Requirements Definition

Dan’s Frappuccino Paradise

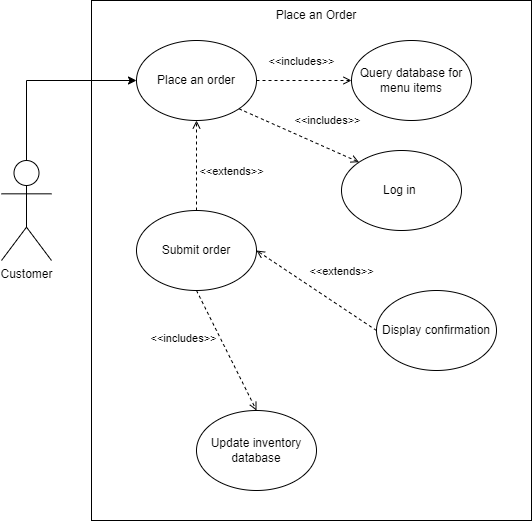
**Introduction and Context**

Dan wants to start a Frappuccino Café. This project aims to build a computer system to make running and managing it easier.

There are three roles a person can have while using the system: customer, barista, or manager. Customers can place orders and manage account balances. Baristas can log hours, place orders for customers, and view and edit pending order progress. The manager can pay, hire and fire employees, buy inventory, edit the menu (including prices) and manage the company’s finances. Baristas can do everything that customers can do, and the manager can do everything that baristas can do. Dan’s enterprise is a success, all thanks to the computer program that practically manages his company for him.

**Users and their Goals**

The following use case diagrams illustrate several actions that a user can take in the system:

****

**User Places an Order**

Participating Actor: Customer

Entry Conditions:

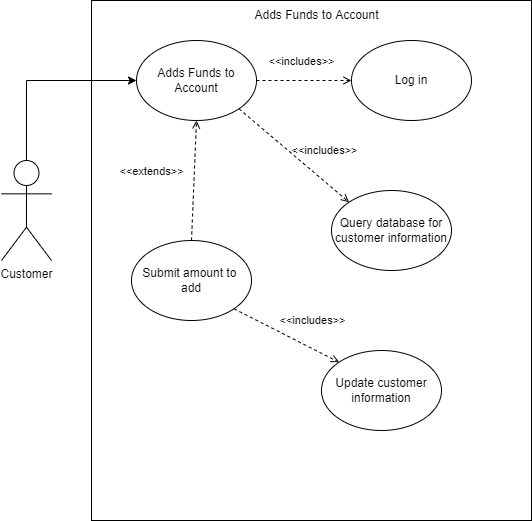
* Customer wishes to place an order

Exit Conditions:

* Customer submits and pays for their order
* Customer cancels order

Event Flow:

1. Customer logs in
2. Menu item database is queried and displayed to Customer
3. Customer selects an item to order
4. Customer customizes the order as desired
5. Customer submits order
6. Money is transferred from Customer account to manager account
7. System displays order confirmation

****

**User Adds Funds to Account**

Participating Actor: Customer

Entry Conditions:

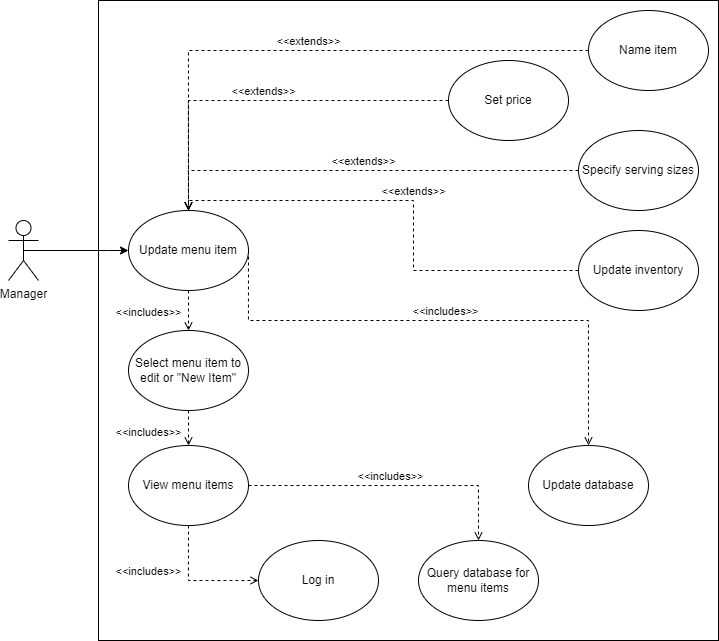
* Customer wishes to add funds to their account

Exit Conditions:

* Customer submits funds to their account
* Customer cancels process

Event Flow:

1. Customer logs in
2. Customer accesses account page
3. Customer information and amount of funds are displayed
4. Customer presses button to add funds
5. Customer inputs amount of funds to add
6. Customer presses button to submit request
7. Customer’s account is updated with new amount of funds

****

**Update Menu**

Participating Actor: Manager

Entry Conditions:

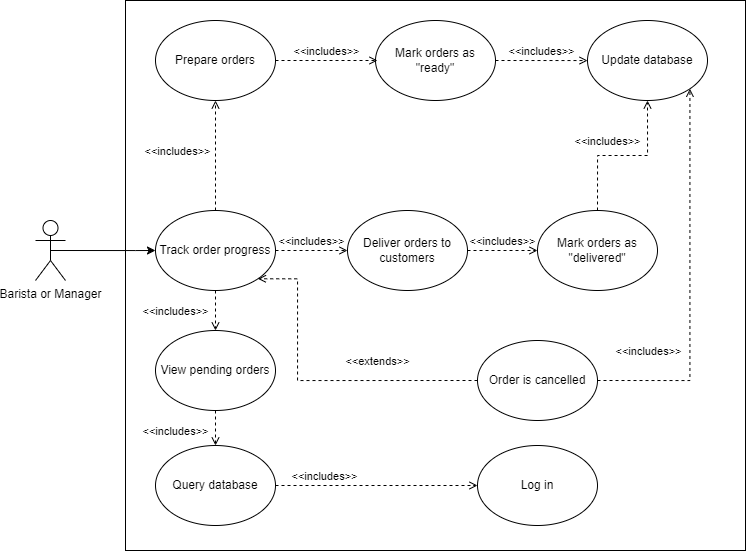
* Manager wishes to add a new menu item
* Manager wishes to update an existing menu item

Exit Conditions:

* Manager saves changes to menu item

Event Flow:

1. Manager logs in
2. Menu item database is queried and displayed to manager
3. Manager selects item to edit
4. Manager edits the item as desired
5. Manager saves changes and the database is updated

****

**Track Order Progress**

Participating Actor: Baristas, Manager

Entry Conditions:

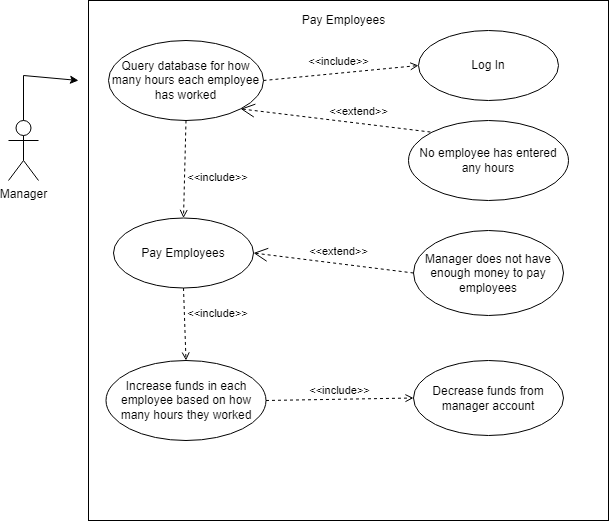
* Barista starts working

Exit Conditions:

* Barista stops working

Event Flow:

1. Barista logs in
2. Pending orders are displayed
3. Barista prepares an order
4. Barista marks order “ready”
5. Barista gives order to customer
6. Barista marks order “delivered”



**Manager Pays the Employees**

Participating Actor: Manager

Entry Conditions:

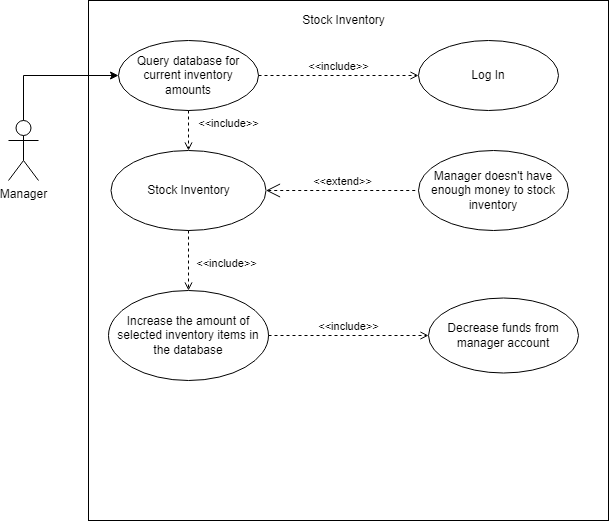
* Manager wishes to pay the employees

Exit Conditions:

* The employees are paid
* No new employee hours have been logged
* The manager doesn’t have enough money to pay the employees

Event Flow:

1. Manager logs on
2. The database is queried to see how much each employee has worked
3. Manager clicks a button to pay employees
4. Increase the funds of the employee accounts
5. Decrease the funds of the manager account
6. Display confirmation



**Manager Stocks the Inventory**

Participating Actor: Manager

Entry Conditions:

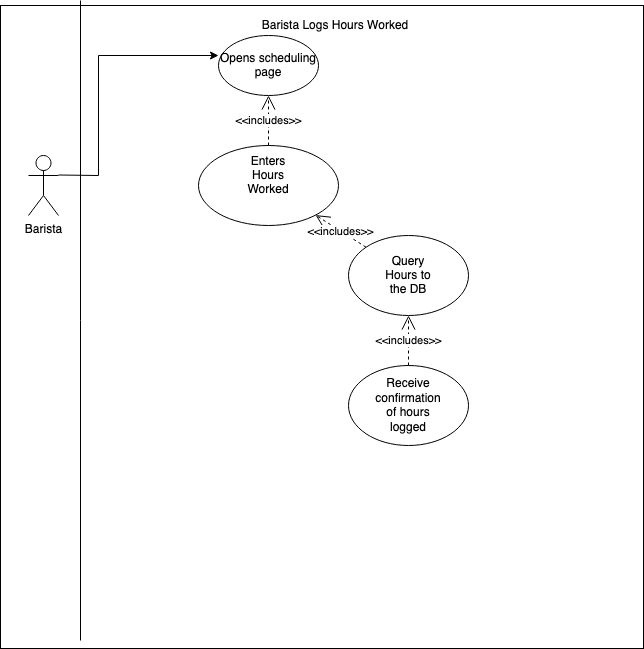
* Manager wishes to stock the inventory

Exit Conditions:

* The inventory has been stocked
* The manager does not have enough money to stock the inventory

Event Flow:

1. Manager logs on
2. The database is queried to see current inventory status
3. Manager selects how much of each inventory item he would like to stock
4. Manager submits the request to stock the inventory
5. Funds are taken from the managers account
6. The inventory is stocked
7. A confirmation page is displayed



**Barista Logs Hours Worked**

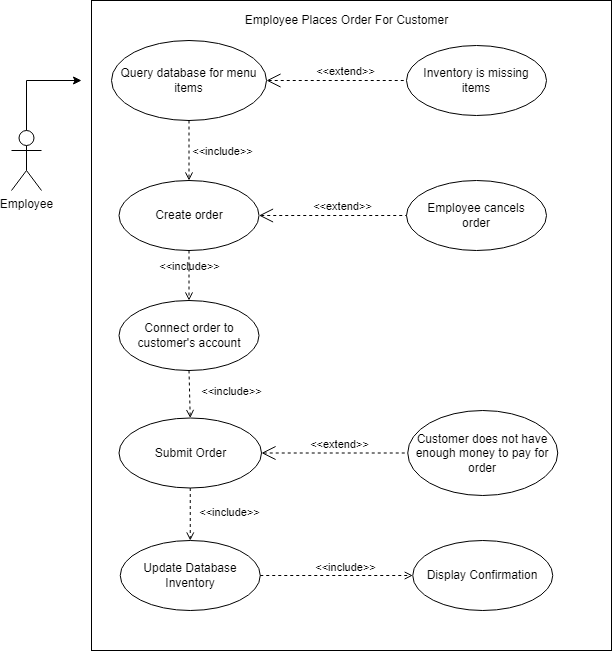
Participating Actor: Barista

Entry Conditions: Barista is logged in

Exit Condition: Barista’s hours worked are logged in the system

Event Flow:

1. Barista opens the scheduling page
2. Barista enters the number of hours worked
3. The inputted hours are sent to the database
4. System displays confirmation that the hours are successfully logged.



**Employee Places Order For Customer**

Participating Actors: Baristas, Manager

Entry Conditions:

* An employee takes a customer’s order

Exit Conditions:

* An order has been placed
* The order is canceled
* The customer does not have enough money

Event Flow:

1. Menu item database is queried and displayed to employee
2. Employee selects an item to order for customer
3. Employee connects the order to the user’s account
4. Employee submits order
5. Money is transferred from customer account to manager account
6. System displays order confirmation

**Functional Requirements**

1. User Authentication and Accounts
   1. Users must be able to create accounts.
   2. All users must log in correctly before doing anything else.
   3. Passwords must be encrypted
   4. Users should be able to manage usernames and passwords.
   5. Users must be able to add funds to accounts.
2. Inventory Management
   1. Inventory should decrease as products are used to fulfill orders.
      1. Customers should not be able to place orders that cannot be fulfilled due to inventory shortages.
   2. The manager must be able to buy and stock the inventory.
      1. The manager should be able to view the current status of the inventory.
      2. Customers and baristas should not be able to buy and stock the inventory.
   3. The manager should be able to update menu information.
      1. The manager should be able to add products to the menu.
      2. The manager should be able to edit product names.
      3. The manager should be able to edit prices.
      4. The manager should be able to edit portion sizes.
3. Order Placement
   1. All users must have the ability to place orders for themselves.
   2. Baristas and the manager should be able to place orders for customers.
   3. Users should be able to customize orders.
4. Order Management
   1. Baristas and the manager must be able to manage orders.
      1. Customers must not be able to manage orders.
   2. Customers, baristas and the manager could track order history.
      1. Customers could only see their order history
      2. Baristas and the manager could see all order history
   3. Baristas and the manager must be able to mark orders as ready for pickup.
      1. Customers must not be able to mark orders as ready for pickup.
   4. Baristas and the manager must be able to mark orders as fulfilled.
5. Employee Management
   1. Baristas must be able to enter in their hours worked.
   2. The manager must be able to pay the baristas for hours worked.
      1. The manager could set an hourly wage for each barista.
   3. The manager could view how many hours each employee has worked.
   4. Baristas could view how many hours they have worked since the last time they were paid.
   5. The manager should be able to hire and fire baristas.
      1. Customers and baristas must not be able to hire nor fire baristas.

**Non-functional Requirements**

1. Development will follow an Agile method with weekly iteration meetings.
2. Front-end development will use the React framework.
3. Back-end development will use the Django framework.
   1. Database management will occur through Django.
   2. A SQLite database will be used.
4. Code will be stored in a GitHub repository.

**Future Features**

1. Customers could create favorite orders for quick ordering.
2. Products other than frappuccinos could be sold.
3. Products could have ratings and reviews.
4. Food or merch could be delivered directly to the customer’s home.
5. The café could expand to multiple locations.

**Glossary**

* Customer: Any person wishing to order a frappuccino from the café.
* Barista: Any employee other than the manager. The system assumes a simple company structure where baristas fulfill the role of cashier and beverage maker.
* Manager: The person in the frappuccino shop who manages inventory and finance.
* User: Any customer, barista or manager.
* Frappuccino: A fancy drink with lots of ingredients.