

## **Phase I: Identification and Analysis**

Nicole Garcia (nswanso6)

### **1. Identify Actors and Use-Cases:**

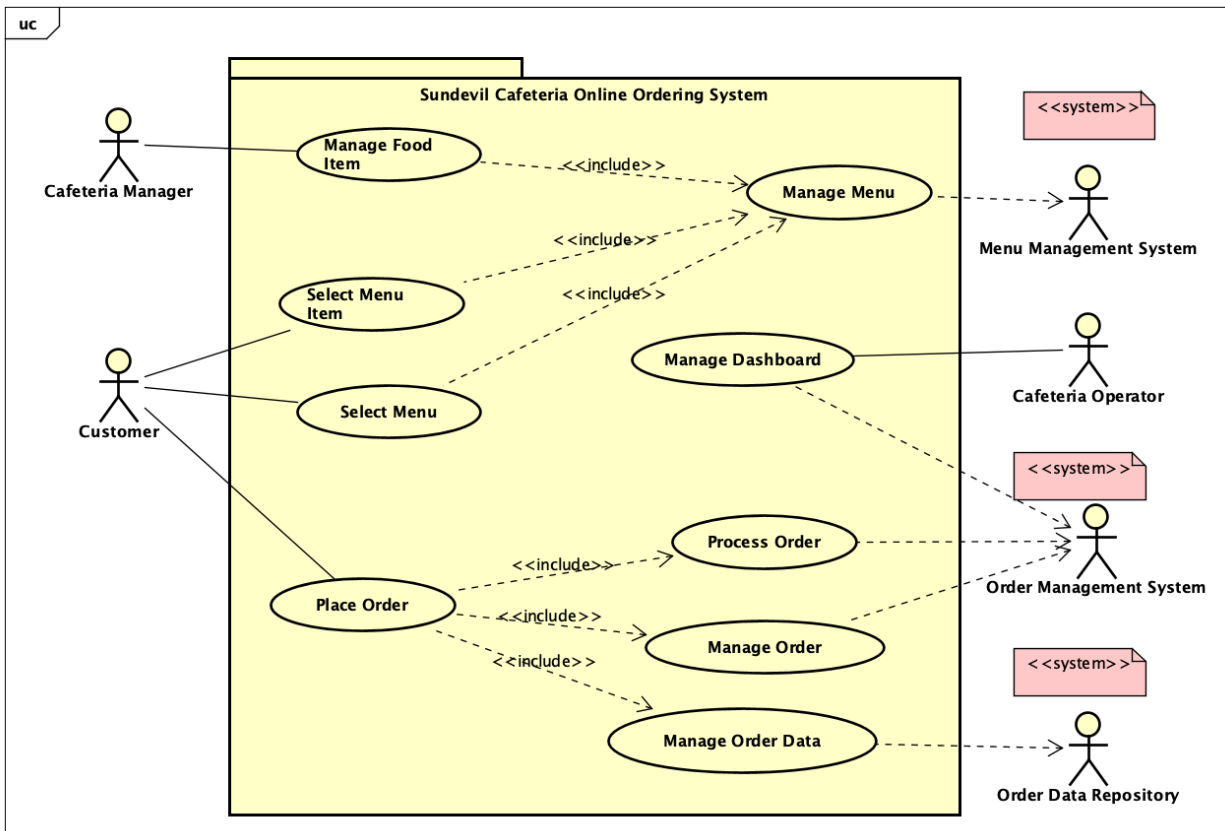
#### **Actors:**

Customer  
Cafeteria Operator  
Cafeteria Manager  
Order Management System  
Menu Management System  
Order Data Repository

#### **Use Cases:**

Manage Food Item  
Select Menu Item  
Select Menu  
Place Order  
Manage Menu  
Manage Dashboard  
Process Order  
Manage Order  
Manage Order Data

## 2. Draw the Use-Case Diagram: Use Case Diagram:



### Description and intuition of the use cases above:

#### Cafeteria Manager Use Cases:

- Manage Food Item: The Cafeteria Manager can manage food items by adding or removing items from the menu. The information is then saved to the Menu Management System which then updates the corresponding menu with the changes.
- Manage Order Data: The Cafeteria Manager is responsible for the Order Data Repository. They make sure orders are being recorded properly.
- Manage Menu: when the Cafeteria Manager updates food items, the corresponding menu gets updated by the Menu Management System.

#### Customer Use Cases:

- Select Menu: Customers can select the breakfast, lunch, dinner, and beverages menu and then browse the current available menu items available to order from that menu. This involves querying the Menu Management System.
- Select Menu Item: Customers can select menu items available on the menu during the ordering process.

- Place Order: Customers can place orders providing their name and the desired menu items. This order is recorded in the Order Data Repository and queued for meal preparation in the Order Management System.

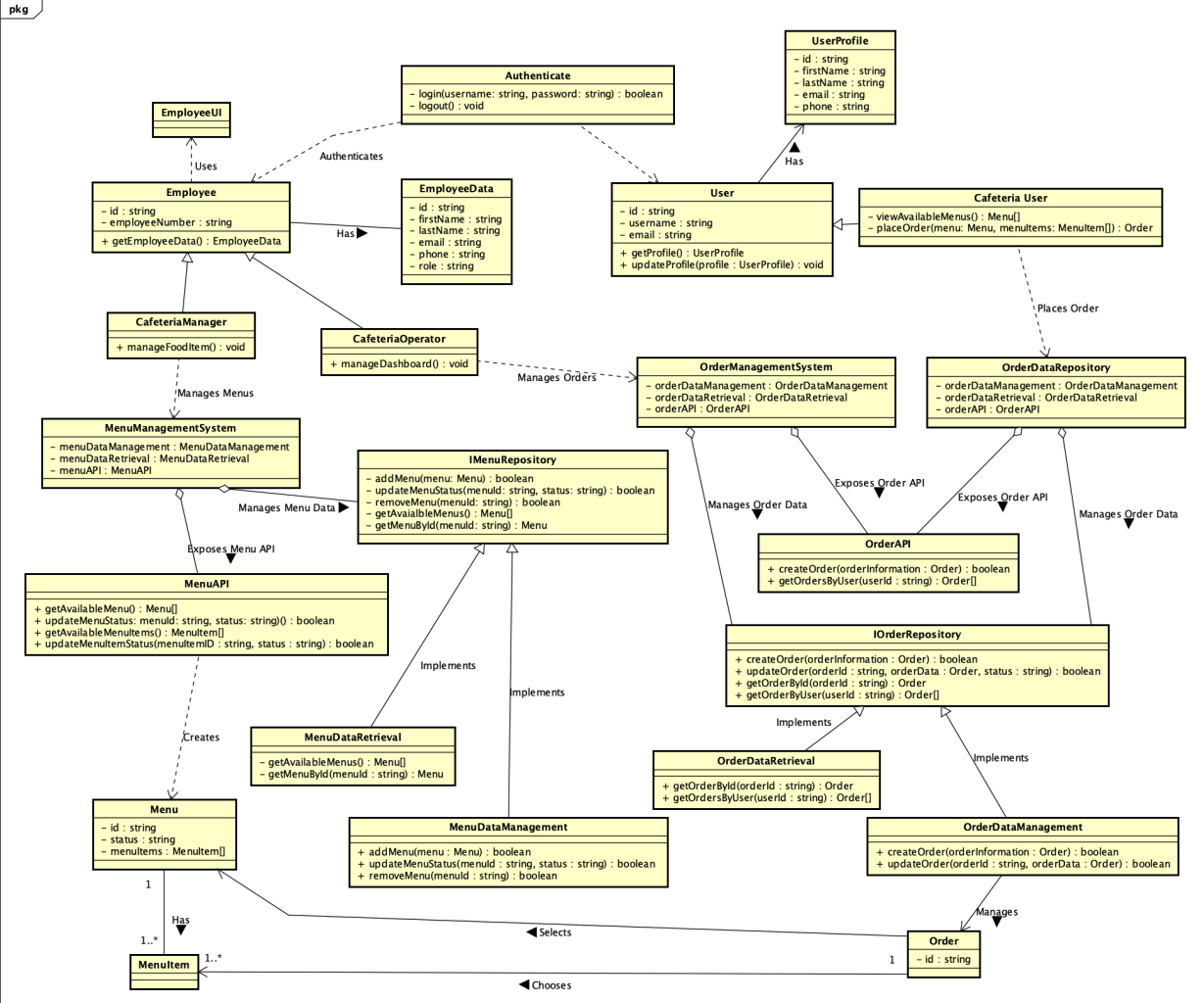
#### Cafeteria Operator Use Cases:

- Manage Dashboard: Cafeteria Operators can access the dashboard to view and monitor the orders. When orders are initially received into the Order Management System, they are queued for meal preparation. At that time, the Order Management System prints off a receipt for the Cafeteria Operator, who then initiates the meal preparation process. The Cafeteria Operator manages the status of the order as it progresses through the various stages. This data is saved in the Order Data Repository.
- Process Order: The Cafeteria Operator receives a receipt of a new order from the Order Management System once it has been processed.
- Manage Order: The Cafeteria Operator updates the status of the order in the Dashboard as it progresses through the stages of preparation in the Order Management System.

#### Systems:

- Order Management System: The system keeps track of orders and their status'. It interacts with the Order Management System.
- Menu Management System: This system keeps track of and manages the menus and menu items. Menus are updated based on changes made by the Cafeteria Manager.
- Order Data Repository: This database stores all order information, including menu items, customer details, payment information, and order status. It is interfaced by the Customer, Cafeteria Operator, and Order Management Systems and is accessed by the web application.

### Class Diagram:



### CRC Cards:

## Class: Authentication

**Description:** Handles user authentication and authorization

## Responsibility:

- Log users in
- Log users out

**Collaborator:**

- User
- Employee

**Class:** User

**Description:** Represents user in system

**Responsibility:**

- Get user profile
- Update user profile

**Collaborator:**

- UserProfile
- 

**Class:** UserProfile

**Description:** Represents user profile data

**Responsibility:**

- Store user profile data

**Collaborator:**

- User
- 

**Class:** Customer

**Description:** Represents a cafeteria customer user

**Responsibility:**

- Select menu item
- Select menu
- Place order

**Collaborator:**

- OrderDataRepository
  - OrderAPI
- 

**Class:** Employee

**Description:** Represents an employee user

**Responsibility:**

- Get employee data

**Collaborator:**

- EmployeeData
- 

**Class:** EmployeeData

**Description:** Represents employee data

**Responsibility:**

- Store employee data

**Collaborator:**

- Employee
-

**Class:** CafeteriaManager

**Description:** Represents a cafeteria manager employee

**Responsibility:**

- Manage food item

**Collaborator:**

- MenuManagementSystem
  - MenuAPI
  - OrderDataRepository
- 

**Class:** CafeteriaOperator

**Description:** Represents a cafeteria operator employee

**Responsibility:**

- Manage dashboard

**Collaborator:**

- OrderManagementSystem
  - OrderAPI
  - OrderDataRepository
- 

**Class:** MenuManagementSystem

**Description:** Manages the menu items and related operations

**Responsibility:**

- Manage menu data
- Retrieve menu data
- Expose menu API

**Collaborator:**

- MenuDataManagement
  - MenuDataRetrieval
  - MenuAPI
- 

**Class:** MenuDataManagement

**Description:** Handles menu data management operations

**Responsibility:**

- Add menu
- Update menu
- Remove menu

**Collaborator:**

- Menu
-

**Class:** MenuDataRetrieval

**Description:** Retrieves menu data

**Responsibility:**

- Get menu
- Get menu by ID

**Collaborator:**

- Menu
- 

**Class:** MenuAPI

**Description:** Provides an API for menu related operations

**Responsibility:**

- Get available menu
- Update menu status
- Get available menu items
- Update menu item status

**Collaborator:**

- EmployeeUI
- 

**Class:** Menu

**Description:** Represents a menu in the system

**Responsibility:**

- Store menu data

**Collaborator:**

- MenuDataManagementSystem
  - Order
- 

**Class:** MenuItem

**Description:** Represents a menu item in the system

**Responsibility:**

- Store menu item data

**Collaborator:**

- Menu
  - MenuDataManagementSystem
  - Order
-

**Class:** OrderDataRepository

**Description:** Manages order data and related operations

**Responsibility:**

- Manage order data
- Retrieve order data
- Expose order API

**Collaborator:**

- OrderDataManagement
  - OrderDataRetrieval
  - OrderAPI
- 

**Class:** OrderDataManagement

**Description:** Handles order data management operations

**Responsibility:**

- Create order
- Update order

**Collaborator:**

- Order
- 

**Class:** OrderDataRetrieval

**Description:** Retrieves order data

**Responsibility:**

- Get order by ID
- Get order by user

**Collaborator:**

- Order
- 

**Class:** OrderAPI

**Description:** Provides an API for order related operations

**Responsibility:**

- Create orders
- Get orders by user

**Collaborator:**

- EmployeeUI
-



**Class:** Order

**Description:** Represents an order in the system

**Responsibility:**

- Store order data

**Collaborator:**

- OrderDataManagement
  - Menu
  - MenuItem
- 

**Class:** UserUI

**Description:** Represents the user interface for regular users

**Responsibility:**

- Display user profile
- Update user profile

**Collaborator:**

- User
- 

**Class:** EmployeeUI

**Description:** Represents the user interface for employees

**Responsibility:**

- Display employee information
- Display available menu and menu items
- Display orders by user

**Collaborator:**

- Employee
  - MenuAPI
  - OrderAPI
-