

## MILESTONE 2: PROJECT PLANNING

# PANTRYPAL

### **Team Lead:**

Kaleigh Mogatas<sup>1</sup>

### **Team Members:**

Michael Chen<sup>2</sup>, Stephanie Xu<sup>3</sup>, Eddie Lu<sup>4</sup>, Vedant Patel<sup>5</sup>, Kenny Nguyen<sup>6</sup>

---

<sup>1</sup> PID: A17051705

<sup>2</sup> PID: A17049399

<sup>3</sup> PID: A17020719

<sup>4</sup> PID: A16350108

<sup>5</sup> PID: A16368980

<sup>6</sup> PID: A16927338

## Table Of Contents

---

### **Project Management |49%|**

#### **Participation Survey |1%|**

### **Risk Analysis |5%|**

1.1 Risks	3
- Team Risks	
- Project Risks	
1.2 Iteration Cycle Lengths	5
1.3 Initial Velocity Calculation	5

### **Planning Poker |2%|**

2.1 Photo Documentation	6
2.2 Table Summary	7

### **User Stories/Features |15%|**

3.1 User Stories	10
3.2 Wire Framing	11
3.3 BDD Scenarios	12

### **Tasks Breakdown |10%|**

17

### **Iterations/Milestones |5%|**

21

### **Scenario Based System Tests |9%|**

25

### **Github |3%|**

7.1 Github Link	26
-----------------	----

## 1. Risk Analysis

---

### 1.1 Risks

#### Team Risks:

- **Risk: API/Language Proficiency**
  - Severity: **Medium**
  - Description: Some team members may not be proficient in the necessary APIs, interfaces, or languages that we are using for this project
  - Resolution: Team members have gained experience using the relevant languages and APIs over the course of the first iteration.
  - Status: **Resolved**
- **Risk: Scheduling Conflicts**
  - Severity: **High**
  - Description: As a team, we may have a hard time scheduling to meet up with each other due to personal reasons or transportation issues. Furthermore, since the holiday season is approaching team members may be traveling and may therefore not be able to meet up.
  - Resolution: As a team, for those unable to meet up in person, there will be a call/zoom for them so we are still able to collaborate as a group.
  - Status: **Resolved**
- **Risk: Workload Management**
  - Severity: **High**
  - Description: Since we are all enrolled in several other classes it may be difficult for us to balance work for this project alongside schoolwork and studying for other classes
  - Resolution: We will utilize user story estimates to better gauge the amount of time we need to put into completing each feature, so that time management will be easier.
  - Status: **Resolved**
  - Resolution: We will maintain communication and conduct weekly stand-up meetings to provide proper support to team members that are overwhelmed. This group has decided to communicate primarily through texts and Discord.
  - Status: **Resolved**

## **Project Risks:**

- **Risk: Time Pressure**
  - Severity: **High**
  - Description: Since we only have a few weeks to complete our project, this may not be enough time to fully develop and flesh out our product
  - Resolution: As a team, we will utilize the GitHub big board/burn-down chart to keep track of our progress and manage our time accordingly.
  - Status: **Resolved**
- **Risk: Completing your Assigned Task**
  - Severity: **High**
  - Description: This would be a problem as a member(s) are not putting in the time and effort to actually complete their task.
  - Resolution: We will host weekly team meetings to communicate progress updates so we know what is behind schedule. This will allow us to allot additional help on that task if needed.
  - Status: **Resolved**
- **Risk: Unclear specifications**
  - Severity: **High**
  - Description: We make a lot of assumptions about what the customer actually wants for their application due to vague instructions that can lead to misunderstandings in our developing process.
  - Resolution: We will reach out to the customer to obtain more information about what they want for their project to get rid of assumptions.
  - Status: **Resolved**
- **Risk: Having a hard time with debugs**
  - Severity: **High**
  - Description: As we continue to code, there are bound to be bugs within our code that will be hard to debug, keeping us from producing a functional application.
  - Resolution: We will run tests and take more time to specifically focus on debugging together, as a team.
  - Status: **Resolved**
  - Resolution: Reach out to TA, discussion hours, or create a post in Piazza
  - Status: **Resolved**

## **1.2 Iteration Cycle Lengths**

**Weeks Remaining until MS2 Turn In : 2 weeks**

Iteration 1 11/27: **1 week**

Iteration 2 12/5: **1 week**

## **1.3 Initial Velocity Calculation**

**Estimated Velocity = 0.7**

**Justification:** Based on an estimated 60 cumulative hours across the next iteration (10 hr/person), we estimate a velocity of 0.7. Our velocity for the last milestone was lower than expected, however, we've had much growth as a team which makes us more optimistic for milestone 2. We've all grasped the basics of GitHub which should significantly reduce the time needed to pull/commit/merge code. Additionally, most of the team ran into some frustrating roadblocks related to Gradle, VSCode, and JavaProjects which made it difficult to develop PantryPal. Many of these issues have been resolved, which should make milestone 2 a smoother process than milestone 1. Thus, this 0.7 velocity is a cautiously optimistic guess based on our velocity for milestone 1.

## 2. Planning Poker

---

### 2.1 Photo Documentation



## 2.2 Table Summary

<b>US#</b>	<b>Name</b>	<b>Hand</b>	<b>False Assumptions Uncovered</b>
8	Accounts	16, 16, 16, 16, 32, 32	<ul style="list-style-type: none"> <li>- Assumed that creating new accounts would be very difficult/would require infrastructure relying on new APIs/technology</li> <li>- Assumed that the account creation page would not require many new UI pages to be implemented</li> </ul>
8	Accounts	32, 32, 32, 32, 32, 32	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>
9	Multi-Platform Support	4,8,16,16,16 16	<ul style="list-style-type: none"> <li>- Assumed that there is not a lot of work as this just works with the accounts. As it just ties to the accounts, it should be able to still connect to the server</li> <li>- Assumed that MongoDB is not a requirement, thus might take a bit more time in comparison to what we anticipate.</li> <li>- Assumed that the tasks would be more tedious and more difficult as we would have to hard code it to the device we are logging into.</li> </ul>
9	Multi-Platform Support	8,8,8,8,8,8	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>
10	Recipe Images	8,8,8,8,8,16	<ul style="list-style-type: none"> <li>- Assumed that UI was not going to be hard because we just have to worry about fitting into the screen</li> <li>- Assumed that it will be difficult to save the image into the server</li> <li>- </li> </ul>
10	Recipe Images	8,8,8,8,8,8	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>
11	Regenerate Recipe	2,4,4,4,8,16	<ul style="list-style-type: none"> <li>- Assumed that it is all on the UI side rather than the coding side</li> <li>- Assumed that it should be easy as we are just changing the prompt for ChatGPT</li> </ul>
11	Regenerate Recipe	8,8,8,8,8,8	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>

12	Recipe Sorting	2,2,4,4,8,16	<ul style="list-style-type: none"> <li>- Assumed that it was simple as we did it from a previous lab and the mini-project</li> <li>- Assumed that we can automatically sort the recipes as soon as the application runs</li> </ul>
12	Recipe Sorting	4,4,4,4,8,8	<ul style="list-style-type: none"> <li>- Assumed that the drop box for the UI would be annoying to implement due to the buttons and the integration portion that comes along with it.</li> <li>- Assumed that we use our recipe list and just recall from that page.</li> <li>- Assumed that we were sorting within the actual UI and not the backend side of the code</li> </ul>
12	Recipe Sorting	8,8,8,8,8,8	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>
13	Filtering Recipes	1,4,4,4,8,8	<ul style="list-style-type: none"> <li>- Assuming that the UI side for this is easy</li> <li>- Assuming that it is easy to create a filter within the MongoDB</li> <li>- Assuming that the server should get the list from MongoDB, then go through that list to create the filter.</li> <li>- Assumed that everything is happening within one part of the code as we are just sorting through the meal types and returning those recipes</li> </ul>
13	Filtering Recipe	4,4,4,4,4,4	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>
14	Sharing through URL link	8,16,16,32,32 32	<ul style="list-style-type: none"> <li>- We do not know how to be able to implement this part of the code</li> <li>- We would have to work with our server, in which we are not confident in working with</li> <li>- Assumed that it would take some time to figure out how to actually code this portion</li> <li>- Assumed that fetching from the database might take a while to actually create a URL from the query we are getting in the URL</li> <li>- Assumed that we are returning a JSON with the link, as a webpage, but we do not know how to follow through with this.</li> <li>- Assumed that we will have to take time out of our work to do some research on how to implement the URL portion</li> </ul>

14	Sharing through URL link	16,16,32,32,32,32	<ul style="list-style-type: none"> <li>- Assumed that there were way too many unknowns while trying to implement the URL.</li> <li>- Assumed that we just fetch the recipe title to generate the URL.</li> <li>- Do not know how to implement the webpage with the URL to open the webpage.</li> <li>- Do not know how to make the image stay with the website, as it may take some time to figure out how to get the image to stay with the recipe as we are sending it through the URL</li> <li>- Creating the recipe page can go both ways in terms of times <ul style="list-style-type: none"> <li>- Assuming that we will not work alone when creating the webpage for the URL</li> </ul> </li> </ul>
14	Sharing through URL link	? , 16,16,32,32,32	<ul style="list-style-type: none"> <li>- Overall, some genuinely do not know what to do for this part of the implementation</li> <li>- Assumed that research is needed for this alongside the implementation</li> <li>- Everything goes through it should not take a lot of time because we can just look up on Google what to do.</li> </ul>
14	Sharing through URL link	16,16,16,16,16,16	<ul style="list-style-type: none"> <li>- CONVERGED</li> </ul>

### 3. User Stories/Features

---

#### 3.1 User Stories

##### US-8: Accounts:

**Priority: HIGH**

As someone who shares a desktop computer, **I want** to be able to keep my saved recipes separate from others behind a login portal **so that** I don't have to worry about other users changing my recipes and their details.

**Time: ~32 hours**

##### US-9: Multi-Platform Support

**Priority: HIGH**

As someone who has multiple devices, **I want** to be able to access the application on all of them and see my recipes **so that** I do not have to constantly use the same device, making it more convenient to access my saved recipes.

**Time: ~16 hours**

##### US-10: Recipe Images:

**Priority: MEDIUM**

As someone who likes to look at pictures of food before ordering at a restaurant, **I want** to be able to see images of the recipe **so that** I can check if a dish looks appetizing.

**Time: ~8 hours**

##### US-11: Regenerate Recipe:

**Priority: HIGH**

As a chef, **I want** to see different food options with the same ingredients **so that** I have a variety of dishes to choose from.

**Time: ~8 hours**

##### US-12: Recipe Sorting:

**Priority: LOW**

As a user with many generated recipes, **I want** to be able to sort my recipes **so that** I can easily alphabetically search for a recipe or find an older recipe.

**Time: ~8 hours**

##### US-13: Filtering Recipes:

**Priority: LOW**

As someone who has so many saved recipes throughout all the meal types, **I want** to be able to only see specific generated recipes based on meal type **so that** I can easily remake my favorite dishes based on the time of day.

**Time: ~4 hours**

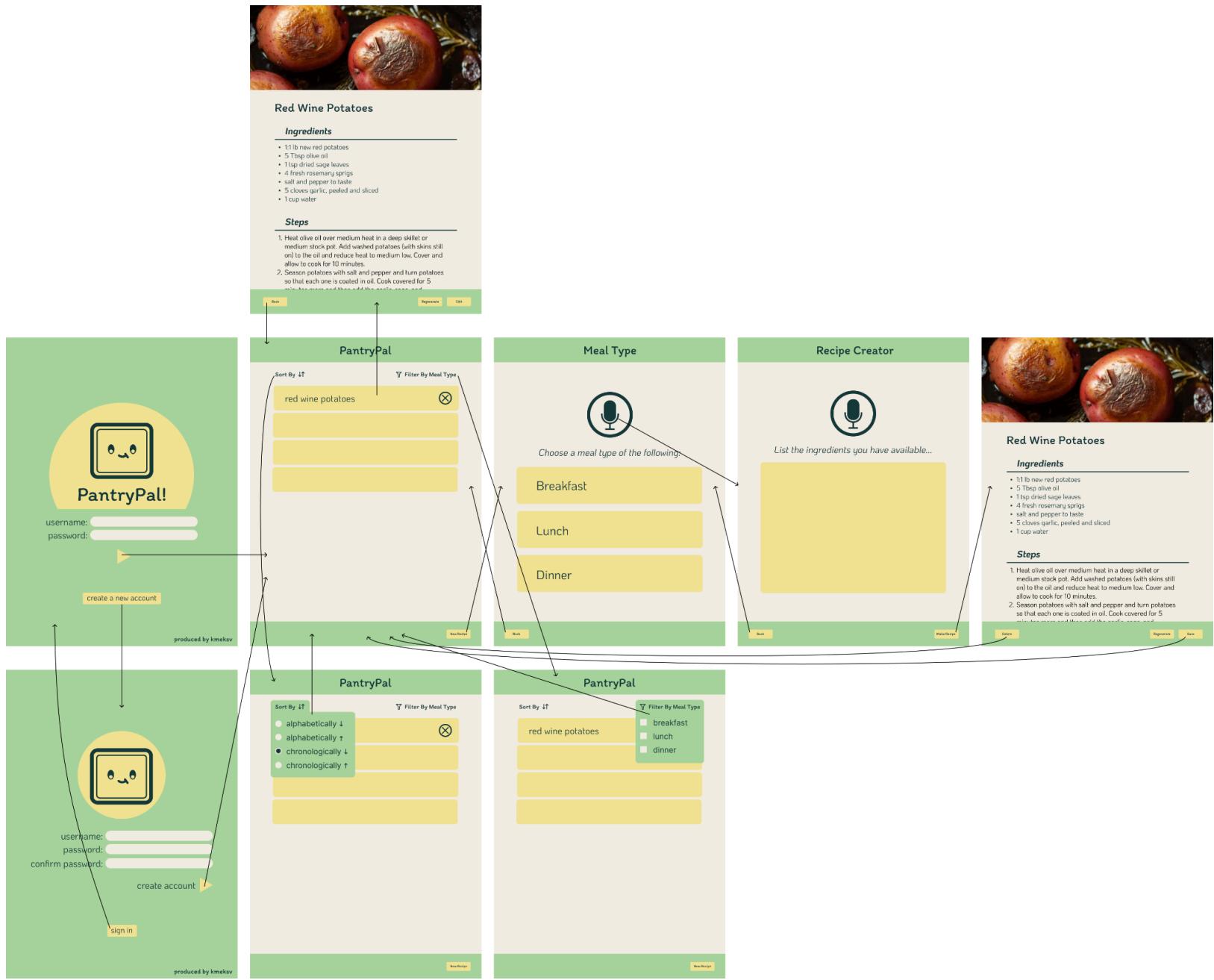
##### US-14: Sharing through URL link

**Priority: MEDIUM**

As someone who likes to send recipes, **I want** to be able to obtain a unique URL for a specific recipe **so that** I can share that recipe with my friends and family.

**Time: ~16 hours**

## 3.2 Wire Framing



### 3.3 BDD Scenarios

---

#### US-8. ACCOUNTS

##### **Scenario 1:** Creating a New Account

**Given** Bob does not have an account with PantryPal  
**When** Bob clicks on the “Sign Up” button on the landing page  
**Then** a page asking for username and password inputs pops up  
**And** Bob inputs in a unique username  
**And** Bob inputs in a password  
**And** Bob clicks on the “Create Account” button  
**Then** the app takes him to his new homepage

##### **Scenario 2:** Creating a New Account with an Already Existing Username

**Given** Bob does not have an account with PantryPal  
**When** Bob clicks on the “Sign Up” button on the landing page  
**Then** a page asking for username and password inputs pops up  
**And** Bob inputs in a username that already exists  
**And** Bob inputs in a password  
**And** Bob clicks on the “Create Account” button  
**Then** text appears on the page informing him that the username already exists and to come up with a new, unique username

##### **Scenario 3:** Manually Logging into an Existing Account

**Given** Bob does have an account with PantryPal  
**When** Bob clicks on the “Login” button on the landing page  
**Then** a page asking for username and password inputs pops up  
**And** Bob inputs in his correct username  
**And** Bob inputs in his correct password  
**And** Bob clicks on the “Sign In” button  
**Then** the app takes him to his homepage

##### **Scenario 4:** Manually Logging into an Existing Account with an Incorrect Username

**Given** Bob does have an account with PantryPal  
**When** Bob clicks on the “Login” button on the landing page  
**Then** a page asking for username and password inputs pops up  
**And** Bob inputs in a username that does not exist  
**And** Bob inputs in a password  
**And** Bob clicks on the “Sign In” button

**Then** text appears on the page informing him that the username does not exist and to try again

**Scenario 5:** Manually Logging into an Existing Account with an Incorrect Password

**Given** Bob does have an account with PantryPal

**When** Bob clicks on the “Login” button on the landing page

**Then** a page asking for username and password inputs pops up

**And** Bob inputs his correct username

**And** Bob inputs an incorrect password

**And** Bob clicks on the “Sign In” button

**Then** text appears on the page informing him that the password he inputted is incorrect and to try again

**Scenario 6:** Manually Logging into an Existing Account and Checking the Auto-Login Feature for Future Logins

**Given** Bob does have an account with PantryPal

**When** Bob clicks on the “Login” button on the landing page

**Then** a page asking for username and password inputs pops up

**And** Bob inputs in his correct username

**And** Bob inputs in his correct password

**And** Bob checks the checkbox labeled “Auto-login on this device for future use”

**And** Bob clicks on the “Sign In” button

**Then** the app takes him to his homepage

**Scenario 7:** Automatically Logging into a Previously Accessed Account

**Given** Bob does have an account with PantryPal and has already opted for automatic login

**When** Bob clicks on the “Login” button on the landing page

**Then** the app takes him to his homepage

## US-9. MULTI-PLATFORM SUPPORT

### Scenario 1: Using PantryPal on PC

**Given** Thomas wants to use PantryPal on his PC

**When** Thomas clicks on the PantryPal run button

**Then** the application is supported and able to launch properly

**And** the UI of the app is auto-adjusted to account for the screen size

## **Scenario 2: Synchronizing Changes Across Multiple Devices on the Same Account**

**Given** Thomas wants to use PantryPal on his Desktop PC  
**And** Thomas has generated a recipe for red wine potatoes on his laptop  
**And** Thomas wants to also use PantryPal on his laptop  
**When** Thomas clicks on the PantryPal run button on his Desktop PC  
**Then** the application displays the red wine potatoes recipe he generated on his laptop

## **US-10. RECIPE IMAGES**

### **Scenario 1: Displaying Image after Recipe Generation**

**Given** Thomas wants to view an image of the recipe he is making.  
**When** Thomas finishes with the Recipe Creator and clicks done.  
**Then** Thomas is moved to his generated recipe, a spicy chicken sandwich, and at the top of the page is an image of a spicy chicken sandwich.

### **Scenario 2: Displaying Image in Recipe List**

**Given** Charlie has an existing recipe in PantryPal for spaghetti and meatballs  
**And** Charlie expands this recipe from within PantryPal's recipe list view  
**Then** the previously generated image of spaghetti and meatballs that was displayed when the recipe was first generated will appear underneath the title of the recipe in the expanded recipe view

## **US-11. REGENERATING RECIPES**

### **Scenario 1: Regeneration of a Recipe**

**Given** Bob has generated a recipe  
**And** Bob has inputted lamb, carrots, and celery  
**And** Bob is on the generated recipe page  
**When** Bob clicks on the "Regenerate" button  
**Then** a different recipes pops up  
**And** this recipe has lamb, carrots, celery, and essential ingredients

## US-12. RECIPE SORTING

### Scenario 1: Sorting by Newest

**Given** Bob has generated recipes  
**When** Bob selects the “Sort By” dropdown  
**And** Bob selects the “Newest” button  
**Then** the list of recipes is sorted chronologically  
**And** the latest recipe is on top  
**And** the oldest recipe is on the bottom

### Scenario 2: Sorting by Oldest

**Given** Bob has generated recipes  
**When** Bob selects the “Sort By” dropdown  
**And** Bob selects the “Oldest” button twice  
**Then** the list of recipes is sorted chronologically  
**And** the latest recipe is on bottom  
**And** the oldest recipe is on the top

### Scenario 3: Sorting by A -> Z

**Given** Bob has generated recipes  
**When** Bob selects the “Sort By” dropdown  
**And** Bob selects the “A -> Z” button  
**Then** the list of recipes is sorted alphabetically  
**And** recipes that start with “A” are on top  
**And** recipes that start with “Z” are on the bottom

### Scenario 4: Sorting by Z -> A

**Given** Bob has generated recipes  
**When** Bob selects the “Sort By” dropdown  
**And** Bob selects the “Z -> A” button  
**Then** the list of recipes is sorted alphabetically  
**And** recipes that start with “Z” are on top  
**And** recipes that start with “A” are on the bottom

## US-13. FILTERING RECIPES

### Scenario 1: Filtering Recipes that Exist

**Given** Bob has generated “Breakfast” recipes  
**When** Bob opens the “Filter by Meal Type” dropdown  
**And** Bob selects the “Breakfast” filter  
**Then** the list only displays recipes with the “Breakfast” tag

**Scenario 2:**

**Given** Bob has generated recipes  
**And** Bob has not generated “Lunch” recipes  
**When** Bob opens the “Filter by Meal Type” dropdown  
**And** Bob selects the “Lunch” filter  
**Then** the list doesn’t display any recipes.

**US-14. SHARING THROUGH URL LINK**

**Scenario 1:** Generating URL

**Given** Bob has generated a recipe  
**And** Bob has saved said recipe  
**When** selects the recipe in the list  
**And** Bob selects the “Share” button  
**Then** a unique URL is generated in PantryPal  
**And** Bob can share this URL  
**And** Bob can view this recipe on a webpage

**Scenario 2:** Viewing Webpage

**Given** Bob has generated a recipe  
**And** Bob has saved said recipe  
**And** Bob has generated a URL for said recipe  
**And** Bob has emailed Jay the URL  
**When** Jay clicks the URL  
**Then** Jay can view this recipe on a webpage

## 4. Tasks Breakdown

---

### 1. Accounts

- a. Title - **Create a Landing Page**
  - i. Description - Create the UI for the landing page of the PantryPal app which prompts users to either sign up to create a new account or login with an existing account, each with functional buttons that lead to their respective pages.  
Time estimate - **4 hours**
- b. Title - **Create UI for Account Sign Up Page**
  - i. Description - Design a UI matching the wireframe that allows the user to create a new account by entering a username and password into JavaFX text fields.  
Time estimate - **4 hours**
- c. Title - **Implement “Create Account” Button Functionality**
  - i. Description - Create the “Create Account” button below the fields on the “Sign Up” page and add an event listener that implements the create functionality to the MongoDB database given that the username does not already exist. Take the user to their new homepage provided the inputted information is valid.  
Time estimate - **12 hours**
- d. Title - **Create UI for Login Page**
  - i. Description - Design a UI matching the wireframe that allows the user to sign into an existing account by entering the username and password into JavaFX text fields.  
Time estimate - **4 hours**
- e. Title - **Implement “Sign In” Button Functionality**
  - i. Description - Create the “Sign In” button below the text fields and auto-registration checkbox and add an event listener that checks the database to compare the inputted username and password and will take the user to their existing homepage provided that the information is a match with an existing account.  
Time estimate - **4 hours**
- f. Title - **Inform User of Invalid Input**

- i. Description - Display a text field on the respective page informing the user if they attempted to sign up with a pre-existing account, or login with a username that does not exist or an incorrect password.

Time estimate - **1 hour**

**g. Title - Implement Automatic Login**

- i. Description - Create a checkbox labeled “Auto-login on this device for future use” below the text fields on the “Login” page and implement functionality so that future logins will be fully automatic and will lead directly to the homepage when the “Login” button on the landing page is clicked.

Time estimate - **1 hour**

**h. Title - Integration Testing**

- i. Description - Create tests within our code to make sure our implementation works properly based on the information provided by the user.

Time estimate - **2 hours**

**2. Multiple Platform Support**

**a. Title - Implement Responsive UI**

- i. Description - Modify the code by replacing the currently hard-coded size, margin, and dimension values for all the UI elements with variables that adapt to various screen sizes and layouts.

Time Estimate - **6 hours**

**b. Title - Allow for Multiple Devices to be on the Same Account**

- i. Description - Figure out how to allow for multiple devices to be logged into the same account and synchronize any updates that are made to a recipe or recipe list on one device to be visible on all devices logged into the same account.

Time Estimate - **1 hour**

**c. Title - Integration Testing**

- i. Description - Set up tests that ensure our implementation of multiple platform support functions properly.

Time Estimate - **1 hour**

### **3. Recipes Images**

- a. Title - **Implement AI to Return an Image**
  - i. Description- Configure ChatGPT to return an image.  
Time Estimate - **2 hours**
- b. Title - **Implement Backend Storage and CRUD for Image**
  - i. Description- Figure out MongoDB and how the image is stored, retrieved, deleted, and updated if necessary.  
Time Estimate - **4 hours**
- c. Title - **Make Image Visible in the UI**
  - i. Description- Integrate the backend to the front so that the UX is consistent with the End-to-end scenario.  
Time Estimate - **1 hour**
- d. Title - **Integration Testing**
  - i. Description- Set up tests for the gray box testing of ChatGPT and data storage. No need to mock test UI.  
Time Estimate - **1 hour**

### **4. Regenerate Recipe**

- a. Title - **Implement “Refresh Recipe” button functionality**
  - i. Description - Design a ‘Refresh Recipe’ button matching the wireframe and add an event listener to allow the user to regenerate a new recipe based on the same ingredients and meal type when they have interacted with the button.  
Time estimate - **3 hours**
- b. Title - **Reconfigure ChatGPT for recipe regeneration**
  - i. Description - Figure out different prompts, temperatures, etc. to get ChatGPT to give different recipes for the same ingredients should people want a different recipe.  
Time estimate - **2.5 hours**
- c. Title - **Integration Testing**
  - i. Description - Creating tests within our code to make sure that the functionality of our code aligns and allows proper recipe regeneration.  
Time estimate - **2.5 hours**

## 5. Iterations/Milestones

---

### 5.1 Milestone 2

#### Iteration 1 - 1 week (10/29 - 11/4)

Be able to generate a working recipe with ChatGPT

- Framework and structure of applications UI - not all buttons need to have implemented functionality
  - US8 - Accounts (**HIGH**)
  - US9 - Multiple Platform Support (**HIGH**)
  - US10 - Recipe Images (**MEDIUM**)
  - US11 - Regenerate Recipe (**HIGH**)

#### Developer Story:

1. Enter the application. The servers are down and you encounter a brief error message. You wait a little bit to try again once the servers are back up.
2. You try again a little later. Given that you already have an account and have logged in on this device before, it takes you directly to your home screen.
3. Proceed to generate a new recipe by clicking the ‘New Recipe’ button in the bottom right.
4. Click the Mic-button and say breakfast, lunch, or dinner
5. Then the screen will change and then click the mic button again and say what ingredients you have, then a recipe will pop up, complete with an image, ingredients, and instructions.
6. Click the “Refresh Recipe” button, where a completely new recipe will then be shown, which still includes your specific ingredients.
7. Click the save button and you will be moved back to the home screen, this recipe should then be saved there at the top.
8. You open the recipe, can edit it, and then save the recipe.
9. You click the ‘X’ button to close the application and decide to move to another device.
10. Enter the application. You should be able to see the main home screen having the user to sign into their account.
11. Click the log-in button and if the incorrect information is entered there should be a message rejecting you and telling you to try again.
12. Once you sign in correctly, you are taken to the home screen again, here you should retain the recipe that you had recently saved on your other device from before.
13. Clicking the ‘X’ button once again closes the application.

If we are able to achieve this we have successfully completed the goals for Iteration 2.

**Justification:** Within the first iteration of milestone 2, we want the priority of the user to be our main priority while working through this iteration. So we want to have the most important user stories to be done within this iteration as it is a huge part of the app changes for milestone 2. Additionally, we want to prioritize these features as these are the most important to the customer, thus they have a higher priority.

As we only have about 40 hours to work within a 60-hour week in this iteration, we chose to still have a 0.7 velocity as it is midterms and Thanksgiving break. Due to these situations, we will not be able to prioritize the project as we would have to focus on other things and would have to get together towards the end of the week to finish the first part of the iteration. With the conflicts of some people being out of town for the holiday, it would be harder to schedule times to meet up as they are also spending time with their families, making our collaboration work very complicated. Additionally, there will be other distractions that are going to come up while we are working and burnout is bound to happen. This having a 0.7 velocity is reasonable. With the time and availability we have as a team, these three user stories will be within our first iteration as they are the most important and we know that with the time we have, we will be able to still get it done, regardless of time conflicts and distractions.

### **Iteration 2 - 1 week (11/5 - 11/11)**

- Implement functionality to existing UI and framework for additional user interactions.
- Focus on major QOL and features of app
  - US12 - Recipe Sorting (**LOW**)
  - US13 - Filtering Recipes (**LOW**)
  - US14 - Sharing Through URL Link (**MEDIUM**)

#### **Developer story:**

1. Enter the application. Given that you already have an account and have logged in on this device before, it takes you directly to your homescreen.
2. Proceed to generate a new recipe by clicking the ‘New Recipe’ button in the bottom right.
3. Click the Mic-button and say breakfast.
4. Then the screen will change and then click the mic button again and say what ingredients you have, then a recipe will pop up, complete with an image, ingredients, and instructions.
5. Click the “Refresh Recipe” button, where a completely new recipe will then be shown, which still includes your specific ingredients.
6. Click the save button and you will be moved back to the home screen, this recipe should then be saved there at the top.
7. You open the recipe, can edit it, and then save the recipe.

8. You click the ‘Filter’ button and decide to filter the meal type to only ‘Breakfast’. Only ‘Breakfast’ recipes should be displayed, amongst them you see the recipe you just generated at the top.
9. Click the ‘sort’ button and decide to change the sorting logic from the default ‘Newest-to-Oldest’ with ‘Oldest-to-Newest’. Your most recent recipe can now be found at the bottom and all other recipes are sorted accordingly.
10. Again, you click the ‘sort’ button and this time decide to sort them ‘Alphabetically’. The recipes should be listed from top to bottom in ‘A-Z’ order.
11. Once you find your recipe again, clicking on it brings you to the full view. At the bottom there is an option to ‘Share’.
12. Clicking on it provides you with a URL link.
13. Sending this to anyone and opening it should provide you with the recipe in a web window.
14. You can now view this recipe with the application closed.

If we are able to achieve this we have successfully completed the goals for Iteration 2, thus completing the goals for a completed PantryPal app.

**Justification:** For this iteration of milestone 2, these are the smaller tasks for the entire milestone. These are the tasks we planned to leave for the end as we already have these implementations from past labs and from our mini project. We are confident that we will be able to get this done in a shorter amount of time in comparison to those in iteration 1. Lastly, we have placed these as a lower priority so we placed them in the second iteration.

Due to not being able to prioritize the project as much for the upcoming iteration 2 for milestone 2, collaboratively as a group, we still believe that we will still be able to finish all of the user stories for the milestone. However, since finals week is coming up, studying would become a top priority for some people, but overall, we would still have a velocity of 0.7. Outside of the collaborative hours we are planning on working together, we are able to still work and do our own parts. Additionally, we chose these four user stories to be a part of iteration 2 because we practically have these implementations from past labs, thus making it a little bit easier to tie them to our overall integration. Lastly, these are smaller user stories, thus it would be the least priority in comparison to the user stories we prioritize for iteration 1.

## 6. Scenario-Based System Tests

---

### 6.1 End-to-End Scenarios

#### **Scenario 1:** Brand New User Creates Account and Generates and Shares Recipe

(Covers US8 - Accounts, US10 - Recipe Images, US11 - Regenerate Recipe, US12 Recipe Sorting, US13 - Filtering Recipes, and US14 - Sharing through URL link)

1. Open the application. Assuming that you have never created an account with the application before, the application will prompt you to create a new user account with a username and password. (**US8 - Accounts**)
2. Create a new account by entering a new valid username and password. The application takes you to the home screen upon accepting these new credentials.
3. From the home screen, generate a new recipe: First, you say 'breakfast'; then, on the next menu, you click the microphone and list the ingredients you have aloud. When you finish reading off your ingredients, you click the microphone and end the transcription.
4. Once you are satisfied with the transcription, you click the right-arrow and you should be presented with a full recipe. The recipe screen will have the name, prep, and cook times of the recipe on the top with ingredients and steps below it. There are also buttons for saving, and deleting the recipe. In addition, there is a generated image of the recipe displayed. (**US10 - Recipe Images**)
5. Click on the regenerate button and you should see a new recipe, still consisting of the same previously specified ingredients but which is distinct from the previous recipe. (**US11 - Regenerate Recipe**)
6. Click on the save button and you should see the home screen that you started on. Now, the new recipe you have saved is at the top of your list of recipes.
7. Repeat steps 3 - 8 for a 'lunch' recipe and a 'dinner' recipe.
8. You can now see three recipes in the list and each recipe is clearly labeled with its name and either a 'breakfast', 'lunch', or a 'dinner' tag.
9. Click on the 'Filter By' button and select 'breakfast'. You can only see a single recipe labeled by its name and the 'breakfast' tag. (**US-13 - Filtering Recipes**)
10. Deselect the 'breakfast' filter and all three recipes reappear on the list. (**US-13 - Filtering Recipes**)
11. Click on the 'Sort By' button and select 'A -> Z'. You can now see three recipes in alphabetical order in the list with their respective tags. (**US-12: Recipe Sorting**)
12. Click on the dinner recipe and a full detailed recipe pops up.
13. Select the "Share" option from the currently selected recipe. The application copies a URL to the clipboard. This URL brings you to a recipe view page containing the previously selected recipe and its accompanying image. (**US-14 - Sharing through URL link**)

14. Once you finish you click the X button. The window closes and the app shuts down.
15. Return to the URL and you can see that the recipe is still there. (**US-14 - Sharing through URL link**)

## **Scenario 2:** Returning User Logs into their Account, Looks for Saved Breakfast Recipes, And Sorts by the Oldest Addition

(Covers US8 - Accounts, US9 - Multi-Platform Support, US10 - Recipe Images, US12 - Sorting Recipes, US13 - Filtering Recipes)

1. Open the application. Given that you have previously logged into this account on this device before, the application opens directly to your Home Page. (**US8 - Accounts**)
2. Your latest recipe should be at the top and you can scroll down to view older saved recipes.
3. Given that you opened PantryPal to start making breakfast, you click on the ‘Filter By’ option at the top of your HomePage and select the Breakfast option. (**US-13 - Filtering Recipes**)
4. Your HomePage should update to only display recipes that fit into the ‘Breakfast’ meal type. (**US-13 - Filtering Recipes**)
5. You remember an old recipe that you want to revisit, so you click on the ‘Sort’ button at the top and sort by ‘Oldest’ recipes. The display of recipes should update to sort from ‘Oldest’ to ‘Newest’ (**US-12 - Sorting Recipes**)
6. You find your recipe and click on it to open the recipe instructions, at the top is a clear image of what you are making, and below it the ingredients and instructions. (**US10 - Recipe Images**)
7. Now you can just start cooking. Once you finish you click the X button. The window closes and the app shuts down.
8. Hop on another computer and open the application. The app has an option to create a new account or to log in. (**US8 - Accounts**)
9. Click the ‘log in’ button and enter your username and password. (**US8 - Accounts / US9 - Multi-Platform Support**)
10. The list displays all the recipes generated on the previous device. (**US9 - Multi-Platform Support**)
11. Click on the ‘Sort By’ button and select ‘oldest’, the oldest recipe is now at the top of the list. (**US-12 - Sorting Recipes**)
12. You select your oldest recipe and click on it to open the recipe instructions, at the top is a clear image of what you are making, and below it the ingredients and instructions. (**US10 - Recipe Images**)
13. Once you finish you click the X button. The window closes and the app shuts down.

## 7. GitHub

---

Link to GitHub repository: <https://github.com/ucsd-cse110-fa23/cse-110-project-team-6>

Link to GitHub Project: <https://github.com/orgs/ucsd-cse110-fa23/projects/57>