Project Title: ReciPull
Team Name: AJARA
Authors: Alexis Cole, Justin Dong, Rhianna So, Austin Wong, Allen Yao
Member Emails: alexisrosecole@gmail.com, jsdong@ucsb.edu, rhiannaso@ucsb.edu, austinwong@ucsb.edu, allenxueyiyao@gmail.com

Vision Statement

Project Summary: ReciPull is a web application designed by college students for college students. A common problem of college students around the nation is the inability to produce quality food on a tight budget. Many students find themselves at a loss for inspiration in different recipes to cook. ReciPull is the perfect application to solve this crisis. A user is able to input their available ingredients to see a list of various recipes to cook from a database. With this application, college students can easily access recipes that they can make with the ingredients they already have, which makes it easy for them to cook when they are on a tight budget or do not have time to go grocery shopping.

Outcome: ReciPull will be a working web-app that will take user input on what ingredients they are working with and will output a list of recipes that they could cook using their available ingredients. Each recipe will be accompanied with essential information: nutritional facts and the number of owned ingredients included. Recipes can also be upvoted or liked, allowing users to see the most favorited recipes, along with other options to filter recipes. Finally, Recipull will integrate Echo voice recognition for easier search capabilities and employ a social media platform that will allow users to connect with their friends for recipe and ingredient sharing.

Project Milestones: We intend on having all initial project specifications solidified by April 12. Specifications can be modified as the project progresses, however. We plan to have a rough
design of the overall layout of the web-app by April 24. We want to have a working prototype that resembles the final product by May 10. The working prototype should have the front-end, back-end, and database components all connected and functioning properly together. One or two key features should be implemented and working according to specifications. Testing should be done throughout the process of implementing the web-app, especially after the implementation of each new feature. Final testing should be done by June 5 to ensure complete functionality, leaving time to resolve minor issues or improve upon certain features.

**Implementation Platform and Technologies:** We plan on developing a web application for ReciPull and would like to develop ReciPull into an Alexa Skill as well. The technologies we will utilize for the client side code of the web application are React, HTML, and CSS. The technology we will be utilizing for the server side code is Java. For the database, we will be using Amazon Web Services and MySQL. We will utilize Pinterest API for the social media aspect of our web-app. For the Alexa skill, we will be using Python (may switch once we implement the web framework) and JSON.

**Initial Milestones (Sprints/Dates) and Planned Implementation:** By the end of Sprint 1, on 4/26, we want our web app to have the following functionalities: the ability to take a list of ingredients as user input and run a search to find recipes with said ingredients. This is the main purpose of ReciPull, thus we want to focus on this first. We plan to use an API to implement this. If possible, this would also be where we would integrate the echo, allowing the user to speak ingredients that trigger the search. By the end of Sprint 2, on 5/10, we want to implement a social networking aspect of ReciPull. We plan to use the Pinterest API to allow users to share which recipes they have tried. We would also like to create a platform for users to be able to borrow
and lend ingredients. By the end of Sprint 3, on 5/22, we want to implement health and tracking functionalities of the web app. These would include nutritional facts for recipes, information on what is in season, and meal planning and tracking capabilities.

**Completed Tasks During Sprints:** By the end of Sprint 1, on 4/26, we were able to set up the recipe and ingredient databases, as well as retrieve applicable recipes given an ingredient and retrieve ingredients given a recipe. We achieved this by connecting the back-end (Java) to the database and created dummy Java programs to demonstrate this, since we had not yet connected the front-end to the back-end. In terms of front-end, we finished each individual component of the user interface (ingredient list, recipe list, header) but did not get the chance to combine all components into a cohesive page. In terms of the back-end, we created templates for recipe objects and set up the framework to process through recipes retrieved from the database. We also began to experiment with the Amazon Alexa. By the end of Sprint 2, on 5/10, we were able to update the database to handle recipe upvoting and downvoting, as well as filters. We created methods to sort through the recipes according to filters, returning recipes that contained a specific filter. We also were able to directly create recipe objects in Java directly from the data we retrieved from the database. We figured out a way to update the actual database information when receiving upvotes or downvotes. Regarding Echo integration, we were able to create a basic Alexa skill and created an Alexa lambda function. For front-end, we combined the previously completed components of the webpage into a seamless user interface. We are in the process of working on combining our front-end and back-end together, as it has been a major blocker during the sprint.
System Architecture

- High-level overview
- Detailed Design
  - UML Diagrams

- Sequence Diagrams
  - User Interaction
- Synchronous Interactions

1. Get Recipes Containing User-inputted Ingredients

1.1 Create Database Connection

1.2 Get Ingredient ID(s) from Given Ingredient Names

1.3 Get Recipe ID(s) that Contain Ingredient ID(s)

1.4 Sort Recipe ID(s) From Containing Most of the Inputted Ingredients to the Least!

1.5 Get Recipe Name(s) from Recipe ID(s)

1.6 Apply Any Filters and Sort Recipes Accordingly

- UI Mockups

![Ingredient List](image)

- Spaghetti & Meatballs
  Basic recipe for Spaghetti & Meatballs. Perfect, quick recipe for a college student.
  See the Recipe

- Yogurt with Granola
  Basic recipe for a balanced breakfast meal. Perfect, quick recipe for a college student to jumpstart their day.
  See the Recipe

- Spaghetti
  This is a Spaghetti recipe
  See the Recipe
Requirements

- Use Cases found on team Trello
  - Use cases under each Sprint task (i.e. UI Card in Sprint 1 List)
- GitHub
- Travis-CI

Appendix

- Technologies employed
  - Java
  - Python
  - JSON
  - React
  - HTML
  - CSS
  - Amazon Web Services
    - Amazon RDS (MySQL)
    - Alexa API
  - Pinterest API