Do’s 8 - Deliverable 4 - Report

Requirements

Desktop

This increment was dedicated largely to database functionality - this is the core of the app since the data you input / manipulate on the desktop *ultimately* needs to end on the mobile phone for when you go shopping.

- Recipe browser database functionality
    - Recipes are pulled from database on program launch
    - Recipe table updated when recipe is added
    - Recipe table updated when recipe is deleted
- Meal plan database functionality
    - Meal plan is pulled from database on program launch
    - Meal plan pushes to database
- Shopping list generated from meal plan
    - Considers inventory
    - Considers additional items

Mobile

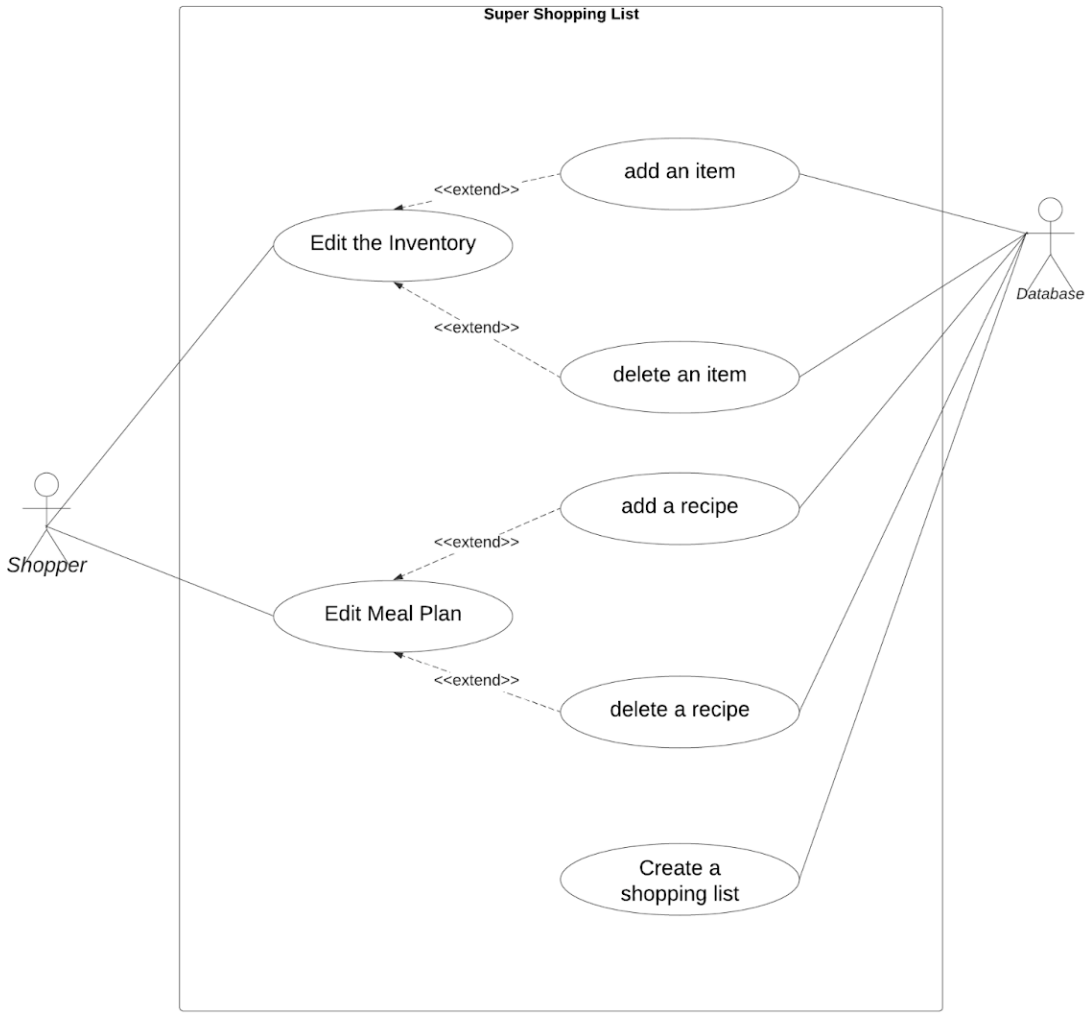
This increment was dedicated to having our database built and having the backend code to be able to connect to the MySQL database. We wanted to display data on the application that was pulled from the database.

- Shopping List generated from the ingredients table from the database
    - Allows additional items
- Inventory generated using ingredients available and shopped for
- Database connection successful
- One of the android people dropped the course so we are behind schedule wise.

UML Diagram

Use Case Diagram

(I did this diagram)



Use case text: edit the Inventory

Main success scenario:

1. Shopper clicks on the Inventory tab

2. Shopper selects the item on the ingredients list

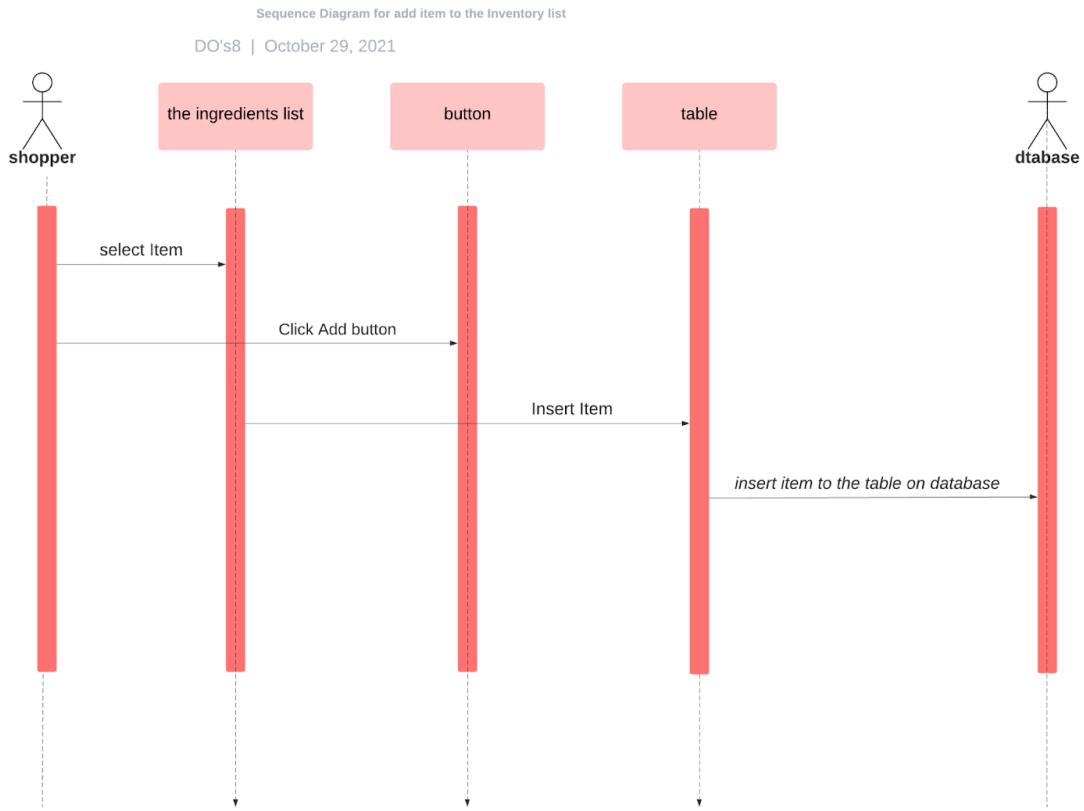
3. Shopper clicks the Add button

Extensions:

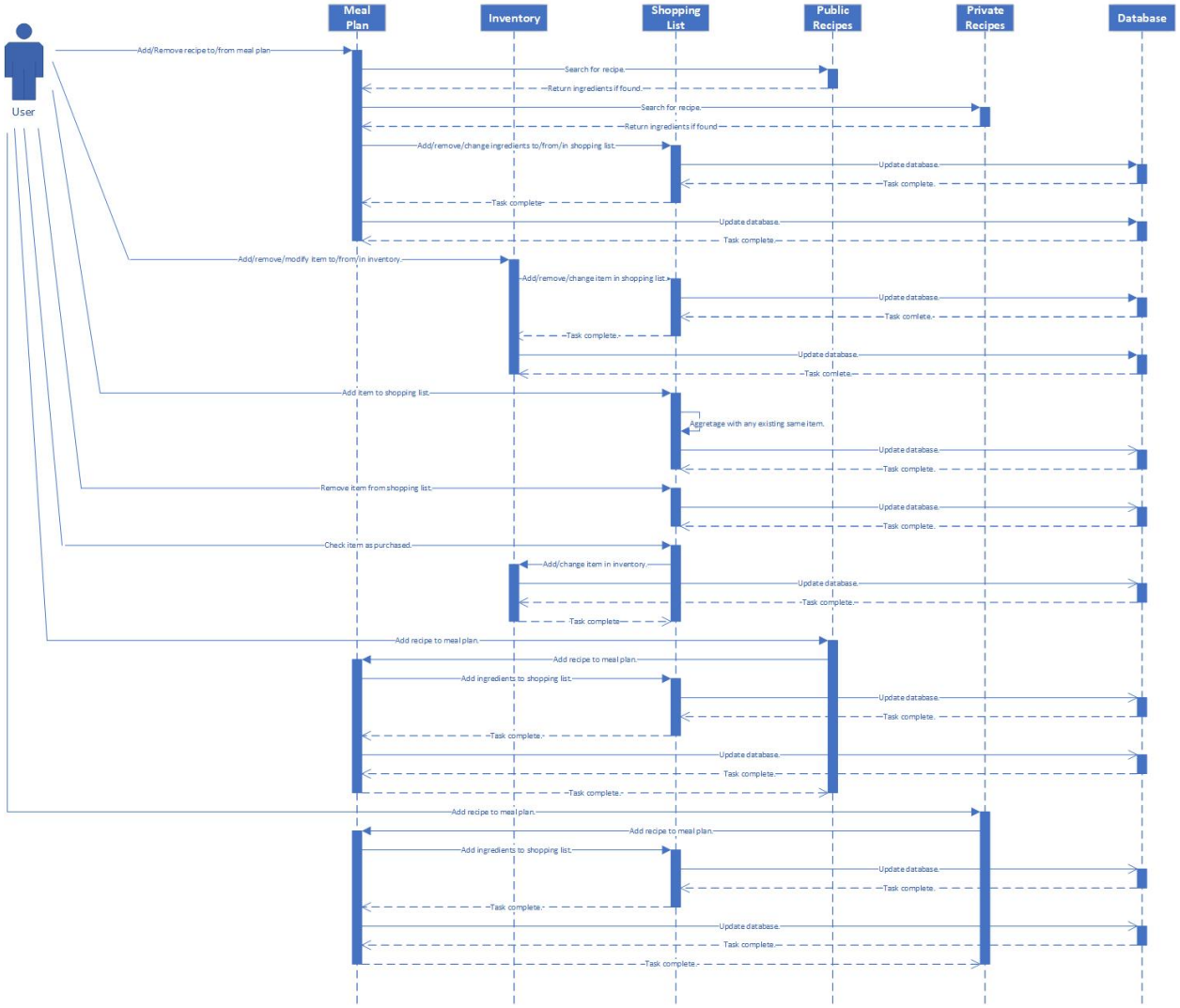
2a. Shopper selects the item on the table

3a. Shopper clicks the Delete button

Sequence Diagram

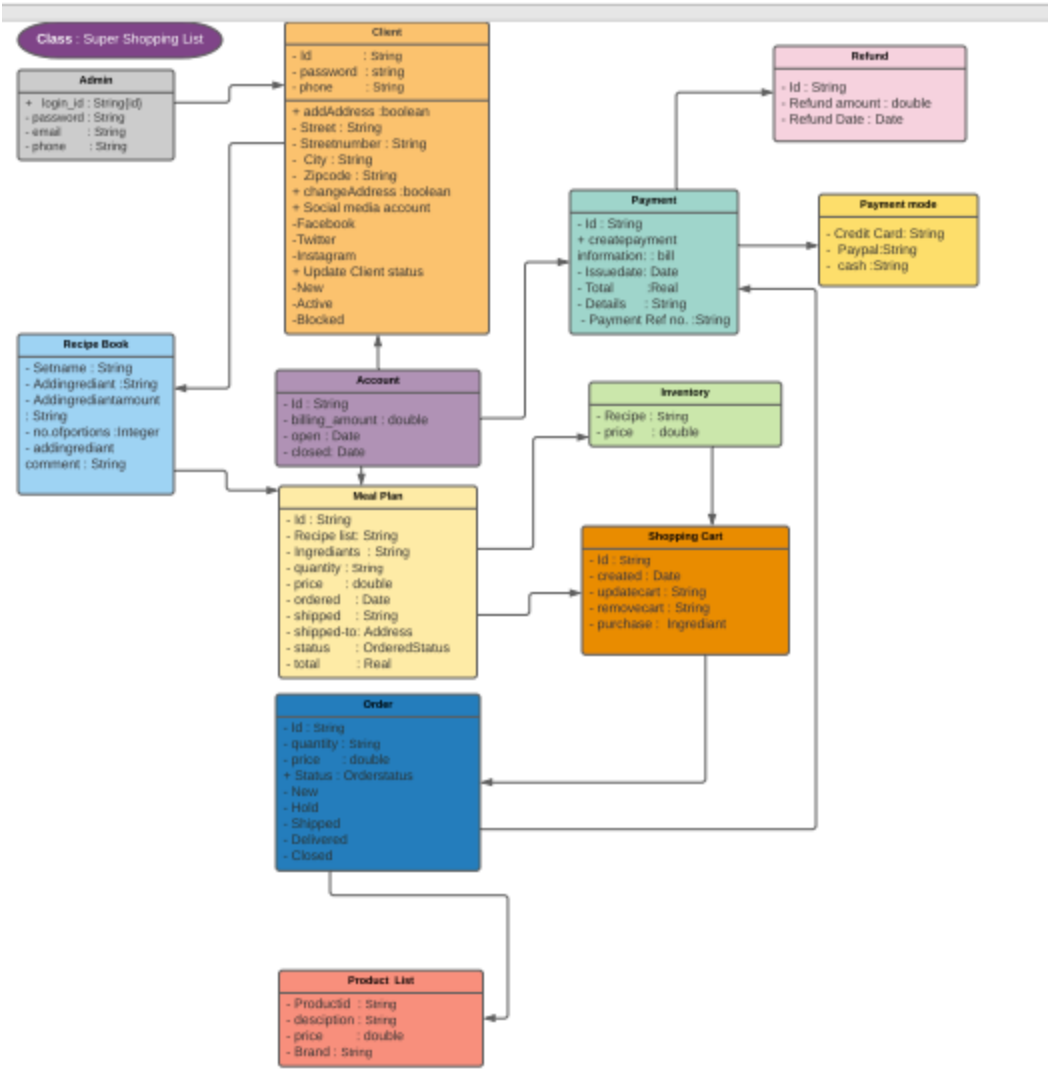


Naveen's Sequence Diagram



Mica's Sequence Diagram

UML Class Diagram



Manual

Mobile

Once you install the mobile application on an android device, the first page you see is the dashboard. From there you can read the instructions on how you can use the application and

what functionalities the application provides. From the dashboard, you can navigate to one of the six pages which include inventory, meal plan, recipe book root, public recipes, favorite recipes, and the shopping list.
For phase one, there is no actual functionality in the app, for now, you can access different activities and the layouts consist of data that is not pulled from the database (will be implemented in phase 2). The inventory activity helps the user to keep track of their inventory for different ingredients. In this activity, there are three sections being fridge, pantry, and spice cabinet. Secondly, the meal plan activity consists of meals planned for different days, for example, Cake for Tuesday and Rice and beans for Thursday. Thirdly, the recipe book root allows you to access the recipe book pages which are the public recipes and the favorite recipes. In the public recipes, you have access to all the different recipes uploaded by different users. And from there you can select recipes to be your favorite and they will appear in the favorite recipes activity. While the shopping list activity pulls ingredients from the recipes of the meals you have in your meal plan and suggests you shop for them. The shopping list can be manually edited.

Desktop

The desktop app has a tab layout to navigate through the different views, these are:

- Dashboard
- Meal plan
- Shopping list
- Recipe
- Inventory
- User Profile

The dashboard has a list of recipes that you can browse through, collapse with the arrows on the left, and select your meal plan with the select box on the right. When you select a recipe, you will see it populate in the meal plan window in that same tab.
If you change tabs with the tab buttons at the top, you can change to the meal plan view. The meal plan view shows the same recipe browser and a meal plan browser that are editable in the same way. If you edit from here, you will see the changes populate in the other tab.

Instructions on How To Compile

Mobile

This is an Android app written with Java. To run it you will need to install the latest version of Android Studio, which can be found on developer.android.com. Once the android studio is opened, you have to open the project with the source code of this project. Make sure to separate the mobile app folder from the git repository so that android can run it. Once the app source code is opened in Android Studio, you can hit run. If all dependencies are up-to-date, an Android phone emulator will run the app. One thing to always make sure of is to have all

activities stated in the AndroidManifest.xml. If not, the respective activity would not open once directed from within the emulator. If something fails, the most likely culprit is Gradle. Deleting the .gradle file and rebuilding the app may help. Also, make sure that Gradle is up to date as well.

Desktop

This is a Qt application written with qt for python - to run it you will need to install pyside6 and qt creator.
Getting qt-creator will require an account (it's free).

- Go to qt.io
- Click download try
- Go open source
- Scroll to and click download the qt installer
- Run the installer
- Install qt 5.15
    - Include qtcore
    - Qtdesigner
    - You can exclude anything you already have (like visual studio or mingw)
    - You can disregard anything like mobile/android/web/etc (this is a qt app made with python - let that guide what you install to keep the install to keep it as small as possible)

Dependencies

- Python 3.8 or later
- Install (using pip) pyside6
- Install (using pip) mysql-connector
- Ensure that your environment is set up correctly such that when you run a python command it is indeed the python for which pyside was installed
- If you are not using python3.9 you will need to pip install data classes (they are built into 3.9)

Running

- You can run the program by using `python mainwindow.py` or by opening the project in an ide and running / pointing config at mainwindow.py.
- Using the build.bat script - this will compile all the ui files into python code (these files are committed to the git so you shouldn't *need* to do this, but you can)
    - For linux systems, you can use a shell script (just remove that "@echo off")
- If you prefer to use an ide to run your code rather than the command line, you can have your ide of choice point its run configuration towards that build script.

Tests

Desktop

- Unit conversions
    - Case: same units
    - Case: different units
    - Case: unit not known
        - The expected behavior is an error value or -1 that programmer checks
- Comparing quantities
    - Case: same units, the same value
    - Case: same units, different values
        - Test both ways - *a* being large and then *b* being larger
    - Case: different units
        - Test both ways - *a* being large and then *b* being larger
    - Case: units not known
- Shopping List generation
    - Case: recipes with different ingredients
    - Case: two recipes that share ingredients
    - Case: two recipes that share ingredients with an inventory that contains needed ingredients
    - Case: recipes with “other items” added to shopping list
    - Case: “other items” *and* an inventory that contains something we need
- User Authentication
    - Case: make sure that existing user is found
    - Case: make sure that existing user cannot sign up again and redirect to login
    - Case: make sure that non existing user is not found
    - Case: make sure that non existing user cannot login and redirect to sign up tab
    - Case: make sure user cannot navigate to other tabs if not signed up or logged in

Mobile

- Connection to the Database
    - Case: Connection successful
        - Put in mock data and delete it
- Test retrieving data for different activities
    - Example: Retrieve recipe details for recipe ID: 1 and compare them to expected output.
- Test pushing mock user data and testing the retrieved results
    - Push in a demo user and add a recipe for them and retrieve and compare it.
- Test Shopping List ingredients being added and test removing them
    - Add eggs, cake rusk, etc to the list and check them off.
- User Authentication

    - Case: make sure actual user is found
- Data Retrieval
    - Case: make sure recipe retrieval from database works
    - Case: make sure all public recipes retrieval from database works
    - Case: make sure inventory retrieval from database works
    - Case: make sure meal plan retrieval from database works
    - Case: make sure extra items retrieval from database works

Feedback-based Details

The only feedback we got from the code inspection was that we didn't have enough comments. We are adding more comments/documentation as a result.

Reflection

During this increment, we accomplished the majority of the core functionality of the project. This process was made fairly smooth due to the extensive discussion we have had with regard to database and UI design - it was easy to know what needed to happen and achieve flow.

One issue that has come up is that the mobile app development has been slower than the desktop app due to uneven workloads. In the future, this difficulty will be minimized by team members from the desktop side helping the mobile development when necessary. We may have spread ourselves too thin. The app side is struggling since experience with Android development is lower than previously thought.

Contributions

Names	Contribution Description	Contribution (%)	Notes
Mustafa Memon	Worked on UI and backend for the shopping list activity for android. Worked on debugging the database connection alongside Nestor. Provided tutorials for android members. Built unit tests for android. Document contribution includes tests, Instructions, Requirements and manual.	16	
Brice Brosig	Meal plan database functionality. Added unit conversion / comparison code. Wrote the	15	

	shopping list generation algorithm. Wrote unit tests.		
Mica	Wrote create a recipe page. Connected desktop app to the database. Initialized the recipe list and inventory from the database. Generated dummy data. Made the second sequence diagram. Wrote feedback-based details and half of the reflections section.	14	
Nestor Molina	Database connection for android project. Built unit tests for android.	15	
Tam Doan	Inventory page for Desktop. Use case diagram for document.	9	
Vandana Sinha	Inventory page for android project and database objects.	13	
Nikhil Gaur	Created a user sign up, login page, Added users to database Authenticated user login to database, Added user authentication test cases	10	
Naveen	Made the meal plan functionality for the mobile app. Created the first sequence diagram.	8	