San Francisco State University
SW Engineering CSC 648 – 848
MealSight
Section 01 - Team Interpreter

Aaron Singh (Team-Lead) | Harrison Rondeau (Front-End) | Banaz Sinjary (Front-end)

Zachary Colbert (Back-end) | William Chin (Scrum-Master)

Ramit Singh (GitHub-Master) | Vikram Rai Singh (Front- End)

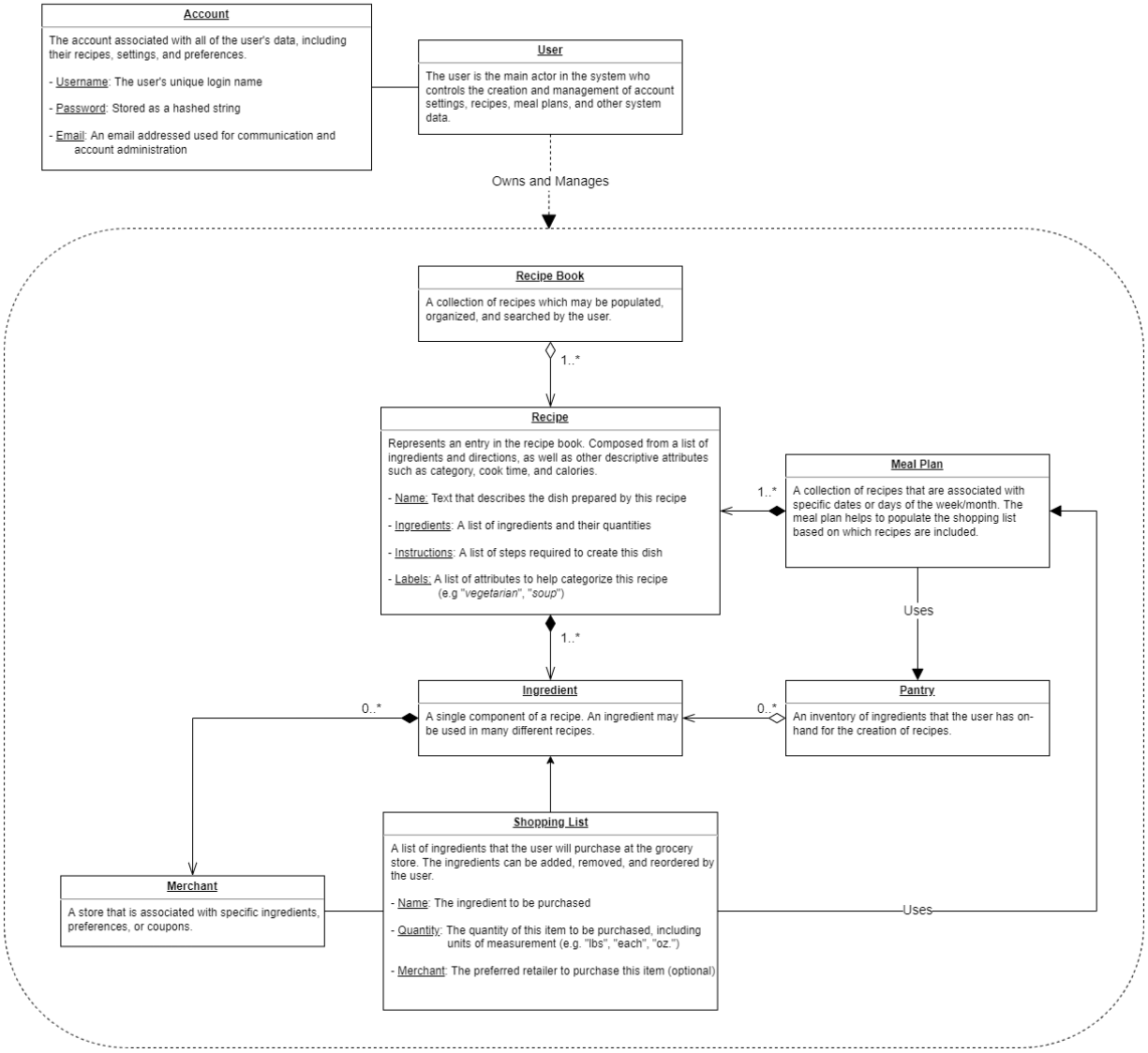
Milestone 2
10/5/2022

Revision History Table

Revision ID	Revision Date	Revised By

1. Data Definitions V2

This should be reasonably consistent with Milestone 1 but should be expanded as needed and refined as per instructors' feedback. Major data items that comprise of sub-data items must be defined in full (list all its sub-data items, and for images/video list formats, max size etc.). **You must use all the data definitions and names consistently in all documents and SW, including UI text, naming for main variables, classes and database elements etc.** Focus on data items unique and important to your implementation. Be sure to cover ALL items critical to your project and especially those providing a competitive advantage. At this stage data describing user privileges, and main info (raw data, metadata, supporting data) must be fully defined (as much as it is possible at this stage)



2. Functional Requirements V2

Expand functional requirements from Milestone 1 into Milestone 2, with more details as necessary. Keep the same reference numbers with respect to Milestone 1 (i.e. if high level requirement was number R.3. in Milestone 1, then Milestone 2 more detailed requirements of requirement R.3. are R.3.1., R.3.2. etc.).

Prioritize each requirement/spec with 1, 2, 3. (1-*must have*; 2 – *desired*; 3 – *opportunistic* as defined in the class). To develop these priorities on behalf of the user, and make your application complete for usability, marketing and business aspects. The priorities you set later in Milestone 3 and 4 will constitute your commitment (especially priorities of 1).

There are three major components:

- Recipe Book: API used to store database of recipes
- Meal Planner: Tool to assist users in the selection of meals
- Shopping List: Asynchronous tool that lists the ingredients for desired recipes.

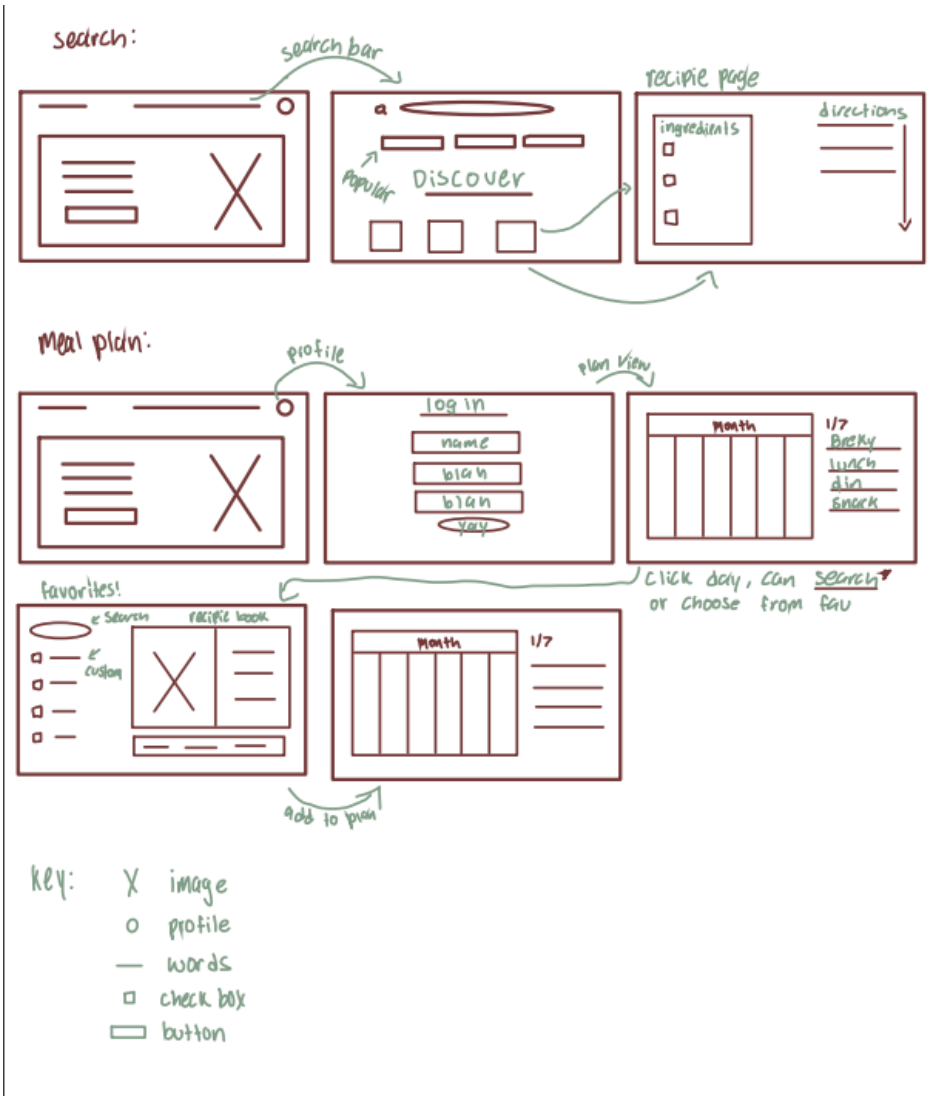
ID	Component	Description	Priority
0001	Recipe book	Recipes can be imported from another website	Low
0001.1	Recipe book	Retrieve the text from the URL where the recipe is located	Low
0001.2	Recipe book	Parse the recipe ingredients and directions from webpage text	Low
0002	Recipe book	Users can manually create and edit their recipes	High
0003	Recipe book	Users can rate their recipes	Low
0004	Recipe book	Users can add notes and comments to recipes	Medium
0005	Recipe book	Must be searchable	High
0006	Meal planner	Meals are assignable to specific dates or weekdays	High
0007	Meal planner	User editable: User can add, remove, or move meals to different dates.	High
0008	Meal planner	Suggestions for meals are generated based on preferences	Low
0008.1	Meal planner	Keep a history of user's past recipes from meal plan	Low
0008.2	Meal planner	Use recipe history and ratings to suggest favorite meals	Low
0008.3	Meal planner	Suggest recipes with ingredients similar to the user's highly rated recipes	Low
0009	Meal planner	Calendar and list view	Medium
0010	Shopping list	Must be user editable	High
0011	Shopping list	Populate ingredients from meal plan	Low
0012	Recipe book	Recipe view with ingredient checklist	Medium
0013	User Account	Keeps track of the Recipe book, Meal Planner, and shopping list	High

3. UI Mockups and Storyboards (high level only)

Create storyboards for 5~6 major functional requirements (e.g., Priority 1 requirements). Start with black and white wire diagrams focusing on basic UX flows for 5~6 major functional requirements. Create simple "storyboards" (sequence of mockups) which represent the functional requirements.

The format for UI mockups is very flexible. Do not use graphics or colors yet (unless absolutely necessary), it draws attention from basic UI concepts (functions, behaviors, layouts, flow…).

Once the storyboard is developed, perform the UX validation meeting so that UX principles are ensured. Include the summary of your UX validation meetings in your M2 documentation report.



4. High level Architecture, Database Organization

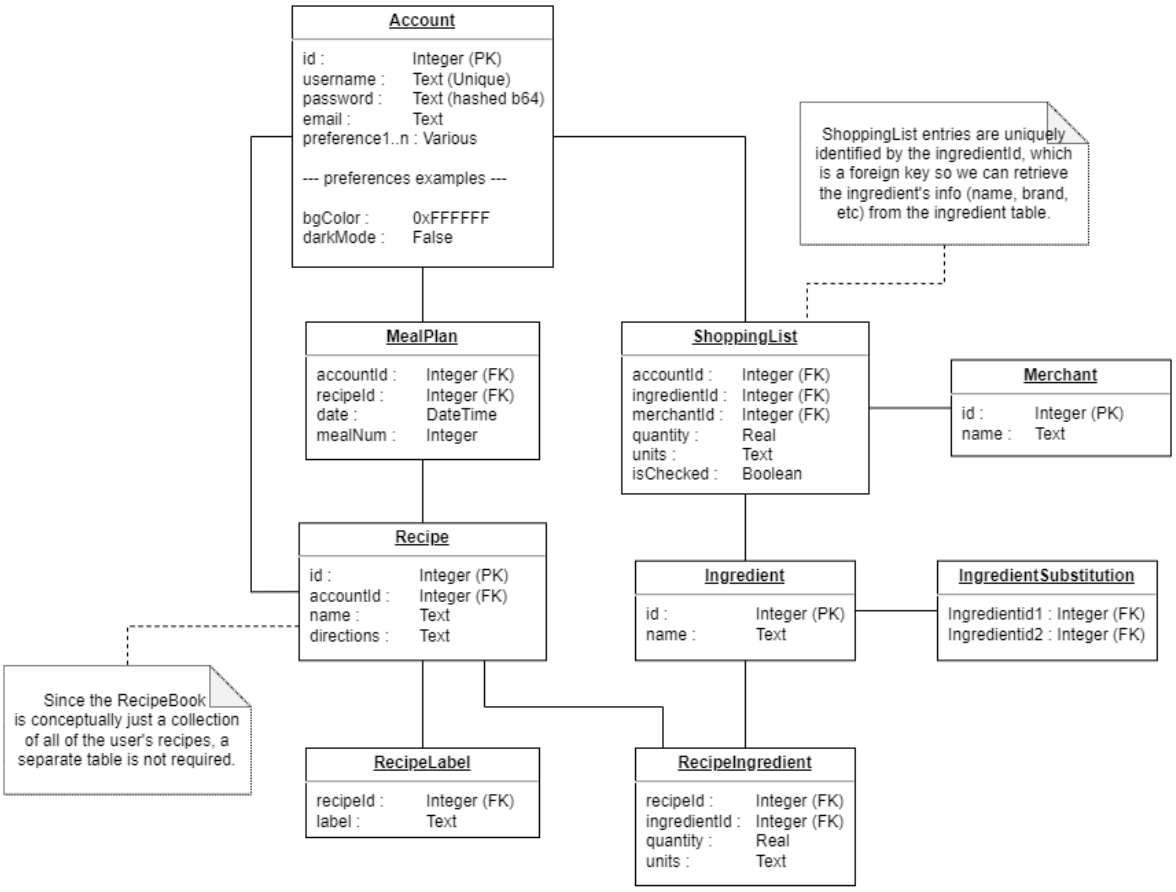
DB organization: Describe the main database schema/organization (high-level), e.g., list main DB tables (e.g. their titles) and columns in each DB table. Make sure the titles and column names are easy to understand in plain English and consistent with data definitions in Section 1 above.

Add/Delete/Search architecture:

In regard to the functional requirements, please specify which DB operations are permitted on DB (e.g., what DB entries will be added, searched, deleted, and displayed)

The user will be allowed to create, edit, and delete recipes. They can create ingredients (by adding a new ingredient as part of a recipe) but ingredients cannot be deleted. The user may interact with the meal plan by adding existing recipes to the meal plan, moving those recipes to different dates, or removing those recipes from the meal plan. The shopping list is populated from the existing list of ingredients by adding a recipe to the meal plan, or via manual input by the user. The user may edit or remove these shopping list entries at will.

Database Diagram including major tables and their relationships:



4.5 APIs @Harrison

Your own APIs : describe and define at high-level major APIs that you will create among your modules.

Package	Usage
tsc-watch	used to automatically restart the server when a typescript file is changed, recompiling to javascript if needed
ts-node	used to run the backend typescript files
express	used to handle server routes and HTTP methods
bcrypt	used to hash passwords
mysql2	used to interact with SQL database

nodemon	used to watch any files not watched by tsc-watch

What HTTP request and responses are defined across backend & frontend

In the backend, what route functions are needed to implement

If you are using the 3rd party API, please describe them in your architecture.

If you are using open-source components, please describe them in your architecture.

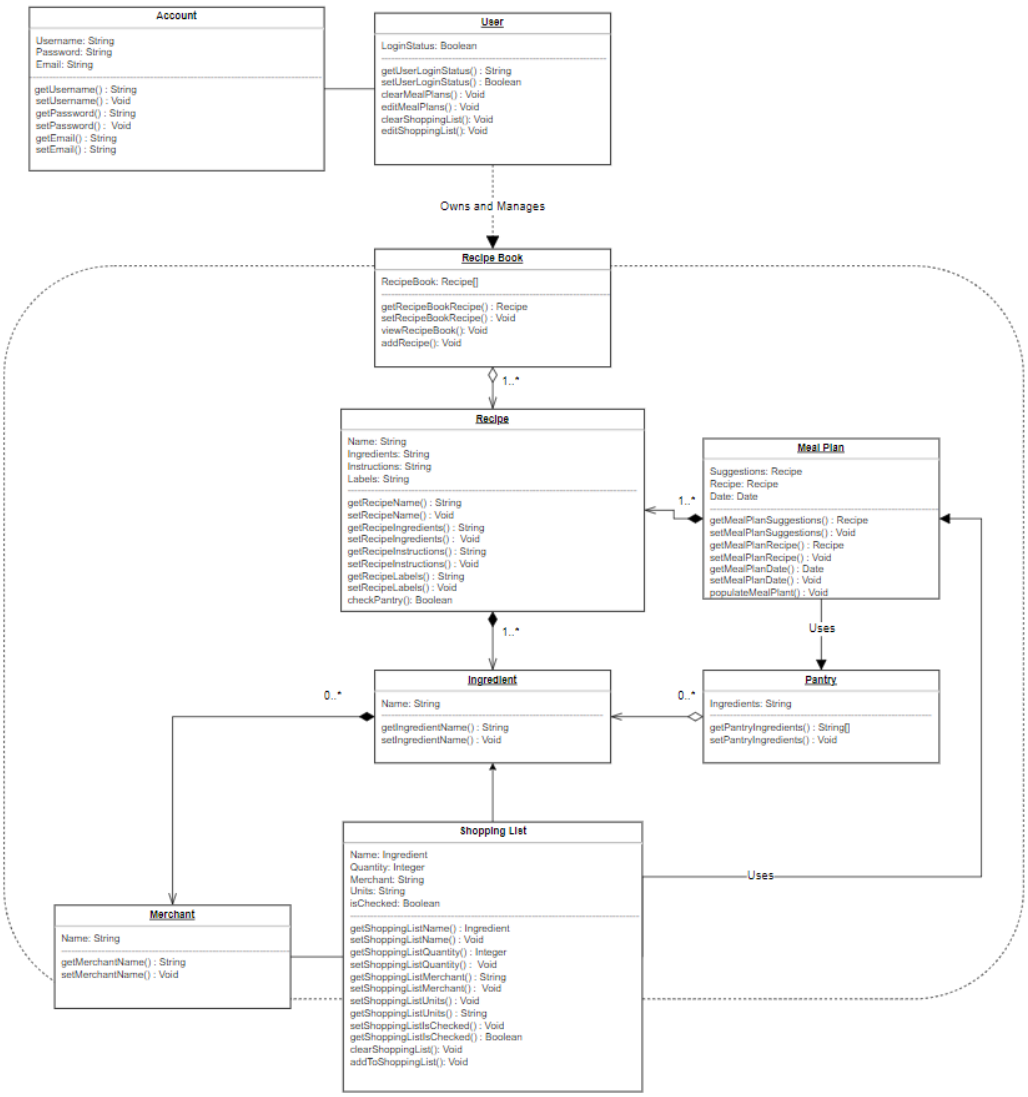
If you have changed SW tools and frameworks or added any new one, please describe it.

5. High Level UML Diagrams @William

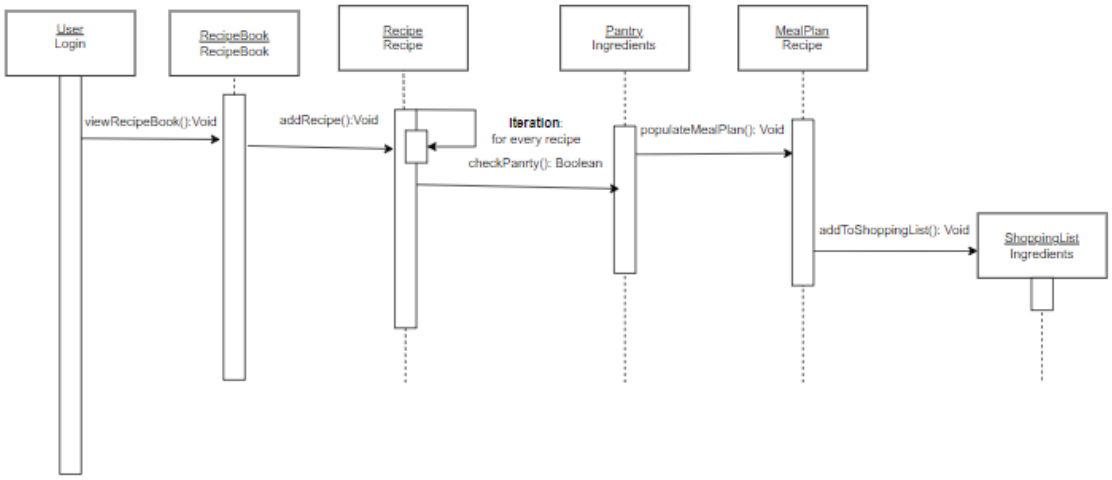
Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet.

For Milestone 2, provide only:

High-level UML class diagrams for implementation classes of core functionality, Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your web site. For UML, you could find many references including http://edn.embarcadero.com/article/31863.



High-level sequence diagrams: for ~5 major functional requirements, please develop UML sequence diagram.



6. Identify *actual* key risks for your project currently

Identify *only actual and specific* risks in your current work such as

- *skills* risks and mitigation plan
    - *Do you have a proper study plan to cover all the necessary technologies?*
    - Many Members don't have knowledge of mongoDB and how to connect the database to nodeJS
    - Not familiar with converting API data into a column for our database
    - Each backend will take 1-2 hours to learn how to properly implement the Apis into the Database
- *schedule* risks
    - Does your team have a team schedule for every member including their detailed task?
    - Notion is being used for tasks, however, no schedules for each detailed task
- *teamwork* risks (any issues related to teamwork);
    - *Everybody is on the meeting regularly?*
    - *Everybody keeps his/her pace? If not, what is your plan to mitigate the risks?*
    - Implement 1 more meeting around Friday to better communicate with the rest of the team
- *legal/content* risks (can you obtain content/SW you need legally with proper licensing, copyright).

Tell us how do you plan to resolve each actual risk you have. The key is to resolve risks as soon as possible. Categorizing risk as above helps a lot in managing them. Be brief: identify the risk and explain (2-3 lines), list how you will address this issues' (2-3 lines), and share with every team members.

7. Project management

Describe in no more than half a page how your team managed M2 tasks including

- how each member's progress is being shared in scrum meeting? Is it transparently shared?
- Outside the scrum meeting, do you have a tool to manage each member's task?
- Currently we are using Notion to manage each members tasks. We are splitting tasks up for each member, but as there are 7 members, integrating the workload can be a little challenging.