

System Specifications Document

Troy Renaud

Table of Contents

Problem Summary:	2
Motivation and Research:	9
Specifications of Program Function:	11
Specification of Interface (Including User Interface):	13
Specification of Help:	28
Specification of Permanent Data Storage:	30
Hardware and Software Requirements (including additional hardware):	33

Problem Summary:

1.1 Problem Summary		3
A brief description of the project including the purpose of the project, summary of functions and description of target user group(s)		
[3]	All items well-described and every item completely specified.	
[2]	Most of the items well described but at least one is not completely specified.	
[1]	Only one well described item or part of each is not fully complete.	
[0]	No summary or completely inadequate.	

Description:

Restaurants often struggle with inefficient inventory tracking, resulting in either stock shortages or excesses that can disrupt service quality. Additionally, the ordering process can be cumbersome without transparent communication. That is why order management is crucial for maintaining an enjoyable customer experience.

This project is focused on developing a Meal and Snack Ordering System. The system aims to streamline the ordering process, allowing staff to easily place orders for various meals and snacks through a user-friendly interface. Staff members can log into the system, and when a new customer visits, they can create a customer profile by securely registering essential details such as the customer's first name, surname, and cell phone number or email address. Once registered, the ordering process becomes straightforward, with staff selecting from a variety of meal options and snacks. A search function will allow staff to find specific products quickly, as well as search for previous customers (if the customer is not new to the Wyvern Club) who are placing orders. The system also allows customization, including the selection of bread, patties, and the ability to add or remove toppings by entering additional notes.

After completing a customer's order, staff will see a detailed summary on the right side of the same interface, displaying the total price and all added items. All payments are required at the time of ordering or upon collection.

On the kitchen side, the system provides a display screen listing all current orders. Kitchen staff can navigate through the orders using up and down arrows (to accommodate messy hands) and press enter twice to mark an order as selected. Once the kitchen completes an order, it is marked as "ready" for collection. Additionally, a custom WhatsApp message is automatically sent to the customer, notifying them that their order is ready. The system also includes a public collections screen outside the Wyvern Club, allowing customers to see when their order is ready for pickup.

On the administration side, staff can efficiently delete or update meal and snack options using barcode or QR code scanning, streamlining the product management process. Admins can also modify customer and user information as well as view sales records of past orders, reducing manual entry and minimising errors.

The main objective of this project is to provide an effective, user-friendly system that enhances order management. The system can easily be adapted for use in other restaurants or fast-food outlets that focus on meal and snack services.

Purpose:

The primary goal of this project is to develop an effective and user-friendly Meal and Snack Ordering System. This system aims to enhance the efficiency of order management, streamline the registration of customers, and facilitate the management of stock within a restaurant. By providing a robust interface for staff to manage orders and customer profiles, the system will ensure a seamless experience for guests while maintaining organized records of transactions and inventory.

Display:

Upon entering the program, users will be greeted with the **Login Page (LoginPageUI)**. This page allows users to enter their username and password to proceed into the system. Depending on the credentials entered, users will be directed to either the **Admin Dashboard (AdminDashboardUI)** or the **Customer Selection UI (CustomerSelectionUI)**.

From the **Customer Selection UI**, users can navigate to several options:

- **New Customer UI (NewCustomerUI)** for adding new customers.
- **Previous Customer UI (PreviousCustomerUI)** to access details of returning customers.
- **Collections UI (CollectionsUI)** for managing tab collections.
- **Kitchen Display UI (KitchenDisplayUI)** for viewing current kitchen orders.

In the **Admin Dashboard**, users can access:

- **Admin Product Management UI (AdminProductManagementUI)** for managing products.
- **Admin Customer Management UI (AdminCustomerManagementUI)** for managing customer information.
- **Admin Sales Records UI (AdminSalesRecordsUI)** for viewing sales records.

- **Admin User Management UI (AdminUserManagementUI)** for managing user accounts.

Each UI includes a **Help Button** that provides navigation and assistance specific to that interface.

Summary of Functions:

- **LoginPageUI:**
 - **Username TextField:** Enter username.
 - **Password TextField:** Enter password.
 - **Show Password Checkbox:** Toggle visibility of the password.
 - **Help Button:** Displays a help dialog.
 - **Login Button:** Validates credentials and redirects to the Admin/User dashboard.
- **CustomerSelectionUI:**
 - **New Customer Button:** Opens NewCustomerUI to add a new customer.
 - **Previous Customer Button:** Opens PreviousCustomerUI to select previous customers.
 - **Collections Button:** Opens CollectionsUI to manage customer tabs.
 - **Kitchen Display Button:** Opens KitchenDisplayUI for order management.
 - **Help Button:** Shows help dialog.
 - **Back Button:** Returns to LoginPageUI.
- **NewCustomerUI:**
 - **First Name Input Field:** Enter customer's first name.
 - **Surname Input Field:** Enter customer's surname.
 - **Cell phone Number Input Field:** Enter customer's cellphone number.

- **Email Input Field:** Enter customer's email.
- **Back Button:** Returns to CustomerSelectionUI.
- **Confirm Button:** Validates input and saves the customer to the database.
- **Help Button:** Displays usage instructions.

- **PreviousCustomerUI:**
 - **Customer Table:** Displays a list of previous customers.
 - **Help Button:** Opens help dialog.
 - **Back Button:** Returns to CustomerSelectionUI.
 - **Confirm Button:** Proceeds to SelectProductsForTabUI for selected customer.

- **SelectProductsForTabUI:**
 - **Product Buttons:** Add selected products to the tab.
 - **Submit Products Button:** Saves selected products to the tab.
 - **Remove Product Button:** Removes selected product from the tab.
 - **Close Tab Button:** Closes the current tab.
 - **Back Button:** Returns to CustomerSelectionUI.
 - **Help Button:** Provides help information.
 - **Paid Before Collection Checkbox:** Indicates payment status.

- **KitchenDisplayUI:**
 - **Order Table:** Displays current kitchen orders with details.
 - **Enter Key:** Marks order as complete and sends notification to customer.
 - **Back Button:** Returns to CustomerSelectionUI.
 - **Help Button:** Displays usage instructions.

- **CollectionsUI:**

- **Customer Orders Table:** Displays orders for collection with statuses.
- **Confirm Button:** Updates 'Paid' and 'Collected' statuses in the database.
- **Back Button:** Saves changes and returns to CustomerSelectionUI.
- **Help Button:** Displays help dialog.
- **Checkboxes:** Toggle 'Paid' and 'Collected' statuses.

- **AdminDashboardUI:**
 - **Manage Products Button:** Opens AdminProductManagementUI.
 - **Manage Customers Button:** Opens AdminCustomerManagementUI.
 - **Sales Records Button:** Opens AdminSalesRecordsUI.
 - **Manage Users Button:** Opens AdminUserManagementUI.
 - **Logout Button:** Logs out and returns to LoginPageUI.
 - **Back Button:** Navigates to the previous screen.
 - **Help Button:** Displays help instructions.

- **AdminProductManagementUI:**
 - **Add Product Button:** Opens AdminNewProductUI to add new products.
 - **Edit Button:** Updates selected product details.
 - **Delete Button:** Removes the selected product after confirmation.
 - **Back Button:** Returns to AdminDashboardUI.
 - **Help Button:** Displays usage instructions.

- **AdminNewProductUI:**
 - **Product Name Input Field:** Enter product name.
 - **Category Input Field:** Enter product category.
 - **Barcode Input Field:** Enter product barcode.
 - **Price Input Field:** Enter product price.
 - **Confirm Button:** Validates and inserts product into the database.

- **Back Button:** Returns to AdminProductManagementUI.
- **Help Button:** Displays usage instructions.

- **AdminCustomerManagementUI:**
 - **Customer Table:** Displays a list of customers.
 - **Add Customer Button:** Opens AdminNewCustomerUI to add a new customer.
 - **Edit Button:** Updates selected customer's details.
 - **Delete Button:** Removes selected customer.
 - **Back Button:** Returns to AdminDashboardUI.
 - **Help Button:** Displays usage instructions.

- **AdminNewCustomerUI:**
 - **Name Input Field:** Enter customer's name.
 - **Surname Input Field:** Enter customer's surname.
 - **Cellphone Number Input Field:** Enter customer's cellphone number.
 - **Email Input Field:** Enter customer's email.
 - **Confirm Button:** Validates input and creates a new customer.
 - **Back Button:** Returns to AdminCustomerManagementUI.
 - **Help Button:** Displays usage instructions.

- **AdminSalesRecordsUI:**
 - **Sales Records Table:** Displays sales records of collected tabs.
 - **Back Button:** Returns to AdminDashboardUI.
 - **Help Button:** Displays usage instructions.
 - **Checkboxes:** Toggle 'Paid' and 'Collected' statuses.

- **AdminUserManagementUI:**

- **User Table:** Displays a list of users.
 - **Add User Button:** Opens AdminNewUserUI to add a new user.
 - **Edit Button:** Updates selected user's information.
 - **Delete Button:** Deletes selected user after confirmation.
 - **Back Button:** Returns to AdminDashboardUI.
 - **Help Button:** Displays usage instructions.
-
- **AdminNewUserUI:**
 - **Username Input Field:** Enter new user's username.
 - **Password Input Field:** Enter new user's password.
 - **Role Input Field:** Enter new user's role.
 - **Confirm Button:** Creates a new user and saves to the database.
 - **Back Button:** Returns to AdminUserManagementUI.
 - **Help Button:** Displays usage instructions.

Motivation and Research:

1.2 Motivation and Research A brief discussion of similar projects as a result of research. Research is correctly referenced and cited. A motivation is included for the project indicating how this project differs from existing projects. This section should NOT be longer than a page.		3
[3]	Similar projects are described, cited and referenced. A motivation is supplied to include an explanation of how the project will differ from previous projects.	
[1-2]	Similar projects are described, cited and referenced. A motivation is supplied to include an explanation of how the project will differ from existing projects. Deduct one mark for each incomplete aspect of research, citing, referencing and insufficient motivation.	
[0]	No motivation or research provided.	

Motivation and Research:

In developing a Stock Management System, my research focused on identifying existing platforms that could inform and enhance various aspects of the project. The motivation behind this initiative stems from the unique operational needs of a restaurant, which serves as a social hub for staff, alumni, and friends, where efficient ordering, registration of customers and inventory management is crucial to ensure smooth service and customer satisfaction.

Several relevant projects were examined during my research:

1. **StockManagement.com:** This platform offers comprehensive features for managing stock inventory across various industries, including food and beverage. However, it lacks specific functionalities such as integrating barcode scanning and a tab-based ordering system, both of which are essential for the Wyvern Club's operations.
2. **RestaurantOwner.com:** While this website provides valuable resources and articles on restaurant management, including inventory control and cost management techniques, it does not offer a dedicated software solution. Instead, it delivers insights and strategies that can inform the development of our Stock Management System.
3. **Stock and Inventory Simple App:** This app features a distinguished layout that effectively organises stock and inventory management. Its user-friendly interface can serve as a model for designing our system, ensuring that it runs efficiently and meets user expectations.

How This Project Differs

The primary motivation behind this project is to address the unique challenges faced by restaurants, creating a solution specifically designed for its environment. Unlike existing platforms, this Stock Management System will focus on features tailored to restaurants and bars, such as:

- **Barcode Scanning:** This feature will streamline inventory management, enabling quick updates and accurate stock levels.
- **Tab-Focused Ordering:** By implementing a tab-based system, the project aims to enhance customer convenience, allowing seamless tracking of orders and payments.

This project intends to deliver a user-friendly approach to stock management, integrating essential functions into a cohesive system that simplifies the operational workflow.

References

- StockManagement.com (n.d.). Stock Management Software - Inventory Management System. [online] Available at: <https://www.stockmanagement.com/>.
- RestaurantOwner.com (n.d.). Restaurant Management Resources for Owners. [online] Available at: <https://www.restaurantowner.com/>.
- Chester Software (Xaltos Technologies Ltd) (2016) *Stock and Inventory Simple* (2.1.43r4) [Mobile App]. Available at: Google Play (Downloaded: 25 January 2024).

Specifications of Program Function:

1.3 Specifications of Program Function		3
[3]	Function list is complete and detailed. The project has sufficient complexity (see taxonomy 2.6.2).	
[2]	The function list is a substantial list of appropriate outcomes but is insufficient in complexity or the functions list is not complete.	
[1]	The function list is four or less points. The project is too simplistic.	
[0]	No Functions listed.	

The program will be able to do the following:

1. User Authentication:

- Allow users to log in via the **LoginPageUI**, which includes fields for entering a username and password, along with options to show the password and access help.

2. Customer Management:

- Provide a **CustomerSelectionUI** for authenticated users to manage customers, including:
 - **Add New Customer:** Open the **NewCustomerUI** to enter and save new customer details.
 - **View Previous Customers:** Open **PreviousCustomerUI** to display a list of past customers, allowing selection for further actions.
 - **Manage Collections:** Open the **CollectionsUI** to handle customer tabs and payment statuses.
 - **Access Kitchen Display:** Navigate to **KitchenDisplayUI** to manage and view orders.

3. Product Management:

- Admin users can access the **AdminDashboardUI** to manage products, including:
 - **Add New Product:** Open the **AdminNewProductUI** to enter product details such as name, category, barcode, and price.
 - **Edit Product Details:** Update existing products through **AdminProductManagementUI**.
 - **Delete Products:** Remove products from inventory as needed.

4. **Order Management:**

- Staff can utilise **SelectProductsForTabUI** to:
 - Select products for a customer's tab from categorized product lists.
 - Customise orders by selecting specific ingredients and adding special notes.
 - Submit the selected products and manage the tab (including closing it).

5. **Order Notifications:**

- Kitchen staff will use the **KitchenDisplayUI** to:
 - View current orders, marking them as complete when finished.
 - Automatically send notifications to customers via WhatsApp upon order completion.

6. **Sales and Records Management:**

- Admin users will have access to:
 - **AdminSalesRecordsUI** to review and manage sales records, including viewing tab statuses (Paid and Collected).
 - **AdminUserManagementUI** to manage user accounts, including adding, editing, and deleting users.

7. **Help and Navigation:**

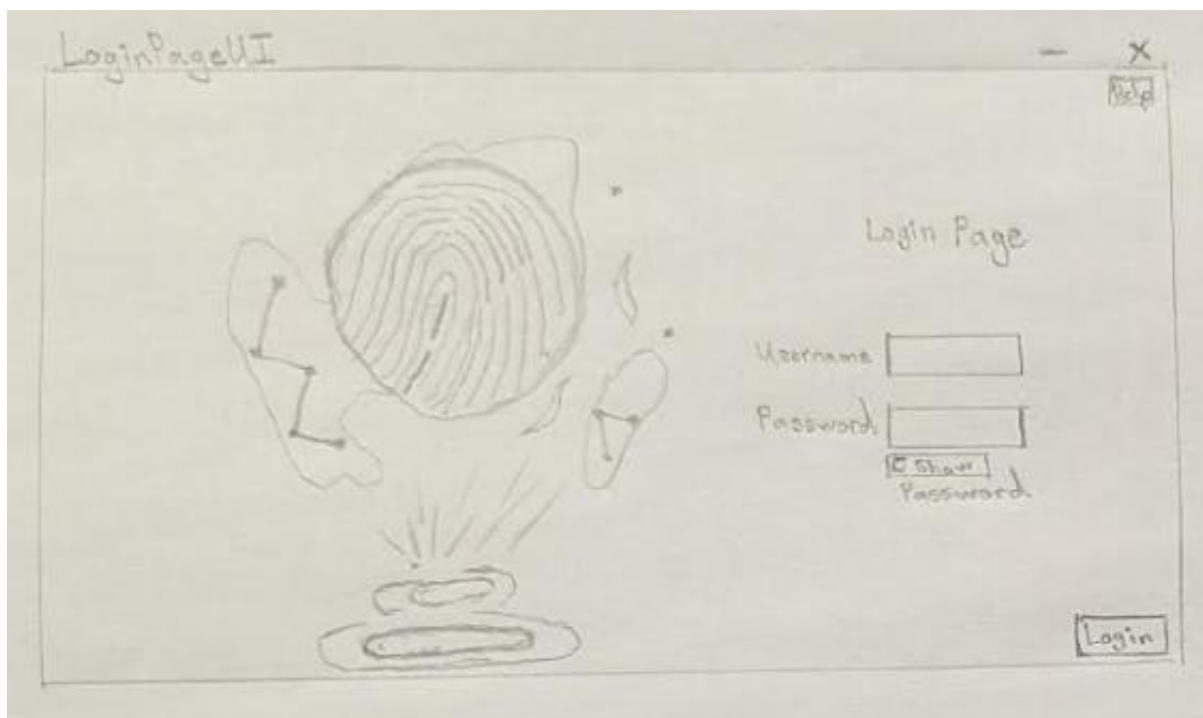
- All UIs include a **Help Button** for user guidance, ensuring smooth navigation through the system.
- Back buttons in each UI will enable users to return to previous screens without losing their progress.

8. **Customer Registration:**

- Support the manual addition of new customers, capturing essential details for future reference and ensuring quick access to customer profiles.

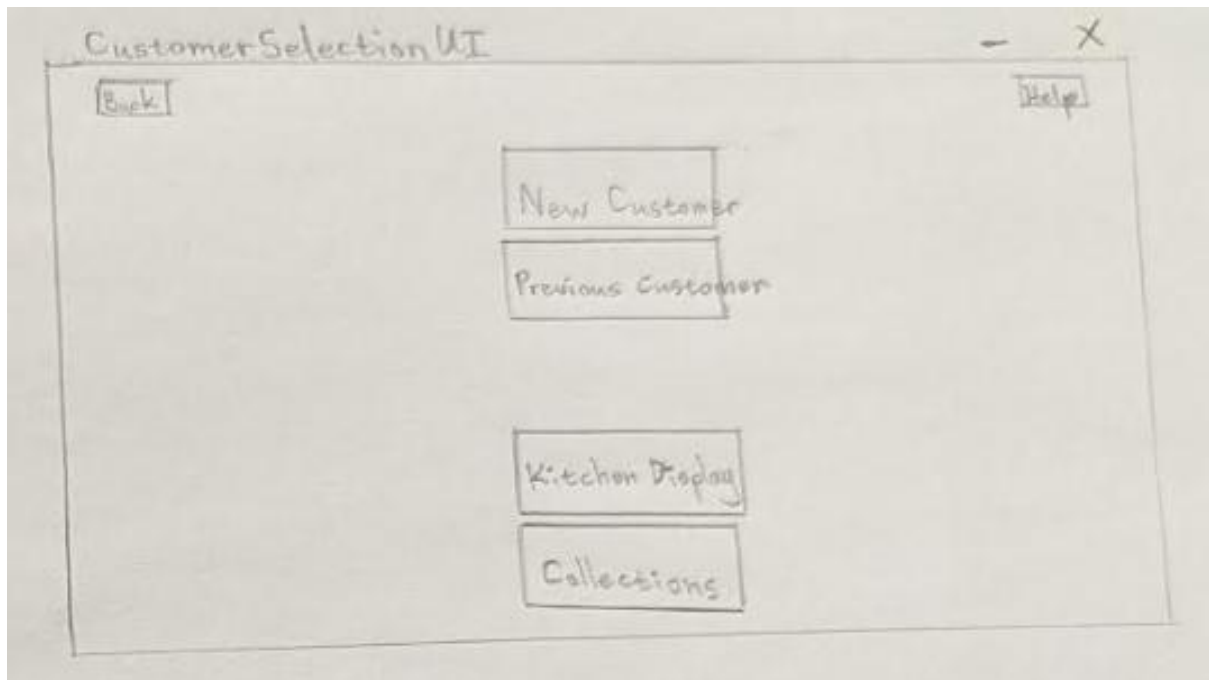
Specification of Interface (Including User Interface):

1.4 Specification of Interface (including User Interface) (See Appendix G – 10.2.5, 10.4.12, 11.4.12)		2
Specify the user interface and if relevant, the input/output from external hardware such as sensors.		
[2]	Interface completely specified.	
[1]	One or two items are inadequately specified.	
[0]	User Interface not specified or incorrectly specified.	



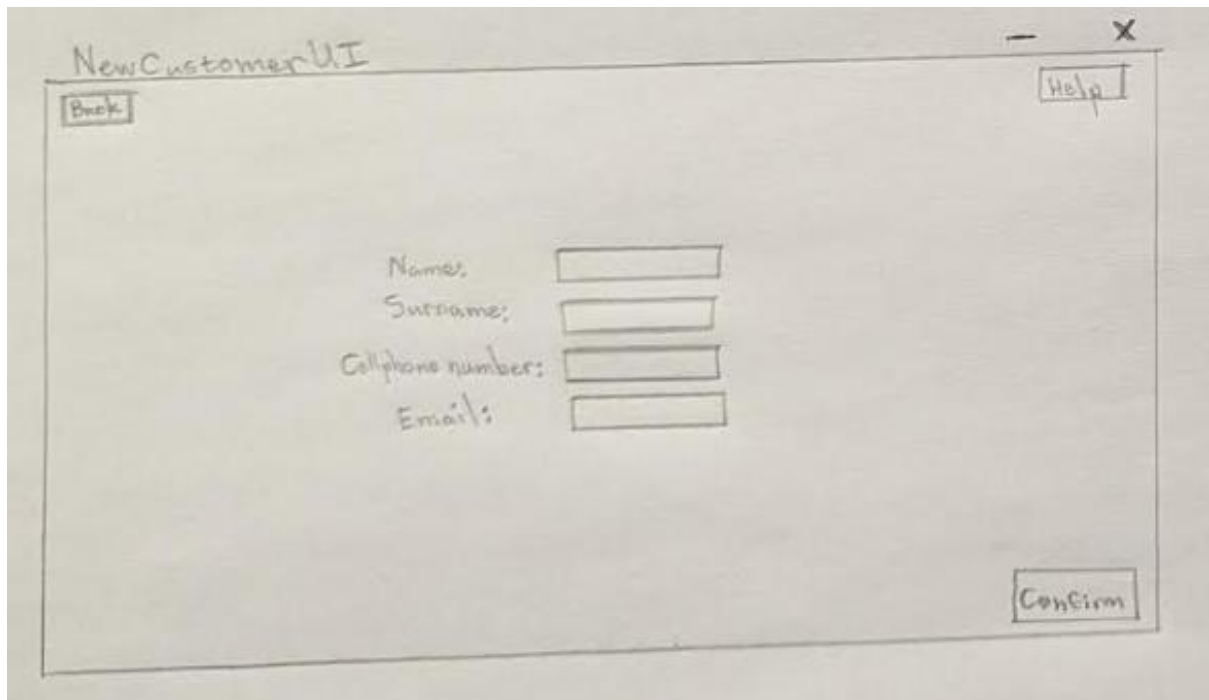
LoginPageUI:

- **"Show Password" Checkbox** pressed -> Toggles visibility of the entered password.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.
- **"Login" Button** pressed -> Validates user credentials and redirects to the Admin or User dashboard if successful; otherwise, displays an error message.



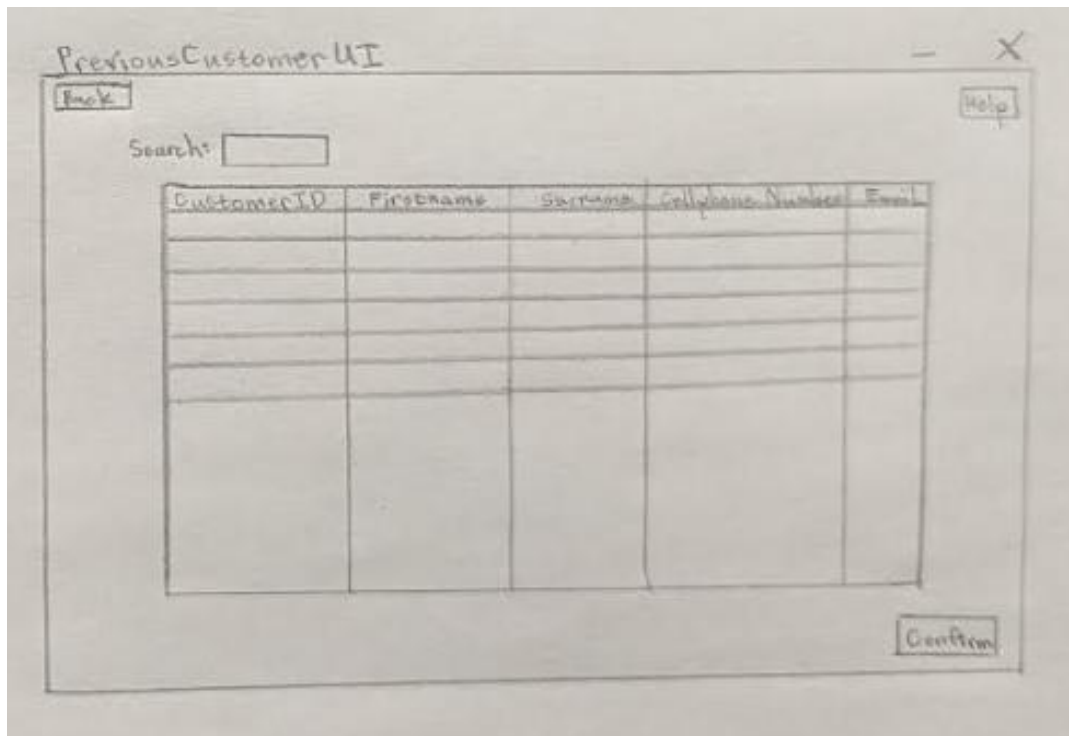
CustomerSelectionUI:

- **"New Customer" Button** pressed -> Opens the NewCustomerUI for adding a new customer.
- **"Previous Customer" Button** pressed -> Opens the PreviousCustomerUI to view and select previous customers.
- **"Collections" Button** pressed -> Opens CollectionsUI for managing customer tab collections.
- **"Kitchen Display" Button** pressed -> Opens KitchenDisplayUI to view current kitchen orders.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.
- **"Back" Button** pressed -> Returns to the LoginPageUI.



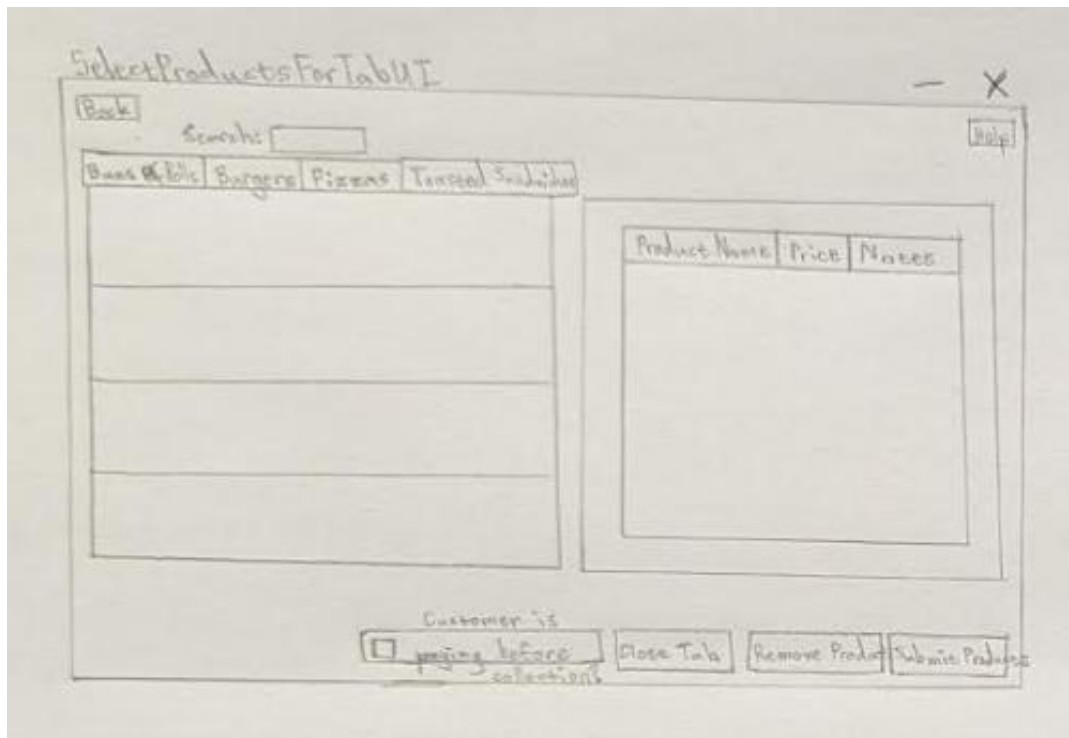
NewCustomerUI:

- User must enter their details in the respective fields for name, surname, cellphone number, and email to create a new customer record.
- **"Back" Button** pressed -> Returns to CustomerSelectionUI.
- **"Confirm" Button** pressed -> Validates input, saves the new customer details to the database, and closes the UI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



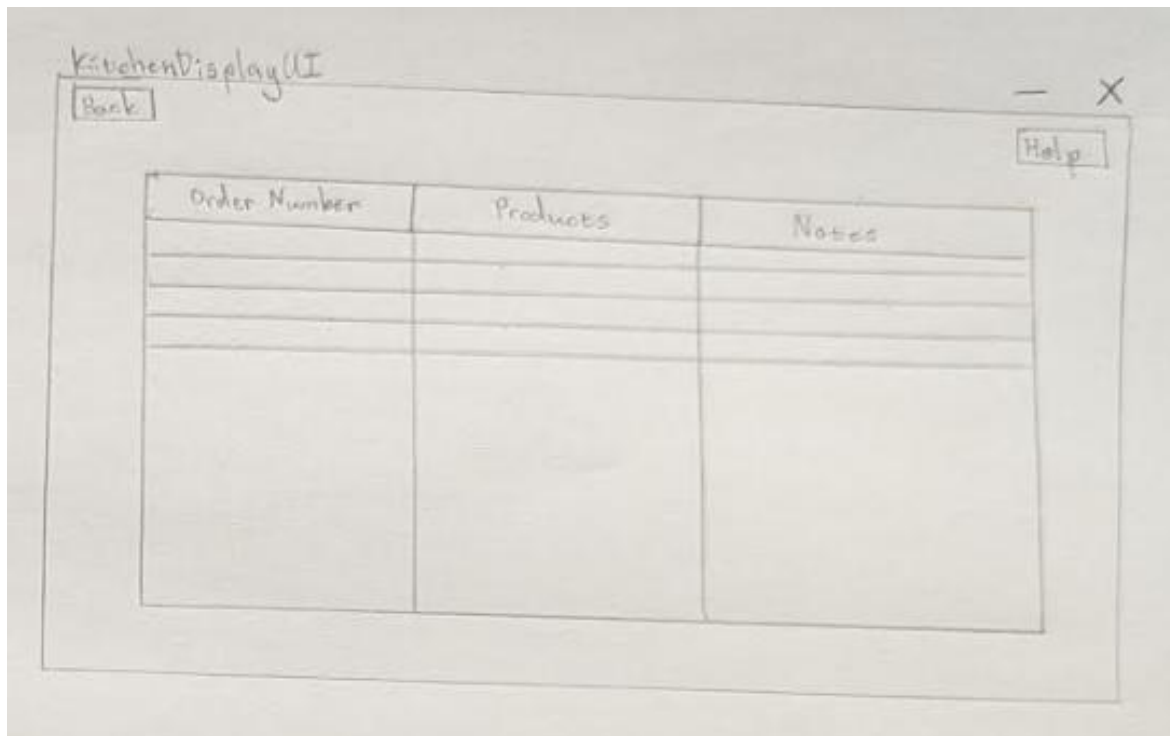
PreviousCustomerUI:

- Users can search for previous customers using the search functionality to find specific records.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.
- **"Back" Button** pressed -> Returns to CustomerSelectionUI.
- **"Confirm" Button** pressed -> Proceeds to the product selection screen for the selected customer.
- **Customer Table** is displayed for selecting a customer by clicking on a row.



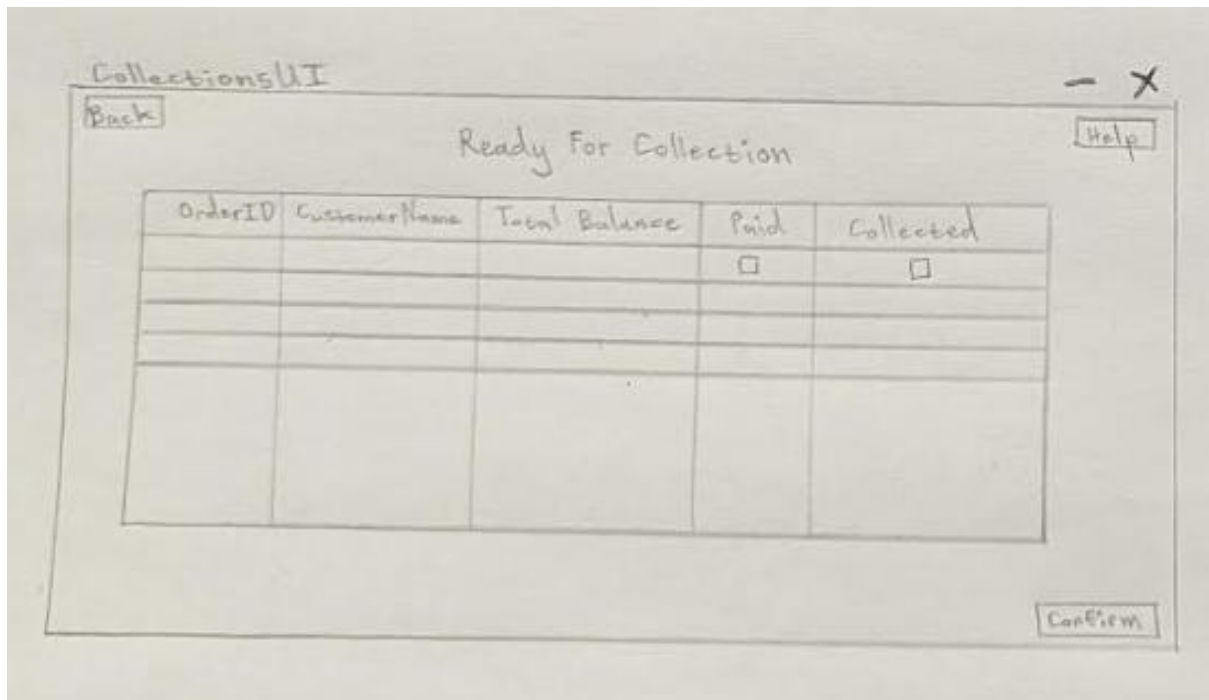
SelectProductsForTabUI:

- Users can view a list of products categorized by type, including product name, price, and notes.
- **"Product Buttons"** pressed -> Adds the selected product to the current customer's tab.
- **"Submit Products" Button** pressed -> Saves the selected products and their details to the tab.
- **"Remove Product" Button** pressed -> Removes the selected product from the tab.
- **"Close Tab" Button** pressed -> Closes the current tab and deletes it from the database.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.
- **"Paid Before Collection" Checkbox** toggled -> Indicates payment before collection and updates the paid status.



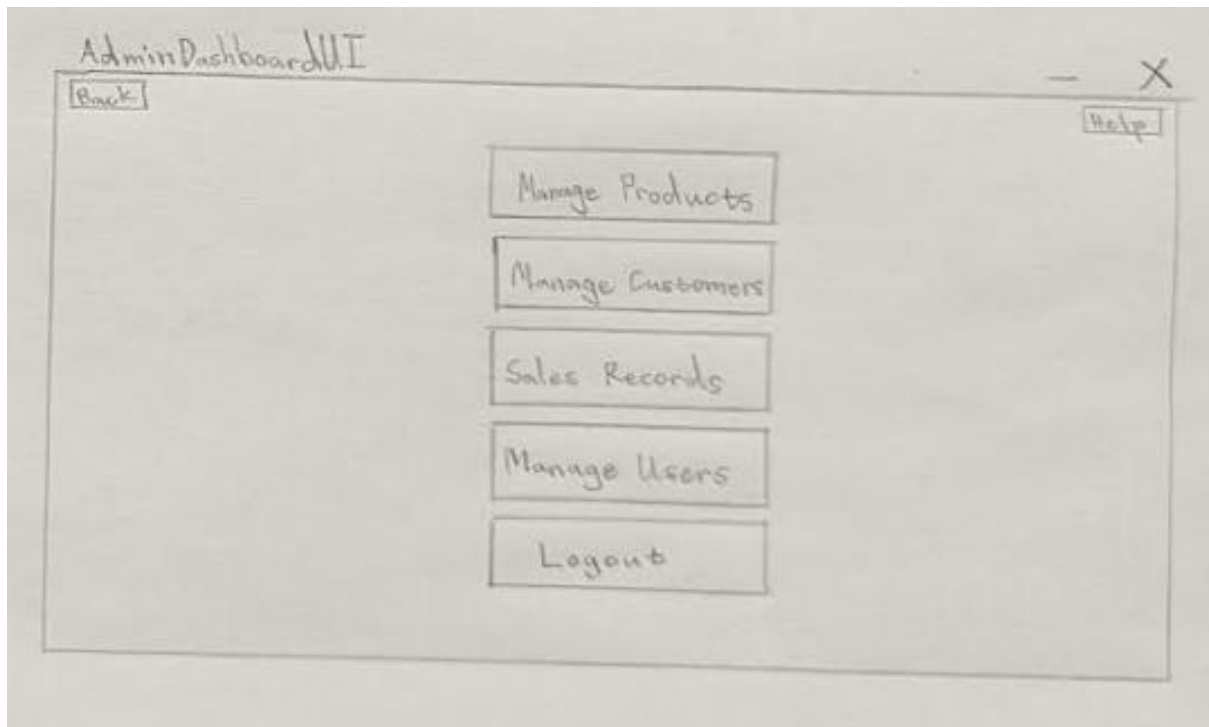
KitchenDisplayUI:

- Displays a table of current kitchen orders, including product details and associated notes.
- **"Order Table"** -> Clicking on a row allows interaction and updates the order status.
- **"Enter Key"** pressed -> Marks the selected order as complete and sends a notification to the customer.
- **"Back" Button** pressed -> Returns to CustomerSelectionUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



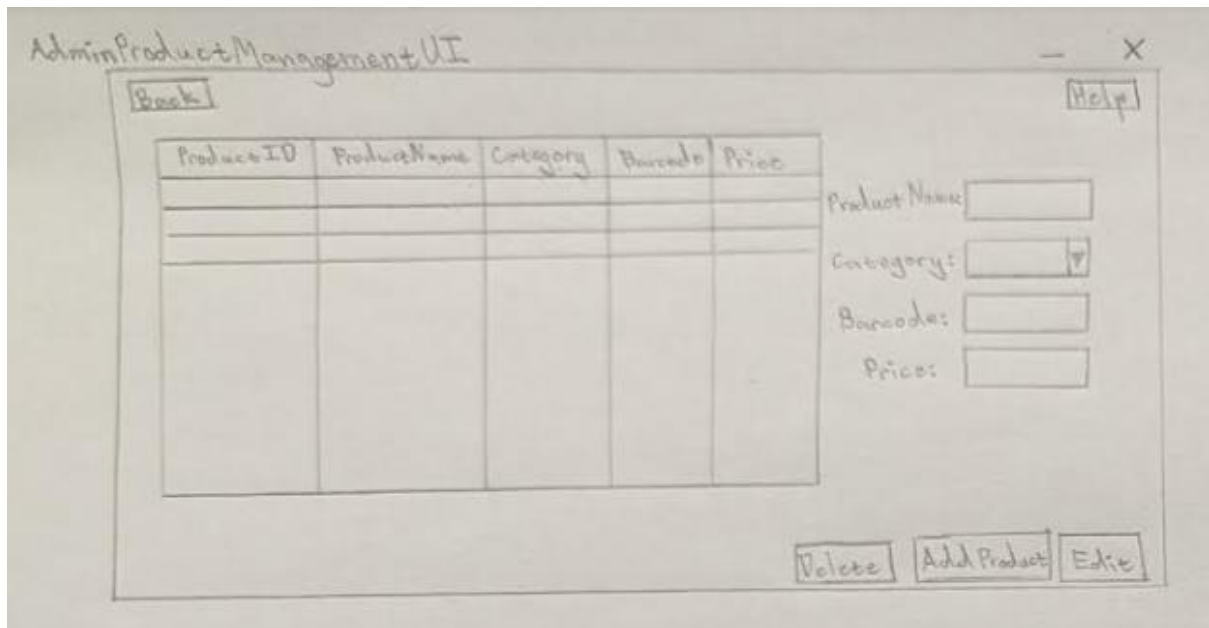
CollectionsUI:

- Displays customer orders, including order ID, customer name, total balance, and paid/collected checkboxes.
- **"Confirm" Button** pressed -> Updates the 'Paid' and 'Collected' statuses in the database.
- **"Back" Button** pressed -> Saves changes and navigates back to CustomerSelectionUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



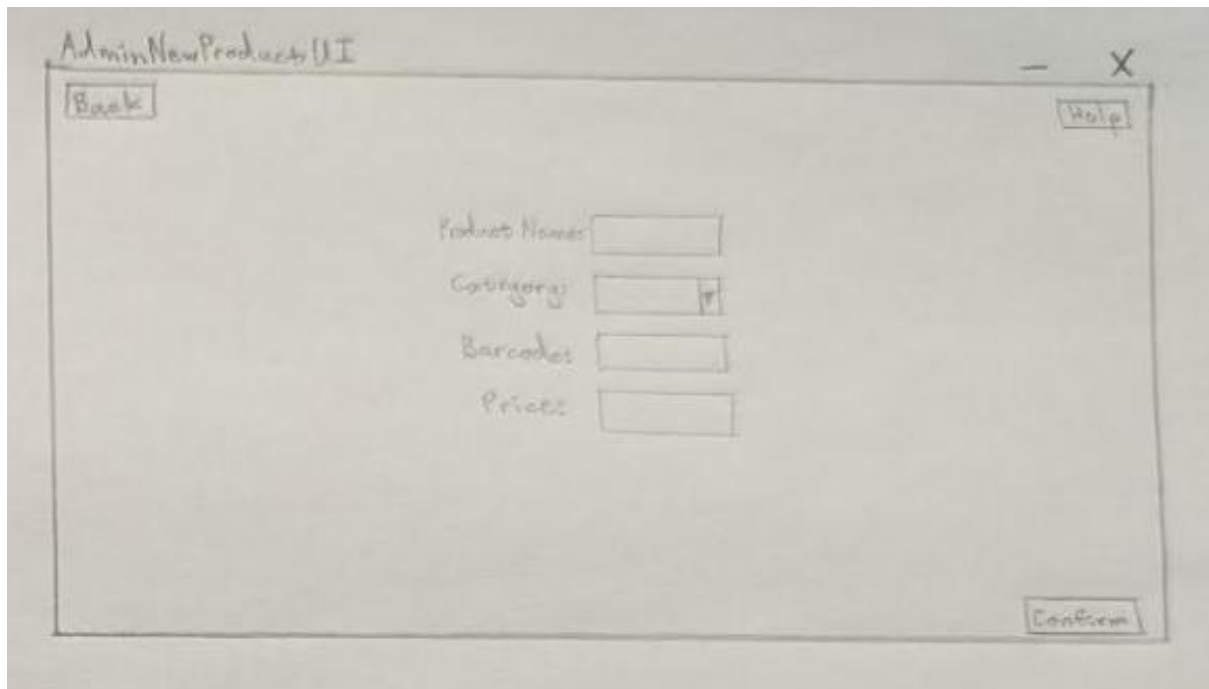
AdminDashboardUI:

- **"Manage Products" Button** pressed -> Opens AdminProductManagementUI for product management.
- **"Manage Customers" Button** pressed -> Opens AdminCustomerManagementUI for customer management.
- **"Sales Records" Button** pressed -> Opens AdminSalesRecordsUI to view sales records.
- **"Manage Users" Button** pressed -> Opens AdminUserManagementUI to manage users.
- **"Logout" Button** pressed -> Logs out the user and returns to LoginPageUI.
- **"Back" Button** pressed -> Navigates to the previous screen.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



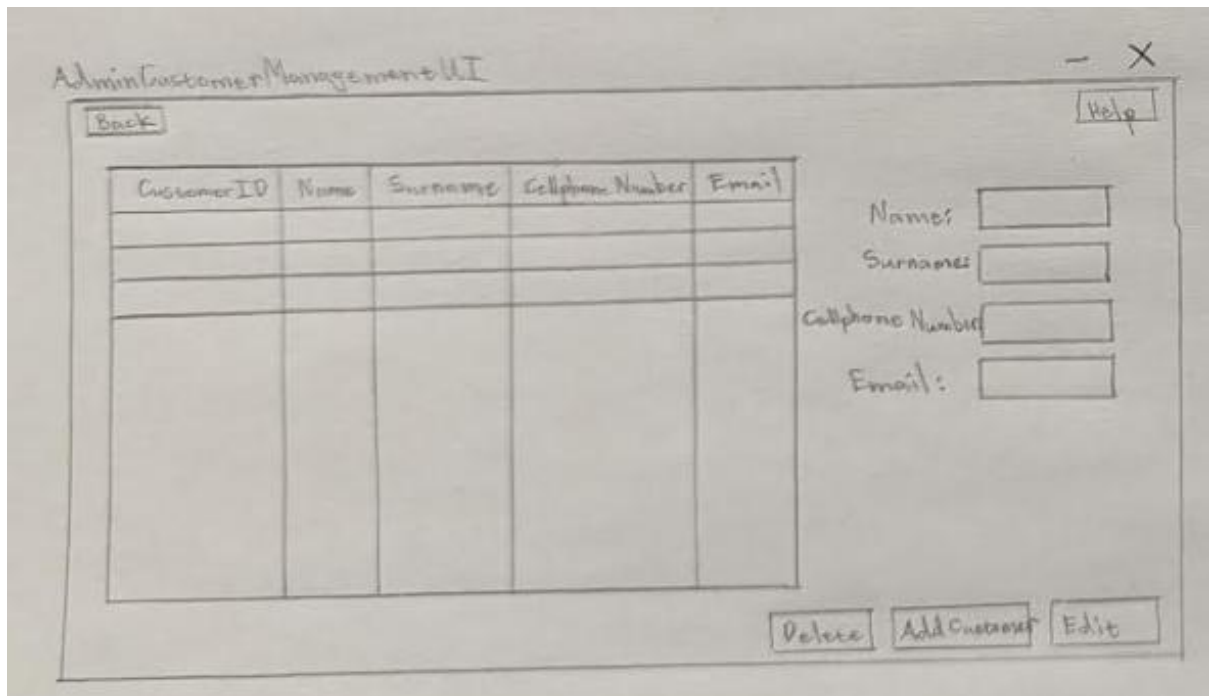
AdminProductManagementUI:

- **"Add Product" Button** pressed -> Opens AdminNewProductUI to add a new product.
- **"Edit" Button** pressed -> Updates the selected product's details in the database.
- **"Delete" Button** pressed -> Removes the selected product after confirmation.
- **"Back" Button** pressed -> Navigates back to AdminDashboardUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



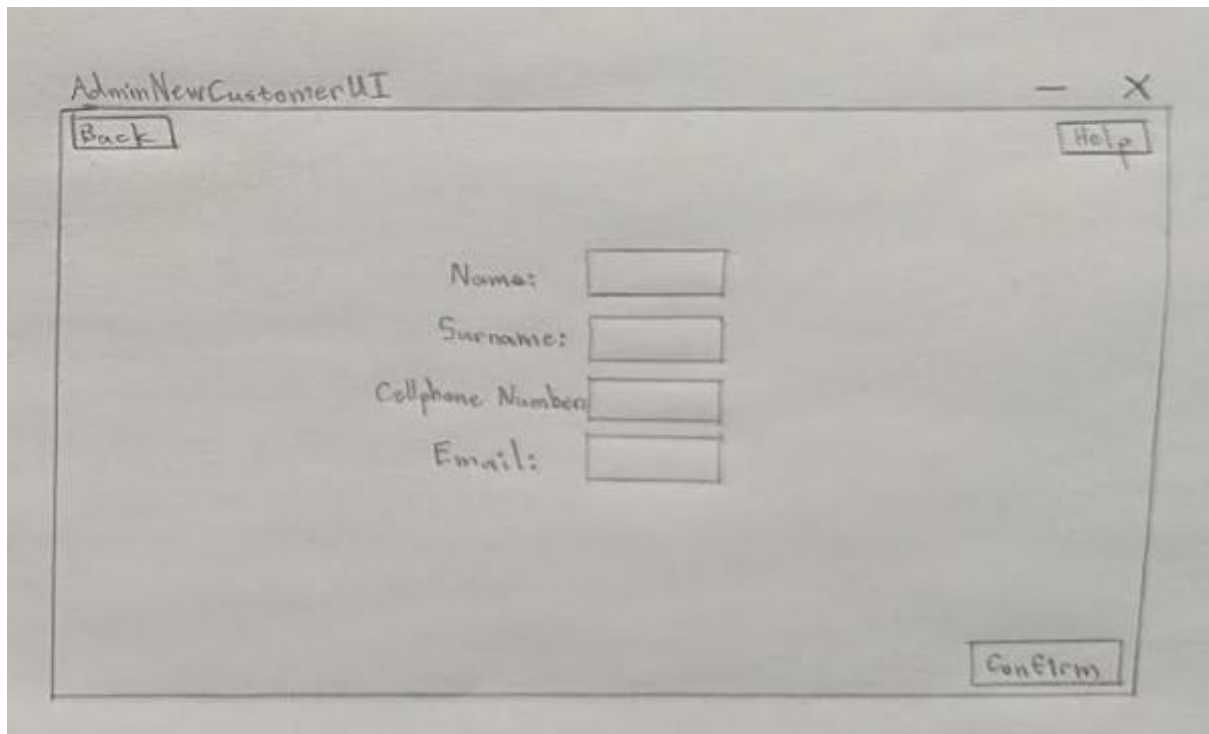
AdminNewProductUI:

- Users must enter details for a new product in the respective fields.
- **"Confirm" Button** pressed -> Validates input and saves the new product to the database.
- **"Back" Button** pressed -> Returns to AdminProductManagementUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



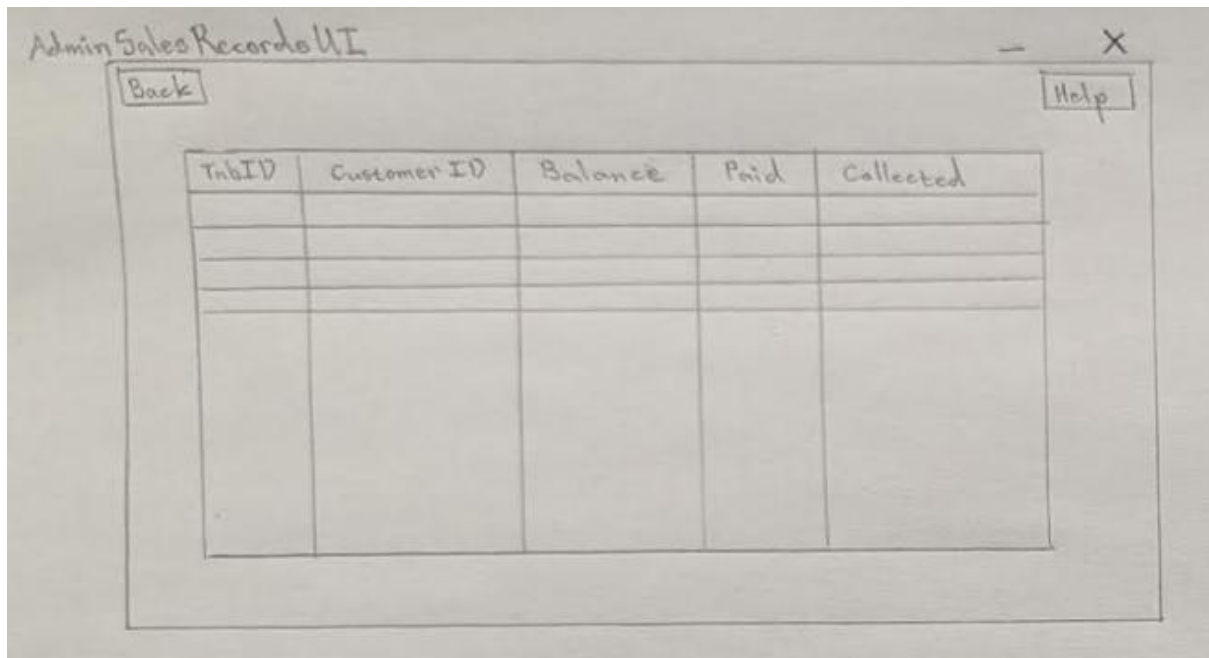
AdminCustomerManagementUI:

- **"Add Customer" Button** pressed -> Opens AdminNewCustomerUI to add a new customer.
- **"Edit" Button** pressed -> Updates the selected customer's details in the database.
- **"Delete" Button** pressed -> Removes the selected customer after confirmation.
- **"Back" Button** pressed -> Navigates back to AdminDashboardUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



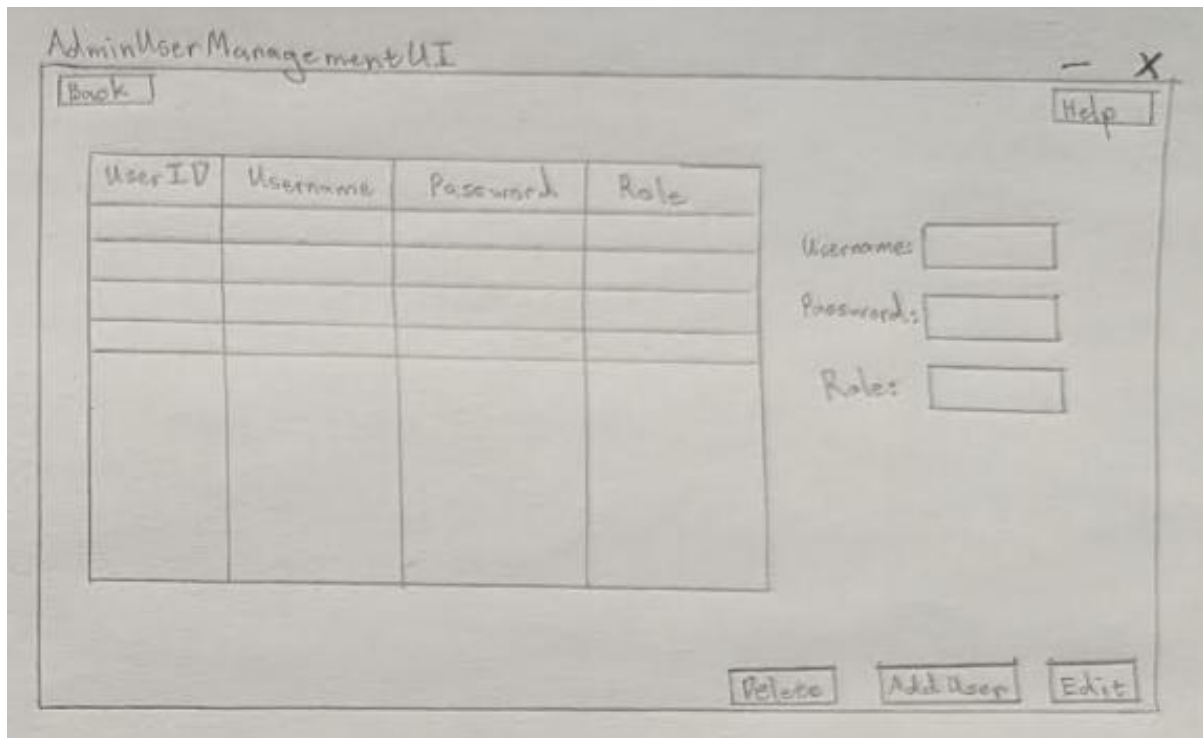
AdminNewCustomerUI:

- Users must enter details for a new customer in the respective fields.
- **"Confirm" Button** pressed -> Validates input and creates a new customer in the database.
- **"Back" Button** pressed -> Returns to AdminCustomerManagementUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



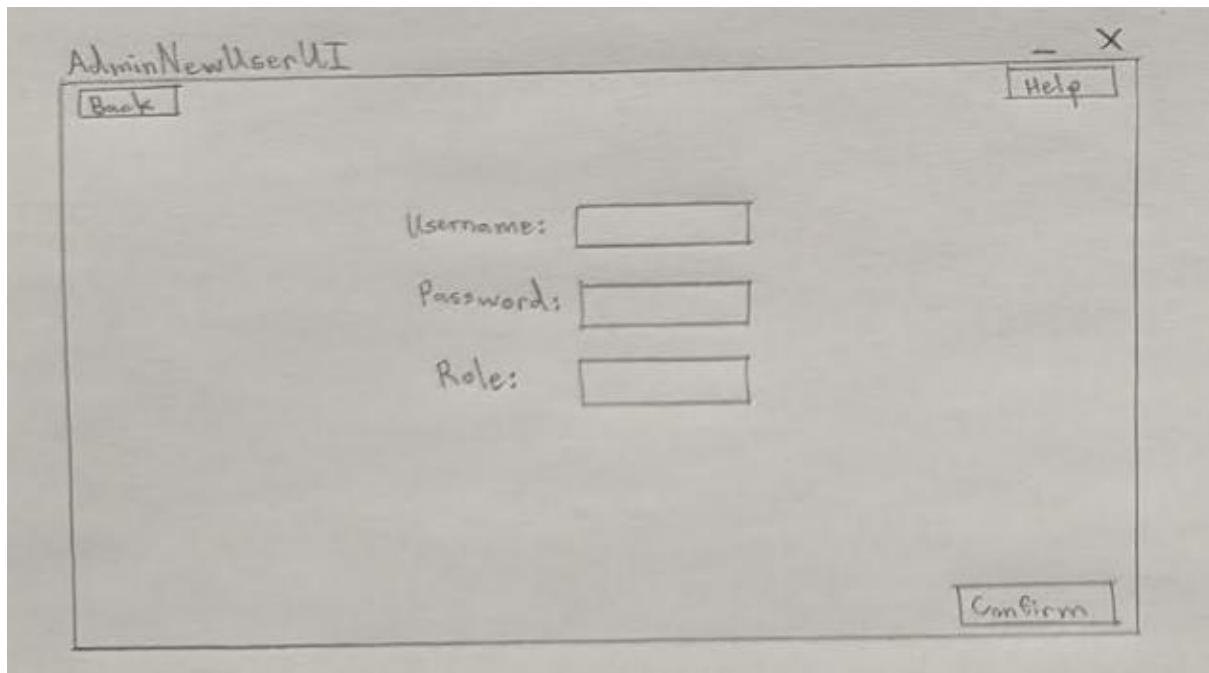
AdminSalesRecordsUI:

- Displays sales records including tab details such as TabID, CustomerID, Balance, Paid, and Collected statuses.
- **"Back" Button** pressed -> Opens AdminDashboardUI and closes the current window.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



AdminUserManagementUI:

- **"Add User" Button** pressed -> Opens AdminNewUserUI for adding a new user.
- **"Delete" Button** pressed -> Deletes the selected user after confirmation.
- **"Edit" Button** pressed -> Updates the selected user's information.
- **"Back" Button** pressed -> Saves changes and navigates back to the previous screen.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.



AdminNewUserUI:

- Users must enter details for a new user in the respective fields.
- **"Confirm" Button** pressed -> Creates a new user and saves the data to the database.
- **"Back" Button** pressed -> Returns to AdminUserManagementUI.
- **"Help" Button** pressed -> Displays a custom help dialog with instructions for this UI.

Specification of Help:

1.5 Specification of Help (see Appendix G – 12.4.15)		
[2]	Help completely described and a variety of help is described. Storage of help related data has been specified. Help provision appropriate for application type.	2
[1]	Help partially discussed with omissions and/or errors. Help is not appropriate for the type of project (e.g. printed manual is not required for mobile applications).	
[0]	Help not discussed.	

Help Function Specification

The Help feature in the Meal and Snack Ordering System serves as a friendly guide for users, assisting them in navigating and utilising the software effectively and correctly. Accessible through a dedicated "Help" button located in the top right corner of the user interface, this feature offers users tailored assistance depending on the specific UI context they are in.

Each help button has a consistent yet customisable appearance, ensuring it aligns with the overall UI design while providing users with relevant and easy-to-understand guidance. For example, the help button for the "Select Products for Tab" UI is designed as follows:

```
private void spftHelpBtnActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String helpMessage = "Welcome to the Select Products for Tab interface! Here's how  
to navigate:\n\n"  
  
        + "1. **Product Selection:**\n"  
  
        + " - Use the buttons to select products from the following categories:\n"  
  
        + " - Buns & Rolls\n"  
  
        + " - Burgers\n"  
  
        + " - Pizzas\n"  
  
        + " - Toasted Sandwiches\n"  
  
        + " - Snacks\n"  
  
        + " - Other\n\n"  
  
        + "2. **Adding Products:**\n"  
  
        + " - Click on a product button to add it to your customer tab.\n"  
  
        + " - A new tab will be created automatically if none exists.\n\n"  
  
        + "3. **Viewing Current Tab:**\n"
```

+ " - Your current tab is displayed on the right side, showing all added products.\n"

+ " - Edit product notes directly in the table as needed.\n\n"

+ "4. **Submitting Products:**\n"

+ " - Click the 'Submit Products' button to save your selections to the tab.\n"

+ " - Feel free to remove any products from the tab before finalizing.\n\n"

+ "5. **Closing the Tab:**\n"

+ " - Use the 'Close Tab' button to finalize your tab after submission.\n\n"

+ "6. **Searching for Products:**\n"

+ " - Enter the product name in the 'Search' text field and press 'Enter' to locate it.\n\n"

+ "7. **Navigation:**\n"

+ " - Click the 'Back' button to return to the previous screen at any time.\n\n"

+ "If you have any further questions or need assistance, please don't hesitate to contact support!";

```
JOptionPane.showMessageDialog(null, helpMessage, "Select Products for Tab Help",  
JOptionPane.INFORMATION_MESSAGE);  
  
}
```

Storage of Help Data

The help content is stored directly in the application's source code, ensuring that it is easily maintainable and customisable as the application evolves. This approach allows for quick updates to the help information in response to changes in the software functionality or user feedback.

Help Provision Appropriateness

The help provision is specifically designed for the application type, focusing on in-context assistance that enhances the user experience without requiring extensive printed manuals. The dialog-based help ensures users can access the information they need right at their fingertips, making it appropriate for the digital nature of the software.

Specification of Permanent Data Storage:

1.6 Specification of Permanent Data Storage		2
[2]	All data fields have been correctly described and grouped appropriately.	
[1]	Some data fields are described with a few errors or data is not appropriately grouped (i.e. related fields should be listed together).	
[0]	No information given.	

Storage:

- Uses MySQL Workbench database with 5 tables for storage (tblcustomers, tblproducts, tbltabproducts, tblusers, tbltabs).

Information Stored:

1. Table: tblcustomers

- **Purpose:** Stores customer information.
- **Fields:**
 - **customerID** (int, Primary Key, Auto Increment): Unique identifier for each customer.
 - **customerFirstname** (varchar(45), NOT NULL): Customer's first name.
 - **customerSurname** (varchar(45), NOT NULL): Customer's surname.
 - **customerCellphoneNumber** (varchar(45), NOT NULL): Customer's cellphone number.
 - **customerEmail** (varchar(45), NOT NULL): Customer's email address.

2. Table: tblproducts

- **Purpose:** Stores product information for the Wyvern Club.
- **Fields:**
 - **productID** (int, Primary Key, Auto Increment): Unique identifier for each product.
 - **productName** (varchar(45), NOT NULL): Name of the product.

- **category** (varchar(45), NOT NULL): Category of the product (e.g., Buns & Rolls, Burgers).
- **barcode** (varchar(45), NOT NULL): Barcode for the product, used for scanning.
- **price** (double, NOT NULL): Price of the product.

3. Table: tbltabproducts

- **Purpose:** Manages the products associated with each customer tab, along with assigned notes.
- **Fields:**
 - **entryID** (int, Primary Key, Auto Increment): Unique identifier for each entry in the tab products.
 - **tabID** (int, NOT NULL): Identifier linking this entry to a specific customer tab (foreign key).
 - **productID** (int, NOT NULL): Identifier linking this entry to a specific product (foreign key).
 - **notes** (varchar(45), DEFAULT NULL): Any specific notes related to the product (e.g., customization requests).
 - **ready** (tinyint(1), NOT NULL, DEFAULT '0'): Status indicating if the product is ready (1) or not (0).

4. Table: tblusers

- **Purpose:** Manages user accounts for the system, allowing for admin and staff functionalities.
- **Fields:**
 - **userID** (int, Primary Key, Auto Increment): Unique identifier for each user in the system.
 - **username** (varchar(45), NOT NULL): The username for user authentication before proceeding to logging in.
 - **password** (varchar(255), NOT NULL): The hashed password for user authentication before proceeding to login.

- **role** (varchar(20), NOT NULL): The role of the user (e.g., 'admin' or 'staff') determining access levels.
- **email** (varchar(100), DEFAULT NULL): The email address of the user.

5. Table: tbltabs

- **Purpose:** Manages customer tabs for the Wyvern Club.
- **Fields:**
 - **tabID** (int, Primary Key, Auto Increment): Unique identifier for each tab.
 - **customerID** (int, NOT NULL): Identifier linking this tab to a specific customer (foreign key).
 - **balance** (decimal(10, 2), NOT NULL): The current balance of the tab.
 - **notes** (varchar(255), DEFAULT NULL): Any specific notes related to the tab (e.g., special requests).
 - **paid** (tinyint(1), NOT NULL, DEFAULT '0'): Status indicating if the tab has been paid (1) or not (0).
 - **collected** (tinyint(1), NOT NULL, DEFAULT '0'): Status indicating if the tab has been collected (1) or not (0).

Hardware and Software Requirements (including additional hardware):

1.7 Hardware and Software Requirements (including additional hardware)		
[2]	Hardware and software list is complete including development (IDE software, secondary software such as database software, SQL connectors) and includes software versions.	2
[1]	Hardware and software is incomplete for development (IDE software, secondary software such as database software, SQL connectors) or missing details of required software or versions missing.	
[0]	Hardware and software not discussed.	

Hardware and Software Requirements for Meal and Snack Ordering System:

Programmer's Requirements:

Hardware:

- **Processor:** 1 GHz or faster
- **RAM:** 4GB or higher
- **Storage:** 2GB or higher
- **GPU:** Integrated with CPU (sufficient for development tasks)
- **Monitor:** Generic
- **Keyboard:** Generic
- **Mouse:** Generic

Software:

- **Operating System:** Windows 10 or newer
- **Java 8 or higher** (Required for application development)
- **NetBeans IDE 8.2** (Integrated Development Environment for Java programming)
- **MySQL Workbench 8.0 CE** (Database management for developing and testing the system)
- **MySQL Connector 8.0.22** (Connector to enable communication between Java and the MySQL database)
- **Google Docs** (For documentation and project collaboration)
- **Web Browser** (For accessing online resources, cloud services, or APIs like Twilio)

User's Requirements:

Hardware:

- **Processor:** 1 GHz or faster
- **RAM:** 3GB or higher
- **Storage:** 2GB or higher
- **GPU:** Integrated with CPU (sufficient for running the system)
- **Monitor:** Standard display
- **Keyboard:** Generic or laptop keyboard
- **Mouse:** Standard or laptop touchpad

Software:

- **Operating System:** Windows 10 or newer
- **Java 8 or higher** (To run the Java-based application)
- **MySQL Workbench 8.0 CE** (To view or interact with the database if necessary)
- **MySQL Connector 8.0.22** (To ensure proper database connectivity when interacting with the system)

Additional Considerations:

- **WhatsApp notifications** (For sending collection notifications)
- **Barcode Scanning Software** (For integrating with the product inventory system in the application)

Test PC:

- **Hardware:**
 - CPU: Intel(R) Core(TM) i7-10510U
 - RAM: 16,0 GB
 - Storage: 200gb

- GPU: Integrated with CPU (And High-performance NVIDIA processor)
- Monitor: Laptop
- Keyboard: Laptop keyboard
- Mouse: Logitech
- **Software:**
 - Java(TM) SE Runtime Environment 1.8.0_111-b14