Pantry: Design Document

Team 15

Yanal Abusamen (yabusame@purdue.edu)

Samuel Carbone (scarbon@purdue.edu)

Joseph Dafforn (jdafforn@purdue.edu)

Joseph Davey (daveyj@purdue.edu)

Jackson Didat (jdidat@purdue.edu)

Jon Du (du129@purdue.edu)

1. Purpose

(a).

Pantry is an iOS application written in the Swift programming language. It will incorporate usage of Google's Firebase for data retrieval and storage. User's will be able to create a profile or log into a pre-existing one. User's will be able to search for or create their own recipes and compare the ingredients to what is in the user's "pantry" to see what is needed that the user doesn't have at home. Other features include an upvote/downvote system, a "cookbook" featuring a user's recipes, and a feed of other users' recipes.

2. Design Outline

(a).

The components to Pantry's client-server model include…

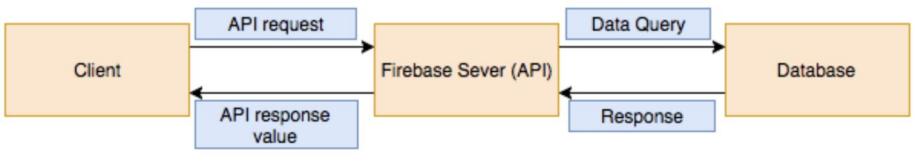
- Client: an iOS Application which users will use to leverage the use of our system.
- Server: A Firebase to communicate with the database.
- Database: A Firebase to keep track of recipes and user profiles.

(b).

Since our system is comprised of a client, a server, and a database interactions will occur as the following:

- Client to server: The client will send requests for information to the server. Ex. A query for spicy recipes or a certain user's favorite recipes.
- Server to database: The server will query the database for the information that was requested by the client.
- Database to server: The database will return the data which was queried by the server.
- Server to client: The server will send back to user the data that it received from the query to the database.

(c).



3. Design Issues

(a).

Issue 1: What feature will the application initially launch on startup?

- Your Profile
- **Feed of Recipes (Discover Tab)**
- Cookbook (ie. The user's saved recipes)
- The User's Pantry (ie. The user's current ingredients)

Solution: We considered launching into the User's Pantry, however we decided to go with option 2 (opening to the feed of recipes) because we feel it will be most comforting for them to immediately see recipes for them to cook, since it is most likely the reason they have launched the app.

Issue 2: How should the discovery page be laid out?

- Traditional Tableview(Multiple recipes in a scrolling view)
- Single Scroll View (Single recipe at a time in a scrolling view)
- **Swiping Left/Right on each displayed recipe (Tinder-like swiping interface)**

Solution: We have decided to display the recipes in the Traditional Tableview, as this allows the user to see a variety of recipes that they are most interested in. Also, this view will allow for the filter results to be displayed very nicely for the user.

Issue 3: What database software should we use?

- MySQL
- **Firebase**
- Postgres

Solution: In this case, we have decided that Firebase is the best option for this application, because it will allow for the recipes to be stored efficiently, and in a reliable manner. FIrebase is an optimal service because it will allow the team to quickly store and retrieve the needed information.

Issue 4: How do we Authenticate Users?

- **Login/password**
- OAuth (Facebook, etc.)

- No login (user is not unique)

Solution: Authentication is best handled with Firebase, since it allows for users to create accounts with their email/password, also signing in with social media. All relevant data and information will be stored in our projects Firebase console, making it very straightforward to implement the projects functionality, giving it a more robust design.

Issue 5: How should we get information about for all the different recipes?

- **Use an existing API that may require a subscription to use certain features.**
- Create our own database of foods and recipes that we can add to over time. The database will be small at the beginning; but it will give us the option to add whatever information we want to each recipe or food.
- Design our own API that pulls information from the web and displays it in our application. This will give us the most control over the features.

Solution: Use an existing API is the best option in this case because it will allow the team to avoid populating all of the recipes ourselves, saving us a lot of time and effort. This will allow us to focus our efforts in the user experience and delivering well on our key features. Unfortunately this could give us less control over what features the API has.

Issue 6: Where should the user find his cookbook (list of saved recipes)?

- Cookbook should have its own tab. This could make the app look cluttered by giving it it's own tab.
- Cookbook should be a part of the user profile page. It reduces clutter and allows more content to be added to the profile page.
- **Cookbook should have its own section in the "Recipes" tab. It reduces clutter and keeps the recipes all in one page.**

Solution: We will design a Recipes tab, which will include the Cookbook, Discovery of recipes, and the shared list of recipes. Combining these features into one Recipes tab will reduce clutter, and is ideal for the user, making the experience very straightforward and simple to use.

Issue 7: How should we rate recipes?

- 5 star scale
- **Thumbs up/ thumbs down**

- Amount of views

Solution: Displaying ratings based on thumbs up/thumbs down we can simplify calculations and it makes it easier for users to interact with the UI.

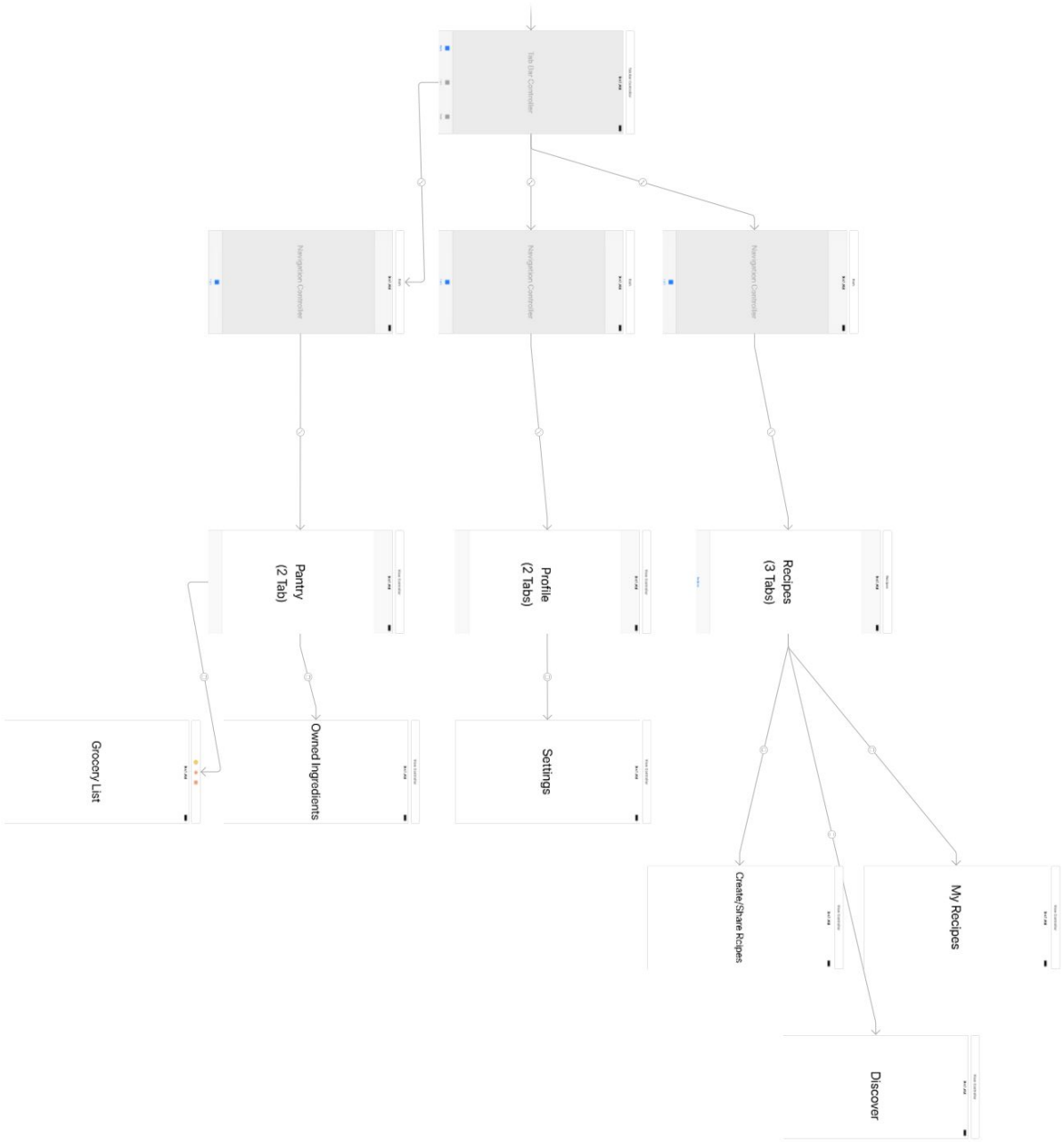
Issue 8: How should the user filter recipes in the discover tab?

- Recipes by location
- Top recipes of the week
- **Sort by Tags**

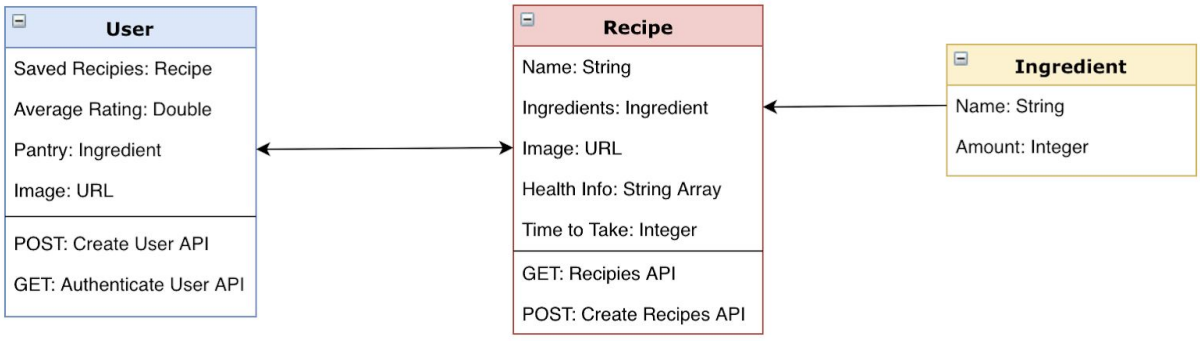
Solution: We decided to show new recipes to discover based on preselected tags. Users will be able to select specific tags and they can see any new popular recipes based on those tags.

3. Design Details

Application Flow (UI Diagram)



Classes



User

- Users query recipes and also have their very own recipes array.
- Users will have saved recipes that they can later query whenever they go to their profile.

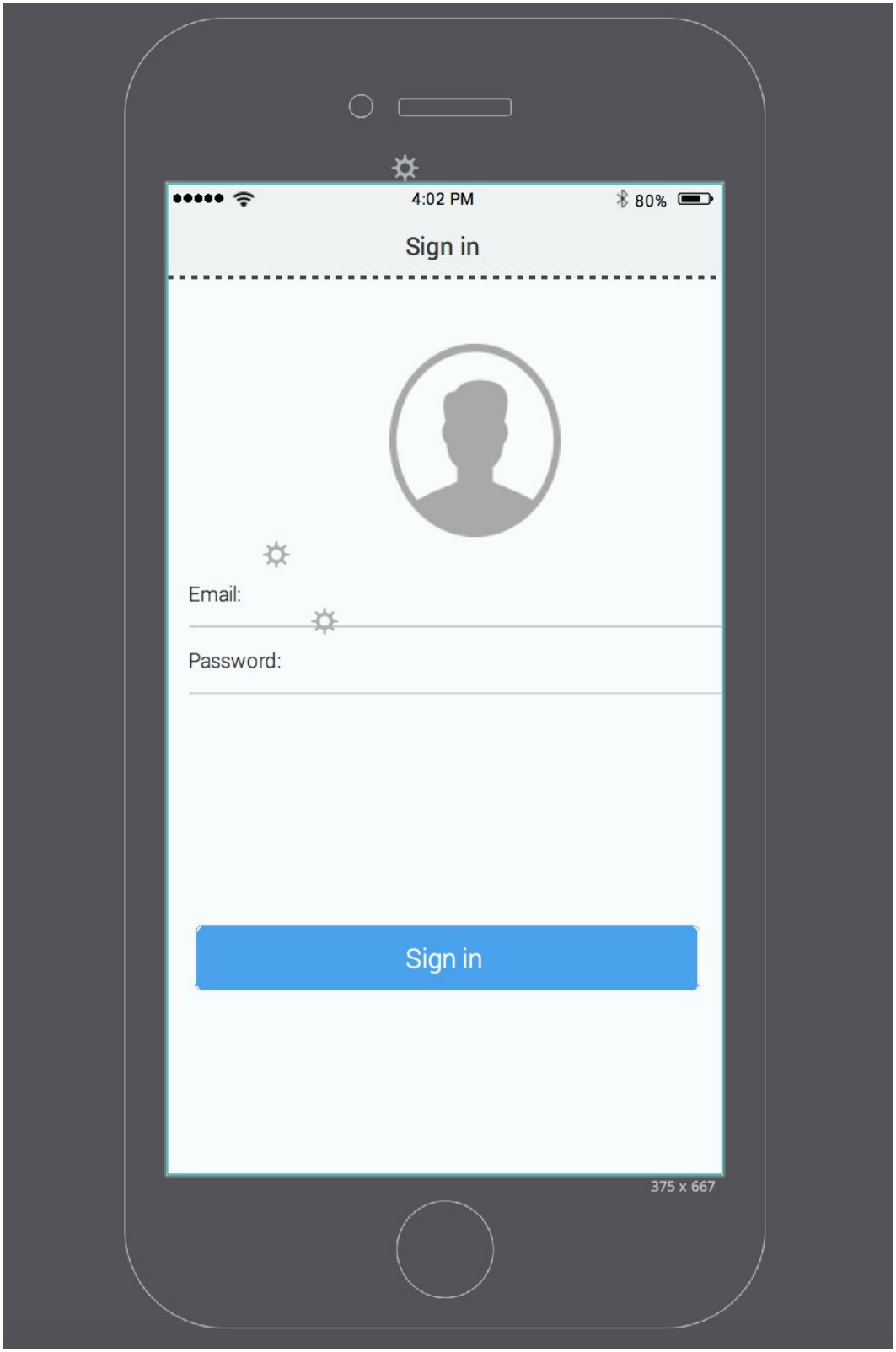
Recipe

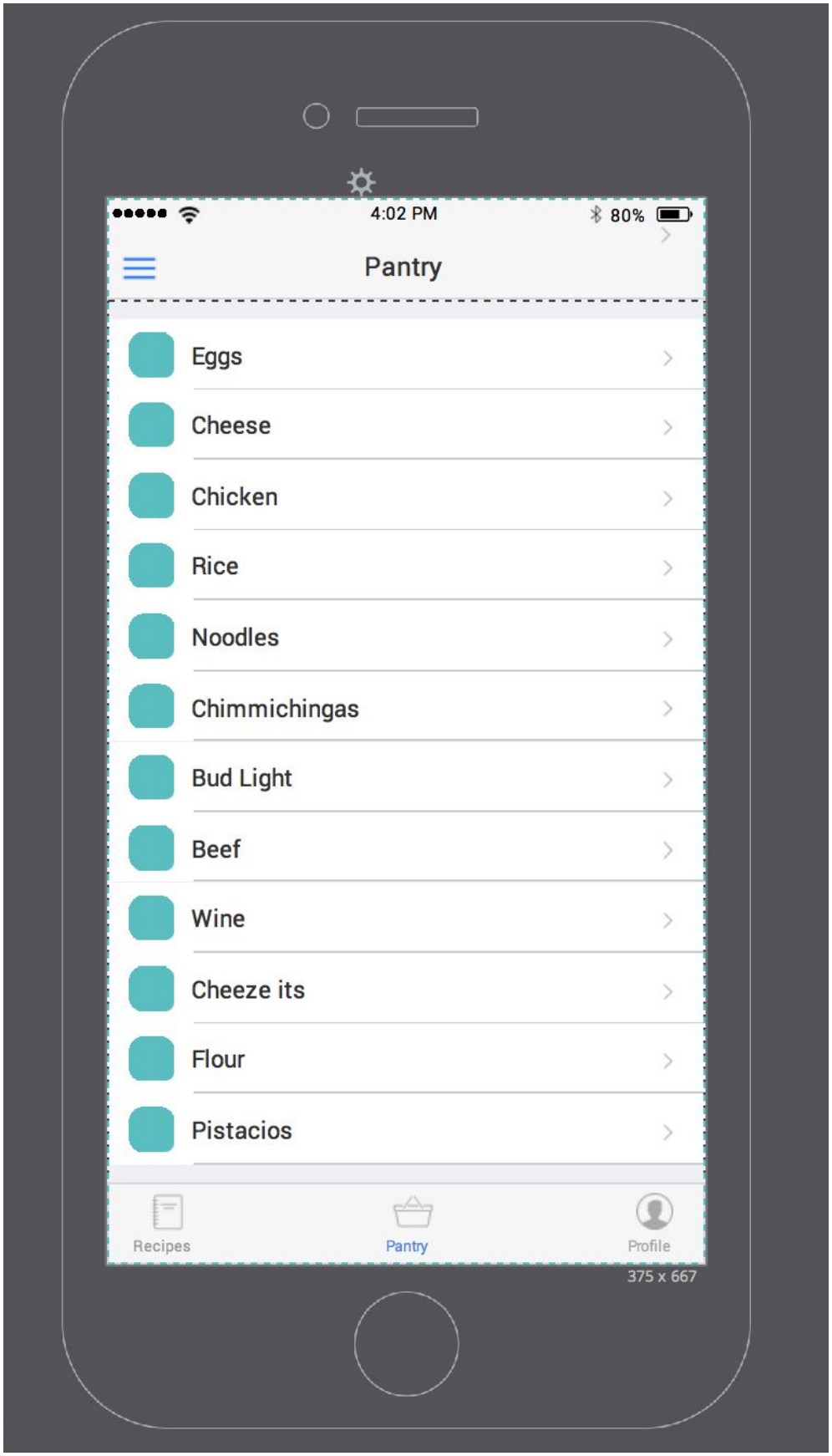
- Recipes will be queries whenever the Recipies tab is used, depending on what the user wants to see.
- Every recipe will have their own ingredient array that contains all ingredients.

Ingredient

- Ingredients contain names and amount currently owned.
- Ingredients data is gathered from the Recipe API call.

Mock-up UI designs




4:02 PM
80%
Pantry
Eggs
Cheese
Chicken
Rice
Noodles
Chimmichingas
Bud Light
Beef
Wine
Cheeze its
Flour
Pistacios
Recipes
Pantry
Profile
375 x 667

4:02 PM

80%

Button

Discover

General Tso's Chicken

Sam Carbone

Asian Cuisine

Grilled Chicken Sandwich

Jackson Didat

Five-minute recipes

Fillet Mignon

Jonathan Du

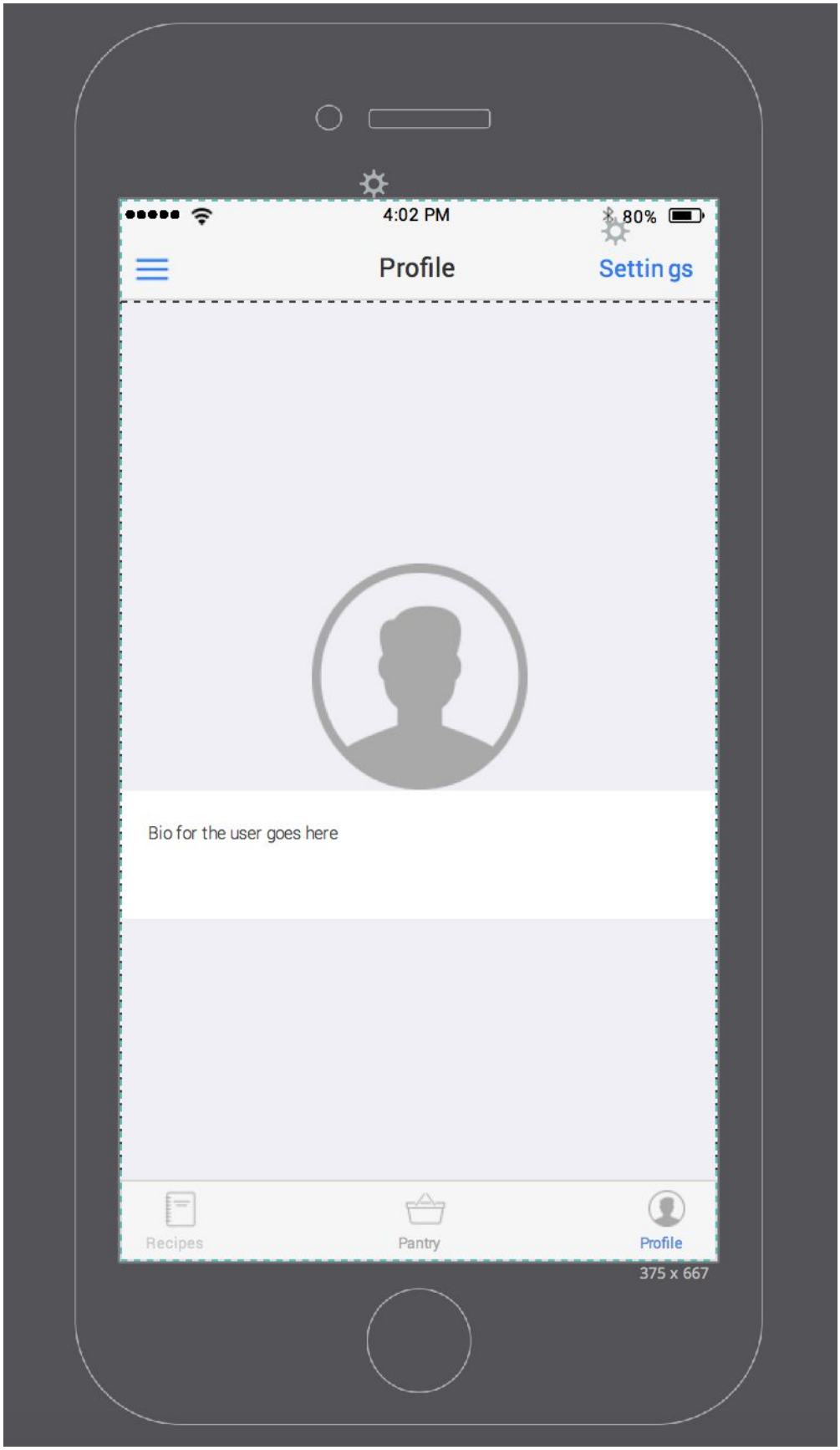
Luxury Meals

Discover

Search

My Recipe s

375 x 667


4:02 PM
80%
Profile
Settin gs
Bio for the user goes here
Recipes
Pantry
Profile
375 x 667