**Project Plan**

**Summary**

The project being built is called *Dan’s Bagel Shop*. It is a web application for a customer who would like to have an online presence for their bagel shop. This app would be usable by both customers and employees of the store.

Customers would use this app to order bagels, check on the status of their orders, and add money to an account that they would use to make purchases.

Employees would use this app to monitor and update orders, update inventory on-hand counts, and manage the overall menu items. There would be three subtypes of employee accounts:

* Chefs - primarily concerned with updating order preparation status and inventory counts
* Cashiers - primarily concerned with handling transactions with customers and updating order pickup status
* Owners/Admins - can perform all operations that chefs and cashiers can, as well as update the menu and hire/fire employees

**Team Organization**

Our team has three members: Jonas Knudsen, Tyler Beck, and Kaden Hellewell.

The job duties of each member may change as the project progresses, but here is what we are planning on working on:

Jonas Knudsen - Primarily focusing on programming in the back-end (i.e., database setup)

Tyler Beck - Primarily focusing on interfacing the front-end with the back-end

Kaden Hellewell - Primarily focusing on programming in the front-end (i.e., webpage / GUI design)

**Software Development Process**

The software development process for this assignment will be as follows:

* Requirements Gathering
  + Identify customers and writing down requirements
* High-Level Design
  + Designing major subsystems and interfaces
* Low-Level Design
  + Refine high-level design until pieces can be implemented
* Development
  + Write the code
* Testing
  + Unit tests - testing individual components (classes, functions, etc.)
  + System tests - testing the overall application as a whole
  + Regression tests - running unit tests after integration to make sure nothing new broke the old code
* Deployment
  + Install application for the user
  + In our case, we can just run the Django server locally and demonstrate functionality via Zoom screen sharing

In a real-world context (i.e., if this weren’t just a school assignment for one class for one semester), we would also have the following steps:

* Maintenance
  + Changes, additions, bug fixes, and enhancements
* Wrap-Up
  + Gather information about the project, what went well, what could have been done better

**Policies, procedures, and tools for communication**

Our group is somewhat informal in how we are defining “policies” and “procedures.” When discussing with our group about this, nothing immediately came to mind about how we will implement official policies or procedures. However, we do believe that the need for formal policies or procedures may arise in the future; if and when this happens, we will implement them when needed.

Our group’s primary form of communication is through a Discord chat room. (Discord is an online chatting service that provides text, audio, and video communication channels.) As COVID-19 is limiting the amount of in-person gathering that may take place, we found virtual meetings to suit our group much better than in-person meetings.

**Risk Analysis**

There are no major identifiable risks associated with doing this project. We are not using real customer or credit card information. We are not planning on hosting our project on a live web server. Therefore, the risk for this project is virtually nonexistent.

**Configuration management plan**

Refer to the README.md for the configuration management plan.