DankMemes

Final Report

Uneeb Islam	**8268634**
Abdullah Baig	**7266683**
Henry Wu	**8247774**
Jiaqi Yuan	**7815849**
Amer Adas	**5946239**

Table of Contents

Introduction

This is a revision of all pre-development deliverables for our android app, Cook Helper. Detailed below are all of the improvements and fixes. In the “Software Requirements”, we have revised each requirement and made them consistent with the app. We have also listed all features that were deemed unnecessary and that had to be cut from the final product. Furthermore, we have expanded our domain UML diagram into a system model by including the UI and utility classes. Finally, we have included updated screenshots of each activity of our final application.

Contributions and Corrections

Member	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Uneeb Islam	25%	25%	25%	25%
Abdullah Baig	25%	25%	25%	25%
Henry Wu	25%	25%	25%	25%
Jiaqi Yuan	25%	25%	25%	25%
Amer Adas	25%	25%	25%	25%

Deliverable	Changes/Correction
Deliverable 1	Some requirements cut, some revised
Deliverable 2	Made full system model
Deliverable 3	New UI classes added

Software Requirements

Cut from final product due to time restraints

1. The system shall allow the user to share recipes to social media
2. The system shall have a section linking the user to online cooking tutorials
3. The system shall allow the user to favourite or bookmark recipes.
4. The system shall allow the user to rate recipes.
5. The system shall have a section for users to set dietary restrictions or preferences.
6. The system shall allow the user to compare two different recipes for the same dish.
7. The system shall allow the user to get a random recipe from the list of available recipes.
8. The system shall allow the user to label the calorie intake of the meal.
9. The system shall provide 5 sample recipes.

Revised functional requirements

1. The system shall allow the user to see the list of all ingredients.
2. The system shall allow the user to add recipes.
3. The system shall allow the user to modify recipes.
4. The system shall allow the user to delete recipes.
5. The system shall show confirmation when a recipe is added, deleted, or modified.
6. The system shall allow the user to search for recipes using:
    a. ingredients
    b. ethnicity
    c. category
7. The system shall allow the user to filter out ingredients in the search list.
8. The system shall display the recipe list alphabetically
9. The system shall allow the user to make a shopping list of ingredients to buy.
10. The system shall provide step by step directions on how to cook the dish when a specific recipe is retrieved.
11. The system shall allow recipes created by the user to be saved in a database.

Revised Non-functional requirements

1. The system shall allow the user to arrive at a recipe in less than 30 seconds.
2. The system shall be compatible with Android devices running Ice Cream Sandwich and up.
3. The system shall require the user to login to use the app.
4. The system shall be finished development by December 5th.
5. The system shall require no more than 50MB of storage.

Use Cases (3 distinct cases start to finish)

Use Case: Add a recipe

Actors: The person/people using the system

Goals: Add a new recipe into the system

Steps:

Actor Actions:	*System Response:*
1. Tap ‘Add a New Recipe’	Open new recipe page
2. Type the name of the recipe	
3. Select the categories of the recipe	
4. Add the ingredients/equipment needed	
5. Add steps	
6. Tap ‘Done’	The page closes

Postconditions: The new recipe now appears in the list and can be searched for

Use Case: Searching for a recipe by name

Actors: The person/people using the system

Goals: Look for a recipe

Related Use Cases:

Specialization of: Searching for a recipe

Steps:

Actor Actions:	*System Response:*
1. Tap ‘Search for a Recipe’	Displays search bar
2. Type in recipe name	
3. Tap ‘Search’	Displays recipes with that name
4. Tap a recipe	Displays that recipe

Postconditions: The recipe the user wanted is open

<u>Use Case: Modifying a Recipe</u>

Actors: The person/people using the system

Goals: Edit an already existing recipe

Related Use Cases:

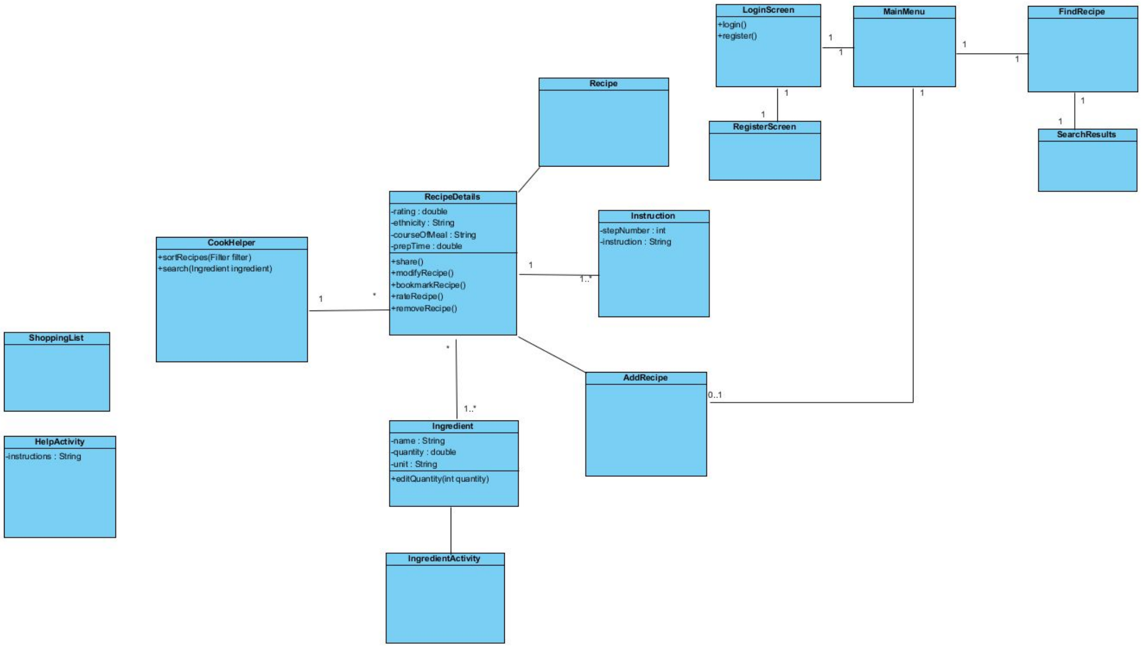
Includes: Searching for a recipe

Steps:

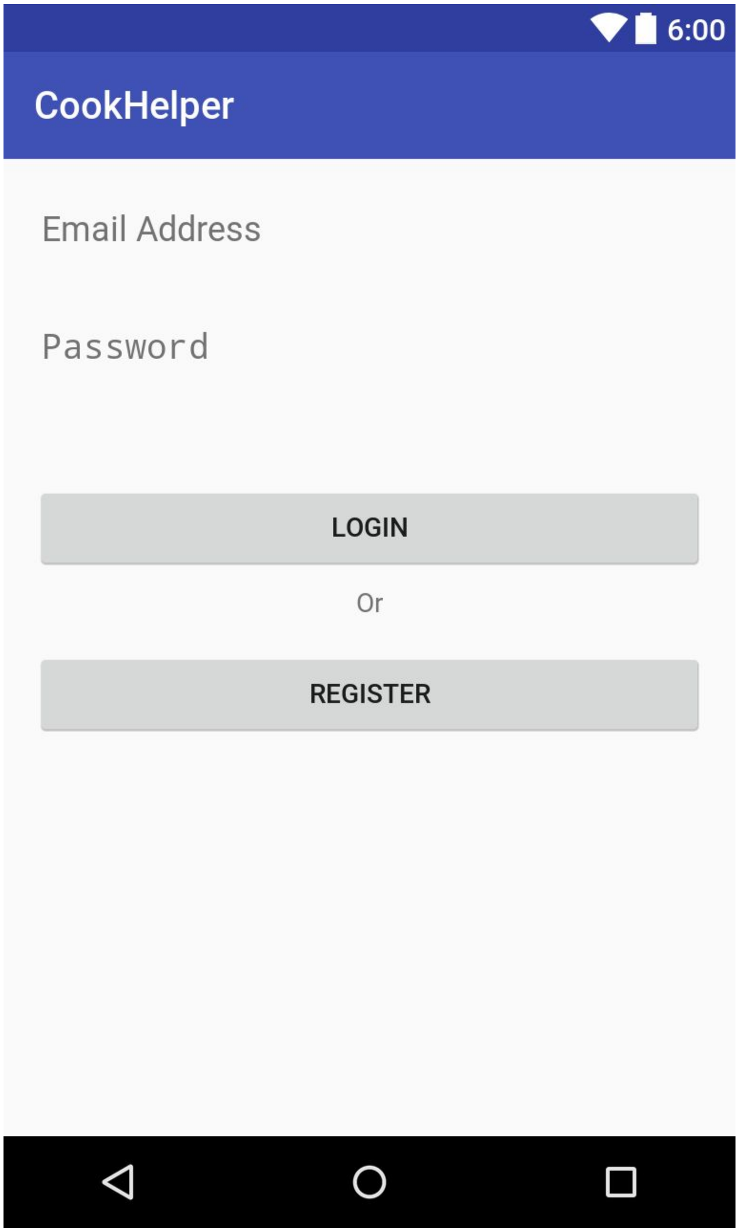
Actor Actions:	*System Response:*
1. Search for a recipe	
2. Click edit	Opens a new dialog box
3. Edit the recipe as desired	
4. Tap ’Done’	Dialog box closes
5. Tap ‘Done’ again	Recipe closes

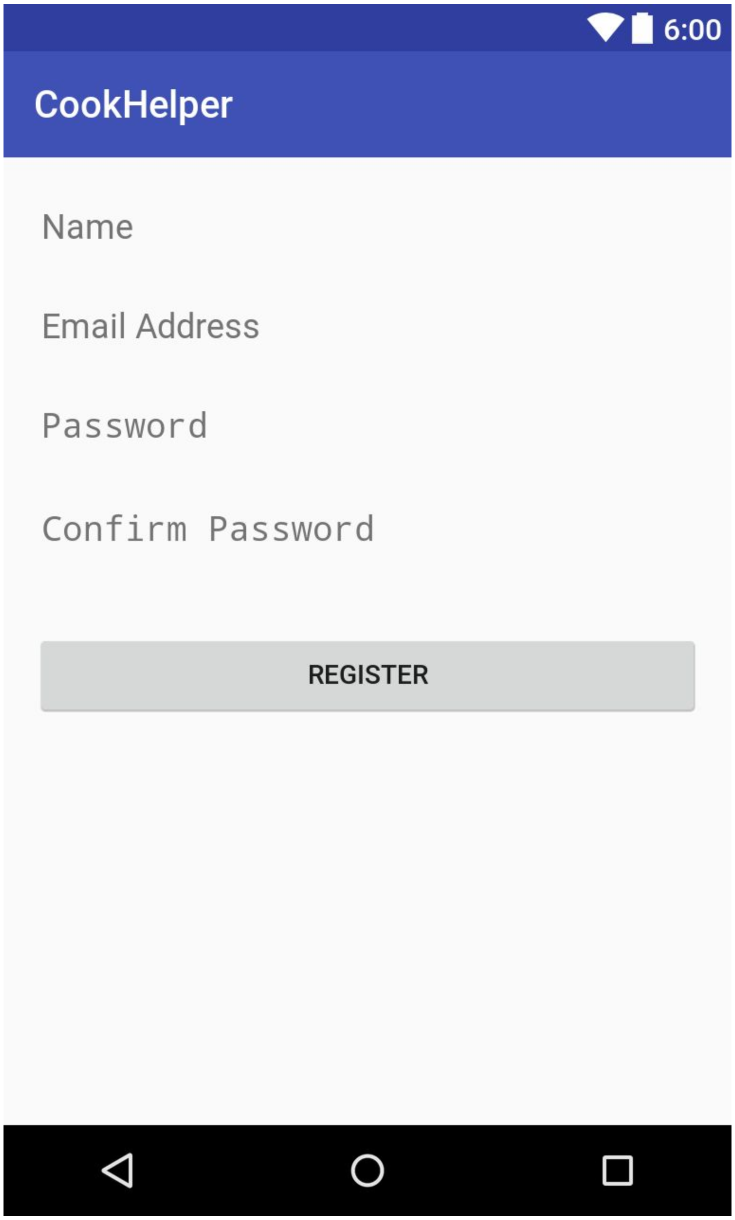
Postconditions: The recipe changed according to the edits

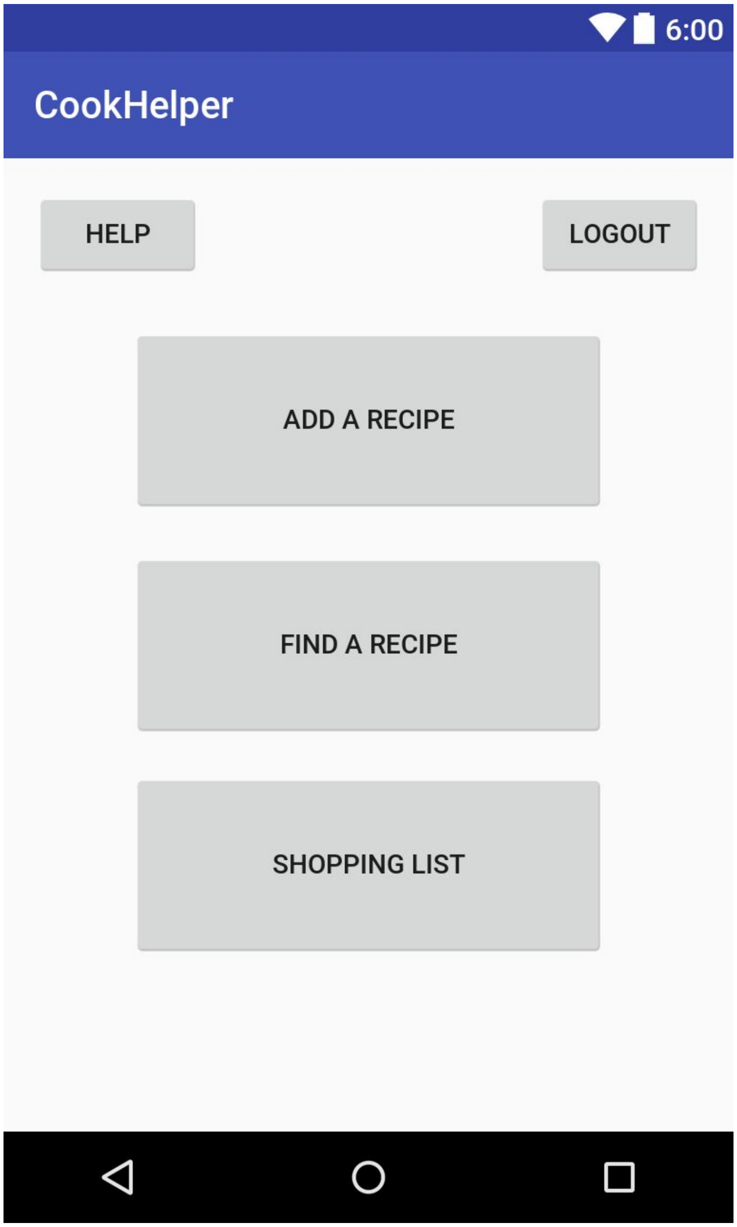
UML System Diagram

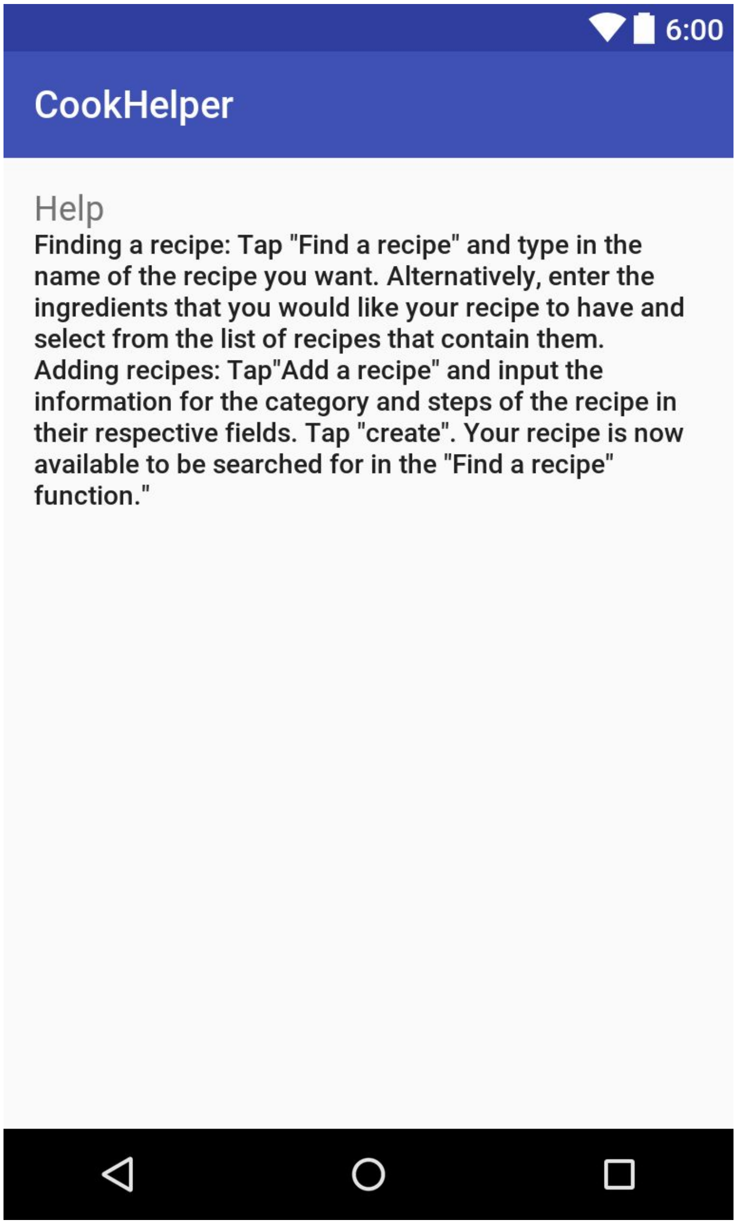


Screenshots of UI


6:00
CookHelper
Email Address
Password
LOGIN
Or
REGISTER

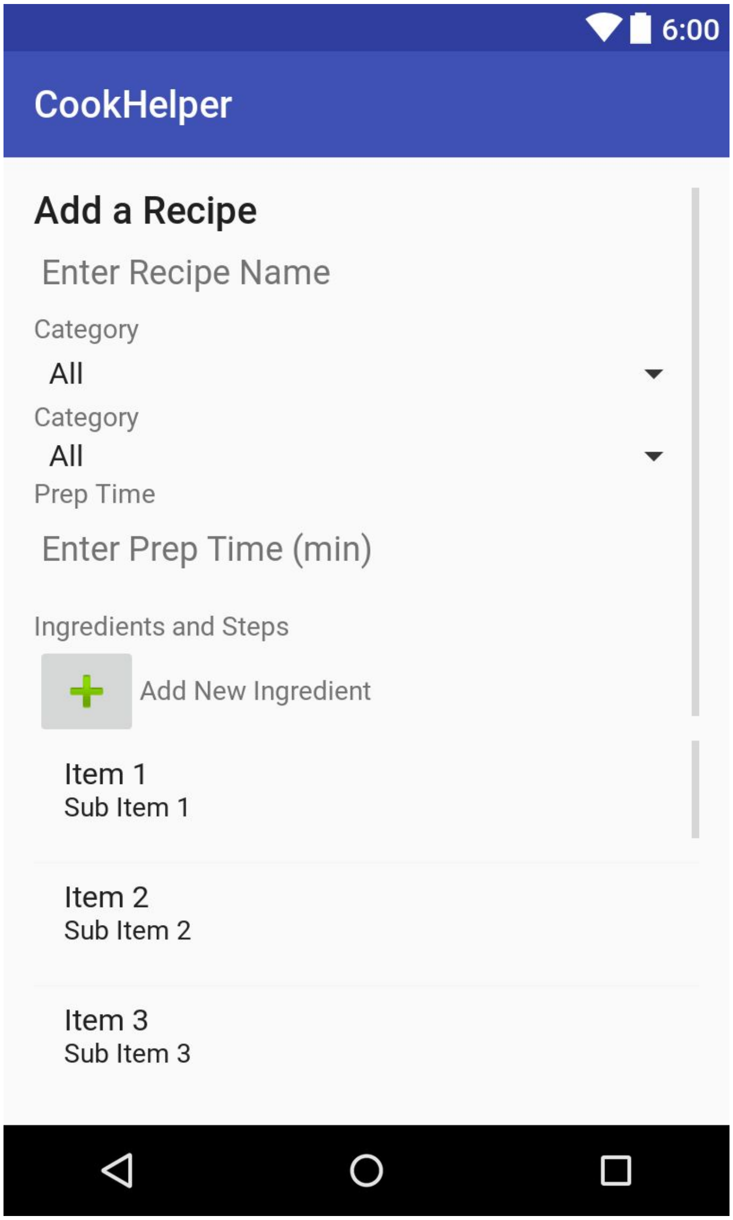

6:00
CookHelper
Name
Email Address
Password
Confirm Password
REGISTER


6:00
CookHelper
HELP
LOGOUT
ADD A RECIPE
FIND A RECIPE
SHOPPING LIST



Help

Finding a recipe: Tap "Find a recipe" and type in the name of the recipe you want. Alternatively, enter the ingredients that you would like your recipe to have and select from the list of recipes that contain them.
Adding recipes: Tap"Add a recipe" and input the information for the category and steps of the recipe in their respective fields. Tap "create". Your recipe is now available to be searched for in the "Find a recipe" function."



6:00

CookHelper

Add a Recipe

Enter Recipe Name

Category

All

Category

All

Prep Time

Enter Prep Time (min)

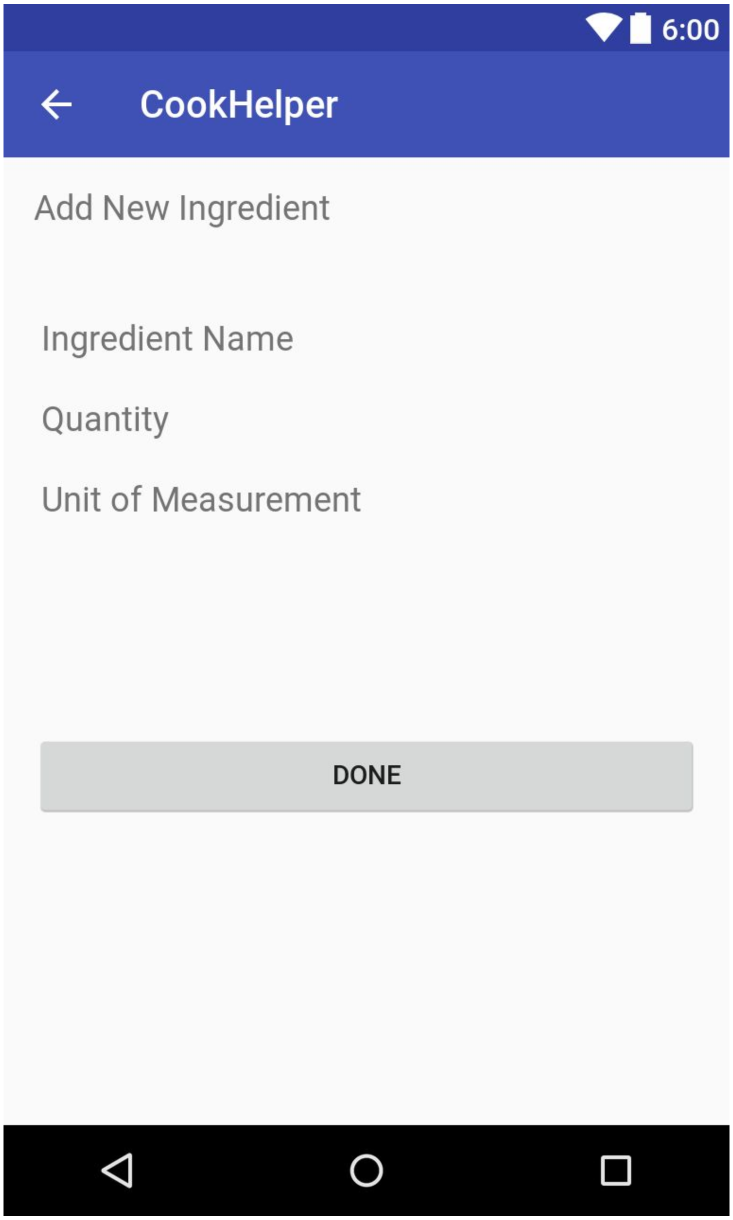
Ingredients and Steps

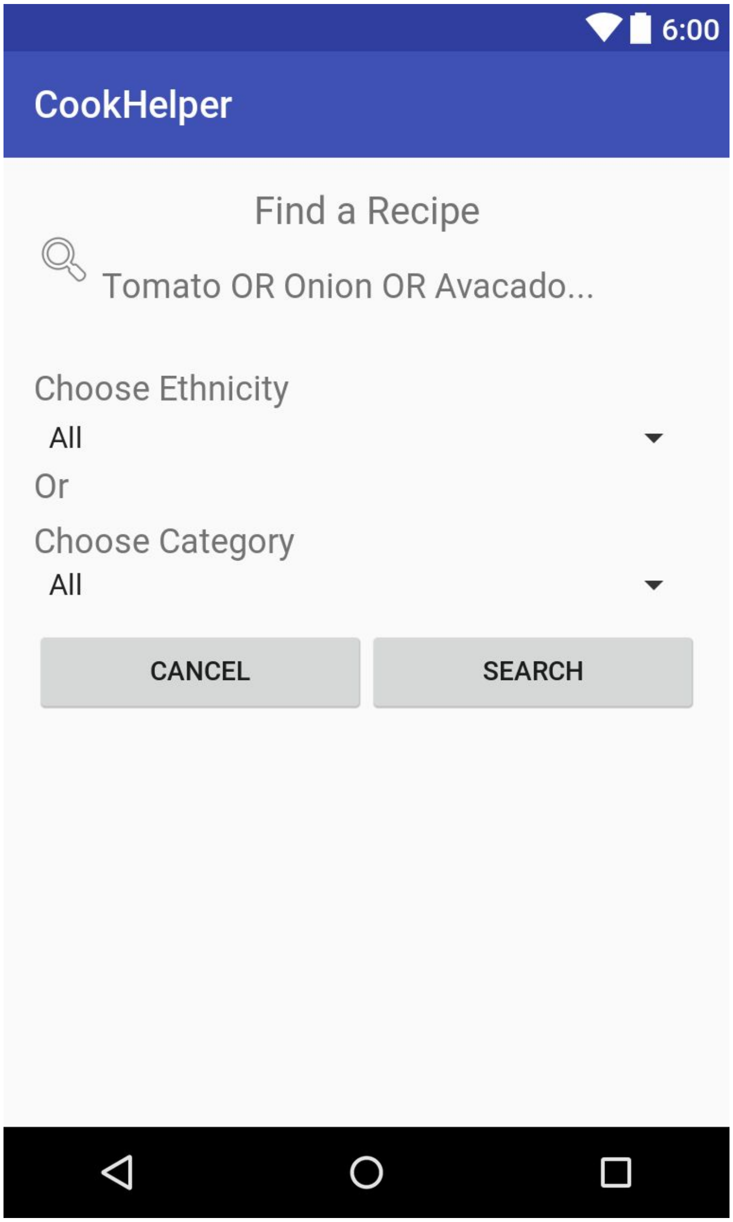
Add New Ingredient

Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3


6:00
CookHelper
Add New Ingredient
Ingredient Name
Quantity
Unit of Measurement
DONE



Find a Recipe

Tomato OR Onion OR Avacado...

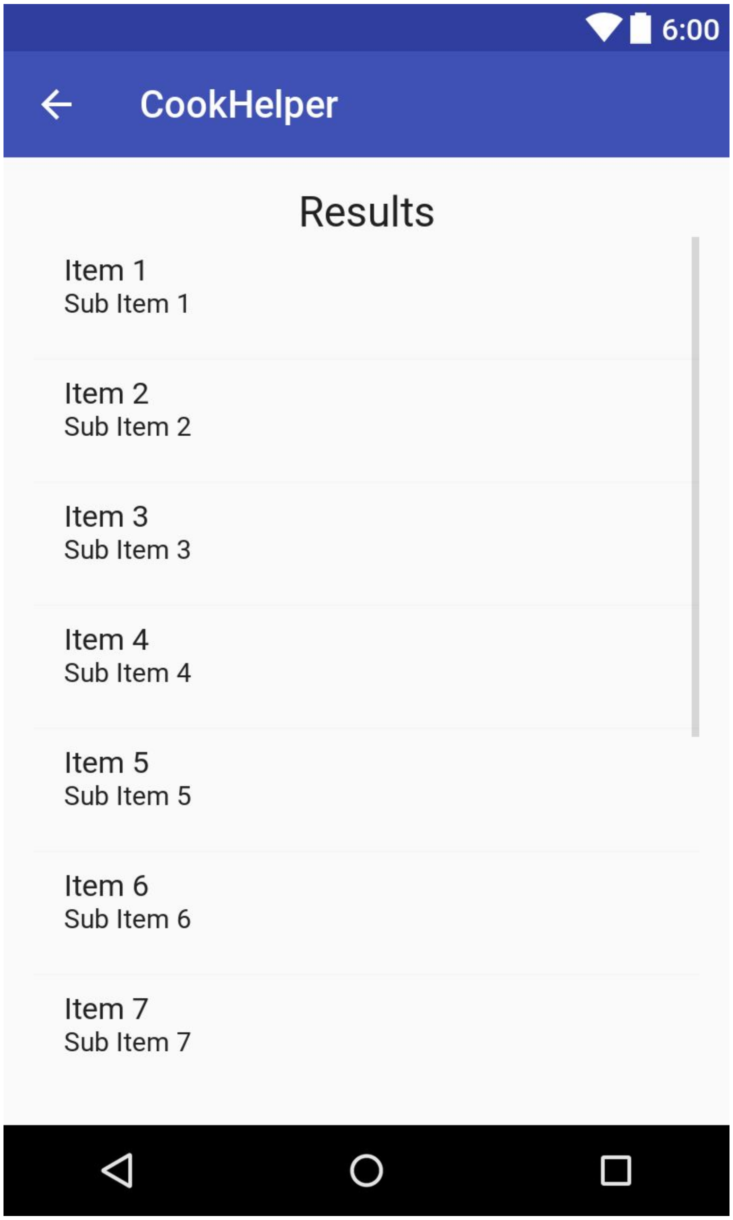
Choose Ethnicity

All

Or

Choose Category

All



Results

Item 1
Sub Item 1

Item 2
Sub Item 2

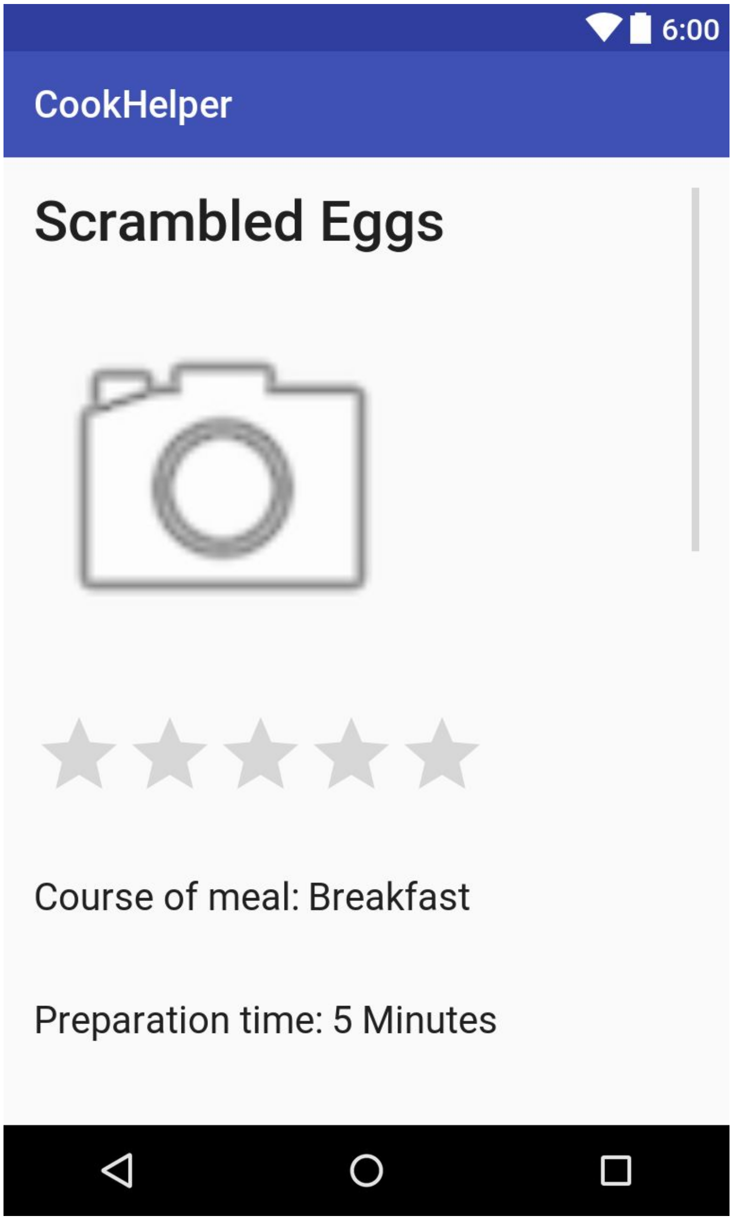
Item 3
Sub Item 3

Item 4
Sub Item 4

Item 5
Sub Item 5

Item 6
Sub Item 6

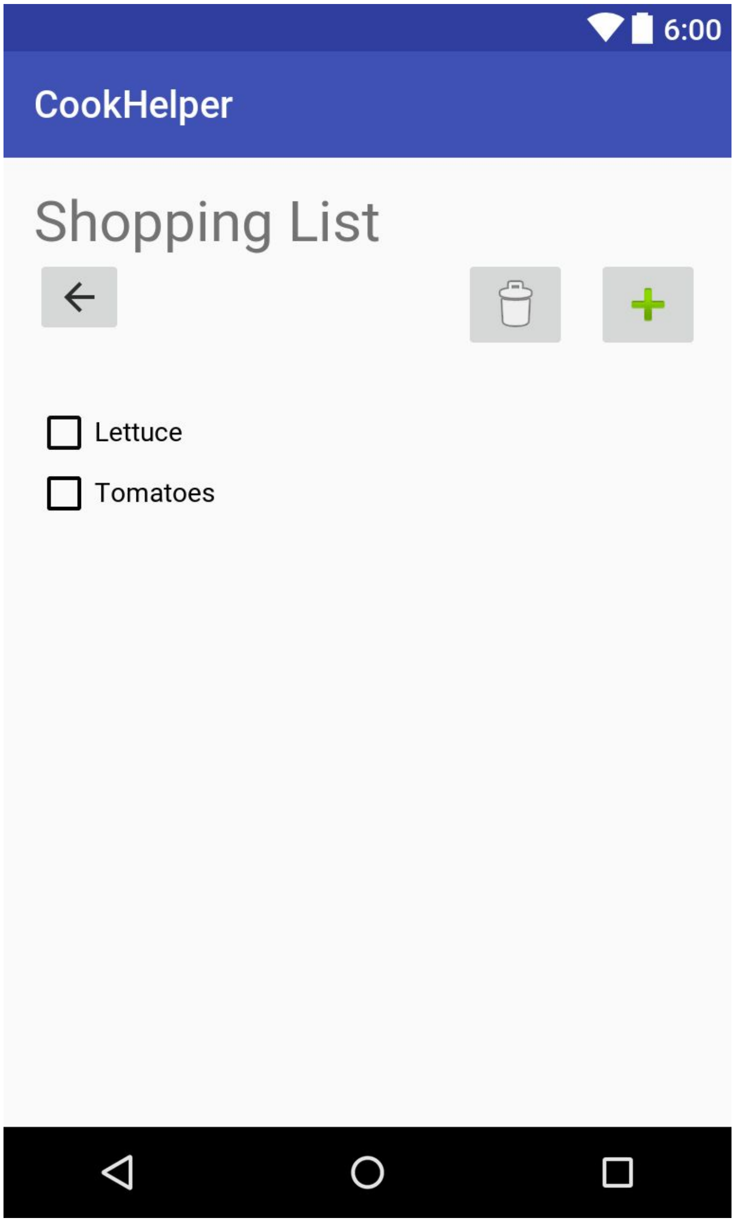
Item 7
Sub Item 7



Scrambled Eggs

Course of meal: Breakfast

Preparation time: 5 Minutes



Shopping List

□ Lettuce

□ Tomatoes