ADDIS ABABA UNIVERSITY, BAHIRDAR UNIVERSITY

MINISTRY OF INNOVATION AND TECHNOLOGY (MINT)

Use Cases and User Testings Cases for Mint Cafe System

Authors:

- Alexander Kiros
- Berihun Tessema
- Birehanu Assmamaw
- Melktekrstos Mengiste
- Ruth Dereje

Addis Ababa, Ethiopia. Monday 25th September, 2023

Contents

1 Introduction **1**

1.1 Mint Cafe App: Overview 1

2 Use case and Testing for the Admin Desktop App **3**

2.1 Menu Management 3

2.1.1 Use Case: Adding Menu Item 3

2.1.2 Use Case: Deleting Menu Item 4

2.1.3 Use Case: Updating Menu Item 5

2.1.4 Use Case: Set Availability of Menu Item 6

2.1.5 Use Case: Viewing Menu Items by Filtering 7

2.2 Inventory Management 8

2.2.1 Use Case: Request for stock 8

2.2.2 Use Case: Request for Ingredient Purchase 9

2.2.3 Use Case: Approval for stock request 10

2.2.4 Use Case: Approval for ingredient request 11

2.2.5 Use Case: View approved stock requests 12

2.2.6 Use Case: View approved ingredient requests 13

2.2.7 Use Case: Registration for inventory 13

2.3 Order Management 14

2.3.1 Use Case: Buy Food 15
2.3.2 Use Case: Updating Order Status 16
2.3.3 Use Case: Viewing Orders 17
2.4 Employee Management 17
2.4.1 UseCase: Add Employee 18
2.4.2 UseCase: View Employee 19
2.4.3 UseCase: Update Employee 19
2.4.4 UseCase: Delete Employee 20
2.5 Report Management 21
2.5.1 UseCase: View Report 21
2.5.2 UseCase: ViewEmployee 22
2.5.3 UseCase: Update Employee 22
2.5.4 UseCase: Delete Employee 23
2.6 Payment Management 23
2.6.1 UseCase: Deposit Money 24
2.6.2 UseCase: Withdraw Money 25
2.6.3 UseCase: Refund Money 25
2.7 Constraint Management 26
2.7.1 UseCase: Set Constraints 26

3 Use Case and Testing for the Mobile Application 28

3.1 Authentication Management 28
3.1.1 Use Case: Employee Login 28

3.1.2 Use Case: Employee Sign-up 29
3.1.3 Use Case: OTP Verification 29
3.1.4 Use Case: Password Reset 30
3.2 Mobile Menu Management 31
3.2.1 Use Case: View Available Food Items 31
3.2.2 Use Case: Add Foods to Cart 32
3.3 Order Management 33
3.3.1 Use Case: Order Placement 33
3.4 Account Activity Management 34
3.4.1 Use Case: View Profile 34
3.4.2 Use Case: View Transaction History 35
3.4.3 Use Case: View Order History 36

1. Introduction

User Acceptance Testing (UAT) stands as a pivotal phase within software development. During this stage, end-users or representatives from the user community take on the role of validating the software's functionality, usability, and alignment with their specific requirements. Typically conducted post-system testing but before the software's deployment to the production environment, UAT holds the crucial role of confirming that the software effectively meets business needs and user expectations. The primary objective of UAT is to ensure that the software aligns with business requirements and adequately addresses the end-users' needs. This phase allows users to assess how the software behaves in real-world scenarios and enables them to provide feedback prior to the software's final release. UAT plays a crucial role in quality control by identifying any flaws, usability issues, or deviations from anticipated performance that could have been missed in preceding testing stages.

1.1. Mint Cafe App: Overview

The Cafe Management System is a software solution designed to streamline cafe operations. Through a mobile app, customers can place orders, handle payments, and manage their accounts. Cafe staff use the system to manage orders, update order status, and oversee inventory. The entire system is overseen by a cafe admin responsible for menus, inventory, payments, and customer accounts. This system involves various actors: Customers, Cafe Staff, Cafe Admin, and Cafe Manager. Customers use it to place orders, manage accounts, and make payments. Cafe Staff handle order updates, inventory management, and order viewing. Cafe Admin takes care of menus, inventory, payments, and customer accounts. The Central Server acts as the system's core, processing orders, storing data, and enabling real-time communication between the mobile and desktop apps. The Payment Gateway is an ex-

ternal entity responsible for secure payment processing through the mobile app. The system offers multiple use cases, including Order Placement, Order Management, Menu Management, User Authentication, Report Viewing, Inventory Management, Payment Processing, and View employee information. Each use case is performed by specific actors based on their roles. Customers can place orders, manage their accounts, and make payments via the mobile app, while Cafe Staff update order status, manage inventory, and access orders. Cafe Admin handles menu updates, inventory management, payment tracking, and customer account management. In summary, the Cafe Management System is a comprehensive solution for optimizing cafe operations. It provides a variety of features and tools to streamline tasks and enhance efficiency. Additionally, it offers a user-friendly interface for both customers and staff.

2. Use case and Testing for the Admin Desktop App

2.1. Menu Management

Menu Management is a fundamental component of a cafe management system, ensuring that authorized personnel can efficiently manage the menu items and offerings. It is responsible for the overall menu management including, adding, updating, deleting menu items, uploading images, setting prices and enabling/disabling based on the availability.

Use cases for menu management:

- Adding Menu Item
- Updating Menu Item
- Deleting Menu Item
- Setting available/unavailable menu items
- Viewing Menu Items

2.1.1. Use Case: Adding Menu Item

1. **Test Objective:** To ensure that Cafe Managers can successfully add a new food item to the cafe's menu using the desktop app.
2. **Test Step:**
    - Log in: The cafe manager opens the desktop app and logs in using their valid credentials.

- Access Menu Entry: The cafe manager navigates to the ”Menu Entry” section within the app.
- Fill Menu Item Details: The app presents a form for the cafe manager to enter the details of the new food item. The cafe manager enters the following information: name, description, price for Employee, price for guest, price for department, fasting option, drink option, and available amount.
- Add Menu Item: The cafe manager submits the form to add the new menu item.

3. **Expected Result:**

- A new menu item is successfully added to the cafe menu, providing more options for customers.
- The cafe manager can view and manage the newly added item in the ”Menu Management” section of the desktop app.
- Employees can access the menu item in their mobile app if it’s available for ordering.

2.1.2. Use Case: Deleting Menu Item

1. **Test Objective:** To verify that Cafe Managers can successfully delete a food item from the cafe’s menu using the desktop app.
2. **Test Step:**

- Log in: The cafe manager opens the desktop app and logs in using their valid credentials.
- Access Food Menu: The cafe manager navigates to the ”Food Menu” section within the app.
- List of Existing Items: The app displays the list of existing menu items.Select Menu Item: The cafe manager selects the menu item they want to delete from the list.
- Confirm Deletion: The app prompts the cafe manager to confirm the deletion of the selected menu item and then the café manager confirms the deletion by clicking the “confirm” button.

3. **Expected Result:**
    - The selected menu item is successfully removed from the cafe's menu.
    - The cafe manager can no longer view or manage the deleted item in the "Food Menu" section of the desktop app.
    - Employees will no longer see the deleted menu item in their mobile app

2.1.3. Use Case: Updating Menu Item

1. **Test Objective:** To verify that Cafe Managers can successfully update a food item on the cafe's menu using the desktop app.
2. **Test Step:**
    - Log in: The cafe manager opens the desktop app and logs in using their valid credentials.
    - Access Food Menu: The cafe manager navigates to the "Food Menu" section within the app.
    - List of Existing Items: The app displays the list of existing menu items.
    - Select Menu Item: The cafe manager selects the menu item they want to update from the list.
    - Navigate to Update Form: The cafe manager clicks the "Update" button, which navigates them to the menu entry section. The app presents a form pre-filled with the current details of the selected menu item, including its name, description, price for Employee, price for guest, price for department, fasting option, drink option, and available amount.
    - Modify Information: The cafe manager modifies the desired fields in the form to update the menu item. They can change the name, description, prices, and other relevant details.
    - Submit Updated Form: The cafe manager submits the updated form.

3. **Expected Result:**
    - The selected menu item is successfully updated with the new details in the cafe's menu.
    - The cafe manager can view and manage the updated item in the "Food Menu" section of the desktop app.
    - Employees will see the updated menu item with its revised information in their mobile app.

2.1.4. Use Case: Set Availability of Menu Item

1. **Test Objective:** To verify that Cafe Managers can successfully set the availability of a food item on the cafe's menu using the desktop app.
2. **Test Step:**
    - Log in: The cafe manager opens the desktop app and logs in using their valid credentials.
    - Access Food Menu: The cafe manager navigates to the "Food Menu" section within the app
    - List of Existing Items: The app displays the list of existing menu items.
    - Select Menu Item: The cafe manager selects the menu item for which they want to set the availability from the list.
    - Set Availability: The cafe manager locates the availability setting for the menu item and adjusts it to either "Available" or "Unavailable" based on the item's amount.
    - Save Changes: The cafe manager saves the changes to update the availability of the menu item by clicking the "set" button.
3. **Expected Result:**
    - The availability of the menu item is successfully updated in the cafe's menu based on the cafe manager's selection.
    - The cafe manager can view and manage the updated item's availability in the "Food Menu" section of the desktop app.

- Employees will see the updated availability status of the menu item in their mobile app to place their orders.
- Cashiers will see the updated availability status of the menu item in their desktop app in the buy food section when taking customer orders.

2.1.5. Use Case: Viewing Menu Items by Filtering

1. **Test Objective:** To verify that Cafe Managers can successfully view menu items by applying filters in the desktop app.
2. **Test Step:**
   - Log in: The cafe manager opens the desktop app and logs in using their valid credentials.
   - Access Food Menu: The cafe manager navigates to the ”Food Menu” section within the app.
   - List of Existing Items: The app displays the list of existing menu items.
   - Locate Filter Options: The cafe manager locates the filter options for selecting filters.
   - Select Filter Option: The cafe manager selects the desired filter option from the available choices, such as ”All,” ”Lunch,” ”Breakfast,” ”Drink,” ”Fasting,” ”Non-fasting,” ”Drinks,” or ”Available”.
   - Apply Filter: The app applies the selected filter and updates the menu items displayed accordingly.Browse Filtered Items: The cafe manager can browse through the filtered menu items and view their details, including name, description, image, and availability status.
3. **Expected Result:**
   - The cafe manager can view menu items based on the selected filter, narrowing down the displayed items to the desired category or availability status
   - The cafe manager can browse through the filtered menu items and view their details.

- Employees and cashiers will see the same filtered menu items in their mobile app and desktop app when taking customer orders.

2.2. Inventory Management

Inventory Management encompasses various essential functions, including requesting stock, requesting ingredients, stock approval, ingredient approval, inventory registration, and maintaining an inventory list. Additionally, there is a dedicated page where pending requests must be approved. These functionalities collectively ensure efficient control and organization of the inventory and its associated processes.

Use cases for inventory management:

- Request for stock
- Request for ingredient purchase
- Approval for stock request
- Approval for ingredient request
- View approved for stock request
- View approved for ingredient request
- Registration for inventory
- View inventory list

2.2.1. Use Case: Request for stock

1. **Test Objective:** To ensure that Cafe Mangers can successfully submit a stock request form in the desktop app for items to be taken out from the store.
2. **Test Step:**

- Log in: The Cafe Manger opens the desktop app and logs in using their valid credentials.
- Access Stock Request: The Cafe Manger navigates to the ”Stock Request” section within the app.
- Fill Stock Request Form: The app displays a stock request form. The Cafe Manger enters the following information:
  - Item Name: The name of the item for which stock is being requested.
  - Measured In: The unit of measurement for the requested stock (e.g., kilograms, liters).
  - Quantity: The desired quantity of the item to be requested.
  - Requested By: The name or identifier of the person submitting the stock request.
- Submit Form: The Cafe Manger submits the completed stock request form.
- Add Menu Item: The cafe manager submits the form to add the new menu item.

3. **Expected Result:**
   - The stock request form is successfully submitted in the desktop app.
   - The stock request is recorded and registered in the app’s system.
   - Cafe committees can review and process the stock request.

2.2.2. Use Case: Request for Ingredient Purchase

1. **Test Objective:** To ensure that store keepers can successfully submit ingredient purchase requests in the desktop app
2. **Test Step:**
   - Log in: The store keeper opens the desktop app and logs in using valid credentials.
   - Access Ingredient Request: The store keeper navigates to the ”Ingredient Request” section.

- Fill Stock Request Form: The app displays the ingredient purchase request form. The store keeper fills in all required fields correctly such as:
  - Item Name: The name of the item for which stock is being requested.
  - Measured In: The unit of measurement for the requested stock (e.g., kilograms, liters).
  - Quantity: The desired quantity of the item to be requested.
  - Item Price: The price per unit of the ingredient.
  - Total Price: The calculated total price for the requested quantity of the ingredient.
  - Total Price in Words: The total price written in words.
  - Requested By: The name or identifier of the person submitting the ingredient purchase request.
  - Recommendation: Any additional comments or recommendations related to the ingredient purchase request.
- Submit Form: The store keeper submits the form.

3. **Expected Result:**
   - The ingredient purchase request form is successfully submitted.
   - The request is recorded in the app's system.
   - The request is routed to the approval page for cafe committee review.

2.2.3. Use Case: Approval for stock request

1. **Test Objective:** To ensure that committee members can successfully review and make decisions on stock requests in the desktop app.
2. **Test Step:**
   - Log in: Committee members open the desktop app and log in using valid credentials.
   - Access Ingredient Approval: Committee members navigate to the "Ingredient Approval" section.

- List of Pending stock requests : The app displays a list of pending stock requests.
- Select request: Committee members select a stock request to review.
- Committee members review the request details and make a decision (approve or reject).
- The app updates the request status based on the committee's decision. Committee members repeat the process for other pending stock requests, if any.

3. **Expected Result:**
   - Committee members can efficiently review and make decisions on stock requests.
   - The app correctly updates the status of each stock request based on the committee's decisions.
   - Cafe Mangers receive updates regarding the status of their stock requests.

2.2.4. Use Case: Approval for ingredient request

1. **Test Objective:** To ensure that committee members can successfully review and make decisions on ingredient requests in the desktop app.
2. **Test Step:**
   - Log in: Committee members open the desktop app and log in using valid credentials.
   - Access Ingredient Approval: Committee members navigate to the "Ingredient Approval" section.
   - List of Pending stock requests: The app displays a list of pending ingredient requests.
   - Select request: Committee members select an ingredient request to review.
   - Committee members review the request details and make a decision (approve or reject).

- The app updates the request status based on the committee's decision.
- Committee members repeat the process for other pending ingredient requests, if any.

3. **Expected Result:**
   - Committee members can efficiently review and make decisions on ingredient requests.
   - The app correctly updates the status of each stock request based on the committee's decisions.
   - Storekeepers receive notifications regarding the status of their ingredient requests.

2.2.5. Use Case: View approved stock requests

1. **Test Objective:** To ensure that Cafe Mangers can successfully receive updates and access approved stock requests in the desktop app.
2. **Test Step:**
   - Log in: The Cafe Manger opens the desktop app and logs in using valid credentials.
   - Access approved stock requests: The Cafe Manger navigates to the "Approved Stock Requests" section.
   - List of approved stock requests: The app displays a list of approved stock requests.
   - The Cafe Manger views the approved stock requests.
3. **Expected Result:**
   - The Cafe Manger can efficiently access and view the list of approved stock requests.
   - The app provides accurate and up-to-date updates regarding the status of approved stock requests for the Cafe Manger.

2.2.6. Use Case: View approved ingredient requests

1. **Test Objective:** To ensure that store keepers can successfully receive updates and access approved ingredient requests in the desktop app.
2. **Test Step:**
   - Log in: The store keeper opens the desktop app and logs in using valid credentials.
   - Access approved ingredient requests: The store keeper navigates to the ”Approved Ingredient Requests” section.
   - List of approved ingredient requests: The app displays a list of approved ingredient requests.
   - The storekeeper views the approved ingredient requests.
3. **Expected Result:**
   - The storekeepers can efficiently access and view the list of approved ingredient requests.
   - The app provides accurate and up-to-date updates regarding the status of approved ingredient requests for the store keeper.

2.2.7. Use Case: Registration for inventory

1. **Test Objective:** To ensure that store keepers can successfully register bought inventory items, update expenses, and add items to the inventory list in the desktop app.
2. **Test Step:**
   - Log in: The store keeper opens the desktop app and logs in using valid credentials.
   - Access ingredient entry: The store keeper navigates to the ”Ingredient Entry” section.
   - Fill Inventory Entry Form: The app displays an inventory entry form. The store keeper enters the following information:
     - Item name

  - Quantity
  - Price
  - Measured in
  - Approved by
- Submit Form: The store keeper submits the completed inventory entry form.

3. **Expected Result:**
   - The store keeper can efficiently register bought inventory items and update expenses.
   - The app correctly updates the inventory list with the newly added items and their quantities.
   - The app accurately calculates and maintains the expense total.
   - A confirmation message is displayed to confirm the successful registration of items and expense updates.

2.3. Order Management

Inventory Management encompasses various essential functions, including requesting stock, requesting ingredients, stock approval, ingredient approval, inventory registration, and maintaining an inventory list. Additionally, there is a dedicated page where pending requests must be approved. These functionalities collectively ensure efficient control and organization of the inventory and its associated processes.

Use cases for order management:

- Buy Food
- Update Order Status
- View Order

2.3.1. Use Case: Buy Food

Scenario I : Buy Food for an Employee

1. **Test Objective:** Order Food from the admin for those that are unable to do so for whatever reasons.
2. **Test Step:**
    - Log in to the application as a cashier and navigate to the "Buy Food" feature.
    - Select who to buy as an employee by specifying the id.
    - Select multiple menu items, adjust quantities, and verify that the total price is calculated correctly.
    - Enter the employee ID and place an order.
    - Verify that the order is successful by cross checking that it appeared in the orders section.
    - Check that the total price is correctly deducted from the employee's account.
3. **Expected Result:**
    - The application allows you to select menu items.
    - The total price is calculated accurately and order placement is successful.
    - The employee's account balance is updated accordingly.

Scenario II : Buy Food for a Guest

1. **Test Objective:** To Buy food for guests visiting the ministry, as setting up the mobile app for them is not going to be a pleasant experience.
2. **Test Step:**
    - Log in to the application as a cashier and navigate to the "Buy Food" feature.

- Select who to buy as a guest by specifying their name.
- Select multiple menu items, adjust quantities, and verify that the total price is calculated correctly and verify that the guest order is successfully placed and displayed as a billing in the billing section.
- Check that the total price is correctly displayed for the guest order.

3. **Expected Result:**
    - The application allows you to select menu items.
    - The total price is calculated accurately.
    - Guest order placement is successful.
    - The total price is correctly displayed for the guest order

2.3.2. Use Case: Updating Order Status

1. **Test Objective:** To verify that the cafe Manger can update the status of an order.
2. **Test Step:**
    - Cafe Manager logs into the system with valid cashier access credentials.
    - Cafe Manager navigates to the ”Orders” section.
    - Cafe Manager selects an order from the list.
    - The system displays the order details, including items, quantities and customer information.
    - Cafe Manager updates the order status for the selected order (e.g., from ”served” , ”Preparing” ,”Completed”).
    - Cafe Manager saves the updated order status and verifies that the order status is correctly updated in the system.
3. **Expected Result:**
    - The system displays order details for the selected order.
    - The cafe cashier can update the order status.
    - The updated order status is reflected correctly in the system.

2.3.3. Use Case: Viewing Orders

1. **Test Objective:** To verify that the cafe cashier (no edit privileges) and Cafe Manager can view orders based on time ranges.
2. **Test Step:**
   - Cafe cashier logs into the system with valid cashier access credentials.
   - Cafe cashier navigates to the ”View Orders” section.
   - Cafe cashier selects a specific time range for which they want to see the orders
   - The system displays a list of orders received on the selected date.
   - For each order in the list, the cafe cashier verifies that the following details are displayed:
     - Order ID (coupon)
     - Food Items ordered
     - Quantities for each food items
     - Employee name
     - Order Status
   - Cafe cashier confirms that the order list only includes orders from the selected time range.
3. **Expected Result:**
   - The system allows the cafe cashier to select a time range.
   - The list of orders for the selected time range are displayed, including order details and statuses.
   - The order list only contains orders from the selected time range.

2.4. Employee Management

Inventory Management encompasses various essential functions, including requesting stock, requesting ingredients, stock approval, ingredient approval,

inventory registration, and maintaining an inventory list. Additionally, there is a dedicated page where pending requests must be approved. These functionalities collectively ensure efficient control and organization of the inventory and its associated processes.

Use cases for employee management:

- addEmployee
- viewEmployee
- editEmployee
- deleteEmployee

2.4.1. UseCase: Add Employee

1. **Test Objective:** To verify that the cafe admin can successfully add a new employee with valid information.
2. **Test Step:**
   - Cafe admin logs into the system with valid admin access credentials.
   - Cafe admin navigates to the ”Add Employee” section.
   - Cafe admin enters valid employee information in the following fields:
     - Name
     - Department
     - Position
     - Email
   - Cafe admin clicks the ”Register” button to submit the employee information.
   - The system processes the registration request and displays a notification dialog with a success message.
3. **Expected Result:**
   - The cafe admin can enter valid employee information in all required fields.

- Upon clicking the ”Register” button, the system processes the registration request without errors.
- The system displays a notification with a success message indicating that the employee has been added successfully.

2.4.2. UseCase: View Employee

1. **Test Objective:** To verify that the system can display a list of registered employees.
2. **Test Step:**
   - Admin navigates to the ”Employee List” section.
   - The system displays a list of registered employees in a table, including the following details for each employee:
     - ID
     - Name
     - Email
     - Department
     - Position
     - Joined At (Date)
3. **Expected Result:**
   - Admin can view a list of registered employees with their details, which would prove to be useful to extract useful information.

2.4.3. UseCase: Update Employee

1. **Test Objective:** To verify that the system allows an admin to edit employee information.
2. **Test Step:**
   - Admin navigates to the ”Employee List” section.
   - The system displays a list of registered employees in a table.

- Admin clicks the ”Edit” icon/button for a specific employee.
- The system navigates to the ”Add Employee” page with the selected employee’s information pre-filled for editing.
- Admin makes changes to the employee’s information and saves the changes.
- The system updates the employee’s information.

3. **Expected Result:**
   - The admin can click the ”Edit” icon/button to edit an employee’s information.
   - The system allows the admin to make changes to the employee’s information and successfully updates it.

2.4.4. UseCase: Delete Employee

1. **Test Objective:** To verify that the system allows an admin to delete an employee when an employee leaves the ministry and is not entitled to use the cafe.
2. **Test Step:**
   - Admin navigates to the ”Employee List” section.
   - The system displays a list of registered employees in a table.
   - Admin clicks the ”Delete” icon/button for a specific employee.
   - The system displays a confirmation.
   - Admin clicks ”Yes” in the confirmation dialog and proceeds to deleting the selected employee from the list.
3. **Expected Result:**
   - The system deletes the employee from the list.

2.5. Report Management

Inventory Management encompasses various essential functions, including requesting stock, requesting ingredients, stock approval, ingredient approval, inventory registration, and maintaining an inventory list. Additionally, there is a dedicated page where pending requests must be approved. These functionalities collectively ensure efficient control and organization of the inventory and its associated processes.

Use cases for report management:

- View Report

2.5.1. UseCase: View Report

1. **Test Objective:** To verify that the system can display reports for a selected time range and provide detailed information about orders, revenue, expenses, and net earnings.
2. **Test Step:**
    - User navigates to the ”Report List” section within the system.
    - The system displays a section for generating reports
    - User selects a specific year from the ”Time Range” dropdown menu.
    - The system updates the report data based on the selected year and displays it in the table.
    - The table shows a row for each month within the selected year, containing information for each month’s expenses from the inventory, revenue from the order and calculates the net earning based on that.
    - Users can review the report data for different months within the selected year and has option to have the report printed as a pdf.
3. **Expected Result:**
    - The system displays updated the report data for the selected year with option to have it in a pdf format as well.

2.5.2. UseCase: ViewEmployee

1. **Test Objective:** To verify that the system can display a list of registered employees.
2. **Test Step:**
   - Admin navigates to the ”Employee List” section.
   - The system displays a list of registered employees in a table, including the following details for each employee:
     - ID
     - Name
     - Email
     - Department
     - Position
     - Joined At (Date)
3. **Expected Result:**
   - Admin can view a list of registered employees with their details, which would prove to be useful to extract useful information.

2.5.3. UseCase: Update Employee

1. **Test Objective:** To verify that the system allows an admin to edit employee information.
2. **Test Step:**
   - Admin navigates to the ”Employee List” section.
   - The system displays a list of registered employees in a table.
   - Admin clicks the ”Edit” icon/button for a specific employee.
   - The system navigates to the ”Add Employee” page with the selected employee’s information pre-filled for editing.
   - Admin makes changes to the employee’s information and saves the changes.

- The system updates the employee's information.

3. **Expected Result:**
   - The admin can click the "Edit" icon/button to edit an employee's information.
   - The system allows the admin to make changes to the employee's information and successfully updates it.

2.5.4. UseCase: Delete Employee

1. **Test Objective:** To verify that the system allows an admin to delete an employee when an employee leaves the ministry and is not entitled to use the cafe.
2. **Test Step:**
   - Admin navigates to the "Employee List" section.
   - The system displays a list of registered employees in a table.
   - Admin clicks the "Delete" icon/button for a specific employee.
   - The system displays a confirmation.
   - Admin clicks "Yes" in the confirmation dialog and proceeds to deleting the selected employee from the list.
3. **Expected Result:**
   - The system deletes the employee from the list.

2.6. Payment Management

Inventory Management encompasses various essential functions, including requesting stock, requesting ingredients, stock approval, ingredient approval, inventory registration, and maintaining an inventory list. Additionally, there is a dedicated page where pending requests must be approved. These functionalities collectively ensure efficient control and organization of the inventory and its associated processes.

Use cases for payment management:

- Withdraw Money
- Deposit Money
- Refund Money

2.6.1. UseCase: Deposit Money

1. **Test Objective:** :To verify that the system allows a cashier to deposit money to a specific employee's account.
2. **Test Step:**
    - Cashier navigates to the "Deposit" section of the system.
    - Cashier enters a valid Employee ID in the "Employee ID" field.
    - Cashier clicks the "Find Employee" button.
    - The system retrieves and displays the full name of the employee associated with the provided employee ID.
    - Cashier enters a valid amount of money to deposit in the "Money Amount" field.
    - Cashier clicks the "Deposit" button.
    - The system processes the deposit transaction and displays a notification message confirming the successful deposit.
3. **Expected Result:**
    - The Cashier can successfully deposit money to the employee's account.
    - The system accurately retrieves and displays the employee's full name.
    - The Cashier receives a confirmation message of the successful deposit.

2.6.2. UseCase: Withdraw Money

1. **Test Objective:** To verify that the system allows a cashier to withdraw money from a specific employee's account.
2. **Test Step:**
   - Cashier navigates to the "Deposit" section.
   - Cashier enters a valid Employee ID in the "Employee ID" field.
   - Cashier clicks the "Find Employee" button.
   - Cashier system retrieves and displays the full name of the employee associated with the provided employee ID.
   - Cashier enters a valid amount of money to withdraw in the "Money Amount" field and clicks the "Withdraw" button.
   - The system processes the withdrawal transaction and displays a notification message confirming the successful withdrawal.
3. **Expected Result:**
   - The Cashier can successfully withdraw money from the employee's account.
   - The system accurately retrieves and displays the employee's full name.
   - The Cashier receives a confirmation message of the successful withdrawal.

2.6.3. UseCase: Refund Money

1. **Test Objective:** To verify that the system allows a cashier to refund money to a specific employee's account.
2. **Test Step:**
   - Cashier navigates to the "Deposit" section.
   - Cashier enters a valid Employee ID in the "Employee ID" field.
   - Cashier clicks the "Find Employee" button.

- The system retrieves and displays the full name of the employee associated with the provided Employee ID.
- Cashier enters a valid amount of money to refund in the ”Money Amount” field.
- Cashier clicks the ”Refund” button.
- The system processes the refund transaction and displays a notification message confirming the successful refund.

3. **Expected Result:**
    - The Cashier can successfully refund money to the employee’s account.
    - The system accurately retrieves and displays the employee’s full name.
    - The Cashier receives a confirmation message of the successful refund.

2.7. Constraint Management

Use cases for constraint management:

- Set Constraints

2.7.1. UseCase: Set Constraints

1. **Test Objective:** :To verify that the system can display and edit constraint settings related to order amounts, allowed order time,and the cafes open hours.
2. **Test Step:**
    - Cashier navigates to the ”Constraint” section within the system.
    - The system displays a section with the constraint settings.
    - Cashier can view the current constraint settings. These settings are initially non-editable.

- Cashier clicks the ”Edit” button.
- The system enables the editing of constraint settings.
- Cashier makes changes to any of the constraint settings (at least one).
- Cashier clicks the ”Save” button.
- The system validates and saves the edited constraint settings.

3. **Expected Result:**
   - The changes on the constraints should be updated throughout the system and be binding for all employees using their mobile applications.

3. Use Case and Testing for the Mobile Application

3.1. Authentication Management

Authentication Management is a crucial component of the mobile application, ensuring that employees can securely log in or sign up for access to the application's features. This section outlines the use cases and testing objectives for Authentication Management.

3.1.1. Use Case: Employee Login

1. **Test Objective:** To ensure that an employee can successfully log in to the mobile application using their credentials.
2. **Test Steps:**
    - Launch App: The employee opens the mobile app on their device.
    - Access Login Screen: The employee navigates to the login screen within the app.
    - Enter Credentials: The employee enters their valid email and password into the respective fields.
    - Submit Login: The employee submits the login form.
3. **Expected Result:**
    - The employee is successfully logged in, gaining access to the application's features and personalized content. Once logged in, any other time when an employee opens the app , they are directed to

home page and wont see the authentication screen again (unless the user logs out).

- If the credentials are incorrect, the employee receives an error message indicating they provided incorrect credentials.

3.1.2. Use Case: Employee Sign-up

1. **Test Objective:** To verify that an employee can successfully sign up for a new account in the mobile application.
2. **Test Steps:**
   - Launch App: The user opens the mobile app on their device.
   - Access Sign-up Screen: The user navigates to the sign-up screen within the app.
   - Fill Registration Form: The app presents a registration form for the employee to enter their details, including their email, password. Other employee information is provided by the admins.
   - Submit Sign-up: The user submits the registration form to create a new account.
3. **Expected Result:**
   - The employee is then passed to OTP verification screen. where they confirm their identity
   - If the employee is not registered by the admins, the user receives appropriate error messages.
   - Once signed in, any other time when an employee opens the app , they are directed to home page and wont see the authentication screen again (unless the user logs out).

3.1.3. Use Case: OTP Verification

1. **Test Objective:** To verify that an employee can successfully complete OTP (One-Time Password) verification during the sign-up process.

2. **Test Steps:**
    - After successfully submitting the registration form (as described in the ”Employee Sign-up” use case), the employee is directed to the OTP verification screen.
    - Access OTP Screen: The employee is presented with a screen to enter the OTP received on their registered email address.
    - Enter OTP: The employee enters the OTP received via email into the appropriate field on the screen.
    - Verify OTP: The employee submits the entered OTP for verification.
3. **Expected Result:**
    - If the entered OTP matches the OTP sent to the employee’s registered email address, the verification is successful, and the employee gains access to their account.
    - If the entered OTP is incorrect or has expired, the employee receives an appropriate error message.

3.1.4. Use Case: Password Reset

1. **Test Objective:** To ensure that employees can successfully reset their password in case of forgotten credentials.
2. **Test Steps:**
    - Launch App: The employee opens the mobile app on their device.
    - Access Password reset: The employee navigates to the forgot password section which resides in the login page.
    - Enter Email: The employee enters their registered email address into the appropriate field.
    - Request Password Reset: The employee submits a request to reset their password.
    - The employee provides the new password together with the Otp code sent to their email.
3. **Expected Result:**

- The employee receives an Otp code to the email they provided.
- If the Otp is incorrect system gives incorrect Otp message, and password recovery fails

3.2. Mobile Menu Management

3.2.1. Use Case: View Available Food Items

1. **Test Objective:** To verify that an employee can successfully view available food items within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.
   - Home Screen: The employee is presented with the home screen of the app.
   - Filter Food Items: The employee can filter food items by categories such as breakfast, lunch, fasting, non-fasting, and drinks.
   - Search Food Items: If the employee is having a hard time finding a menu item. They cna use the search icon on the right side of the app bar and search the menu item. The system search returns all related menu items related (are similar) to what the employee has entered.
   - See More: The employee has the option to view all available food items.
   - Food Item Details: When a food item is pressed, the app displays food's information as
     - Name
     - Description
     - Price for employees
     - Price for guests
     - The quantity to add to the cart.
3. **Expected Result:**

- employee can filter and view food items by category.
- employee can see detailed information about each food item.
- Only food items made available by the cashier are shown in the app.
- When employee opens app and places an available food item, but in the middle the cashier makes the food item unavailable. When placing the order, it is validated, and the employee is shown with dialog saying the ordered menu item is unavailable.

3.2.2. Use Case: Add Foods to Cart

1. **Test Objective:** To verify that an employee can successfully add food items to their cart within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.
   - Home Screen: The employee is presented with the home screen of the app.
   - Select Food Item: The user chooses a food item they want to add to their cart.
   - Quantity Selection: The user selects the quantity they want to add to the cart.
   - Add to Cart: The employee clicks the ”Add to Cart” button.
   - Cart View: The app displays the employee’s cart with the added food item(s), their respective quantities and price for the menu item.
   - Quantity Update: If the same food item is added again, the quantity in the cart is updated instead of creating a separate entry.
   - Minimum Quantity: If the employee tries to add a food item with a quantity of 0, the app displays a dialog stating that they must choose at least one quantity.
   - Remove from Cart: The employee has the option to remove a food item from the cart.
3. **Expected Result:**

- Employees can easily add food items to their cart with the desired quantity.
- The cart accurately reflects the added food items and quantities.
- If the same food item is added, the quantity in the cart is updated.
- Employees are prevented from adding items with a quantity of 0.

3.3. Order Management

3.3.1. Use Case: Order Placement

1. **Test Objective:** To verify that employees can successfully place an order within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.
   - Select Food Items: The employee selects food items they want to order, specifying the quantity for each item.
   - Proceed to Checkout: The employee proceeds to the checkout screen to review their order.
   - Review Order: In the checkout screen, the employee reviews their order, which includes
     - The employee's name
     - The date
     - Food items ordered with quantities
     - The price for each food item.

     ,
   - Calculate Total: The app calculates and displays the total price for the entire order.
   - Place Order: The employee clicks the "Place Order" button to confirm the order.
   - Order Validation (as set by the cashier):

  - If the cafe is closed, the app displays a dialog stating that the cafe is closed.
  - If any of the ordered menu items are unavailable, the app shows a dialog indicating that some menu items are not available.
  - If the order limit is reached, a dialog informs the user to try buying food as a guest.
  - If the order includes lunch or breakfast menu items but is not the appropriate time, a dialog points out the constraint set by the cashier regarding meal times.
- Deduct Account Balance: After successful validation, the app deducts the corresponding amount from the employee's account balance.
- Reflect in Profile: The profile screen is updated to show the new account balance.

3. **Expected Result:**
   - Employee can review and confirm their order in the checkout screen.
   - The app validates the order for various conditions such as cafe hours, menu item availability, order limit, and meal times.
   - If the order is valid, the order is placed, and the employee's account balance is updated accordingly.
   - Employee can see the updated account balance in their profile screen.

3.4. Account Activity Management

3.4.1. Use Case: View Profile

1. **Test Objective:** To notify that an employee can successfully view their profile information within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.

- Access Profile: The employee navigates to the profile section within the home screen of the app by pressing profile icon on the left side of the app bar.
- View Profile: The app displays the employee's profile information, including their:
  - Name
  - Email
  - Department
  - Position
  - account in their balance
- Buy food as Guest: The app gives employee choice of buying food as a guest, when an employee has reached his/her limit to buy food with the employee price. A toggle button is present to enable this functionality.
- Log out: The app allows to logout of the system.

3. **Expected Result:**
   - The employee can view their profile details accurately, ensuring the information is up-to-date.
   - When buy food as guest is enabled, prices added to the cart take price set for a guest and not for the employee.

3.4.2. Use Case: View Transaction History

1. **Test Objective:** To verify that an employee can successfully view their transaction history within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.
   - Access Transaction History: The employee navigates to the transaction history section within the app, found on the left side of the app bar.
   - View Transactions: The app displays a list of the employee's past transactions, including details such as

  - Transaction type
  - Amount
  - Date
  - Transaction ID.

3. **Expected Result:**
   - The employee can view their transaction history accurately, ensuring all past transactions are listed correctly.
   - Transactions are sorted chronologically with the latest transactions appearing at the top.

3.4.3. Use Case: View Order History

1. **Test Objective:** To verify that an employee can successfully view their order history within the mobile application.
2. **Test Steps:**
   - Launch App: The employee opens the mobile app on their device.
   - Access Order History: The employee navigates to the Order history section within the home screen of the app by order icon on the right side of the app bar.
   - View Orders: The app displays a list of the user's past food orders, including details such as
     - Order coupon
     - Ordered food items with their quantity when card is pressed
     - Total price amount
     - Date
3. **Expected Result:**
   - The employee can view their order history accurately, ensuring all past orders are listed correctly.
   - Order history includes all the orders associated with the employee using the app (Employee orders and Guest orders).
   - Orders are sorted chronologically with the latest orders appearing at the top.