*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

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/Outputs
      1. When starting on the main menu screen, the user will see a list of categories, including specials, iced drinks, and espressos among others. They can then select a category and the next page will display the list of items that fall under that category. For example, items under specials include a lavender vanilla latte, a breakfast in bed latte, and a chocolate nut latte. After selecting an item, the user will be taken to a page containing the item, a description of the item, and the ability to order the item. They should also have the ability to go back a page at any point.
   2. Processes
      1. Takes the category/item input and brings the user to the page for that category/input.
   3. Performance
      1. Should be able to change between pages and begin to load items within 3 seconds.
   4. Security
      1. None
2. Ordering: Yet another staple. In the order section, the user will have the option to choose the items that they would like to order, how many of them they want, the size, any extras, a pickup time, and the choice for notifications about an order being ready. This is also where they can mark favorites. The business has not yet grown to a level where delivery is an option. After everything has been established, the user will be able to check out by providing your payment, upon which they will get an email confirmation and a receipt.
   1. Inputs
      1. When beginning the order, the user must decide on customization options. The first option is size, which will be between regular and large. Next is extras, which has the options of an espresso shot, a flavor shot, or a special kind of milk. Finally, the user will decide on the amount of the item and the pickup time. They will then choose to checkout, upon which a screen will be displayed, showing payment options, the overall price, and notification information. The user will choose what kind of payment they want to use, choosing between a card, Apple Pay, Paypal, Google Pay, or a gift card. More options may be supported in the future. The user will then need to put in an email to get sent a confirmation message and a phone number for notifications if they want. If the user has an account, the payment method, email, and phone number will all be prefilled. After checking out, a confirmation screen will show, saying an email was sent to the used address. If a phone number was entered, a text will be sent out too.
      2. Size
      3. Extras (Optional)
      4. Pickup Time
      5. Payment
      6. Phone Number (Optional)
      7. Order
      8. Favorite (Optional)
   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 payment information and email 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 they don’t have to search the entirety of the menu for their preferred items, thus providing ease when ordering. They can also use it for items or specialized orders that they want to remember.
   1. Inputs/Outputs
      1. While you are on the page for a specific item, the user should have the ability to toggle whether or not the item (and its customizations) is a favorite. If the user adds it as a favorite, they should be taken to the favorites list, now updated with the new item. If the user wants to untoggle the favorite marker, the page should update and display the list with the item now removed.
   2. Processes
      1. Should be able to toggle an item as a favorite or not.
      2. Moves to item screen (with any extras/presets) or back to the favorites list.
   3. 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 the item screen and return to list within 3 seconds.
   4. Security
      1. None
4. Location: Important, especially for newcomers. This tab lets the user know where they are picking up their order. It will also be able to navigate to the location of pickup with Google Maps, although the address is also provided should the user want to use another navigation app. As the business expands, new stores have the opportunity to open, allowing for more pickup spots. Therefore, the user will be able to establish a default location for travel and pickup. They 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/Outputs
      1. The Location Home screen will have the default location on it. The default will always be the original location, no matter the circumstance. The page will contain its address, the store’s unique deal, its hours, and a navigation option. If the user chooses to navigate to the location, the app will move to Google Maps or another navigation app of choice to begin navigation. If the business expands into a chain in the future, there will be the implementation of the ability to change store addresses.
   2. Processes
      1. Leads you to Google Maps for navigation if the user chooses to do so.
   3. Performance
      1. Should be able to transfer over to and start navigation in Google Maps in about 10 seconds.
   4. 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, they will be able to hold your personal information that is beneficial for using the app, including name and payment information. They 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 their account section, the user will be able to see how many points they have. They will be able to apply these points from the points menu.
   1. Inputs/Outputs
      1. The home screen for My Account will start by only having a create account option if the user has not logged in to their account on this device previously. In order to make their account, they must input their name, email, and password. They may then confirm their account creation, upon which an email will be sent to the given address containing the terms and conditions for having an account. Once those terms and conditions are accepted, the new home page will have all account information, including the information listed before as well as payment information, rewards points, and phone number if the user chooses to add one. Within this page, the user can also claim rewards. By choosing to claim rewards, all reward options will come up. After the reward is chosen, you will be moved to the order section
   2. 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.
   3. 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.
   4. 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, the user will have access to receipts from the past 4 years. This is helpful for keeping track of transaction history. First, they will be able to know if someone else got into their account if they have one. This is similar to bank statements, as users can see charges they know aren’t theirs. It also lets them look back at and favorite their orders. After trying something, if the user wants to see what it was so they 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 in terms of differences in the order.
   1. Inputs/Outputs
      1. The home screen for receipts will be your receipt list. While on this screen, the user will have the option to favorite the order and keep it in their favorites list. They may also select a receipt to view the details of. This will move them to a screen containing the order number, location, date, time, price, rewards points earned, payment method, and items ordered. They should be able to move back to the home page to view another receipt.
   2. 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.
   3. Performance
      1. Should know in advance if user has an account to track receipts with.
      2. Should know in advance if user account 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.
   4. 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 [o1] 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. If SSD does not open in visio, try using diagrams.net.