**Project 27: Lakeside Espresso Café Online Management System**

**About the Client**

Lakeside Espresso Café is a specialty coffee roaster and retailer, dedicated to sourcing, roasting and sharing exceptional coffee in a sustainable, respectful and responsible way. We want to establish our first coffee shop in Melbourne CBD starting January 2025 and we need our own POS system to manage our shop.

Lakeside Espresso Café is a specialty coffee roaster and retailer, born and based in Melbourne. We are dedicated to sourcing, roasting and making beautiful coffee, and we’re committed to doing so in a responsible, sustainable and respectful way.

Lakeside Espresso Café was founded in 2009 by Fleur Studd and Jason Scheltus. It's hard to fathom now, but at that time, finding fresh, in-season, traceable, high quality coffee in Melbourne (or Australia) was practically impossible. We wanted to help address that and, in doing so, ignite positive change in the industry by redefining what coffee was and could be, and to build a market for, and appreciation of, high quality specialty coffee.

We built our first shop and roastery in the wonderful, bustling Broadmeadows. We fell in love with this location for many reasons, the most important being that it allows us to engage with a community of shoppers who are seeking out quality produce, and who care about seasonality and provenance.

We realised early on that we wanted to deliver Melbourne's best coffee experience and to do this, we needed to put all our energy into sourcing, roasting and sharing the very best coffee we could find. Our goal was to make these coffees accessible and exciting, easy to understand and appreciate, and simple to brew and enjoy.

**Project Brief & Business Problem Specifications:**

## This project should provide an automated platform for coffee shop, so that Lakeside Espresso Café could handle their daily sales transactions in a more efficient and productive manner. The application should provide the user with the capability to save, manage, and retrieve the sales transactions associated with the specified business. The management of the Lakeside Espresso Café is the only ones who have access to this POS for their shop. It is necessary for the management to log in to the system using their current valid system credentials in order for them to be granted access to the features and functions that are available. The Administrator, the Staff, and the Cashier are the three distinct sorts of user roles that are able to log in and use this system. The Admin user should have the ability to access and administer all of the features and operations of the system, including the management of users and system information. This capability should only available to the Admin user. The Cashier users should only access the sales transactions that they themselves have generated, whereas the Staff users should have restricted capabilities. Additionally, the project should provide a daily sales report as well as a printable receipt.

## The user of the system must provide his or her user credential in order to access and use the features and functionalities of this straightforward project because it will possess a secure login feature. This means that the user can only access and use the features and functionalities of this straightforward project. In addition to storing the sales transactions, the system should also have the capability of storing a list of the many types of water jars and containers, as well as their prices.

**System Modules**

**Administrator**

* **Home Page**
  + Display the summary and images.
* **Category Management**
  + Add New Category
  + List All Categories
  + View Category Details
  + Delete Category
* **Product Management**
  + Add New Product
  + List All Products
  + View Product Details
  + Update Product Details
  + Delete Product
* **Sale Management**
  + Add New Sale
  + List All Sales
  + View Sale Details
  + Update Sale Details
  + Print Sale Receipt
  + Delete Sale
* **Report**
  + Generate Printable Daily Sales Report
  + Filter Daily Sales Report by Date
  + Filter Daily Sales Report by User
* **User Management**
  + Add New User
  + List All Users
  + View User Details
  + Edit User Details
  + Delete User Details
* **Update System Information**
* **Update Account Details/Credentials**
* **Login and Logout**

**Staff**

* **Home Page**
  + Display the summary and images.
  + After successfully checking in to the **Coffee Shop POS System**, users will automatically be led to this page after being brought there by the system.
* **Category Management**
  + Add New Category
  + List All Categories
  + View Category Details
  + Delete Category
* **Product Management**
  + Add New Product
  + List All Products
  + View Product Details
  + Update Product Details
  + Delete Product
* **Sale Management**
  + Add New Sale
  + List All Sales
  + View Sale Details
  + Update Sale Details
  + Print Sale Receipt
  + Delete Sale
  + This feature is used to have the POS, which can then be used to calculate the daily and overall sales of the Coffee Shop.
* **Report**
  + Generate Printable Daily Sales Report
  + Filter Daily Sales Report by Date
  + Filter Daily Sales Report by User
* **Update Account Details/Credentials**
* **Login and Logout**
  + Users of the system will be directed to this page to provide their credentials before being granted access to the systemâ€™s data and functionality.

**Cashier**

* **Home Page**
  + Display the summary and images.
* **Sale Management**
  + Add New Sale
  + List All Sales
  + View Sale Details
  + Update Sale Details
  + Print Sale Receipt
  + Delete Sale
* **Report**
  + Generate Printable Daily Sales Report
  + Filter Daily Sales Report by Date
* **Update Account Details/Credentials**
* **Login and Logout**

**User Modules (User Frontend):**

Developers need to research and discuss with the client to finalise the modules and requirements.

**Input Data and Validation Requirements**

- All the fields such as Car, Payment, Loan should validated and should not take invalid values

* + Each form for Car, Customer, Booking should not accept blank value fields
  + Avoiding errors in data
  + Controlling amount of input

## - Integration of all the modules/forms in the system.

* + Preparation of the test cases.
  + Preparation of the possible test data with all the validation checks.
  + Black-box/White-box testing.

# Recording of all the reproduced errors.

* + Modifications should be done for the errors found during testing.
  + Prepare the test result scripts after rectification of the errors
  + Functionality of the entire modules/forms.

# Validations for user input.

* + Checking of the Coding standards to be maintained during coding.
  + Testing the modules with all the possible test data.
  + Testing of the functionality involving all type of calculations.
  + Commenting standard in the source files.
* Easy & fast retrieval of information.
* Well-designed reports.
* Decrease the load of the person involve in existing manual system.
* Access of any information individually.
* Work become very speedy.
* Easy to update information

**UI Design**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

**UI Design Requirements**

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error.

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

* + Security of data.
  + Ensure data accuracy's
  + Proper control of the higher officials.
  + Minimize manual data entry.
  + Minimum time needed for the various processing.
* Greater efficiency.
* Better service.
  + User friendliness and interactive.
  + Minimum time required.

**Functional Requirements**

User can add/change a coffee:

- Type

- Size: s/m/l

- Number of cups

- Customize the flavor: sweetness and milk

User can choose a shipping method:

- Pick up/deliver

- Time

- Location

User can place order online and get confirmation:

- Order and cost preview

- Payment

- Confirmation: success/failure

User should have an easy access to the coffee he/she often purchases or recently purchases

User has the chance to cancel order

User can maintain their account and access to order history

User can reuse shipping address and card information

User should be notified when delivery is unavailable:

- Address is too far away

- Not reach minimum cost

User should be notified when pick-up service is unavailable:

- Out of business hour

User should be notified the state of the order

User can complain/remind if the delivery doesn’t come on time

User reads the same menu with that in the store, and should be notified when some drink is sold out

User should be informed with the latest promotions and new/seasonal products

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear for the stakeholders. Generally, functional requirements describe system behavior under specific conditions. The developers of this system must enhance the performance and efficiency of the system by adding 15 to 20 more functional requirements. Students need to do their own research to find how they can improve the system and which FRs need to added. The group must need a prior approval from the stakeholders/project supervisor before finalizing these Functional Requirements. These enhanced FRs must be reflected separately in Final SRS Report after the approval.

**Non-functional Requirements**

Apart from other NFRs following NFRs must be implemented in the system:

Less clicks and keyboard interactions

Consistent look between website/app and the style in store

Account/phone number verification

**Hardware Requirement: Should be recommended by the developers.**

**Software Requirement: Should be recommended by the developers.**