**Design II**

**Search and Sort Functions**

1. Create a button called search

2. Create a button called sort

3. Create a global variable called Sorting

4. Add an event handler to the Sort button that operates as follows

a. When pressed bring down a segmented drop down list

b. When an option is selected, all other options are deselected, and the selected option is saved to the variable Sorting

c. If the selected option is already selected, deselect it and set the Sorting array as Null

5. Add an event handler to the search button that operates as follows

a. When pressed first make a connection to the server that stores recipes

b. If we cannot make a connection, return an error that the app cannot connect to the internet and do not continue

c. Pull the list of recipes and store them into a temporary array of Recipe objects

d. Sort the list by what is stored in the global variable Sorting

e. If the Sorting array is empty, sort the list alphabetically by the recipe name

f. Once the list is sorted, display the results in the window underneath the two buttons

g. Attach an event handler to each of the recipes added that says when clicked, display the Recipe's associated window

**Post Public Recipes**

1. Add a post recipe button
2. User clicks the post recipe button
3. System displays fields for recipe name, ingredients, instructions, description, image and waits for user input, as well as for private vs public. The system also creates an instance of the Recipe class to store the information related to the recipe.
4. As the program receives input in the different fields, it updates the instance of the Recipe class. The user will make the recipe public.
5. Possible outcomes:
   1. If the user chooses to exit out of the recipe creating screen, the program prompts the user if they want to save it as a draft or not.
      1. If the user selects that they want to save it, the system will save the recipe in the database.
      2. If they say they do not want to save it as a draft, the system will collect the instance of the Recipe class and not save it anywhere.
   2. If the user is disconnected from the application, then the system will save the current recipe as a draft in the database so the user can see it when they open the application again.
   3. If the user tries to post the recipe without filling out a required field (everything except for picture), then the application will send an alert to the user telling them that they need to fill out the fields that they haven’t before they can post the recipe.
   4. If the user tries to post the recipe with all the required fields, the application will store the recipe object in the database and make it available to everyone

**Sorting based on rating:**

1. Create a button called Sort

2. Create a global variable called Sorting

3. Add an event handler to the Sort button that opens up a dropdown list when the Sort button is pressed

4. When the Rating option is selected, deselect all other options and save Rating to the variable Sorting

5. Pull the list of recipes and store them in a temporary array of Array objects

6. Sort the list by the Rating field of the recipe from top to bottom

7. Place the top 100 recipes based on this rating into a separate array

8. Display the first 10 recipes of this list in the window

9. Add a next page button

10. Attach an event handler to this button, that when clicked, displays the next 10 recipes

11. Attach an event handler to each of the recipes added that says when clicked, displays the recipe's associated window

**List updated monthly for popular recipes**

1. Create a timer for every month

2. Once the timer goes to 0, pull the list of recipes and store them into a temporary array of Array objects

3. Sort this array based on the Views field of each recipe from top to bottom

4. Place the top 10 recipes based on this views field into a separate array

5. Add a Popular button

6. If that button is clicked, display the recipes in order on the window

7. Attach an event handler to each of the recipes added that says when clicked, displays the recipe's associated window

8. Restart the timer

**Inventory Management Functions**

1. Add an “Add ingredient” button
2. Add an event handler to the “Add ingredient” button that operate as follows:
   1. When pressed, make a connection to the server that stores recipes
      1. If we cannot make a connection, return an error that the app cannot connect to the internet and do not continue
   2. When pressed, pop up a keyboard and custom prompt that has slots for name of ingredient, expiration date, amount, and any other information
   3. After all the information for the ingredient is added, add the ingredient to the list
   4. Add the ingredient to the location in the list that it is supposed to go based on the currently selected sort category
3. Add a “Search for ingredient” text box
4. Add an event handler to the “Search for ingredient” text box
   1. When pressed, pop up a keyboard for the user to input something to search for in the text box
   2. When the user enters in the ingredient they want to search for, search their list of ingredients by name and return all of the ingredients with a matching name
      1. If no ingredients have a matching name to the one they searched, display a prompt stating that “No ingredients were found with that name”

**Logging In Functionality (Michael Guyer):**

1. Add an event handler to the “Login” button that when pressed operates as follows:
   1. Start a counter at 0 for login attempts
   2. Database of accounts is queried using the entered username
      1. If username cannot be found…
         1. Display a prompt stating that the “Username and/or password does not exist”
   3. If the username is found in the database…
      1. Check that the entered password matches the password connected to the queried username
         1. If the passwords do not match, then display a prompt stating that “Username and/or password does not exist”
         2. Increment counter for login attempts
         3. If login attempts counter is 10, then message to email of account with entered username an email with a message making them aware that someone is trying to enter their account
   4. Give the user access to the account with the entered login credentials
2. Add an event handler to the “Guest Mode” button that when pressed operates as follows:
   1. Load the default information and data that has been saved on the device
   2. Give the user access to the guest account of the device

**Posting comment function**

1. Add a “Submit” button
2. Add an event handler onClick to the button
   1. When it is clicked:
      1. Connect with the backend to upload the comment to the database
      2. Call a newly updated list of comments from the database
      3. Display new comments on the window

**Report function**

1. Add a flag button
2. Add an event handler onClick to the button
   1. When it is clicked:
      1. Check the id for the recipe itself and the author
      2. Find all the recipes from that specific author
      3. Move those recipes to the “hidden” section in the database
      4. Call newly updated recipes and display them

**Rating function**

1. Add five stars button
2. Add an event handler onClick to the button
   1. When it is clicked:
      1. Using the loop, increase the index based on how many stars were clicked
      2. Connect with the backend to upload the rating to the database
      3. Display filled rating stars based on how many a user has clicked.
   2. When it is clicked again:
      1. Deletes the current rating from the database
      2. Update with the new rating
      3. Display filled rating stars based on how many a user has clicked.

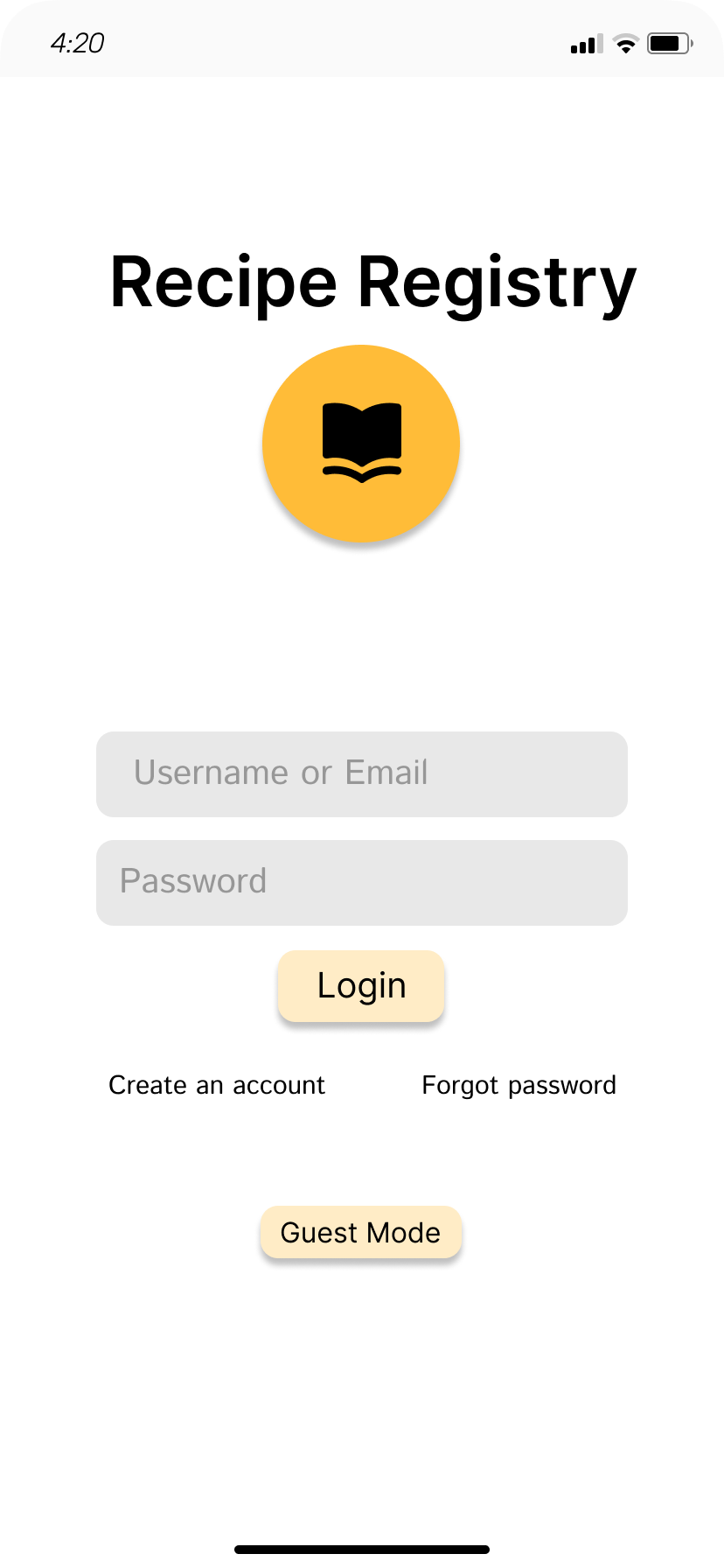
**Notification function**

1. Ask user permission
2. Add an event handler to the recipe comment section
   1. When the other user uploads a new comment:
      1. Check the notification type
      2. Display notification
      3. Closing the notification after 5 seconds

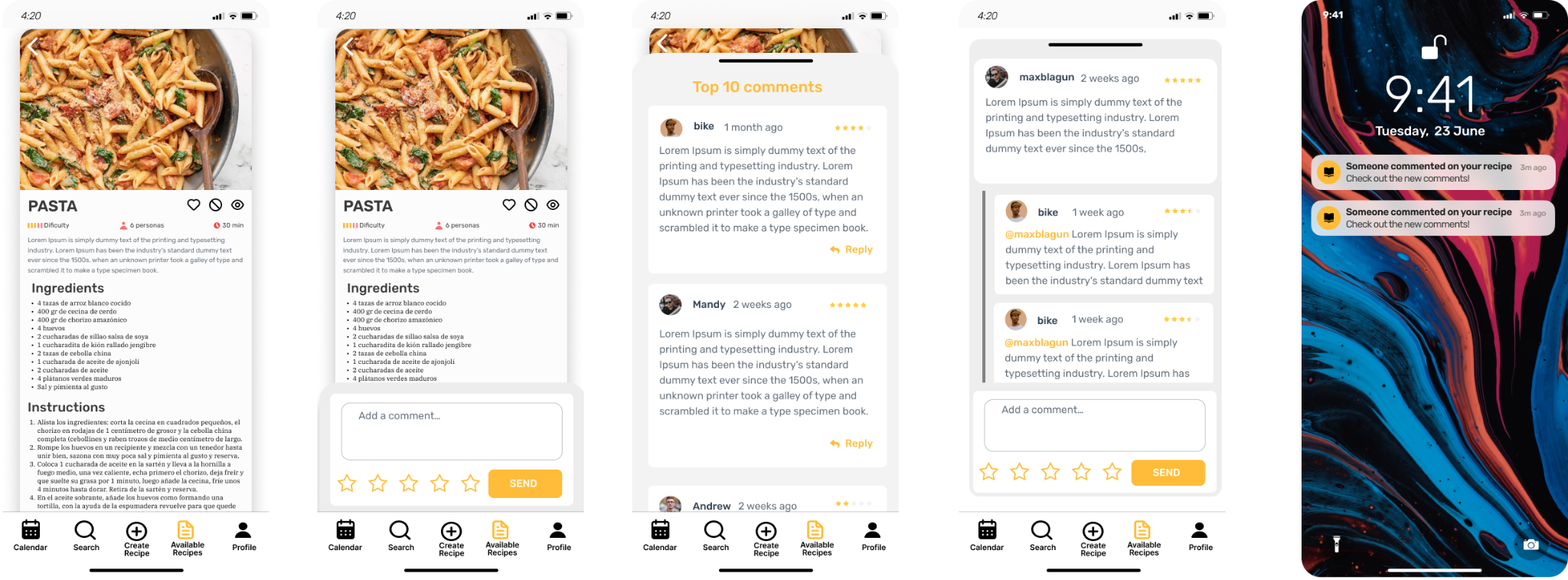
**Updating top 10 comments function**

1. Create a timer for every minute
2. Once the timer goes to 0, call the list of comments from the database
3. Sort all the comments based on the rating from 5 to 0
4. Call the first 10 comments from the list and display them on the screen
5. Reset the timer

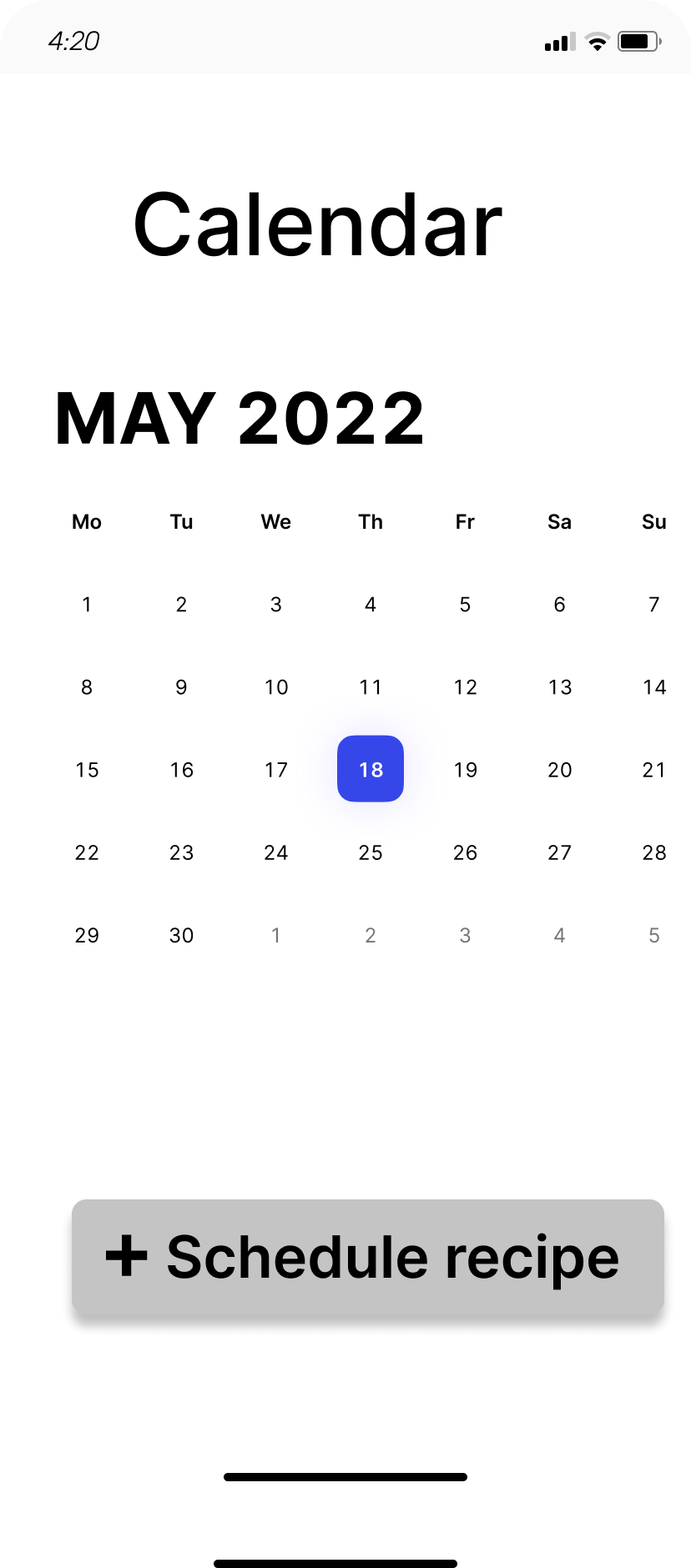
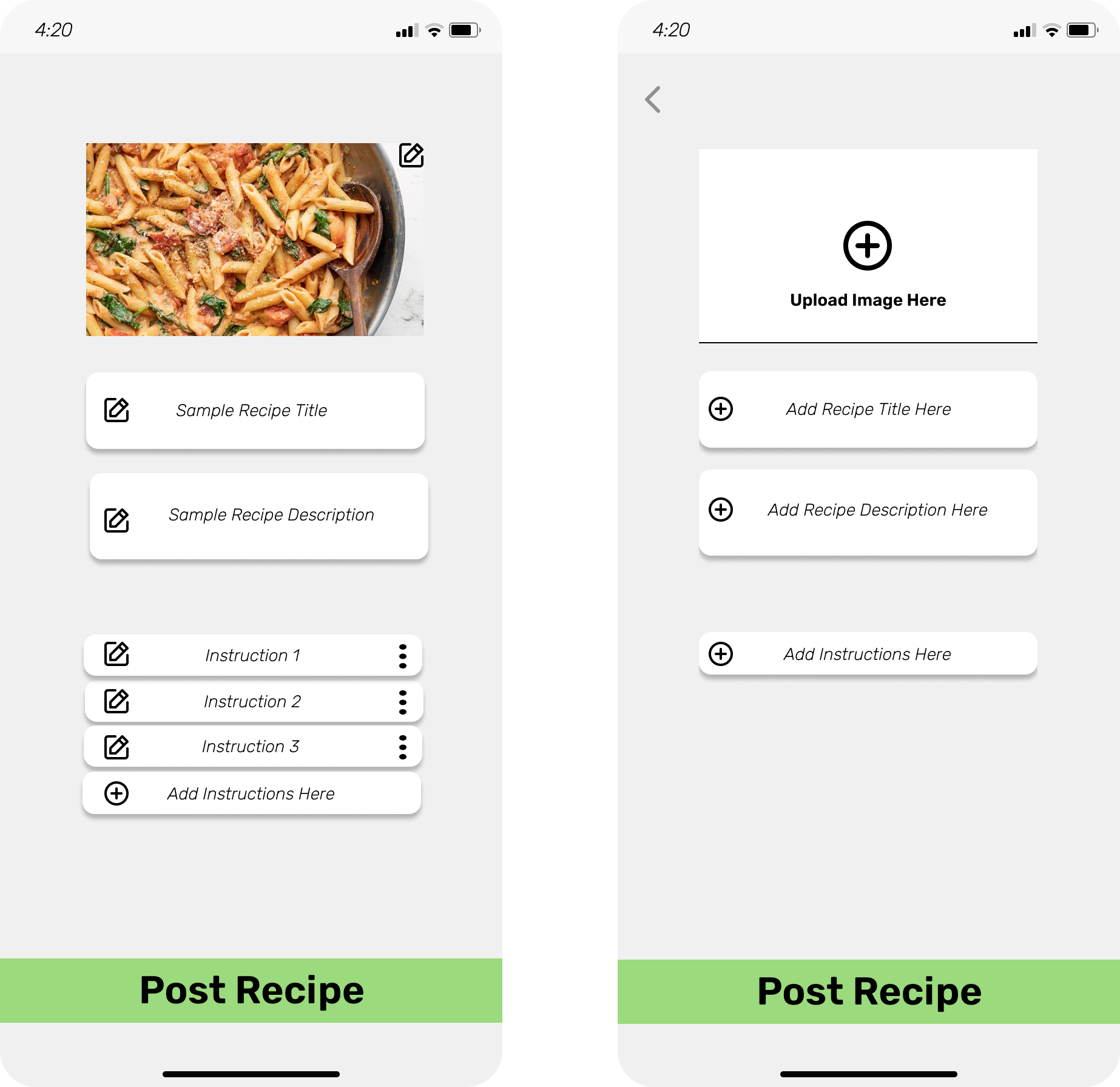
**Login Page UI**



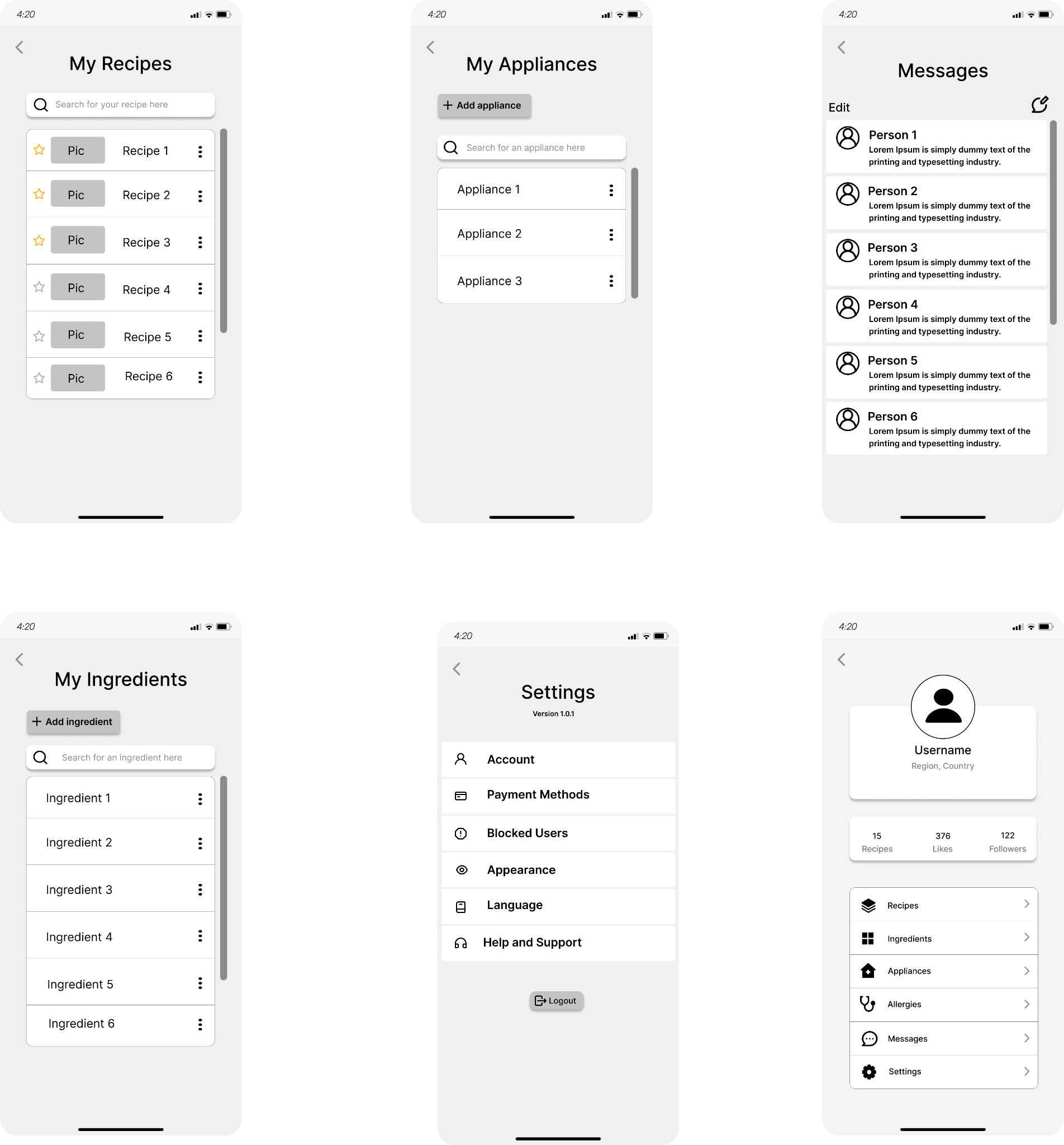
**Recipe Viewing Page, Comment Viewing/Posting Pages, and Notifications UI**



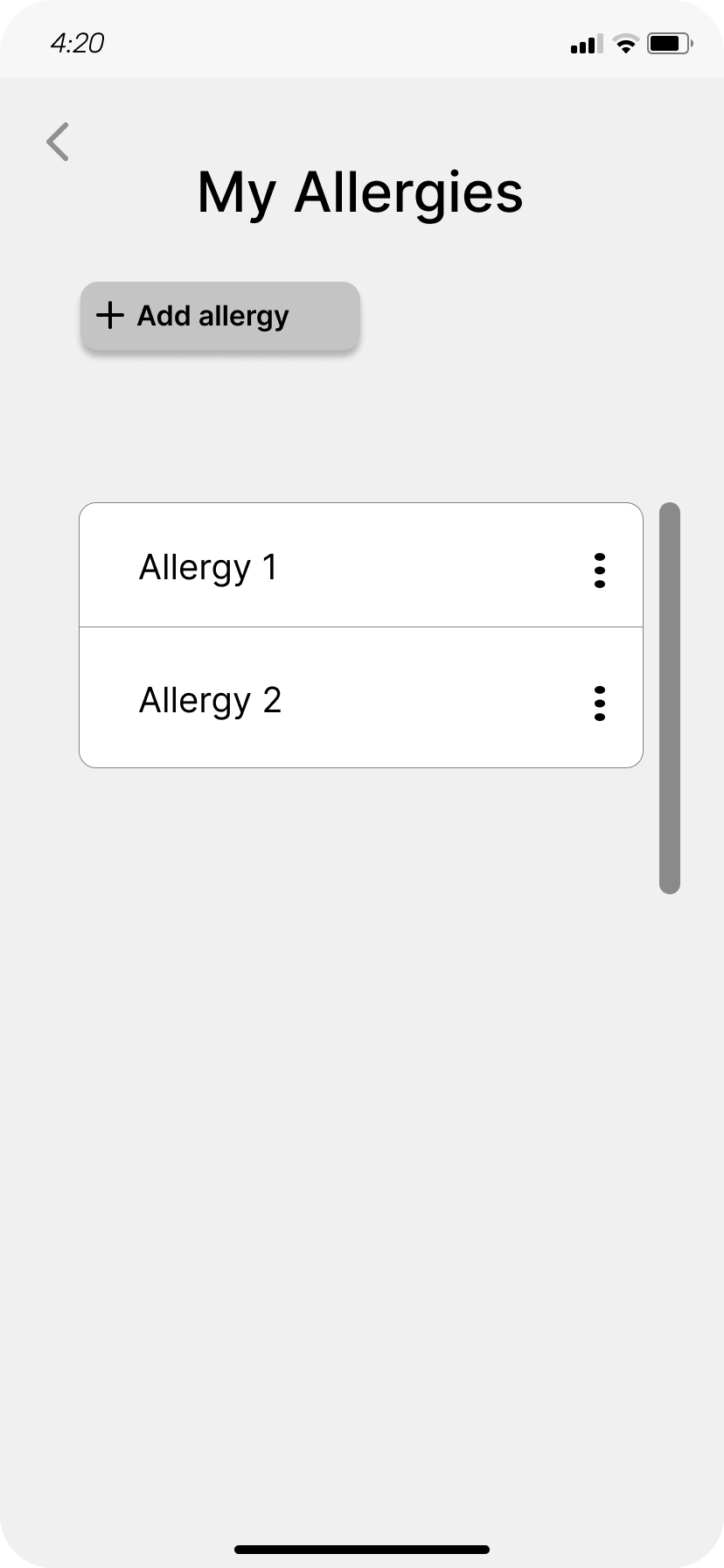
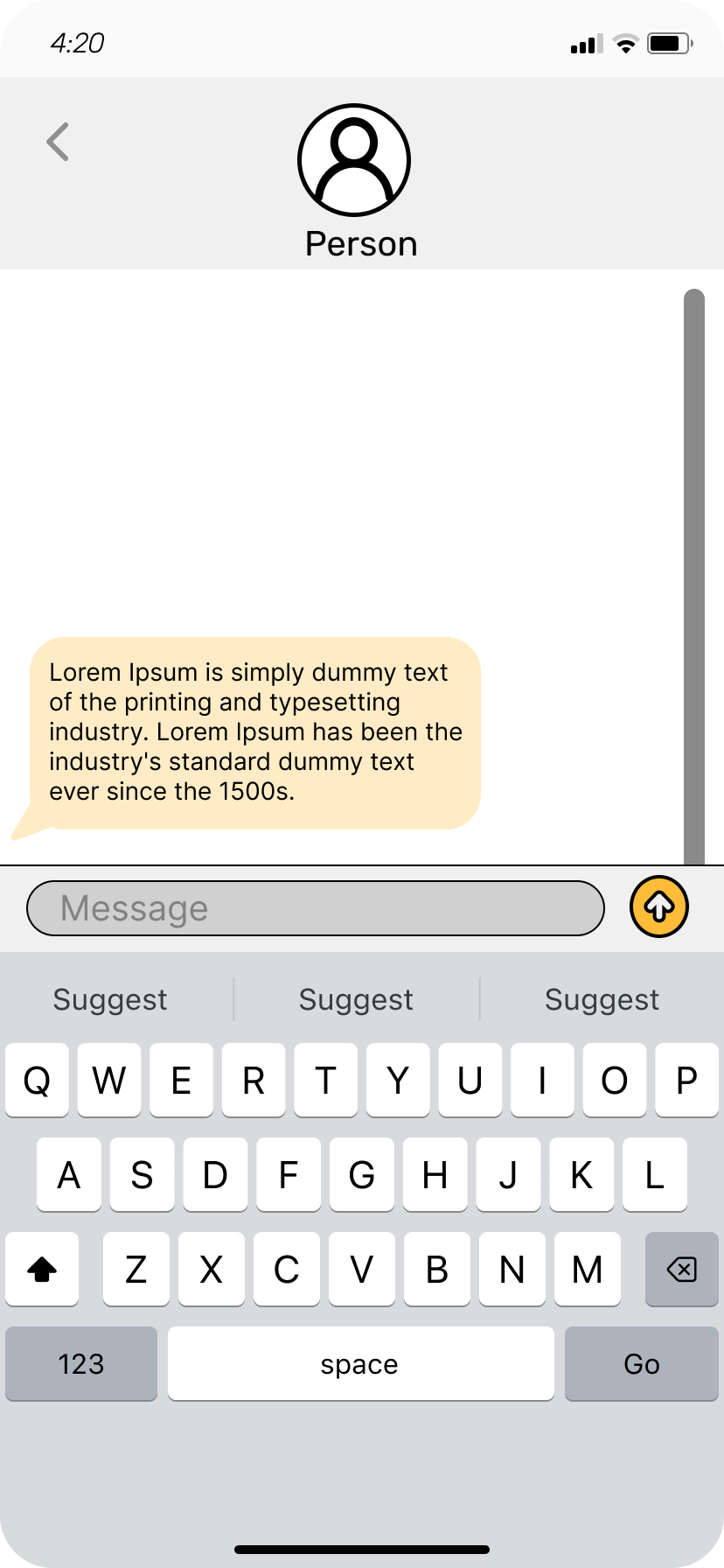
**Recipe Posting Pages UI Calendar Page UI**



**Profile Pages UI**



**Profile Page UI**  **Messages Page UI**

**Recipe Searching Pages UI**

