

# Dan's Bagel Shop

---

## Project Overview

This project aims to build a system for Dan's Bagel Shop to allow for online orders.

The system will allow for users to place orders and the staff to see the ongoing orders and inventory. The staff will be able to fulfill orders and manage inventory levels from their respective positions. The users will be able to select a time for a pickup, order their food, and pay for their items from their account they set up. The system will support online ordering from the store's website from the user's PC or the in-store kiosk.

## Team Organization

Angel and Rachel will be focused on the front end side of the project while Dylan and Thomas will be on the back end side of it. If needed, every member will hop back and forth between the two to help wherever is needed. We will be using the egoless programming philosophy.

## Software Development Process

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a "backlog", and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

Phase	Iteration
1.	Phase 1 - Requirements Capture
2.	Phase 2 - Analysis, Architectural, UI, and DB Design
3	Phase 3 - Implementation, and Unit Testing
4	Phase 4 - More Implementation and Testing

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

## Communication policies, procedures, and tools

To communicate we will use Discord group chats and messaging. In addition, we have exchanged emails and phone numbers so that we have more than one tool to get in touch with each other.

## Risk Analysis

A risk would be file management or merging issues with files. To overcome this, we will update the group when we are merging files or updating the repository in any way. This will keep everyone on the same page and it ensures that everyone has the latest code they are working with.

Time management for a particular focus is another risk. We need to watch that we don't individually spend large amounts of time on a particular focus if we can not figure it out.

Functional requirements:

- Users will be able to create a username and password
  - Validate that it is a correct username and password
- Application will allow user to create a username and password and will store that information in the database
  - Make sure the database is storing the correct username and password.
- Users will be able to customize order and place order
  - Make sure the database and menu is up to date with all the inventory items.
- Users can view different spreads and toppings to go on bagel
  - N/A
- Users can put in payment information
  - Make sure the user's account balance reflects what they add.
- Application will process payment information
  - Make sure the user account balance is updated at the end of a payment.
- Users can set a time to pick it up from the shop
  - Make sure the time selected is within the correct time slot (5 min to 1 week).
- Cashiers will be able to see the current list of all open orders
  - N/A
- Employees can change status of order
  - Make sure that status of order is up to date
- Employees can order as customers
  - N/A
- Employees can update the inventory
  - Make sure the database and menu is up to date with all the inventory items.
- Managers can update menu

- Make sure the database and menu is up to date with all the inventory items.
- Managers can update inventory
  - Make sure the database and menu is up to date with all the inventory items.
- Managers can create rewards programs (discounts, promotions)
  - N/A

#### Non-Functional Requirements:

- Application should be able to store thousands of login credentials
  - The number of credentials could exceed the space on the server. Validate that database is storing login credentials correctly and that the database has enough space to store that information.
- Application will force user to create a secure password
  - Validate that the password is meeting the proper criteria.
- Application should be able to allow potentially hundreds of users to use the application at once
  - If too many people get on at a time, it could crash the site. Need to make sure we have the proper hardware to handle this case.
- Application must allow at least 15 users to order at the same time?
  - If too many orders come in at once, some orders may be lost. Need to make sure that the hardware can handle multiple orders coming in 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"
  - Validate that if they try to cancel an order after their order status changes to "In preparation", they can not cancel
- Purchase is made with balance on account, default is 100. User is able to add more to account
  - N/A
- Allow user to edit account information at any time
  - N/A
- Cashier needs to be able to see current list of all orders and orders that are completed will be at the top
  - N/A
- Inventory is decremented as soon as order is in preparation
  - Validate that orders are processed and that status is updated
- If order is unclaimed by the customer then it is closed
  - Make sure that order status updates correctly
- Chefs must be able to view all open orders and update the orders status as soon as completed
  - Make sure that chef can login and has the ability to view all open orders and view their status
- Chefs can submit an order as a user
  - N/A
- Chefs should be able to decrement or increment inventory
  - Make sure that inventory is updated

- Manager can submit an order as a user
  - N/A
- Managers can update menu and update inventory
  - N/A
- Managers will have a Big Button to fill everything and a button to fill individual items
  - N/A

## **Configuration Management**

See the README.md in the Git repository.