

# *Dan's Bagel Shop*

## **1. Introduction and Context**

*This is a brief introduction to the purpose system that describes the environment or context of the proposed system, the problem it addresses, and how the proposed system will solve that problem. It should be less than a page. A couple of paragraphs to a ½ of page would be sufficient in most cases. For the most part, this description should be very similar to the project over in the project plan.*

This document describes the functional and non-functional requirements for a web based application called, Dan's Bagel Shop, that will allow customers the ability to order bagels online. It describes how users and staff will interact with the web application.

On the user side, they will be able to use the web application to place an order on the website by selecting an option from the menu. They would then choose a time for pickup and proceed to pay for their items using a preloaded balance on their account. Before a customer orders, they would need to set up an account on the web application which requires a username, password, email, etc. (further details found below). The customer would be given access to a balance from which they purchase from the bagel shop.

The staff at Dan's Bagel Shop will be given three different positions. The first position will be the manager. The system will allow the manager to see the current inventory and given the ability to refill the inventory if necessary. The cashiers will use the system to see a current list of open orders and given the ability to finish an order once a customer has picked up their food. Chefs will be allowed to see and fulfil an order. Upon making the order, they will be given the ability to decrement from the inventory, updating the system's inventory list.

## **2. Users and their Goals**

*This section contains identifies of the users of the proposed system and their goals, illustrated and supported by Use Case diagrams. Here "users" is a broad term that could include other software systems.*

The UML Use Case Diagrams in the following figures describe the staff positions and the user goals for Dan's Bagel Shop.

Figure 1 - Staff Members and Users

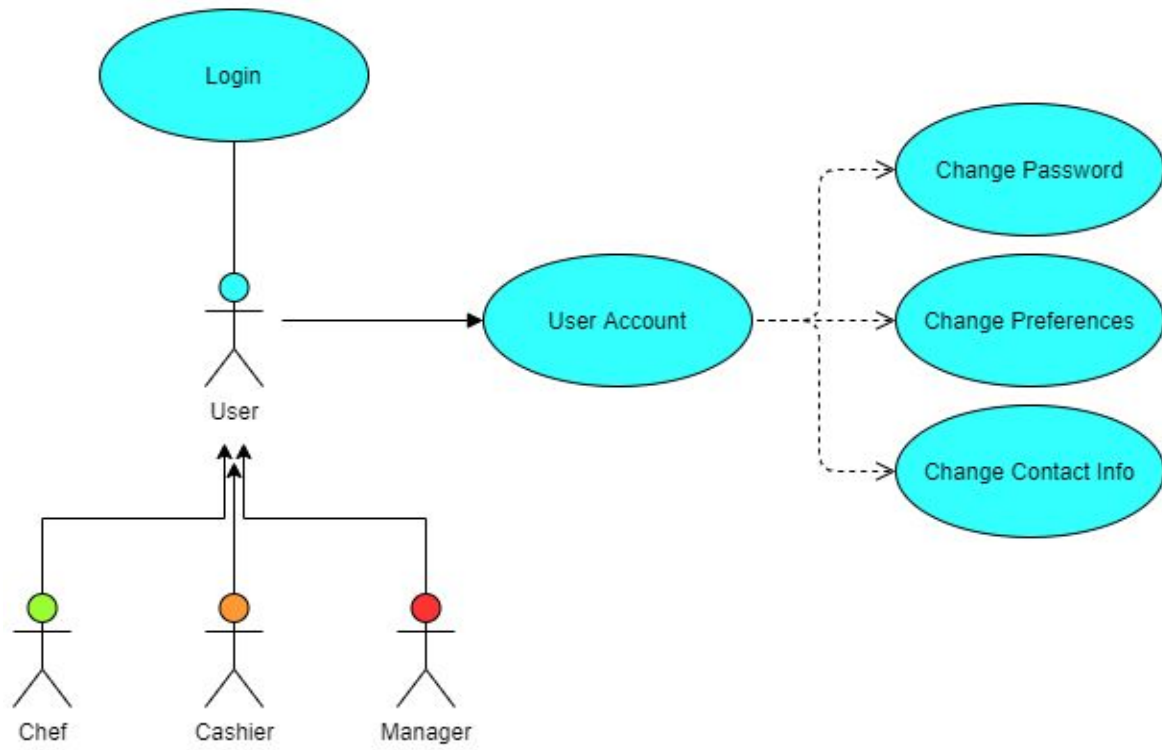


Figure 2 - User Goals

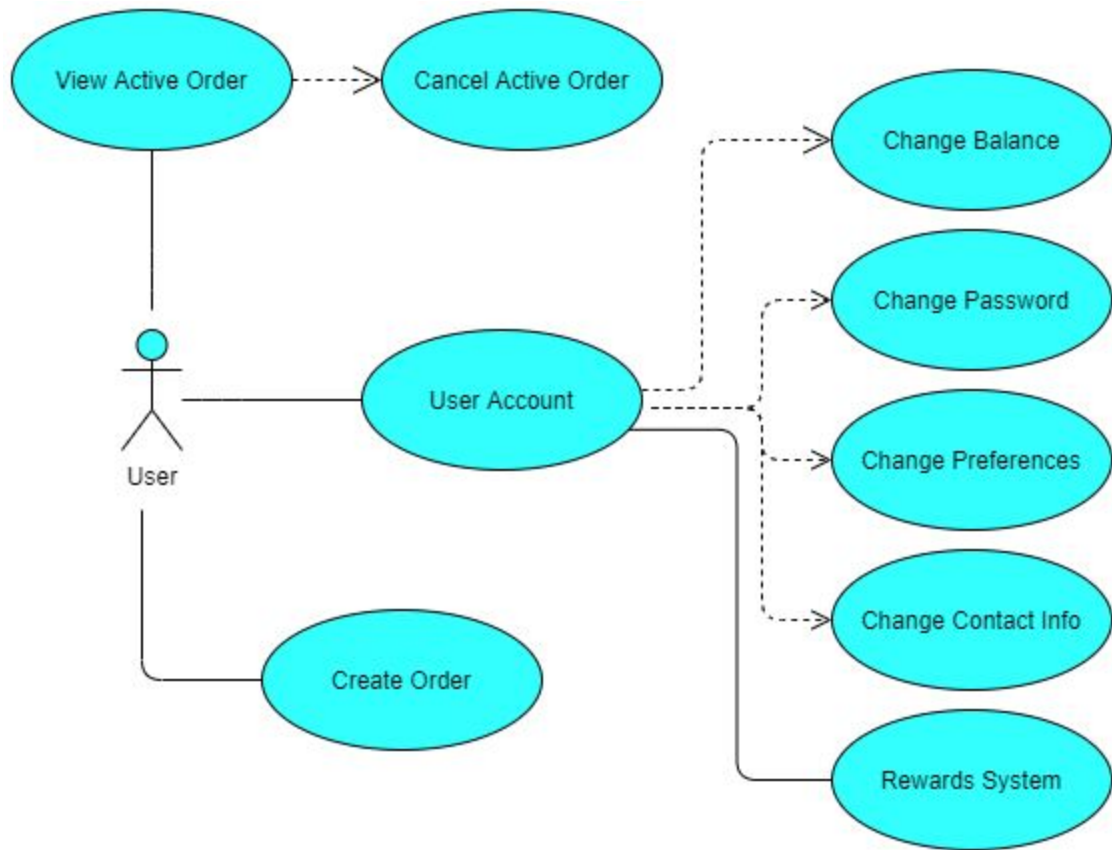


Figure 3 - Chef Goals

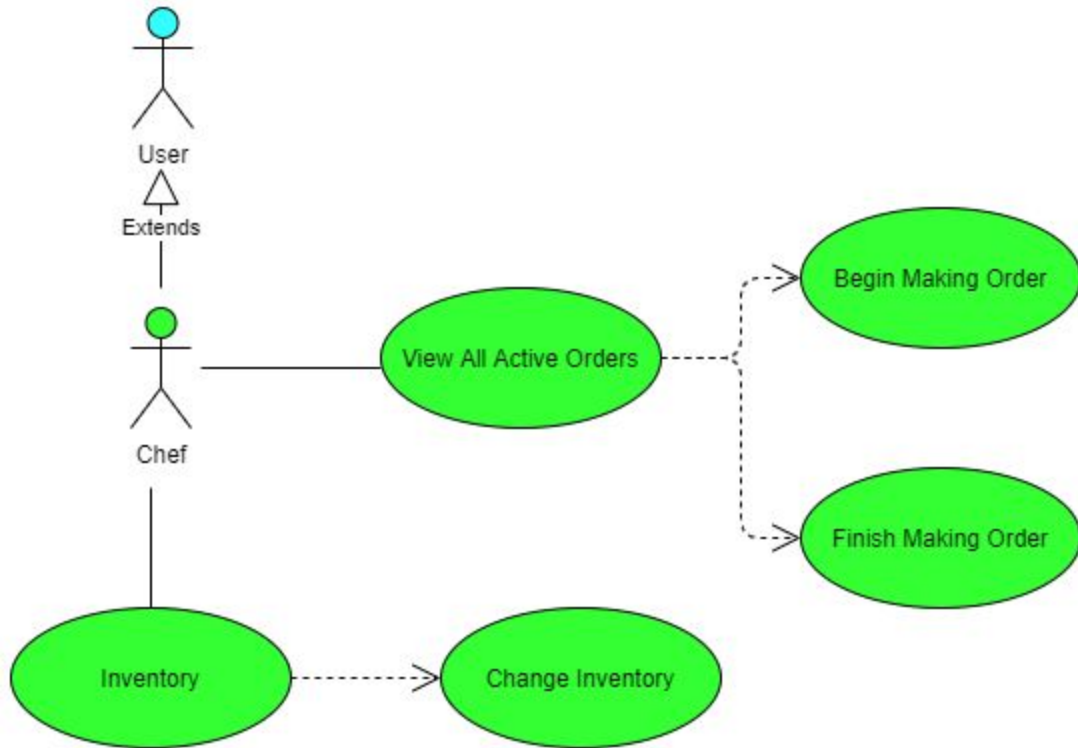


Figure 4 - Cashier Goals

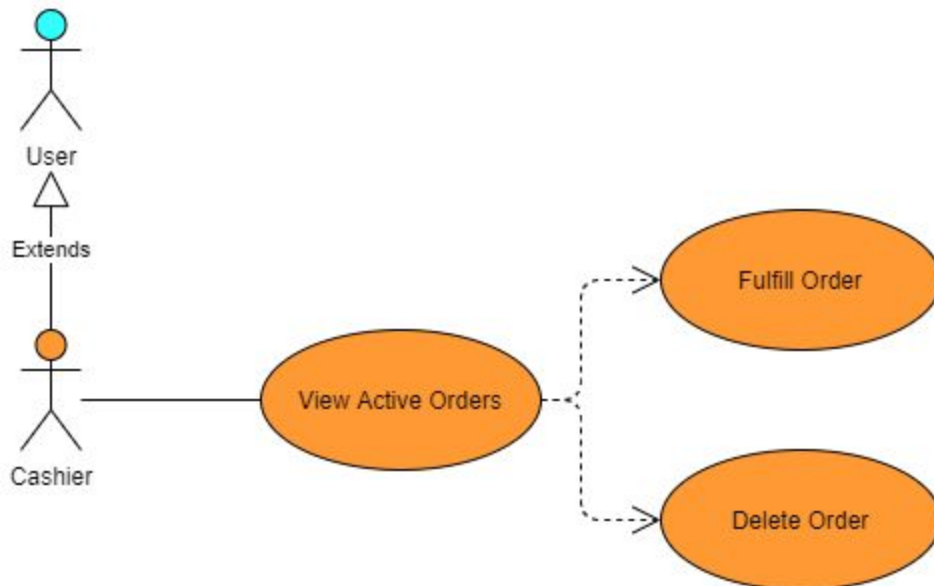
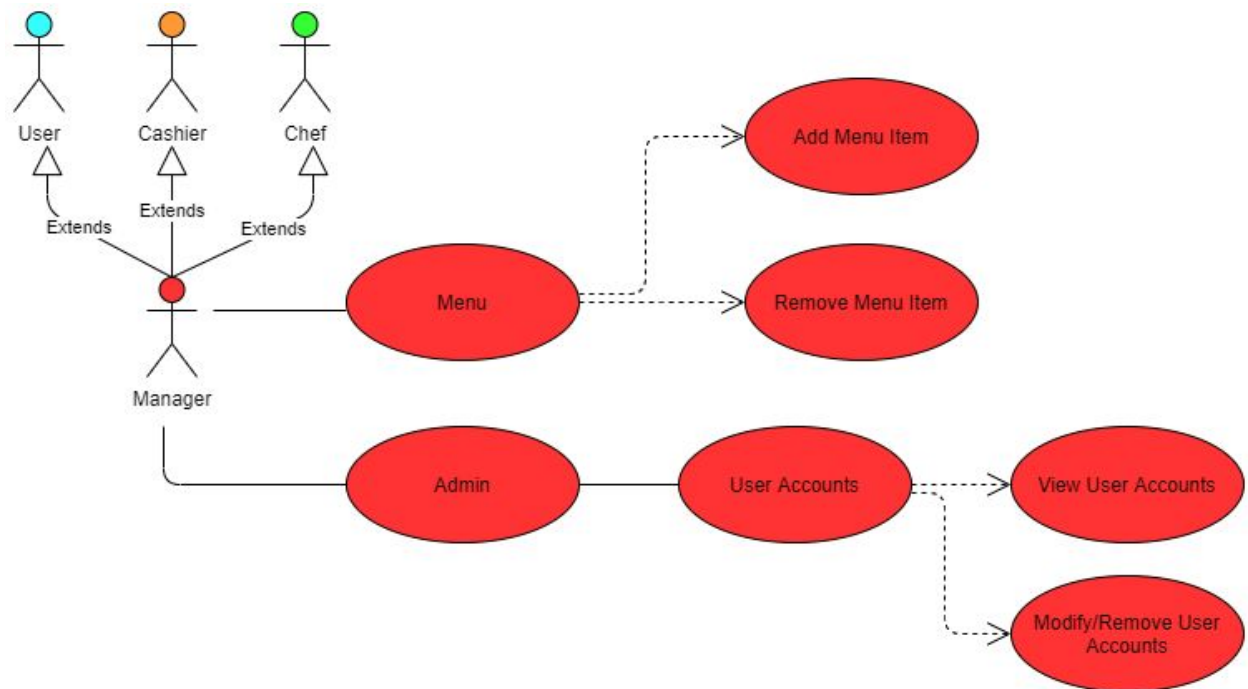


Figure 5 - Manager Goals



Below are a couple example scenarios that resemble user goals.

#### Scenario #1

1. John, a customer, logs onto his account from the home page of Dan's Bagel Shop using his username and password.
2. John can see his active orders in the middle of the page. At the top, he sees a change preferences button and a create order button.
3. John selects Create Order and it allows him to choose a pick-up time for his order.
4. John then selects his desired order from the provided menu and adds it to his total.
5. John then selects checkout and is taken to the checkout screen where he confirms his order and money is taken out of his account balance.
6. John is then returned to the main page where he sees his active order is now showing along with the current status of it, and the option to cancel his order.

#### Scenario #2

1. Sam, a chef, logs into his account using his name and password.
2. Sam can see all the active orders, sorted by time on the main page.
3. At the top of the screen, he can see multiple buttons: one to take you to the inventory page, one to take you to the personal preferences page, and one to create an order.
4. Sam takes the first active order and clicks Accepted, and begins to make it.

5. Sam accidentally drops one of the inventory items and has to grab another. He goes to the inventory page and decrements the item from the inventory count. He then returns to the main screen.
6. Sam finishes the order and updates the status to Order Made, and hands the order to the cashier.
7. Sam can then repeat the same process with the next active order in line.

### Scenario #3

1. Dan, the manager, logs into his account using his username and password.
2. After he is logged in, displayed on the screen is multiple options for Dan to pick: Create an Order, Inventory, Admin, Personal Active Orders, Shop Active Orders, My Account, and Menu.
3. Dan has access to the inventory and has the ability to update it. There is a big button to fill in everything at once as well as buttons to fill individual items.
4. Dan is also able to update and make changes to the menu.
5. Dan decides to remove an item from the menu and replace it with a brand new item.
6. He updates the menu and then navigates to the inventory page to update it with the ingredients necessary for the new item and removes the ingredients for the old one.
7. Dan then, saves changes and returns to the main page.

### 3. Functional Requirements

*This section contains requirements or constraints on the functioning of the proposed system, written in a way that doesn't presuppose "how" the system will accomplish those requirements. The requirements should be organized in a hierarchy of increasing specificity and presented in outline form so they are easy to reference. Each requirement must stand on its own because it could be referenced or quoted in other documents without the benefit of contextual information. Where appropriate, the requirements could include statements about the rationale (motivation) and/or priority (importance to the client.)*

#### Functional requirements:

- Users will be able to create a username and password
- Application will allow user to create a username and password and will store that information in the database
- Users will be able to customize order and place order
- Users can view different spreads and toppings to go on bagel
- Users can put in payment information
- Application will process payment information
- Users can set a time to pick it up from the shop
- Cashiers will be able to see the current list of all open orders
- Employees can change status of order
- Employees can order as customers
- Employees can update the inventory
- Managers can update menu
- Managers can update inventory

- Managers can create rewards programs (discounts, promotions)

#### **4. Non-functional Requirements**

*This section contains requirements that describe and/or constrain the development process. For example, a non-functional requirement may state that development will follow an Agile method with weekly iteration meetings.*

Non-Functional Requirements:

- Application should be able to store thousands of login credentials
- Application will force user to create a secure password
- Application should be able to allow potentially hundreds of users to use the application at once
- Application must allow at least 15 users to order at the same time?
- Application can show the user the status of their order; Ordered, In preparation, and Ready. I will allow the user to cancel their order right after they've ordered, up until the status changes to "In Preparation"
- Purchase is made with balance on account, default is 100. User is able to add more to account
- Allow user to edit account information at any time
- Cashier needs to be able to see current list of all orders and orders that are completed will be at the top
- Inventory is decremented as soon as order is in preparation
- If order is unclaimed by the customer then it is closed
- Chefs must be able to view all open orders and update the orders status as soon as completed
- Chefs can submit an order as a user
- Chefs should be able to decrement or increment inventory
- Manager can submit an order as a user
- Managers can update menu and update inventory
- Managers will have a Big Button to fill everything and a button to fill individual items

#### **5. Future Features**

*This section contains a list of ideas or features that are beyond the scope of the project.*

Make pre-made Party Packs available

Allow multiple users to pay for single purchase

Allow payment plans for expensive purchases

Include a menu suggestion box

Allow users to send bagel gifts

Make a mobile application

## **6. Glossary**

Functional- detailed statements of the project's desired capabilities. They are the user requirements but won't necessarily be seen by the user. They are the things the application should do. Ex) Most user requirements

Non-Functional- Statements about the quality of the application's behavior or constraints on how it produces a desired result. These requirements specify things such as the application's performance, reliability, and security characteristics.