**Organic Food Store**

**Low-level Design Document**

# **1. Component design Details**

1. **Store location recommendation system**

* Google map API

Obtain the population density

Divide the maps into blocks, each with an population center

Divide the map into smaller grid

Iterate through each grid

Iterate through all population center within certain distance of the grid

Calculate the distance between each grid and population center

Compute the recommandation value with population / distance

Sort all values

Suggest the top solutions for the user to pick

Display the necessary information for each solution

* Display page

Display the map using google map API

Iterate through all the population centers

Display the population center on the mao in forms of heat map

The higher the population density, the deeper the color and and larger the circle

Iterate through all the solutions

Draw an clickable icon on suggested store location

When user click on the icon

Display the reachable population centerby distance.

1. **Online shopping website**

* Homepage
  + List of items:

Display name, price, weight, quantity in shopping cart and quantity left in inventory.

Initialize weight of order to 0

Initialize price of order to 0

Initialize Quantity in shopping cart to 0;

Get the quantity left in inventory from 4) Inventory management System;

While user doesn’t click on Shopping Cart Symbol or Tracking Order Button

If user click on the item

If Quantity left in inventory <= 0

Pops up warning: Not enough in stock

Back to Homepage

Else

Update Quantity in shopping cart: + 1

Update Quantity left in inventory: - 1

Update Weight of Order: + weight of the item

Update Price of Order: + price of the item

* + Shopping Cart Symbol:

If user click on the Shopping Cart Symbol

Proceed to Shopping Cart Page

* + Tracking Order Button:

If user click on the Tracking Order Button

Proceed to Tracking Order Page

* Shopping Cart Page
  + List of items added, with “+”, “-” and “Delete” button:

While user doesn’t click on Checkout Button

If user clicks on “+” button

Update Quantity left in inventory: - 1

If Quantity left in inventory <= 0

Pops up warning: Not enough in stock

Back to Shopping Cart Page

Else

Update Quantity in shopping cart: + 1

Update Weight of Order: + weight of the item

Update Price of Order: + price of the item

Else if user clicks on the “-” button

If Quantity in shopping cart <= 1

Pops up warning: can’t decrease the quantity any more

Back to Shopping Cart Page

Else

Update Quantity in shopping cart: - 1

Update Quantity left in inventory: + 1

Update Weight of Order: - weight of the item

Update Price of Order: - price of the item

Else if user clicks on the “Delete” button

Update Quantity left in inventory: + Quantity in Shopping Cart

Update Weight of Order: - weight of the item \* Quantity in Shopping Cart

Update Price of Order: - price of the item \* Quantity in Shopping Cart

Update Quantity in shopping cart to 0

Delete the item from the list

* + Checkout Button

If user click on the Checkout Button

If Weight of Order < 20 pounds

Proceed to Checkout Page

Else

Pops up warning: weight is over 20 pounds

Back to shopping cart page

* Checkout Page
  + Read and Store user inputs

First Name, Last Name, Email, Phone number

Address

Payment info: credit/debit card number, expiration date, CVV, billing address

* + Confirm Button

If user clicks the Confirm Button

Check user address info with Google Map

Check user payment info with Bank

If both address info and payment info are correct

Generate Order number

Generate Estimated Delivery time

Proceed to Thanks for order page

Else

Pops up warning: Wrong info, check and enter again

Back to checkout Page

* Thanks for order page
  + Get and Display Order info:

Order number, First Name, Last Name, Email, Phone number, Address, Estimated Delivery time

* + Track Order Button

If user click the Track Order Button

Proceed to Tracking Order Page

* Tracking Order Page
  + Read and Store user inputs:

Order number, phone number OR email

* + Match user input to find the order
  + Tracking Order using 3) Delivery Tracking system

1. **Delivery tracking system**

* Find drone assigned to delivery
  + Input: address
  + Return: drone number, delivery time, address
* Get delivery information from Google Maps
  + Input: address, drone number
  + Return: route, drone position, estimated remaining time

1. **Inventory management System**

* Display List of items in Inventory:

Name, Quantity, and “Update” button

* Employee login
  + Get Employee ID and password
  + Inform user if credentials are invalid, reset fields
* Restock item
  + Verify employee is logged in
  + If employee not logged in, display login page

If user clicks the “Update” button

Promote user to enter the Quantity Restored

Read and store the user input for Quantity Restocked

Update Quantity: + Quantity Restocked

* If item quantity <= 10
  + Get all items with quantity <=10
  + Display item names

**2. Backlog**

|  |  |  |
| --- | --- | --- |
| **Component** | **Tasks/Functions** | **Estimated time of completion** |
| Store location | Get Population Info | 2 day |
|  | Calculate delivery time? | 3 day |
|  | Decide location | 3 day |
| Display page | Display the information | 3 day |
| Shopping Website | Design web pages | 3 days |
|  | Build home page | 4 days |
|  | Build shopping cart page | 3 days |
|  | Build checkout page | 2 days |
|  | Build thanks page | 2 days |
|  | Build track page | 2 days |
|  | Test web pages | 3 days |
| Delivery Tracking | Create drone database | 2 days |
|  | Connect with Google Maps | 3 days |
|  | Display delivery information(route, positional data) | 3 days |
|  |  |  |
| Inventory management | Create inventory database | 2 days |
|  | Build inventory update page | 2 days |
|  | Low inventory notification (pop-up upon opening inventory page) | 2 days |
|  | Employee verification | 1 day |
|  |  |  |