Requirements Definition

*Dan’s Bagel Shop*

**Introduction and Context**

Currently orders can only be made in person at Dan’s Bagel Shop. With an online store Dan’s Bagel Shop will be more accessible, attracting a bigger consumer base. With a bigger consumer base there needs to be better tools for handling the workflow and inventory of Dan’s Bagel Shop. This project would provide tools for management and employees, and provide an online storefront for customers to make purchases.

For store employees the project would keep track of orders to ensure they’re ready in time for the customers. Management will be able to view the store’s inventory anywhere, and order ingredients as needed. Customers will be able to place orders online, and track their progress. They will also have access to their order history so they can easily order their favorite bagels.

**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 board term that could include other software systems.*

**Functional Requirements**

The majority of the functional requirements for this project relate to what user accounts can do. The first item on the list defines what all accounts should be able to do.items two through five define functions specific to a certain account type. There are four account types in our program which are: customer, chef, cashier, and admin(owner). The last items just detail how certain parts of the online interface will look.

1. **All Account Details**
   1. Each account must have an Email associated with it
   2. A first and last name must be provided for the account
   3. Passwords will be required
   4. All accounts will have a unique numeric ID
   5. Users will be able to change account info
      1. Passwords, name, and email
2. **Customer Account Details**
   1. $100 automatically added upon account creation
   2. Customers will be able to add funds to their account balance
   3. The ability to place orders online
   4. View order history
      1. Reorder from order history
   5. View order status
      1. Orders may be canceled until bagel status is “Ready”
      2. Orders may **not** be changed once placed.
3. **Chef Account Details**
   1. Will be able to see incoming orders
      1. Orders will be organized by pick up time
      2. The chef will be able to update the order status
      3. Orders will show up no later than 10 minutes before it needs to be ready
   2. Chefs will be able to update the inventory
      1. remove items that have been used for an order
      2. remove items that have spoiled
4. **Cashier Account Details**
   1. Be able to lookup customer’s numeric ID
   2. See list of orders and which customers they are associated with
5. **Admin (Owner) Account Details**
   1. Add and remove employee accounts
   2. Update the menu
   3. Order inventory
6. **Menu display**
   1. Able to be sorted by most popular items
      1. each item on the menu must keep track of how many times it was order in the last month
   2. Able to be sorted sorted by price
   3. Default menu presentation will be sorted alphabetically within categories
      1. Menu categories are bagels, spreads, sandwiches, and beverages
   4. No premade combos all items are sold individually
7. **Orders**
   1. Pick up times for an order may be specified
      1. Orders can only be placed one week in advance
      2. Orders without a specified pickup time will have a pickup time of when the order was placed plus 10 minutes
   2. Orders not picked up within 30 minutes of being ready are discarded
8. **Inventory**
   1. Items ordered for are the same as items sold on the menu
   2. Each item can be ordered individually
   3. Provide an analytics screen
      1. See what items need to be ordered
      2. See the sale figures of each item
9. **Existing Customers**
   1. Integrate existing customers into the new system
      1. Waiting for details, currently unknown how this existing database looks

**Non-functional Requirements**

This product will only be available as a website

1. Both front and back ends of the website will developed with the Django framework
2. Weekly meetings will be held to review team progress
3. Goal milestones will be hammered out in high-level design
4. Team communication will occur in discord
5. Google drive will be used to store documents that the team worked on concurrently
6. A GitHub repository has been setup to keep track of code changes

**Future Features**

In the future we could develop a phone app which would allow customers to order from their phones. This would also have the added benefit of us being able to send them updates about their orders.

**Glossary**

*This section contains a list important terms and their definition.*