

## Lab 5: Question 3

### The Canteen Automation Project

---

#### Group: 31

Hemang Nakarani :	201801158
Dharmin Hirapara :	201801135
Shah Yash Pareshkumar :	201801131
Sudarshan Kundnani :	201801140
Shantanu Tyagi :	201801015
Kiran Ravuri :	201801421
Srinivas Talnikar :	201801406
Sudhanshu Mishra :	201801114
Harshil Joshi :	201801022
Urvish Pandya :	201801170

---

#### 1. Stakeholders and Users:

##### i. Canteen Staff : [User,Actor]

- The manager and the waiters are the canteen staff. They will manage the account book of all customers. They can update the Menu anytime they want. They will accept the food order of customers. They will update the status of the food. They can log in and logout from the system.

##### ii. Customers : [User,Actor]

- They can search for products, order items from the canteens. They can log in, logout from the system. They can update their profile and passwords. They can give the rating and the overview of the parched item.

##### iii. Admin : [User,Actor]

- They will provide the statistical data such as their monthly Turnover, Profit/Loss, Overall rating, and reviews to canteen managers and will be in charge of authorizing valid(registered) users.

##### iv. Tracker database: [Actor]

- It will consist of whole data of total orders and payment transactions, and it will provide corresponding data to the users and canteen owners.

v. Payment GateWay: [Actor]

- This will help for online payment transactions from customers' bank account to Canteen Bank Account.

## **2. Features exercised by various users are:**

- A. Authentication Token System: This will take care that the user using the system is verified. Customers, canteen staff, an admin will have different authentication to separate their access over their network.
- B. Update Profile: The functionality ensures the user can update their profiles and preferences successfully. This can be done through the edit profile tab in the app UI. Managers and users can update their profile photos, names, and contact details. OTP functionality will be implemented to authorize the changes.
- C. Change Password: The user can also change their login passwords from the profile tab in the app UI. The user will be asked to authorize himself through OTP functionality.
- D. Search Food Items: Customers can search for different products before ordering the food. This can be done from the trending list, manually navigating through the menu, or using the search bar to filter the item or search for its name.
- E. Showing Frequently Ordered Food Items: Based on what food items are being ordered currently or what food items are popular at that time of the day,
- F. Place Orders: This functionality renders customers the ability to order through the application. Order carts will be maintained which can be edited. After selecting the order's payment mode, the details will be notified to the canteen manager who will accept and update the order status.
- G. Order Food: This functionality ensures the ability to order through the application. The Order cart will be maintained, which can, later on, be edited, payment mode selection, and placing the order which the canteen staff will confirm.
- H. Order Status: After the customer has placed the order, Canteen staff can accept it. Eventually, it can be put on a waiting or cooking list and later be updated to cooked status and delivered respectively. All the changes will be reflected on the customer's side.

- I. Accept / Update Order: The canteen manager will be notified of any new order being requested by a customer and will now be prompted to accept the order and update the order status.
- J. Notifications: A notifications page would be provided. Both manager and customer will have their personalized reports. Coupon codes and new offers will be notified to the customers, and managers will receive notifications about the orders placed and order cooking status.
- K. Manage User Accounts and Payments: The manager will be provided with a feature to access the record of payments done by users and the remaining dues of users.
- L. Add new items: Managers will have access to the database so they can update the items in the menu, which will then be reflected in the system.
- M. Manage database: Admin will have access to the entire database and can make the changes if necessary. Take care of the corruption of the database.
- N. Manage Users: Admin can view all the users and their profiles and make changes if necessary, approve new canteen requests, and provides then canteen account credentials.
- O. Manage System: Admin can manage the system and take care of any bugs, customer reports, or faults if any, and take considerations of the overall health of the system.
- P. Make Payment: With the use of this functionality, customers will be able to make payments through online mode for their food. We will add the option for coupons based on order history, if available.
- Q. Online Payment: Customers will be provided with the facility to make the payment online, using standard available payment tools. The cost will be instant, and the updates regarding the amount will be notified both at the customer's and manager's end.
- R. Wallet-based payment: Since each order's amount is not very high, customers can make payments based on credit. In this, they need to pay the total credits collected at the end of the month to the corresponding canteen counter. Canteen counters will log the credit collected for each customer.

### 3. Non-functional requirements:

1. Service should be available at all times.
2. The system should be able to handle multiple users simultaneously searching for books. At least 100 users can access the system together.
3. Passwords should be encrypted and not in plain text for security purposes.
4. Affiliation with Institute's database of institute members.
5. Database for users: For maintaining the record of registered/ valid users.
6. Different users cannot have the same name or password.

### 4. User Interface

For the system we are developing, we will have two separate user interfaces namely for the Customer and Manager:-

**For The Customer:** - Post logging into the system with the appropriate username and password, the customer would be presented with a screen that presents the user with the counters available and are taking orders, a curated list of options based on the user's previous order and the food items ordered the most. In the top left, there will be an option for the user to open a side menu where they will be presented with options such as settings, their Account Book(which contains a list of their expenses and dues), and the option to sign out.

Once their food items have been ordered, the user will be able to see the status of their order and will be getting live updates.

**For the Manager:-** Post logging into the system the manager will be presented with The screen where they will be able to accept orders, manage user payment credits/accounts, add new menu items to the counter they are in charge of, receive orders, and send notifications to the user.

Once an order is accepted/ received, the manager can send live updates and notifications on the status of their food and let them know once it's done.

### 5. Open Issues are:

- A. Scheduling Orders: The canteen counters will receive multiple orders with different cooking times and different cooks cooking different types of dishes. In this case, we need to find out which dish should be made first and which later even if the dish being made later was ordered before the dish being made first.

- B. Non DAIICT Customers: At times, the college canteen has customers that do not have an institutional ID, in this case, we need to ensure a guest account with limited functionality so that these customers can also order the food.
- C. Suppose if the food is prepared and the customer opted for a cash payment but never came to collect the food. In that case, the canteen staff should have an option to charge the customer's account on their next order.

## 6. Use Case Diagram:

*Attached At the end of the PDF*

## 7. Requirements/ Needs/ Objectives:

**Purpose and Need:** The Cafeteria management system is a new system that replaces the current manual processes for ordering and picking up food by queuing up in a line in the canteen. But with the help of this software, you just have to follow a very simple process to order your food online and canteen owners can easily manage the orders thus eliminating the need for standing in queues and making the process automated and more efficient.

**Various useful features,** like discussed above, will be provided in order to ensure smooth and easy functioning. It will also allow the canteen managers to manage their orders and keep a track of payments. Customers will be shown the menus of different canteen counters from which they can select whatever they want to, followed by various payment methods. On the canteen side, the staff can easily access the orders and update their menus, etc. This system aims to accelerate customer orders and customer order systems used by employees to accept customer orders.

The purpose of the system is to develop a simple Canteen automation Software and implement it, which later will be used for a web-based application. Our objective is to make a platform-independent application to maintain a database of all orders ordered from various sources and all the different services required by each of them. Established canteen automation practices should provide the needed connectivity and accountability between those two operational units, and when managed properly, enhances the effectiveness by eliminating the lunchtime chaos at the canteen and simplifying the whole process by robotizing each and every step of canteen operations.

**Hardware System Requirements:**

- i5 Processor-Based Computer or higher
- Memory: 8 GB RAM
- Hard Drive: 50 GB
- Internet Connection

**Software Requirements:**

- Linux 18.00 / Windows 10 / MacOS Sierra
- Visual Studio Code [Frontend Development]
- IntelliJ IDEA [Backend Development]
- PgAdmin (Installed PostgreSQL)
- Postman For API Testing

**Development Team Requirement :**

- UI/UX designing Team for creating designs and assets for the application.
- FrontEnd Engineers Team: Efficient Frontend developers who know Javascript and React Js
- Backend Engineers Team: Java SpringBoot developers who can code stable, scalable, and large systems, Also we need database designers who can design databases as per user requirements.

