**SE-I Course Project (PhasEs 1 & 2 COVER SHEET)**

**Discussions Scheduled for Week 14 *(Monday | Tuesday)***

* Print 1 copy of this cover sheet and attach both to a printed copy of the documentation *(SRS, … etc.)*. You must submit a CD including softcopies of all your documents and Project implementation.
* Please write all your names in English.
* Please make sure that your students’ IDs are correct.
* Handwritten Signatures for the attendance of all team members should be filled in the cover sheet copy before the discussion.
* Please attend the discussion on time *(announced separately)*, late teams will lose 3 grades.

**Project Name: Juice Shop Management System**

**Project Leader Name: Youssef Mohamed Khalil**

**Team Information *(typed not handwritten, except for the attendance signature)*:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ID**  **[Ordered by ID]** | **Full Name**  **[In English]** | **Attendance**  **[Handwritten Signature]** | **Final Grade** |
| **1** | 320210097 | Nada Mossad Mohamed |  |  |
| **2** | 320210107 | Youssef Mohamed Khalil |  |  |
| **3** | 320210113 | Hana Hisham Rashad Shaaban |  |  |
| **4** | 320210124 | Noran Samir Hamid Mohamed |  |  |
| **5** | 320210236 | Marwa Ibrahim Sultan |  |  |
| **6** | 320210332 | Sarah Ali Mohamed |  |  |

**Grading Criteria:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | | **Actual Grade** | **Notes** |
| **Functional Requirements** | **2** |  |  |
| **Non-Functional Requirements** | **2** |  |  |
| **Use-Case Diagram(s)** *including general use-cases for the system, and the detailed use-cases description* | **3** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity Diagram(s)** | **3** |  |  |
| **Database Specification** *(ERD, Tables)* | **1** |  |  |
| **System Architecture –** *including applied Architectural Pattern(s)* | **2** |  |  |
| **Sequence Diagram(s)** | **3** |  |  |
| **System Sequence Diagrams (SSDs)** | **2** |  |  |
| **Collaboration/Communication Diagram(s)** | **3** |  |  |
| **Class Diagram *(2 versions)***   1. **An initial version based on the requirements and Use-Case/Activity diagrams.** 2. **An intermediate version based on the interaction diagrams.** | **4** |  |  |
| **Object Diagrams** *(Including object diagrams that illustrate the preconditions and the post-conditions of selected functions)* | **3** |  |  |
| **Package Diagram(s)** | **2** |  |  |
| **Self-Study Component 1: State-Machine Diagrams** *(for selected state-dependent objects)* | **1** |  |  |
| **Self-Study Component 2: Deployment diagram(s)** | **1** |  |  |
| **Front End Design for all Functions** Desk-Top **or** *(HTML, Bootstrap).* | **2** |  |  |
| **Implementation based on the submitted Requirements & Design. Should include at least 4 of the following modules (in addition of course to modules specific to your individual projects):**   1. **User Role Management Module.** 2. **User manipulation Module** *(Login, Add / Delete / Update / Search, List).* 3. **Controlling Resources Module** *(Rooms, Orders, Products, ... etc.).* 4. **Reservation and Rescheduling Module.** 5. **Generating Reports Module** *(PDFs, … etc.).* 6. **Sending Emails or Notifications Module.** | **8** |  | **2** *per module* |
| **Presentation Skills** | **3** |  |  |

**45**

**Teaching-Assistant’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Introduction:**

The project is a juice shop management system for handling the processes of searching the items and finding the price of each item. The program has two modes: the cashier and admin modes.

In the cashier mode, the customer can use the program in the cashier mode to see all the products that the shop offers and make his own order then the order receipt will be sent to the customer’s email.

In the admin mode, the seller can make orders like the cashier mode in addition to the features, add items, remove items, and update items. Add item is the feature responsible for adding new items in which the admin will need to record the item’s name, price, and quantity in the store. Delete feature, Is the feature through which the admin can completely delete an item from the store’s database if the shop is no longer offering this item. Update item is the feature through which the admin can update any information about the item like its name, price, or quantity.

The user can easily log out from any mode to log into another mode. There is one admin user as the processes of adding, deleting, and updating items is a very critical and private process**.**

* 1. **Scope:**

Our project aims to serve all juice shops all over the world. This version supports the most spoken language in the globe, which is English. The system admin has access to all necessary functional specifications and data sources.

* 1. **Purpose:**

A Juice Shop Management System is software built to handle the primary functions of a juice shop. Juice shops rely on this management system to manage asset ordering as well as update the menu with their needs. Juice shop management systems help managers keep track of the stock and their receipts, as well as customers’ orders and needs.

The juice shop management system also involves maintaining the database for entering new juice types and recording existing ones that have been added in their suitable category.

1. **Technology Used:**

Our project is a desktop application developed in c#, we use SQL server as a dynamic database server to store and add items and their cost constantly. we preferred to use windows forms for GUI because of its features on the dotnet as requesting input from the user when taking orders, and create menus contain items and images. we will use the Guna Widget for customizing themes and dragging & dropping UI controls.

Icon

Description automatically generatedIcon

Description automatically generatedIcon

Description automatically generated

1. **Stakeholders:**
   1. **Cashier:**

The system allows the registration of cashiers; they will be able to manage the shop smoothly with high accuracy so that no mistakes occur, and no orders get mixed. The cashier directly deals with customers as he can place customers’ orders and then he gets to print them (if needed).

* 1. **Admin:**

The system allows the registration of admins; they will be able to manage the shop smoothly with high accuracy so that no mistakes occur, and no orders get mixed. The admin can add items to the menu, remove items from it or update the pre-existing items. The admin can also deal with customers and place their preorders in the system.

* 1. **Government:**

The system will provide easy access to the records of the shop and the total income of the shop as well as the total taxes that should be collected.

* 1. **Shop owner:**

The system will provide a steady flow of work with almost no mistakes or problems. The system will also calculate the customers’ cheques, which will be time-saving for staff-customer interactions. It will also ensure that no calculation mistakes occur.

* 1. **Customer/ Guest:**

The system ensures high accuracy so that the customers will not be disturbed by wrong orders or any mistake in calculating their cheques.

* 1. **Development Team:**

The development process will be as easy as possible, as the system’s requirements are clear, specific, and applicable.

**4.0. Requirements:**

**4.1. Functional Requirements:**

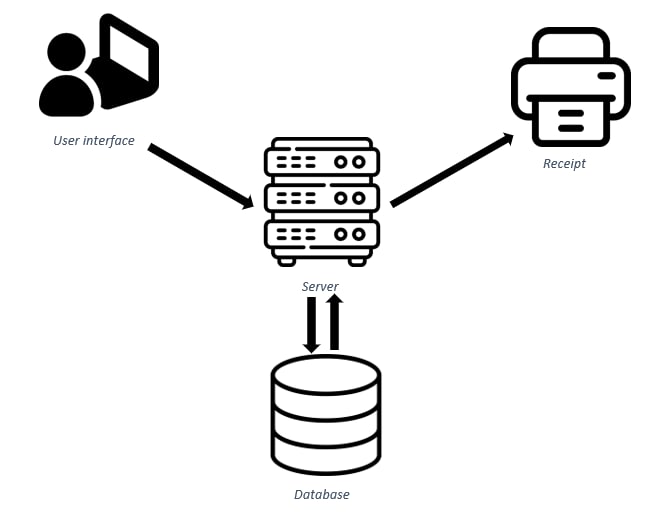
|  |  |
| --- | --- |
| **Admin** | **Requirements:** |
|  | * Log in * Log out * Add items * Remove items * Place orders * Print receipt * Update items (name, price, category) |

|  |  |
| --- | --- |
| **cashier** | **Requirements:** |
|  | * Place orders * Print orders |

**4.2. Non-Functional Requirements:**

|  |  |
| --- | --- |
| **Non-Functional Requirements** | **Description** |
| * **Security** | Only the admin should be able to edit the items list. |
| * **Capacity** | The system is designed to be able to store the records of the very big shops and the big amount of items’ data |
| * **Scalability** | The system should with the same high performance at both small and big cafés |
| * **Usability** | The system should be user-friendly and easy to be used by non-technical people. |
| * **Reliability** | The system is not dealing with many users so it should not fail at any time. |
| * **Availability** | The system should be always available for the user. |

**5.0. System Architecture:**

****

**5.1. Database:**

Collection of information monitored by the system, We will keep the information about login and items in the shop related to what kinds of juice we have right now in this component.

**5.2. Server:**

A piece of computer hardware or software (computer program) that provides functionality for our program, allows the user to log in successfully, and use the system as he or wants. The usercan make an order, add items, update items, and delete items.

**5.3. User Interface:**

It is the point of human-computer interaction and communication in our software, we have an attractive user interface that is easy to use, so the user ca

**5.4. Receipt:**

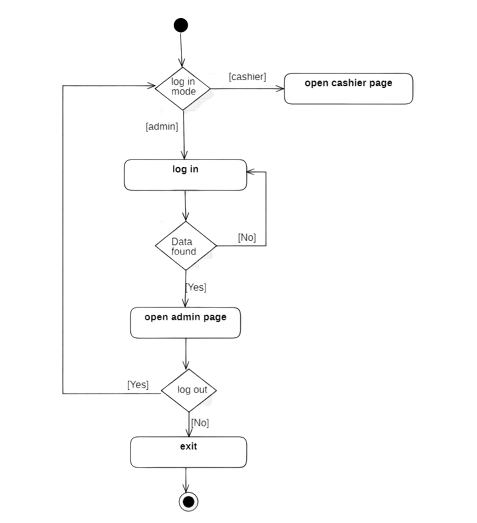
When the customer makes an order, the cashier approves that and gets the required items for the customer. The cashier prints a receipt for the user where all he ordered is mentioned.

**6.0. Activity Diagrams:**

**6.1. Log-in activity diagram:**

We have two login modes, cashier and admin. If the chosen login mode is cashier, then “open cashier page” which will be described in the next activity diagram. Else if the chosen login mode is admin so let him log in with his name and password. Checking the entered data if false so let the user enter the data again, if true then “open admin page”.

If the user wants to log out then open the log-in page again.

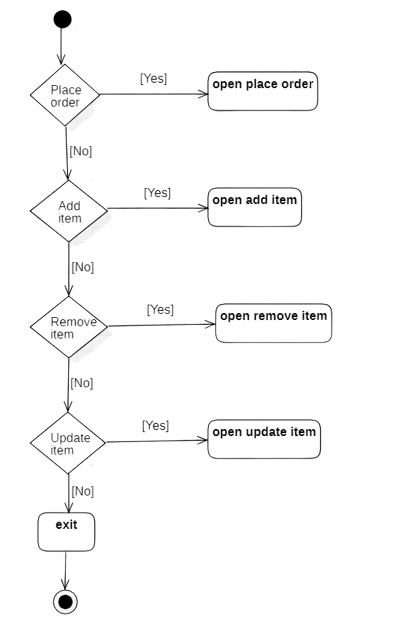
****

**6.2. Admin mode activity diagram:**

if the user is in admin mode he can place an order, add the item to the menu, remove the item from the menu, and update the item to the menu.

when the admin chooses one of them a specific page will

open according to the chosen task.

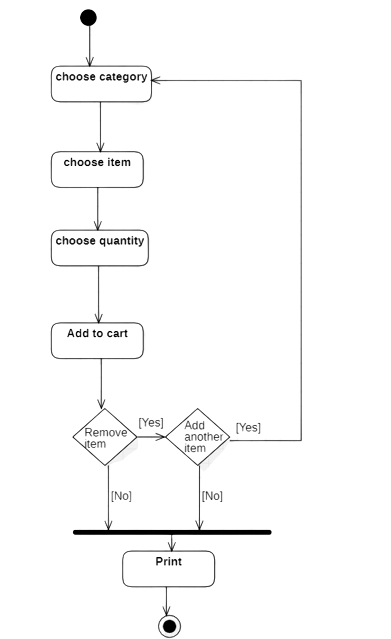


**6.3.Place order activity diagram:**

To place an order, the user should choose the category first, then choose an item from that category. He can specify the quantity needed for that item, then add it to the cart.

Maybe the user changes his mind and removes an item, if yes then ask him if he wants to add another item if yes, then let him choose the category again and soon.

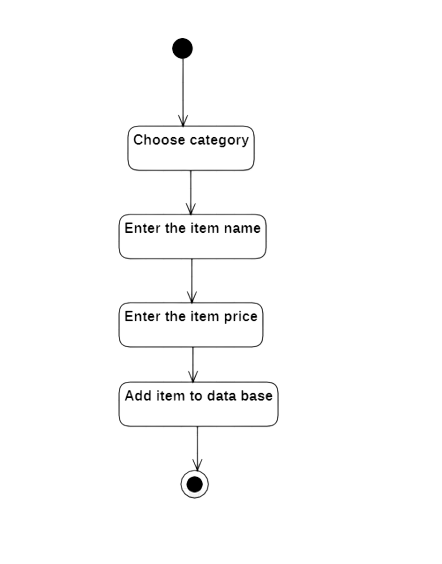
If all is done and the user doesn’t want to remove or add another item then print the pill.



**6.4. Add item to the menu activity diagram:**

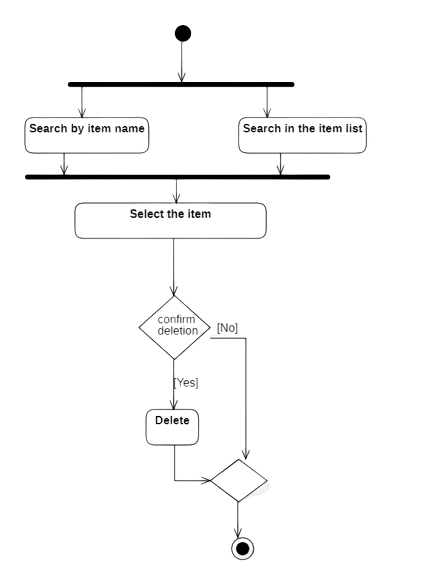
The admin can add a new item to the menu, but he must choose the category that item will be in, the name of the item, and its price.

After doing all of that the item will be added to the database.



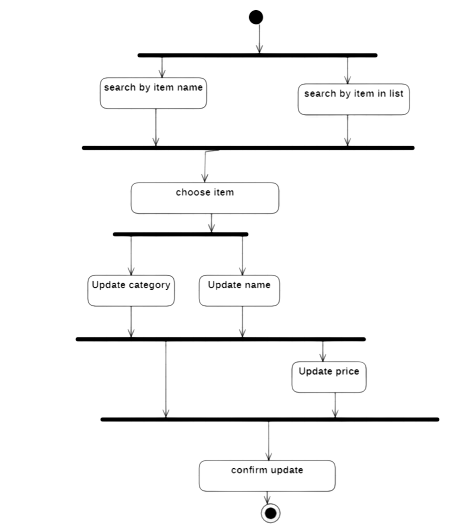
**6.5. remove an item from the menu activity diagram:**

before removing the item we can search for it by item name or search in the item list, if the user found the wanted item and then select it, the page will ask for confirming deletion if true then delete it, if false then the item will not be deleted.

****

**6.6. update item activity diagram:**

admin can update the item, but first, he can search for it by item name or by item in the list. After choosing the item we can update three things the category, name, and price then confirms that update.



**7.0 Use-Case Diagram:**

A picture containing graphical user interface

Description automatically generated

**7.1. Log-in process:**

The admin or the cashier starts the system, they log into the system using their information, which has been previously registered on the system. After they log in each can get the system to perform tasks/actions that are allowed for each.

This scenario fails if the admin/cashier tries to log into the system using invalid data, they will not be allowed to log in.

**7.2. Add item process:**

The admin or the cashier starts the system, then the admin logs into the system using valid data (data that is registered on the system). After that, the admin can add items and they will be registered on the system.

This scenario fails if the admin tries to log into the system using invalid data, they will not be allowed to log into the system.

**7.3. Remove item process:**

The admin or the cashier starts the system, then the admin logs into the system using valid data (data that is registered on the system). While removing a certain item, the item must be registered on the system for the admin to remove it.

This scenario fails if the admin tries to log into the system using invalid data, they will not be allowed to log in. The other case of failure is if the item required to be removed is not registered on the system. Therefore, they cannot remove an item that does not exist on the system.

**7.4. Update item process:**

The admin or the cashier starts the system, then the admin logs into the system using valid data (data that is registered on the system). After that the admin chooses to update a certain item. To update the item the admin must search for it and find it then they can update this item. Then the item will be updated in the system.

This scenario fails if the admin tries to log into the system using invalid data, they will not be allowed to log in. The other case of failure is that the item that the admin is looking for is not registered on the system. Therefore, they cannot update an item that does not exist on the system.

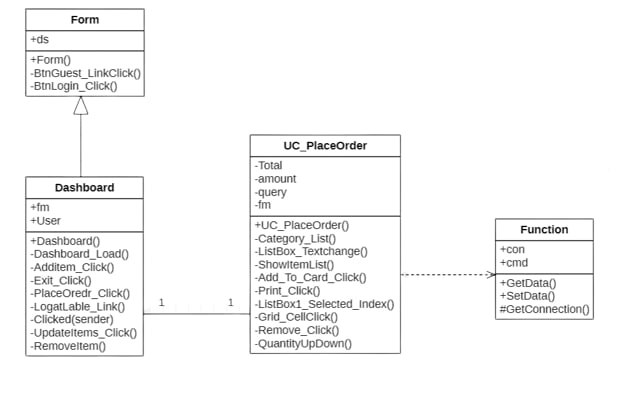
**7.5. Place order process:**

The admin or the cashier starts the system, then the admin/cashier logs into the system using valid data (data that is registered on the system). After that, the admin/cashier chooses to place an order, while placing the order the admin/cashier must add the chosen items to the cart (customer’s order). The items added to the order must be a part of the menu. After placing the order, the admin/cashier has access to print the order or cancel it. In the end, the system will proceed with the order and allow the printing or cancellation of the order.

This scenario fails if the admin/cashier tries to log into the system using invalid data, they will not be allowed to log in. The other case of failure is the attempt of adding items to the order that are not registered in the menu. Therefore, they cannot add an item that does not exist in the system to order.

**8.0. Class Diagram:**

**8.1. Initial Version:**

****

**8.2. Intermediate Version:**

**Diagram

Description automatically generated**

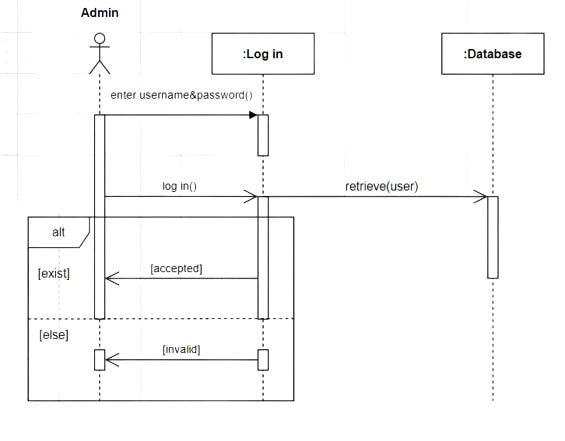
**8.3. Final Version:**

**Diagram

Description automatically generated**

**9.0. Sequence Diagrams:**

**9.1. Log In Sequence Diagram:**

****

**9.2. Add Item Sequence Diagram:**

**Diagram

Description automatically generated**

**9.3. Update Item Sequence Diagram:**

**Diagram

Description automatically generated**

**9.4. Remove Item Sequence Diagram:**

**Diagram

Description automatically generated**

**10. System Sequence Diagrams:**

**10.1. Place Order System Sequence Diagram:**

**Diagram

Description automatically generated**

**10.2. Add Item System Sequence Diagram:**

**Diagram

Description automatically generated**

**10.3. Update Item System Sequence Diagram:**

**Diagram

Description automatically generated**

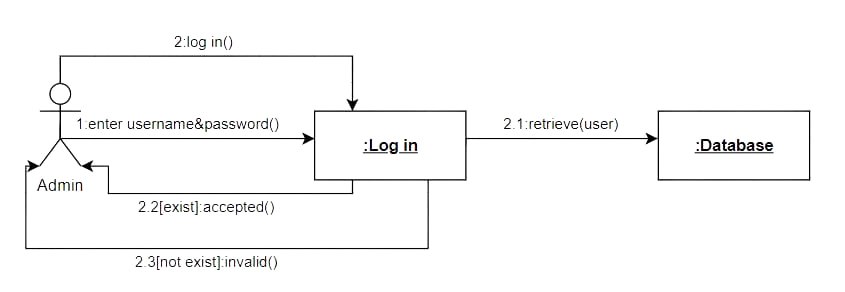
**10.4. Delete Item System Sequence Diagram:**

**Diagram

Description automatically generated**

**11.0. Collaboration / Communication Diagrams:**

**11.1. Log In Collaboration Diagram:**

****

**11.2. Add Item Collaboration Diagram:**

Diagram

Description automatically generated

**11.3. Update Item Collaboration Diagram:**

**Diagram

Description automatically generated**

**11.4. Delete Item Collaboration Diagram:**

**Diagram

Description automatically generated**

**12.0. Deployment Diagrams:**

**12.1. Deployment Diagram i:**

**Diagram

Description automatically generated**

**12.2. Deployment Diagram ii:**

**Diagram

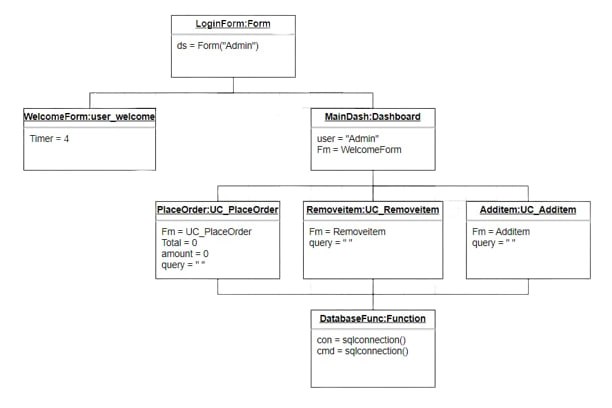
Description automatically generated**

**12.3. Deployment Diagram iii:**

**Diagram

Description automatically generated**

**13.0. Object Diagram:**

****

**14. Package Diagram:**

**Diagram

Description automatically generated**

**15.0 State-Machine Diagram:**

Diagram

Description automatically generated

**16.0. ERD Diagram:**

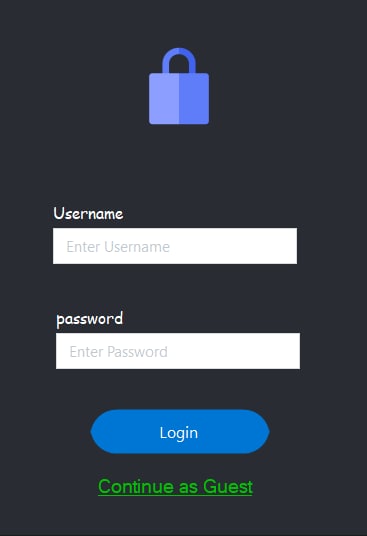
**Diagram

Description automatically generated**

**17.0. Snapshots of the user interface:**

**17.1. login page:**

This page asks the user to enter his username and password and check the database to make sure that the data is correct



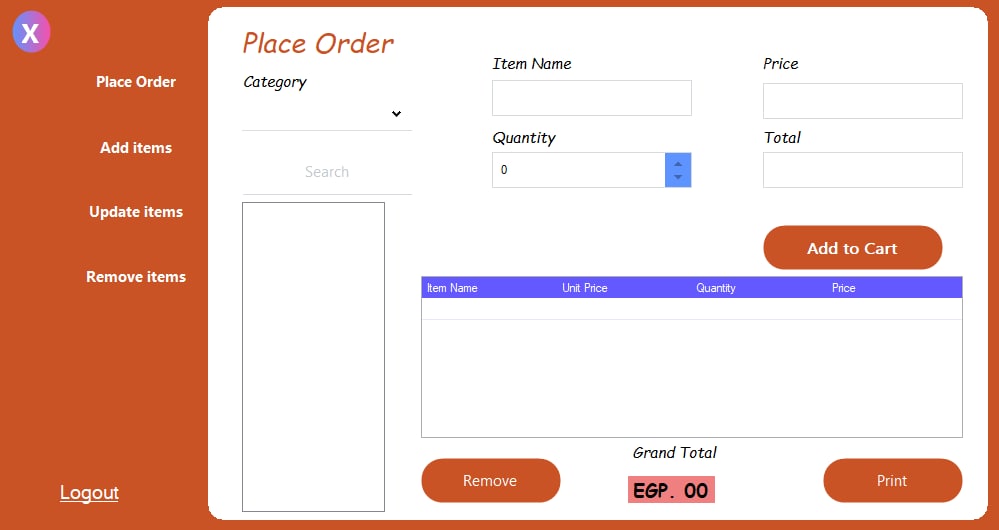
**17.2. Welcome page:**

after the user logs in successfully, this page appears to show him all options of the system such as placing an order, adding items, updating items, and removing items.

****

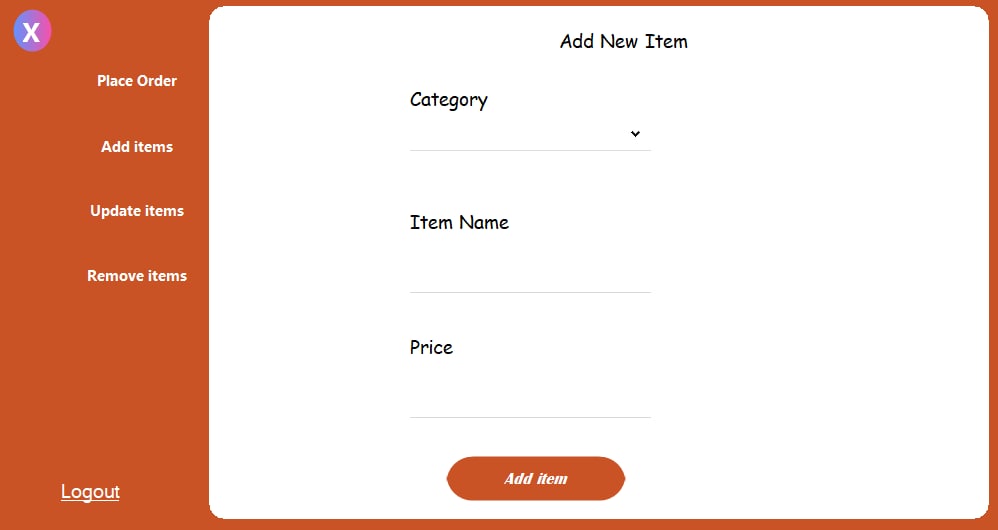
**17.3. place order page:**

this page through which the admin can make the order for the customer and print the order receipt.



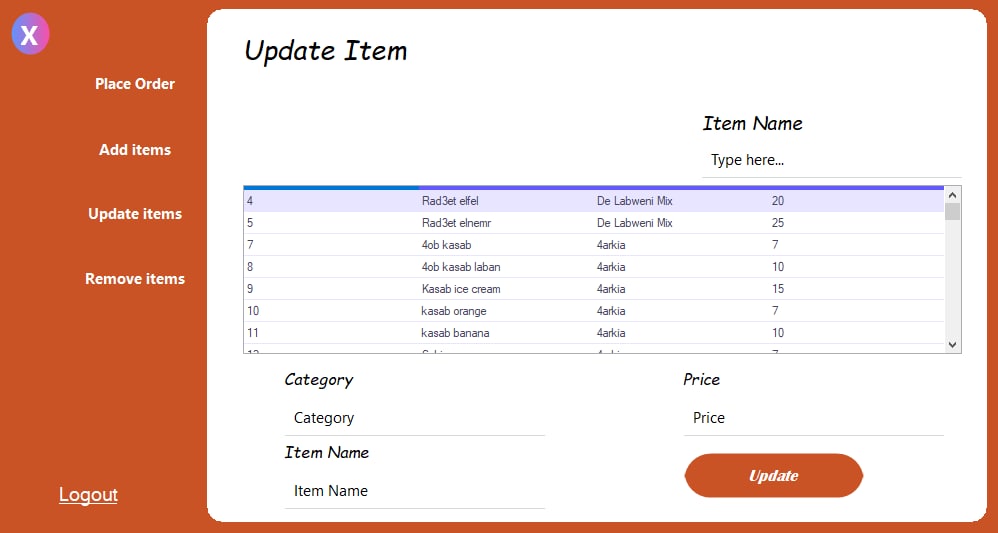
**17.4. Add new item page:**

This page allows the user to add a new item by indicating its category, name, and price.



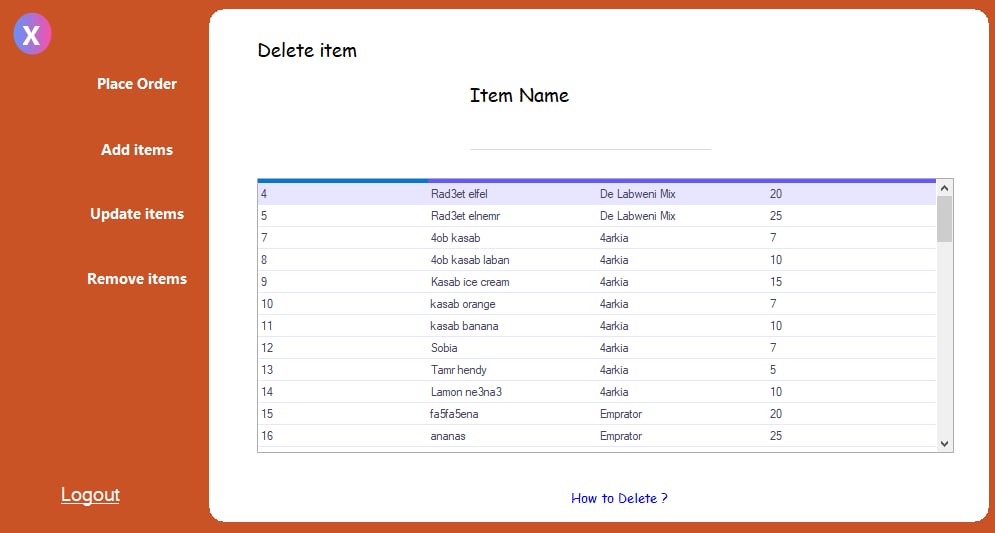
**17.5. Update item page:**

This page allows the user to update items by typing their names, categories, and prices.



**17.6. Delete item page:**

This page allows the user to delete items by writing their names.



**18.0 Glossary:**

|  |  |
| --- | --- |
| Term | Definition |
| customer | **The one that can log into the system as a guest, is allowed to place orders only** |
| Cashier | **One can log in as an admin he can place an order, update the menu, and confirm payment** |
| item | **It is the drink that is tracked by the system**  **Each item has a name, price, and category belonging to** |
| category | **It is a group of items that share something in common** |
| Database | **Collection of information monitored by the system** |
| Stakeholder | **Anyone who has an interest in the project** |
| user | **Cashier or customer** |
| Software requirement specification | **The document that functions and constraints that the system requires to operation** |

**19.0. System Modules:**

**19.1. Admin:**

* + - **Place order**
      * **Print receipt**
      * **Cancel order**
    - **Add item**
    - **Update item**
    - **Delete item**

**19.2. Cashier:**

* **Place order**
  + **Print receipt**
  + **Cancel order**

**INDEX**

A

add-items 3, 4, 7, 9, 10, 11, 16, 17, 22

Admin-Mode 3, 4, 5, 8, 9, 11, 13, 16, 17, 22, 24

C

Cashier-Mode 3, 4, 5, 7, 8, 16, 17

J

Juice-Shop 1, 2, 3

L

Log 3, 7, 8, 16, 17, 24

R

Receipt 3, 5, 7, 22

S

Software-System 1, 3, 4, 5, 6, 7, 16, 17, 21, 24

SQL-Server 4