Jonas Williams-Gilchrist

Jacob Pope

Luke Barton

Braeden Stewart

**Requirements Definition Document**

Updated 9/24/20

The purpose of this document is to provide a list of various project requirements relating to the overall software engineering project for Daniel’s Bagel Shop. Our group, listed above, are tasked with developing software that will assist Dan with the various happenings of his bagel shop, ranging from APIs for customers to place orders for specific types of products as well as for employees of all levels to access the shop’s inventory and manage different aspects of the workplace congruent to their duty.

The following items help to illustrate our vision for the system and the different functions that this system must have to satisfy Dan’s requirements. This document will distinguish between function requirements, as well as the nonfunctional requirements of the project. Additionally, the listed requirements will be categorized into a hierarchy of complexity, with extra information concerning the rationale and/or priority, as necessary.

1. **Functional Requirements**

* Application will handle wide variety of tasks related to the upkeep of the store
  + Only one storefront needs to be managed and the website will correspond with the physical workplace
  + Store inventory should be editable such that goods used to create bagels can be added as they are physically stocked and removed as they are used to create meals and sold to customers
    - Inventory does not need to include raw ingredients such as flour, sesame seeds, etc., but should include whole items, like plain bagel, bottle of soda, unit of bacon, etc.
    - The store wants to stock approximately 100 of each item, so the system should be able to handle quantities at least that large for inventory management.
  + Facilitate the steps that take place in the purchase of a bagel
    - Customer places an order all the way to customer receives their order.
      * Ordered
      * Preparing
      * Ready
        + Send an email to the customer when their order is ready
      * Completed
    - Menu to order from should be dynamic and updatable
      * Add and remove new types of bagels, shmears, etc.
    - Would like to have orders ready within five minutes, ensure the system can handle this pace.
* Application should support multiple different types of users of the system.
  + Each type of employee should have access relative to their role. For example, the manager should have access to all information, cashiers should be able to handle and approve transactions, and customers should only be able to order meals.
  + Customer
    - Limited accessibility, should only be able to order meals
      * Can cancel order as well
    - Should be able to create an account through the web portal
    - Add money to their account
    - View basic status of their orders/transactions
  + Bagel Chef
    - Placed orders should be visible
    - Update inventory of items related to creating bagels
    - Update status of placed orders
  + Cashier
    - View status of orders
    - Facilitate transactions between customers and the company
    - Should handle purchase of shmears and beverages
      * Update inventory of these items
    - Has the ability to discard an order if the customer does not show/pay
  + Manager
    - Approves/creates employee accounts
    - Has master control over the company
    - Add inventory
      * Can edit items one by one or fill all items up to desired amount
  + Application should be able to support at least twenty-five customers, as well as at least ten employee accounts.
    - Data will be kept that relates to individual accounts
      * First name
      * Last name
      * Email
      * Phone number
      * Account balance (money)
        + Created accounts should be credited $100
      * Member number (identification number)
      * Password
      * Record additional data, such as order history, rewards/coupons
* Access to the system should be through a web browser
  + Each type of account should only see pages pertinent to their role
  + Implement a login system
    - Using a user identification and a password
  + Account creation page
* Record additional analytics of the website
  + Expiration of products
  + Ongoing promotions
  + Item popularity
  + Website traffic

1. **Nonfunctional Requirements**

This section deals with the requirements of our software development project that are not directly related to the functionality of our web-based application. These requirements deal more with how we will accomplish the project, rather than what the project will accomplish.

* The project will follow an agile development process
* We will meet at least weekly to update group members about the progress we are making, as well as any potential issues encountered.
  + Decide on important issues that require the presence of multiple people
* Assignment milestones must be met, as outlined by our client, Dan, from Dan’s Bagels
  + Work should be done on a weekly basis to keep up with these requirements and prevent rushing the project over a single day.

1. **Future Features**

Some ideas that arise during the software development process may not currently be realistic or desired in the result of the products but may have potential to be included later or if our client’s requirements change. It is worthwhile to include them here to have a record of and some items to draw inspiration from.

* Record the individual expirations of products to phase old inventory out more easily and bring new inventory in automatically
* A more robust promotion service and customer history, allowing ease of access and the ability for customers to reorder something they have in the past.
* Include additional employee roles to allow for business expansion
  + Allow for multiple storefronts to be managed

1. **Glossary**

This space is used for defining and listing important terms that we use throughout the documentation of this software development process. Currently, the project is complex enough to warrant through definition of terms. Words should be defined as is common sense in the relative context to which they are used. Feel free to add anything to this section as is seen fit.