**MEAL PLANNER SPECIFICATION**

**TEAM FOODIES**

**Jeffrey Hermo**

**Vrajesh Bhavsar**

**Zeel Thakkar**

**Table of Contents**

[**Scope, limitations, and Project Terms** 3](#_Toc121988918)

[**Design Class Diagram** 4](#_Toc121988919)

[**Navigation Diagram** 5](#_Toc121988920)

[**Data Dictionary** 6](#_Toc121988921)

[**Entity Relationship Diagram** 9](#_Toc121988922)

[**Prototypes per Iteration** 10](#_Toc121988923)

[**What’s for Version 2** 33](#_Toc121988924)

[**Iteration Project User Stories** 34](#_Toc121988925)

[**GitHub Project Repository and Commits** 37](#_Toc121988926)

# **Scope, limitations, and Project Terms**

**SCOPE AND LIMITATIONS**

**Meal Planner Version 1.0**

* The application does not have functionality for notification. Orders, for now, would have to be texted back via the information in the email.
* The application does not have functionality for choosing a meal for customers. Choosing a meal would be done by texting back via the information in the email.

**TERMS**

**MEAL –** This is the food. It can be meat, vegetable, or both.

**MENU –** This is the food for a certain day of the week. Example: The meat menu for Wednesday is Hamburger.

# **Design Class Diagram**

**Diagram

Description automatically generated**

# **Navigation Diagram**

**Diagram

Description automatically generated**

# **Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **userCredentials Collection** | | |
| **Field name** | **Field type** | **Description** |
| **email** | **String** | **Email address of the user.** |
| **password** | **String** | **Random strings for login** |
| **status** | **String** | **If user is active or inactive** |
| **userDetails.firstName** | **String** | **First name of the user** |
| **userDetails.lastName** | **String** | **Last name of the user** |
| **userDetails.address** | **String** | **Address of the user** |
| **userDetails.address.houseNumber** | **String** | **House number of the address of the user.** |
| **userDetails.address.street** | **String** | **Street of the address of the user** |
| **userDetails.address.city** | **String** | **City of the address of the user** |
| **userDetails.address.province** | **String** | **Province of the address of the user** |
| **userDetails.address.postalCode** | **String** | **Postal code of the address of the user.** |

|  |  |  |
| --- | --- | --- |
| **meals Collection** | | |
| **Field name** | **Field type** | **Description** |
| **mealName** | **String** | **Name of the meal** |
| **mealDescription** | **String** | **Description of the meal** |
| **mealIngredients** | **String** | **Ingredients of the meal** |
| **mealPrice** | **String** | **Price of the meal** |
| **mealStatus** | **String** | **Status of the meal** |
| **mealType** | **String** | **Type of the meal** |
| **imageUrl** | **String** | **Url of the image** |

|  |  |  |
| --- | --- | --- |
| **menus Collection** | | |
| **Field name** | **Field type** | **Description** |
| **menuName** | **String** | **Name of the menu** |
| **meatMeal.mealsCollection** | **Map** | **Meal from meals Collection** |
| **vegetableMeal.mealsCollection** | **Map** | **Meal from meals Collection** |
| **bothMeal.mealsCollection** | **Map** | **Meal from meals Collection** |
| **imageUrl** | **String** | **Url of the image** |

|  |  |  |
| --- | --- | --- |
| **emails Collection** | | |
| **Field name** | **Field type** | **Description** |
| **deliveryDate** | **String** | **Date of the email delivery** |
| **sent** | **Boolean** | **True if email is already sent** |
| **mealPlanWeek.emailAddressList** | **Array** | **List of email address of customer that are active** |
| **mealPlanWeek.mondayMenu.menusCollection** | **Map** | **Menu from menus Collection** |
| **mealPlanWeek.wednesdayMenu.menusCollection** | **Map** | **Menu from menus Collection** |
| **mealPlanWeek.fridayMenu.menusCollection** | **Map** | **Menu from menus Collection** |
| **notes** | **String** | **Notes from the admin** |

# **Entity Relationship Diagram**

**Diagram

Description automatically generated**

# **Prototypes per Iteration**

**ITERATION 1:**

**Signup and Login Use Case**

1. **Company logo.**
2. **Signup Button – On Click, will go to user signup cases.**
3. **Login Button – On Click, will go to login page.**

**Graphical user interface, application

Description automatically generated**

**CUSTOMER USE CASE**

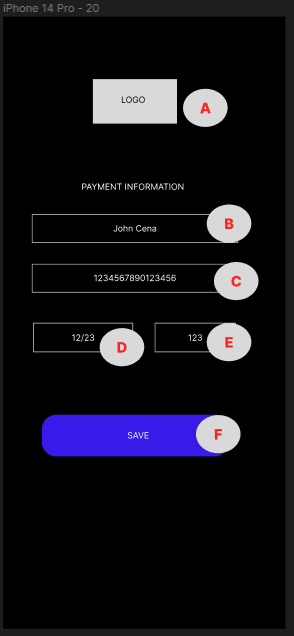
1. **Create an Account – Customer inputs first name, last name, house number, street, city, province, postal code, phone number, email, and password for registration.**
   1. **Company logo.**
   2. **First Name**
   3. **Last Name**
   4. **House Number**
   5. **Street**
   6. **City**
   7. **Province – Dropdown**
   8. **Postal Code**
   9. **Phone Number**
   10. **Email – this will be the username of the user.**
   11. **Password – login password for the account.**
   12. **Show password checkbox – to show the typed password of the user.**
   13. **Next Button – go to the next page (Add Payment Details).**

**A screenshot of a phone

Description automatically generated with low confidence**

**Validations:**

1. **All fields are required.**
2. **Phone number is only number and lock up to 10 numbers only.**
3. **Username (email) is unique. Cannot register the same username that has been used.**
4. **Only valid email format is allowed (example:** [**string@yahoo.com**](mailto:string@yahoo.com)**).**
5. **Add Payment Details – Customer adds card number, expiry date, and security code.**
   1. **Company logo.**
   2. **Full name on card – name on credit card.**
   3. **Card number – user’s credit card number.**
   4. **Expiry date – expiry date on card.**
   5. **Security code – CVC code on card.**
   6. **Save Button – saves the user profile on the system.**

****

**Validations:**

1. **Card number is locked to 16 numbers only.**
2. **Expiry date is locked to 4 digits only.**
3. **Security code is locked to 3 digits only.**
4. **Login User – user inputs an email address and password. This will be either an admin or customer login.**
   1. **Company logo.**
   2. **Username input – user’s email will be inputted as their username.**
   3. **Password input – password corresponding to the username input.**
   4. **Show password – shows the password input.**
   5. **Login Button – logins the user to the system.**

**Graphical user interface, application

Description automatically generated**

**Validations:**

1. **All fields are required.**
2. **Email format should be correct (Example:** [**string@yahoo.com**](mailto:string@yahoo.com)**).**
3. **Password should match the email.**
4. **Email should be existing in the system.**

**ITERATION 2:**

**Prerequisite: Admin has its own page as per below and meal button is available.**

1. **Meals Button – On Click will go to Meals Use Cases.**
2. **Users Button – On Click will go to Users Use Cases.**

**Graphical user interface, text, application

Description automatically generated**

**MEALS USE CASES**

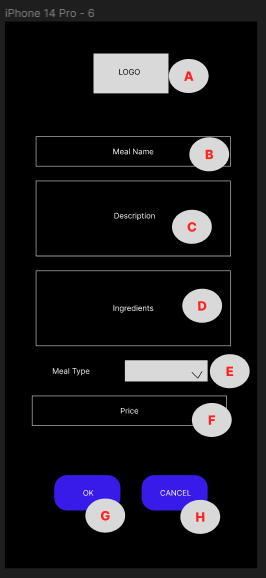
1. **VIEW MEALS – On Click of View meals button on the main meal menu, it should load all the meals available in the application.**
   1. **Search bar – This should show the meal as you type dynamically the name. (Name of the meal should be typed here).**
   2. **SORT– Filtering of list can be done in the search bar and sorting is via the sort dropdown.**
   3. **ADD MEAL button – will go to add meal menu (Add Meal).**
   4. **Location of image.**
   5. **Meal Name – On Click will go to View Meal.**
   6. **Company logo.**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validations: None**

1. **ADD MEAL – On clicking of Add button, should show a new screen with fields to add a new meal.**
   1. **Company logo.**
   2. **Meal Name – Unique.**
   3. **Description – Meal description**
   4. **Ingredients – Meal ingredients.**
   5. **Meal Type – Could be Meat, Vegetable, or Both.**
   6. **Price – Price of the meal.**
   7. **OK – Adds the new meal. Triggers also the validation check.**
   8. **CANCEL – Goes back to the main meal page.**

****

**Validations:**

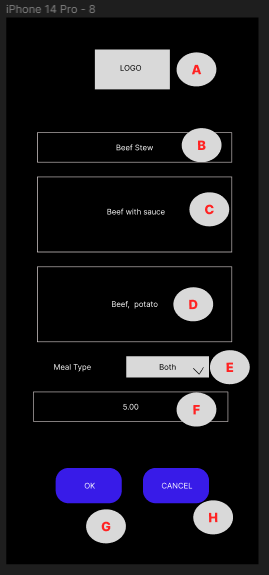
* **If the name of the meal already exists, it should not be allowed to add.**
* **All fields are required to have a value.**

1. **VIEW MEAL – On clicking of any meal in View Meals use case, should show a new screen with fields showing details of the meal. All fields are disabled.**
   1. **Company logo.**
   2. **Meal Name – Unique.**
   3. **Description – Meal description**
   4. **Ingredients – Meal ingredients.**
   5. **Meal Type – Could be Meat, Vegetable, or Both.**
   6. **Price – Price of the meal.**
   7. **UPDATE – Button for triggering update. On Click will go to Update Meal.**

**A screenshot of a phone

Description automatically generated with medium confidence**

1. **UPDATE MEAL – On clicking the update button from View Meal use case, all fields will be enabled except THE MEAL NAME.**
   1. **Company Logo.**
   2. **Meal Name – Unique.**
   3. **Description – Meal description**
   4. **Ingredients – Meal ingredients.**
   5. **Meal Type – Could be Meat, Vegetable, or Both.**
   6. **Price – Price of the meal.**
   7. **OK – Upon clicking will update the details of the meal.**
   8. **CANCEL – Will redirect to the meal view page.**

****

**Validations:**

* **All fields are required to have a value.**

**USER USE CASES**

**Prerequisite – Admin logins.**

1. **VIEW USERS – On the Click of View users button on the main meal menu, it should load all the meals available in the application.**
   1. **Company logo.**
   2. **Search bar – This should show the user as you type dynamically the name. (Name of the meal should be typed here).**
   3. **Users(2) – The number of users available in the application should be shown beside the user title.**
   4. **SORT– Filtering of the list can be done in the search bar and sorting is via the sort dropdown.**
   5. **Image of the user.**
   6. **User Full Name - Each row is clickable. Upon click will show all the details of the user and this is where the update use case will happen (View User).**

**Graphical user interface, application

Description automatically generated**

1. **VIEW USER – On click of any user, it will show the details of that user. Everything is not enabled.**
   1. **Company logo.**
   2. **First Name**
   3. **Last Name**
   4. **House Number**
   5. **Street**
   6. **City**
   7. **Province – Dropdown**
   8. **Postal Code**
   9. **Phone Number**
   10. **User Status - Can be Active or Inactive**
   11. **Update Button – On Click will go to update user (Update User).**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validation:**

* **None.**

1. **UPDATE USER – On clicking of the name of the user, a popup should appear with the details of the user.**
   1. **Company logo.**
   2. **First Name**
   3. **Last Name**
   4. **House Number**
   5. **Street**
   6. **City**
   7. **Province – Dropdown**
   8. **Postal Code**
   9. **Phone Number**
   10. **User Status - Can be Active or Inactive**
   11. **Ok Button – On click will trigger the validation and update the user.**
   12. **Cancel Button – On click will go back to user list page (View Users).**

**A screenshot of a phone

Description automatically generated with low confidence**

**Validations:**

* **Same validations as the Create an Account. All fields required.**
* **House number and phone number are digits only.**
* **Phone number should only accept 10 digits.**

**ITERATION 3:**

**Menu, Customer and Email Use Cases:**

1. **Menu Button – On Click, will go to View Menus.**
2. **Email Button – On Click, will go to Create Email.**

**Graphical user interface, application

Description automatically generated**

**MENU USE CASES**

**Pre-requisite: Admin logins to account.**

1. **VIEW MENUS - On clicking of the Menus button, it should load all the menu available in the application.**
   1. **Company Logo**
   2. **Search bar – This should show the meal as you type dynamically the name. (Name of the menu should be typed here).**
   3. **Sort – List can be sorted by A-Z or Z-A. Dropdown.**
   4. **Add menu – On click, will go to Add Menu use case.**
   5. **Menu list.**

**A screenshot of a phone

Description automatically generated with medium confidence**

1. **VIEW MENU – On click of the menu name, it will go to this page containing all the details of the menu:**
2. **Logo – Company logo**
3. **Menu Name – Menu Name**
4. **Location of Meal List for meat– On update, this will list all the meat meals.**
5. **Meat Meal – Meat meal for the menu.**
6. **Location of the Meal list for vegetable – On update, this will list all the vegetable meals.**
7. **Vegetable Meal – Vegetable meal for the menu.**
8. **Location of the Meal list for meat and vegetable (both) – On update, this will list all the vegetable meals.**
9. **Meat and Vegetable Meal (Both) – Meat and vegetable meal for the menu.**
10. **Update Button – On Click, will go to Update Menu Use case.**

**A screenshot of a phone

Description automatically generated with medium confidence**

1. **UPDATE MENU**
   1. **Logo – Company logo**
   2. **Menu Name – Menu Name**
   3. **Location of Meal List for meat–list all the meat meals.**
   4. **Meat Meal – Meat meal for the menu.**
   5. **Location of the Meal list for vegetable –list all the vegetable meals.**
   6. **Vegetable Meal – Vegetable meal for the menu.**
   7. **Location of the Meal list for meat and vegetable (both) – list all the vegetable meals.**
   8. **Meat and Vegetable Meal (Both) – Meat and vegetable meal for the menu.**
   9. **Ok Button – On Click, will update the menu (if there is any changes).**
   10. **Cancel Button – On click will go back to the previous page (View Menus)**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validations:**

* **All meal fields are required.**
* **If there is no changes and user clicked the OK button, will throw a toast “No field was updated”**

1. **ADD MENU – On click of the add button in the view menus screen, it will go to the menu creation screen.**
2. **Logo – Company logo**
3. **Menu Name – Menu Name**
4. **Location of Meal List for meat–list all the meat meals.**
5. **Meat Meal – Meat meal for the menu.**
6. **Location of the Meal list for vegetable –list all the vegetable meals.**
7. **Vegetable Meal – Vegetable meal for the menu.**
8. **Location of the Meal list for meat and vegetable (both) – list all the vegetable meals.**
9. **Meat and Vegetable Meal (Both) – Meat and vegetable meal for the menu.**
10. **Ok Button – On Click, will add the menu (if there is any changes).**
11. **Cancel Button – On click will go back to the previous page (View Menus)**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validation:**

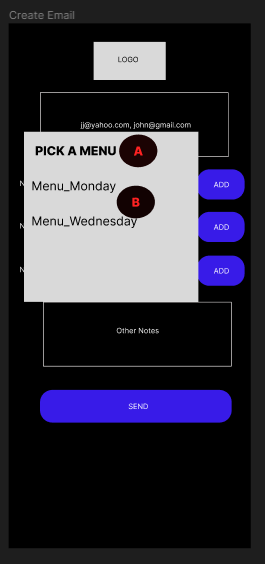
1. **All fields are required.**
2. **Menu name is unique.**
3. **CREATE EMAIL – Create menu email to send to all customers.**
4. **Company Logo**
5. **Email list – This is automatically filled up on load. Will retrieve all active emails of customers.**
6. **Next available date for Monday, Wednesday, and Friday – As we have 3 days of menu deliveries (Monday, Wednesday, and Friday), it will post automatically the next date of those days.**
7. **Menu – This will contain the menu for that day.**
8. **Add Button – On click, will popup a dialog containing menu list (menu created in the system).**
9. **Other notes – Will contain extra messages for the customers.**
10. **Send Button – On click, will send the email to all the listed email.**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validations:**

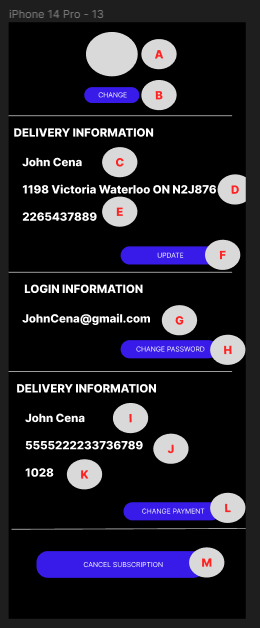
1. **Contains atleast 3 menus.**
2. **There are active customers.**
3. **ADD MENU FOR EMAIL – Adds menu to the email to be sent to the customers. On click of the menu, will be added to the day where the add button was clicked.**
4. **Title.**
5. **Menu Name – Menu available.**

****

**CUSTOMER USE CASE**

**Prerequisite: Customer logins to his/her account.**

1. **VIEW USER PROFILE – Customer views personal information, login information, and payment information.**
   1. **User photo.**
   2. **Change photo button – On click, will open local gallery and can get and change the photo.**
   3. **Full name – User’s full name.**
   4. **Address – User’s full address (delivery address)**
   5. **Phone number**
   6. **Update Button – On click, will show a dialog where user can update user personal information.**
   7. **Email address**
   8. **Change password – On click, will show a dialog where user can change password.**
   9. **Card Name – Name on the card of the user.**
   10. **Card Number – Card number on the credit card.**
   11. **Expiry date – Credit’s card expiry date.**
   12. **Change Payment – On click, will change payment details of customer.**
   13. **Cancel Subscription – Will cancel user’s subscription to service.**

****

1. **UPDATE PERSONAL DETAILS – Customer updates first Name, last Name, house number, street, city, province, postal code, and phone number.**
   1. **Company logo.**
   2. **First Name.**
   3. **Last Name.**
   4. **House Number.**
   5. **Street.**
   6. **City.**
   7. **Province.**
   8. **Postal Code.**
   9. **Phone Number.**
   10. **Cancel Button – On click, will remove the popup.**
   11. **Update Button – On click, will update the personal details of user.**

****

**Validations:**

* **All fields are required.**

1. **UPDATE PASSWORD – User change password on the system.**
   1. **Company logo.**
   2. **Current password – User needs to input current password to change the password.**
   3. **New password – new password that the user will use.**
   4. **Confirm new password – needs to retype new password for confirmation.**
   5. **Show password – On click of the checkbox, will show all the password inputted.**
   6. **Cancel button – will remove the popup.**
   7. **Update button – will update the password in the system.**

**Graphical user interface, application

Description automatically generated**

**Validations:**

* **All fields are required.**
* **Old password must match with current password.**
* **New password must be the same with confirm new password.**

1. **UPDATE PAYMENT DETAILS**
   1. **Company logo.**
   2. **Name on card.**
   3. **Card number.**
   4. **Expiry Date.**
   5. **CVC/Security code.**
   6. **Cancel button – Will remove the popup.**
   7. **Update button – will update the payment details in the system.**

**A screenshot of a phone

Description automatically generated with medium confidence**

**Validations:**

* **All fields are required.**

# **What’s for Version 2**

* The email functionality will now only be used for notification. The texting back for your orders will not be used. Instead, the application will have ordering functionality in the application.
  + Customer will be notified via email and in the application that a menu is now available for the week.
  + Customer can now choose what meals to have in the application.
* UI improvements:
  + Light mode will be available.
  + More customization.

# **Iteration Project User Stories**

**Iteration 1**

**Text

Description automatically generated**

**Iteration 2**

**Text

Description automatically generated**

**Iteration 3**

**Text

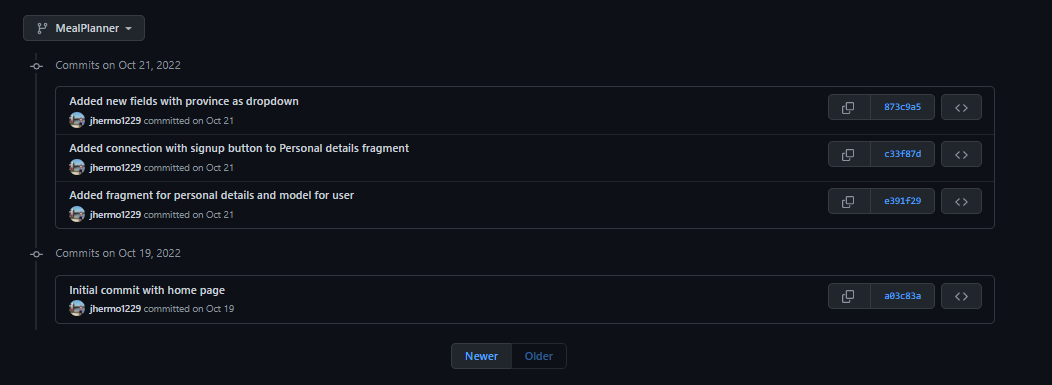
Description automatically generated**

# **GitHub Project Repository and Commits**

**A screenshot of a computer

Description automatically generated with medium confidence**

[**https://github.com/jhermo1229/MealPlanner**](https://github.com/jhermo1229/MealPlanner)

****

**A picture containing graphical user interface

Description automatically generated**

**Total of 109 commits from October 19, 2022 until December 14, 2022**