity databases when opening this application project on their respective desktop.

Codes in Standalone File:

<datasource jndi-name="java:/classicmodels" pool-name="classicmodelsPool">

<connection-url>jdbc:postgresql://localhost:5432/classicmodels</connection-url>

<driver-class>org.postgresql.Driver</driver-class>

<driver>postgresql</driver>

<security>

<user-name>postgres</user-name>

<password>postgresql</password>

</security>

</datasource>

**2.7 Strengths and Limitations**

The approach used in this project follows the Model-View-Controller (MVC) design pattern, which allows for a separation of the application's logic, user interface, and data. This approach offers several advantages, such as the ability to maintain and update each component independently. Consider a scenario where a business owner wishes to access the order details for vehicles. In such cases, the browser dispatches a display request to the Controller (servlet), which then analyzes the request and invokes the Model (Session Bean) that holds the business logic to retrieve information from the database and store it in the bean. Subsequently, the Controller forwards the request to the View (JSP) to display the list of order details. This separation of concerns simplifies the development process, making it easier to manage and update the application over time.

Another strength of this applications is that it uses the Hibernate for implementing CRUD operations. Hibernate improved performance by support features like caching and lazy loading, which can help improve the performance of database operations. It also increased productivity, because of it provides a high-level object-oriented API that can simplify database access.

Furthermore, a Login page also added into the web application for user authentication. This feature serves to differentiate between admin and non-admin users. Admin users can access additional privileges such as monitoring and controlling order details, customer details, employee details, payment details, and other related features. On the other hand, non-admin users are limited to accessing the online store page only. By incorporating a Login page with role-based access control, it can enhance the security and usability of the web application, ensuring that users only have access to the features they need, and that sensitive data is kept secure.

One limitation of the application is that the robustness of the system as users cannot specify the quantity they wish to purchase directly before adding the item to the cart. They can only adjust the quantity in the cart, which may cause confusion or errors if they accidentally add too many or too few items.In addition, the application also has a limitation where the admin is unable to delete multiple records in a row. For instance, if the admin wants to delete 20 employee records, they have to delete them one by one manually, which can be time-consuming and tedious. This limitation could be addressed by implementing a feature that allows the admin to select multiple records and delete them at once.