*Dolphins*

***The Good Bean* Requirements Document**

# Introduction-Nic

The system requirements document's purpose is to clarify the system requirements for The Good Bean app. This document contains a description model that describes the requirements of our system; a class diagram that outlines the system objects, attributes, and methods; a use case diagram that shows all uses of our system; use case scenarios that explain the uses from the use case diagram alongside with any exceptions; and a systems sequence chart that will provide a sequence diagram for each use case scenario.

# Description Model-brian

Using text, describe the requirements for your system. Expand on the function section from your project plan. Include requirements for the following categories: Output, Input, Processes, Performance and Security.

1. Menu: A staple for any restaurant or eatery-based app. It is designed for the user to look at the items that the Good Bean offers. This will include the item itself, a description if necessary, and the price. There will also be the option for a popular items section of the menu, which will rotate depending on what customers have frequently bought in the past. The popular items will rotate on a weekly basis.
   1. Inputs
      1. Category
      2. Item
      3. Back
      4. Home
   2. Outputs
      1. Category List Page
      2. Item List Page
      3. Item Page
   3. Processes
      1. Takes the category/item input and brings the user to the page for that category/input.
   4. Performance
      1. Should be able to change between pages and begin to load items within 3 seconds.
   5. Security
      1. None
2. Ordering: Yet another staple. In the order section, you will have the option to choose the items that you would like to order, how many of them you want, the size, any extras, a pickup time, and the choice for notifications about an order being ready. This is also where you can mark favorites. The business has not yet grown to a level where delivery is an option. After everything has been established, you will be able to check out by providing your payment, upon which you will get an email confirmation and a receipt.
   1. Inputs
      1. Increase quantitiy
      2. Decrease quantity
      3. Size
      4. Extras (Optional)
      5. Pickup Time
      6. Payment
      7. Phone Number (Optional)
      8. Order
      9. Favorite (Optional)
      10. Checkout
      11. Back
   2. Outputs
      1. Item Page
      2. Pricing
      3. Quantity
      4. Checkout Screen
      5. Payment Methods
      6. Payment Information
      7. Email
      8. Phone Number Entry(Optional)
      9. Text Message (Optional)
      10. Confirmation Screen
   3. Processes
      1. Takes checkout-related input and changes pages accordingly.
      2. Takes input in relation to the order and updates details throughout.
      3. Add item to favorites menu if a user chooses to do so.
      4. Calculates the total price of the order based on what user order details.
      5. Sends out email/text message after order is complete.
   4. Performance
      1. Should be able to move to the next page within 3 seconds.
      2. Should be able to determine if an order can be made within 1 second of criteria for not being able to order being fulfilled.
      3. Should move from the confirmation page back to home page after 8 seconds.
      4. Should immediately have pricing ready as part of moving to checkout screen.
      5. Email/text should be sent within 10 seconds of checkout going through.
      6. Should be able to add to favorites within 2 seconds.
   5. Security
      1. Hides parts of sensitive information so it can’t be easily stolen.
3. Favorites: A convenient feature. This will provide instant access to items the user deems one of their favorites. It makes it so you don’t have to search the entirety of the menu for your preferred items, thus providing ease when ordering. You can also use it for items or specialized orders that you want to remember.
   1. Inputs
      1. Item Selection
      2. Remove
      3. Back
   2. Outputs
      1. Favorites Lists
      2. Item Screen
   3. Processes
      1. Removes a favorite item from the list if the user wants to.
      2. Moves to item screen (with any extras/presets) or back to the favorites list.
   4. Performance
      1. Should remove favorite from list and update within 2 seconds.
      2. Should move to item screen (with any extras/presets) in 5 seconds.
      3. Should leave item screen and return to list within 3 seconds.
   5. Security
      1. None
4. Location: Important, especially for newcomers. This tab lets you know where you are picking up your order. It will also be able to navigate to the location of pickup with Google Maps, although the address is also provided should you want to use another navigation app. As the business expands, new stores have the opportunity to open, allowing for more pickup spots. Therefore, you will be able to establish a default location for travel and pickup. You will also be able to see if a specific location has any deals that another does not. Each deal will be seen alongside the location.
   1. Inputs
      1. Navigate
      2. Location
   2. Outputs
      1. Deal
      2. Map
      3. Address
      4. Hours
      5. Navigation
   3. Processes
      1. Leads you to Google Maps for navigation if user chooses to do so.
   4. Performance
      1. Should be able to transfer over to and start navigation in Google Maps in about 10 seconds.
   5. Security
      1. None
5. Account: A necessity for repeat customers. This will allow a user to create an account with the Good Bean. As an account holder, you will be able to hold your personal information that is beneficial for using the app, including name and payment information. You can also earn rewards/loyalty points as an account holder. The points work as follows: 25 for an extra, 75 for a small drink, and 100 for a large drink. In your account section, you will be able to see how many points you have. You will be able to apply these points from the points menu.
   1. Inputs
      1. Claim Reward
      2. Reward Options
      3. Name
      4. Email Address
      5. Password
      6. Phone Number
      7. Payment Information
      8. Log Out
      9. Delete Account
      10. Log In
      11. Update Name (Optional)
      12. Update Email Address (Optional)
      13. Update Password (Optional)
      14. Update Payment Information (Optional)
      15. Update Phone Number (Optional)
      16. Update Payment Information (Optional)
   2. Outputs
      1. Points
      2. Name
      3. Email
      4. Phone Number
      5. Payment Information
      6. Creation Screen
      7. Terms and Conditions Email
      8. Rewards
      9. My Account Screen
      10. Create Account Screen
      11. Update Information Screen
   3. Processes
      1. Creates account using information provided by user.
      2. Updates information if user chooses to do so.
      3. Uses rewards if user chooses to do so.
      4. Sends out email for terms and conditions when user creates account.
      5. Logs in automatically unless user manually logs out.
      6. Deletes account from Good Bean system if user no longer wants it.
   4. Performance
      1. Should send out terms and conditions email within 5 seconds after create account is selected with all information entered.
      2. Should update and display information within 10 seconds after new information is entered.
      3. Should get from home page to update, rewards, or create pages within 3 seconds.
      4. Should be able to log user out within 5 seconds.
      5. Should be able to delete account within 10 seconds.
      6. Should navigate to order menu from rewards within 10 seconds.
   5. Security
      1. Will need a login to access account on a device.
      2. Login password requires at least 6 characters with at least 1 letter and 1 number.
      3. To create a new account, customer must accept terms and conditions by email.
6. View Receipts: A small feature with a lot of value. In the receipts section, you will have access to your past receipts. This is helpful for keeping track of transaction history. First, you will be able to know if someone else got into your account if you have one. This is similar to bank statements as you can see charges you know aren’t yours. It also lets you look back at your orders and favorite them. After trying something, if you want to see what it was so you can favorite it for next time, receipts are a great way to see what that item was. Finally, it clears up any order confusion. Everyone makes mistakes, and the receipts are there to clear up any mistakes, so there are no problems between customers and employees.
   1. Inputs
      1. View Receipt Details
      2. Favorite
   2. Outputs
      1. Receipt List
      2. Receipt Details
      3. Order Number
      4. Order Date/Time
      5. Location
      6. Points Earned
      7. Price
      8. Payment Method
      9. Back
   3. Processes
      1. Return to receipt list if user wants to go back from receipt details.
      2. Display all details of an order on a receipt when user chooses to view details.
   4. Performance
      1. Should know in advance if user has an account to track receipts with.
      2. Should know in advance if account user has any receipts.
      3. Should be able to load receipt list within 5 seconds.
      4. Should be able to load receipt details within 5 seconds.
   5. Security
      1. Hides parts of sensitive information so it can’t be easily stolen.

# Class Diagram-nic

Create a class diagram. The Class Diagram should contain all of the system objects, their attributes, and any known methods. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

# Use Case Diagram-ebru

Create a Use Case Diagram for all of the "uses" of your system. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

# Use Case Scenarios-ebru

Create a full description Use Case Scenario (detailed descriptions) for each use case of the system. This intermediate scenario should include an enumerated list of steps involved in the activity as well as any exception conditions.

# System Sequence Charts-brian

For each Use Case Scenario, provide a sequence diagram. Use your class diagram, use case diagram and scenarios to create the corresponding Sequence Diagram.