Team 5

Section 101  
Team Gopher Milk

Project Milestone 2 -Features, Functional/Non-Functional Requirements, Project Plan

**Features of Shelf Life:**

* **Create account**
  + Creating individual menu items specifying amount of ingredients used in each dish
  + Specify price of each menu item
  + Create username and password
* **Manager/employee login**
  + User Name
  + Password

* **Order form**
  + Used to re-order ingredients
  + Shows a predicted amount of each ingredient to order
  + This predicted value would change based off of the amount of usage of that ingredient through the past week (or any other specified block of time)
  + User can order the predicted amount or a different amount
  + When submitted the inventory and profit is updated.
* **Inventory**
  + Each menu item will have a specific amount of each individual ingredient.
  + When a menu item is ordered, the ingredients will be subtracted from the inventory.
  + When an order form is submitted, the inventory is updated
  + We will display how much of each individual ingredient is left, along with the percentage.
  + When an item runs low(below 30%), an alert is sent to dashboard advising the manager to re-order. The system will recommend how much of the ingredient should be re-ordered based on past usage.

* **Sales**
  + The sales function keeps track of profit/costs for the restaurant. For each dish ordered, the price of the dish is added to the restaurant's profit.
  + It also allows the user to keep track of food costs such as total money spent on food, and money spent on each individual ingredient.
  + Under the sales tab there will be an option to display past sales. The user can view sales from the past day, week, month or year.
* **Dashboard**
  + Acts as the home page
  + Displays low inventory warnings if any exist. Also shows a brief summary of the sales for the current day.
* **Analysis**
  + This would focus on a comparison of specific dishes
  + Would display a graph of a chosen dishes sales of selected amount of time
    - Month, quarter or year
  + Would show which dishes get ordered more in certain seasons/months
  + Would allow managers to see what ingredients they should stock up on for an upcoming month/season
* **Taking an Order**
  + This function is where the server enters a customers order
  + The user clicks on a button with the dishes name to send an order
  + This function sends the information of the dish, cost, ingredients, date and time to other functions

**Functional/Non-Functional Requirements:**

* **Create account**
  + Functional Requirements
    - Allows user to create their username and password
    - User enters their restaurant's name
    - User enters in each dish, the ingredients and amount of each ingredient used in the dish.
    - Take in an initial count of ingredients
    - User enters the price of dish
  + Non-functional Requirements
    - The initial account setup would be determined by if the account hasn’t previously been in existance
    - Saves username and password to database
    - Saves dish name, ingredients and cost to database
* **Order form**
  + Functional Requirements
    - Show/be able to select specific ingredients to order
    - Show an estimated amount of the ingredient to order
    - Display estimated cost
    - Submit cost
  + Non-functional Requirements
    - It should predict the amount of that selected ingredient to order based off of previous week’s (or any other time frame) sales/usage
    - Order form has suggested values loaded within 1.4 seconds of opening page
    - Inventory and profit is updated within 2 seconds of submitting order
    - The ingredient being selected must exist within the database
      * This would be shown through an HTML dropdown menu
* **Inventory**
  + Functional Requirements
    - Show/display how much of a selected ingredient is in stock
    - Display percentage of ingredient left
    - If the stock of a specific ingredient is too low (below a set threshold) it would be highlighted in a specific color (i.e red, orange etc)
  + Non-functional Requirements
    - The stock of the selected ingredient would be shown as a number greater than or equal to 0
    - The ingredient would be highlighted in the specified color if the stock falls below 30% of the stock after the previous order
* **Sales**
  + Functional Requirements
    - Displays the total profit for the restaurant.
    - Allows the user to keep track of food costs such as total money spent on food, and money spent on each individual ingredient.
    - Display past sales in graph. The user can view sales from the past day, week, month or year.
  + Non-functional Requirements
    - For each dish ordered, the price of the dish is added to the restaurant's profit
    - When an order form is submitted, the cost if subtracted from the profit
    - Profit updated within 2 seconds of an order or order form submitted
    - When user wants to view a graph of sales:
      * Add up sales from specific time user selected
      * Graph data within 1.4 seconds
* **Dashboard**
  + Functional Requirements
    - Displays low inventory warnings if any exist.
    - Shows a brief summary of the sales for the current day
  + Non-functional Requirements
    - If no sales happened on that day they would be 0 by default.
    - Alert sent to page if inventory is less than 30% for an ingredient
    - If no low inventory warnings exist it wouldn’t show any warnings
      * If a low inventory warning flag hasn’t been set, then don’t display
* **Analysis**
  + Functional Requirements
    - Allows user to pick a dish to analyse
    - Graphs number of times the dish has been ordered vs profit
      * Time frame could also be selected as previous year, previous three months, or previous month
  + Non-functional Requirements
    - If the account hasn’t existed for more than the time frame selected, then display “Insufficient data”
    - Adds up times that dish has been ordered, and sums up profit from that dish within time frame
    - Graph should be displayed within 1.5 seconds
* **Taking order**
  + Functional Requirements
    - To send an order, the user clicks on button matching dish name
    - Buttons of all the dish names are visible
  + Non-functional Requirements
    - Inventory and sales should be updated within 2 seconds of submitting order

**Project Plan**

Gantt Chart: created using application Teamweek. Link here: <https://app.teamweek.com/#pp/DUueZ8NIyJfDHAZvDZgwqVwNuGy-Q_sn>

Screenshots:

