**Unit 5.3 - Design Dominick George**

**System Overview Diagram**

Users Database

Stores User Details

Image Server

Images Database

Store Image Details

Products Database

Stores Product details

Order Management

Orders Database

Stores order details

Payments Database

Stores payment details

In the system overview diagram, I have created I have tried to show how I want the databases in my system to work together. The rectangles on the right side of the diagram represent the 5 databases my app will use to function. The initial database is the Users Database where it will store all the data associated with each customer of Wye Camping & Leisure. This will include data such as their user\_id, username, password, acc\_type and so on. This will be linked with other databases such as the payments database and the orders database. This is to avoid data redundancy within my app and make it so that the app runs efficiently. I will use the user\_id in the users database as a primary key so that it can link to the other databases as a foreign key.

The next database is the payments database, this database is important to the system as it stores all the data relating to the processing of transactions which is vital to the company’s success. I will use in this database a payments\_id as a primary key as I can link the database to other databases as a foreign key. This is necessary because other databases will need to have access to this data to complete other processes. Meaning to avoid data redundancy it will be more efficient to run the database as I have currently laid it out. It will also link with various databases such as the user and orders database. This is so that there payments processed through the app are linked with the appropriate user and order and is to make sure that each order processed has been paid for by the user that has placed the order. This database will have a lot of data that needs to be stored so that the program has the necessary data needed to function. Some of the data that needs to be stored on the database includes giving every separate transaction a payment\_id, order\_id, user\_id, pay\_method, card\_num, card\_expiry, CVV and name\_on\_card.

The next database is the images database. This database does not have any association with the other databases. This is because the database is just used to store the images that the app needs to display throughout the program. The data base will only need to store two data sets, image\_id and the actual image itself.

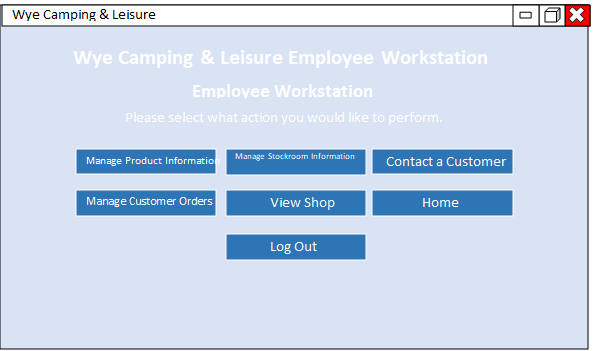
The next database is the orders database. This database will be important to the system as it stores all the information relating to the orders that the company receives. Meaning the company will need to view this database regularly as it needs to complete orders daily and satisfy customers. This database will be used with two other databases in order to function efficiently. These two databases that will be linked with this database are the payments and user databases. All three databases will be vital to the company’s operation as it will allow processes to be conducted as quickly as possible. The data the database will need to store to run is order\_id, user\_id, product\_id, deliv\_date, purch\_date, quantity. I am going to use the order\_id section of the database as the primary key of this database as it will allow me to eliminate data redundancy within my program and assortment of databases. This means the program will use the order\_id whenever it needs to access elements of the order database.

**Input and Output Screens**

Title

Label Widget

Label Widget

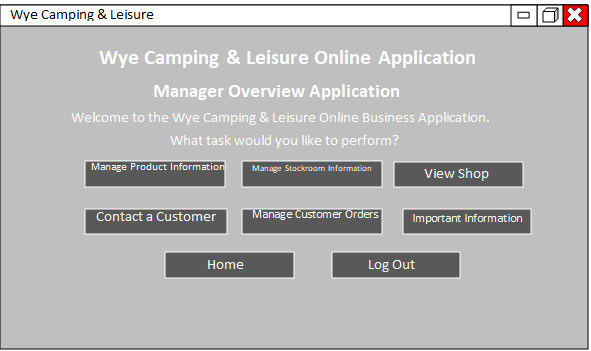


Button Widget

Button Widget

**Functions:**

* **Manage Product Information Button** – This button is to allow the user to manipulate product information within the products database. An example of what they can do in this part of the program is add or lower stock levels, delete old products, add new products, etc.
* **Manage stockroom information Button** – This button is to once pressed give the user the ability to see an overview of all the products on the database allowing the user to easily do stock counts and so on.
* **Contact a customer Button** – This button is to allow the user to access the user database and retrieve the contact information associated with the user and be able to contact them should they need to for whatever reason.
* **Manage customer orders Button** – This button is to give the user the ability to manipulate the orders database. Examples of what the user could do with this section of the program is add or delete orders, view orders, etc.
* **View shop Button** – This button will take the user to the shop page, for quick and easy access.
* **Log out Button** – This button will take the user back to the login screen and delete all the data in the entries.



Button Widget

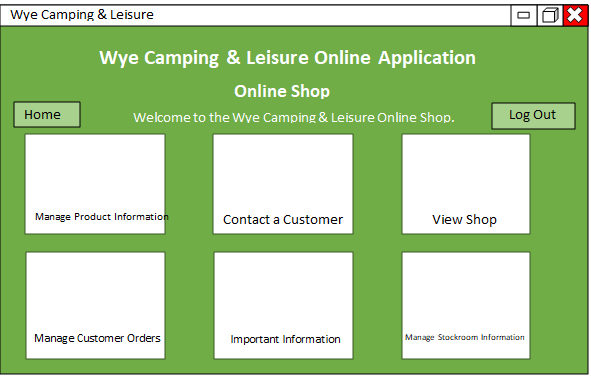
Button Widget

Title

Label Widget

**Button Functions:**

* **Manage Product Information Button** – This button is to allow the user to manipulate product information within the products database. An example of what they can do in this part of the program is add or lower stock levels, delete old products, add new products, etc.
* **Manage stockroom information Button** – This button is to once pressed give the user the ability to see an overview of all the products on the database allowing the user to easily do stock counts and so on.
* **Contact a customer Button** – This button is to allow the user to access the user database and retrieve the contact information associated with the user and be able to contact them should they need to for whatever reason.
* **Manage customer orders Button** – This button is to give the user the ability to manipulate the orders database. Examples of what the user could do with this section of the program is add or delete orders, view orders, etc.
* **View shop Button** – This button will take the user to the shop page, for quick and easy access.
* **Important Information Button** – This button will give the user access to important information that the business has. For example, supplier contact details, account passwords, other contact details, etc.
* **Log out Button** – This button will take the user back to the login screen and delete all the data in the entries.



Title

Image

Button Widget

Button Widget

Button Widget

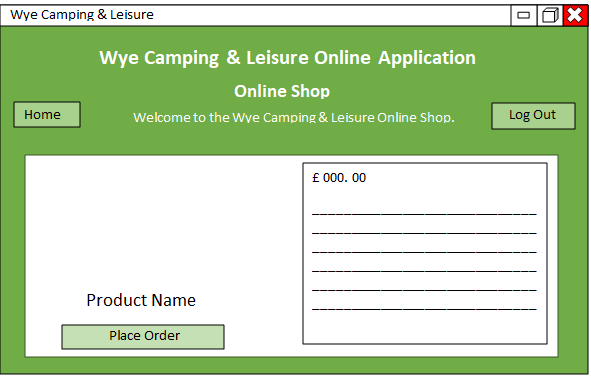
Label Widget

**Function:**

* **Home Button** – This button will take the user back to the top of this page.
* **Logout Button** – This button will take the user back to the login page and will delete all the data in the entries.
* **Image** – This type of image label will display what the item looks like on the shop page. The data will be retrieved from the images database and will be displayed on the white rectangles.
* **Name buttons** – The button underneath the images will show text and will be will be the name of the product. This data will be retrieved from the products database underneath the product\_name data section.

Title

Label Widget



Image

Label Widget

Button Widget

Button Widget

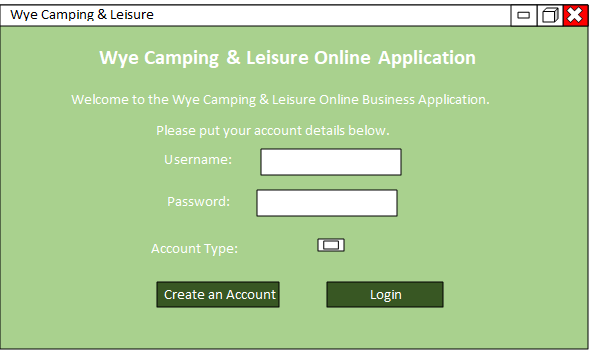
Button Widget

**Functions:**

* **Home Button –** This button will take the user back to the top of the shop page with the list of all the different items.
* **Logout Button** – This button will take the user back to the login page and delete all the data in the entries.
* **Image –** This widget feature will display the corresponding picture that is associated with the product that the user has clicked on. This will retrieve the image from the image from the images database.
* **Place order Button –** This button will add and extra order into the orders database. This will include filling out all the data necessary in that database.

Label Widget

Title



Drop-down Menu Widget

Data Entry Widget

Button Widget

Label Widget

**Functions:**

* **Username Entry** – This will take the input the user provided and compare it with the users database to check whether the user is on the database, if also correct with the password and account type then it will take the user to the appropriate page according to which account type the put in. If incorrect an error message will be displayed.
* **Password Entry** – This will take the users input and compare it with the password the user has on the database. If also correct with the username and account type, it will grant the user access to the appropriate section of the program.
* **Account Type** – This is a drop-down menu that when pressed by the user will give the user three account options. A customer account, employee account or a manager account. Each account type will take the user to a different part of the program and will grant the user different privileges. The users input must match what is already on the database otherwise the user will not be logged in to the correct account type.
* **Create an account Button –** This button will take the user to another screen where the user can create an account and add it to the database.
* **Login** – This button will gather the data the user typed into the different entries and compare it with the data in the databases, if the data matches, it will let the user into the corresponding page and if the data doesn’t match, an error message will be displayed**.**

This screen will be the initial one the user will see when the application is launched. This screen is my final design that I want my project to look like when it is launched. It is visually appealing and includes all the relevant information that I need to display. Additionally, the screen also has the necessary input fields and buttons that I want implemented into my project. I did not want to make the screen too complicated as I think that it would make the screen look confusing to the user and may make the program less effective in generating sales. I managed to incorporate a useful into the interface with the account type section. The button that the section uses, once clicked, displays a drop-down menu which tells the user which account type they want to log into. The same is also displayed on the create an account page.

**Data Structures**

Foreign Key

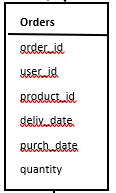
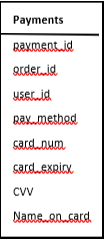
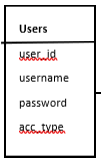
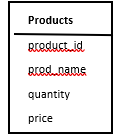
Primary Key

Primary Key

Primary Key

Primary Key

Primary Key

****

****

Foreign Key

Foreign Key

Foreign Key

**ER Diagrams**

**Images**

image\_id

**Payments**

payment\_id

order\_id

user\_id

pay\_method

card\_num

card\_expiry

CVV

Name\_on\_card

**Users**

user\_id

username

password

acc\_type

**Orders**

order\_id

user\_id

product\_id

deliv\_date

purch\_date

quantity

**Products**

product\_id

prod\_name

quantity

price

**Algorithms (Pseudocode)**

user opens program

display user entries for username, password, account type

display create an account and login button

if user fills in login data

user presses login button

data is compared with data in the login database

if user enters customer account

display shop webpage showing an array of items

display buttons on webpage saying basket, home, log out

if user enters employee account

display employee webpage

display buttons with labels manage product info, manage stockroom info, view

shop, contact a customer, manage customer orders, home, logout

if user enters manager account

display manager workstation webpage

display buttons with labels manage product info, manage stockroom info, view

shop, contact a customer, manage customer orders, home, logout, important information

if user presses the create an account button

open create an account screen

display user entries for username, email, password, account type

if user wants a manager or employee account

ask for code before creating account

if user puts in correct code

create account

else

display error message