**Team 5 – MealBrain**

Team Members:

Aaron Voymas (495)

Wessley Monnin (495)

Marcel Demers (394)

Justin Mosley (394)

Jacob Pantall (294)

Mickey Jose (294)

**COMP 495**

**Software Modules**

A description of the key functional modules in the application and how they will communicate and interface with one another. Note that a functional module will have a front-end business logic in the back-end portion.

1. Alert/Notification Module (that we will have notification and confirmation for actions that affect the database?): this module handles the ability to generate notification and alert modals, to display information and/or ask for confirmation.
2. Settings Module (themes etc): this module handles user-level preferences such as app theme.
3. Accessibility Module (a set of standards and practices to apply to the rest of the modules): this is less a module and more to call attention to an intended set of practices that we want to include around accessibility. A primary means to implement these are using the SemanticProperties of the xaml elements.
4. Authentication Module: this module handles authentication services both locally and with authentication outside of the app.
5. User Account Module: this module handles creation, modification, and deletion of local user accounts. This module interacts with the user account tables in the db.
   1. Local User Account: a user local to the app who can create and manage recipes, meal plans, and grocery lists, as well as set their own theme and manage their own account information
   2. User Account Details Page: page where the user enters a display name for the account, selects the account type (Personal or Business), add an email address, and upload a profile image. A user’s information can be saved and updated here.
   3. User Account Page: page where a user can add a local user account, or select a local user account to modify.
6. Recipe Module: this module handles creation and management of recipes. Recipes needs to interact with the ingredients tables and the recipes tables in the database.
   1. Recipe Card: a recipe card contains the information for as single recipe, is linked to a single local user account, and can be used with meal plans and grocery lists. It consists of a name, description, optional image, prep time, cook time, number of servings, list of instructions, and list of ingredients.
   2. Recipe List Page: page where a user can add, or select a recipe card to modify.
   3. Recipe Details Page: page where a user enters a recipe name, description, prep time, cook time, number of servings, list of instructions, list of ingredients, and upload an image. User can also export a recipe to file and Share a Recipe to social media (stretch goal).
7. Meal Plan Module: this module handles creation and management of meal plans. Meal plans interacts with the recipe and grocery list modules. This module interacts with the meal plan tables in the db.
   1. Meal Plan: a meal plan contains information for a single meal plan, is linked to a single local user account, and can be used with grocery lists. It consists of a name, date-range, the days in the week, and the different meal times/spaces (breakfast, lunch, dinner, snacks), and recipes assigned to meals.
   2. Meal Plan List Page: page where a user can add a meal plan or select one to modify.
   3. Meal Plan Details Page: page where a user enters a meal plan name, selects the date-week range, and can add days, meal times, and select from their recipes. User can also export a meal plan to file, start a grocery list with the meal plan, and Share the Meal Plan to social media (stretch goal).
8. Grocery List Module: this module handles creation and management of grocery lists. Grocery lists interact with meal plans and recipes. This module interacts with the grocery list tables in the db.
   1. Grocery List: a grocery list contains information for a grocery list generated based upon the meal plan(s) and recipe(s) selected, including a name, meal plan(s) and recipe(s) selected, and a list of ingredients with the total amounts of each that will be needed to create everything from the meal plans and recipes chosen.
   2. Grocery List Page: page where a user can add a grocery list or select one to modify.
   3. Grocery List Details Page: page where a user enters a grocery list name, selects meal plan(s) and/or recipe(s), and can generate the total ingredients of each kind needed. User can also export a grocery list to file.
9. Measurement Converter Tool Module: this module is for a built-in tool so the user can convert between measurements. This module interacts with the measurements table in the database.
   1. Measurement Converter Tool Page: page where the user can enter a decimal or fraction value, select measurement types to convert from and too, a button to invoke the conversion, a display of the converted value, and a list of common fractional measurements and their decimal equivalents for reference.
10. Social Media Module (stretch goal): this module handles authenticating with social media accounts, invoking their apis to make posts, and packaging the shared item for the post.
11. Meal Brain for Business Module (stretch goal): this module handles the elements involved with the Business Plan account types, including bulk orders and bulk order lists.
12. Cloud Integration Module (stretch of a stretch goal): this module handles registering with the cloud service, linking the cloud account to a user on the device, and syncing the recipes, meal plans, and grocery lists, for that user.

**Use Case Diagrams**

A diagram showing the major use cases that you have already determined from your requirements gathering. This site gives examples of use case diagrams. Use Case diagram/flow required.

A diagram of a company

AI-generated content may be incorrect.

**Database Architecture**

An entity relationship diagram showing the foreign key/primary key relationships and ordinality in the database. ERD required

* Measurement (pk uuid, measurement name, measurement abbreviation)
* Measurement Conversion (pk uuid, fk measurement from, fk measurement to, conversion factor)
* Recipe (pk uuid, recipe name, recipe desc)
* Tags (pk uuid, tag name)
* Recipe Tags (pk uuid, fk recipe, fk tag)
* Recipe Ingredients List (pk uuid, fk recipe, quantity, fk measurement, fk ingredient)
* Recipe Instructions List (pk uuid, fk recipe, step number, instruction)
* Meal Plan (pk uuid, meal plan name, meal plan date-week)
* Meal Plan Details (pk uuid, fk meal plan, fk day of week, fk meal type, fk recipe)
* Grocery List (pk uuid, fk meal plan, name)
* Bulk Order (pk uuid, name, quantity, fk recipe)
* Bulk Order Details (pk uuid, date/date-range, fk bulk order)
* User Account (local) (pk uuid, name, email, fk account type)
* Account Type (pk uuid, name)
* Ingredients (pk uuid, ingredient name)
* Days of the Week (pk uuid, day name)
* Meal Type (Breakfast, Lunch, Dinner, Snack) table or enum? (pk uuid, meal type name)

A computer screen shot of a computer program

AI-generated content may be incorrect.

**Application Architecture**

A diagram or description showing how the architecture of the application will work. For example, you may have a web API with three layers and a client side you the user interface and a back-end database. It should be very clear what layers you have in your code so that people know where they should be putting their code when they work on unassigned use case.

A diagram of a software company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.