Cookable

By:

Copyright Team 5 CSC 648-848 Spring 2019

Trevor Sampson (Team Lead, Front & Back-end) tsampson1@mail.sfsu.edu

Thanh Dip (Back-end Lead Developer, Database & Back-end)

Pyae Naing (Back-end Developer, Github Master)

Tony Rodriguez (Front-end Lead Developer, UI)

Shivam Rai (Front-end Developer, UI)

Surabhi Chavan (Front-end Developer, UI)

Milestone 1

March 14, 2019	Milestone Version 1
March 28, 2019	Milestone Version 2

1. Executive Summary

Team 5 is designing a web client application called “Cookable” which will be the number one place for accessing food recipes on the web. We aim to create a user friendly and intuitive interface that will allow you to easily find recipes based on the ingredients that users currently own in their kitchens. No more browsing random websites endlessly in search of the right recipe. With Cookable, we will bring the right recipe to you.

Cookable will offer general users the ability to search for recipes based on ingredients they provide to the application. This functionality will come standard to all users who are accessing the website. What really makes Cookable stand out is the functionality that we offer to Registered Users. One main feature includes the ability to store the ingredients you own at home. This allows the website to suggest recipes to the user based on what they currently own. Another feature includes the option to “favorite/like” recipes that users would like to save for later use. Most importantly, registered users will also be able to create and upload their own recipes to the website, which they can then share with their friends, family, and fellow Cookable users. How people prepare and cook their food is a very personal thing, which is why it is important to offer them the option to add and share their own recipes. As more people join and use Cookable, more fun and exciting recipes will become available for people to use and enjoy.

Team 5 consists of 6 San Francisco State students who are pursuing degrees in Computer Science. The team lead is Trevor Sampson. The front-end development team consists of Tony Rodriguez, Shivam Rai, and Surabhi Chavan. The back-end development team consists of Thanh Dip, and Pyae Nang. As a unit, team 5 brings a wide variety of skill sets to the table, and are very enthusiastic and passionate about developing products that users are happy to use in their day to day lives. Team 5 aims to showcase their skills and passion on this project and all their future software development endeavors.

2. Use Cases

Actors:

- Casual user - Users who do not sign in and only use the site’s basic functions.
- Registered users - Users who sign in and save their ingredients.

Persona:

- Emily - A busy college student who wants to cook more.
    - Very familiar with using web applications.
    - Want’s to cook but doesn’t know where and how to start.
    - Can’t afford certain ingredients.
    - Forgetful and always forgets to make a shopping list.
- Haley - A mother who is is well in her late 40s.

- She is not familiar with web applications.
- She gets easily confused by complex UI.
- She is a seasoned cook and wishes to share her recipes with her niece Emily.

Case:

Casual

- A casual user, Emily, wants to cook something at home but she doesn't know what she can cook and is not familiar with recipes. She goes onto cookable to find possible recipes she can cook with the food only in her fridge. She finds that she can make a lot of different food with the simple ingredients in her fridge.
- Emily is at the grocery store and she wants to cook something at home but doesn't know what to get. Her budget is limited as well so she can only buy cheap ingredients. She opens up the website and enters in the ingredients she can buy and it gives her all the recipes she can cook under a budget.

Registered

- Emily wishes to get a family recipe from Haley so she registered and asked Haley to do the same.
- Haley decided to create an account with cookable because she wishes to share the family's recipes with her niece Emily to cook. She finds it easy to sign up and then uploads her recipe to cookable and sent it to Emily.
- Emily receives the recipe with the instructions and amount of ingredient needed to cook her family's recipe and she has an easy to access buy list.

3. Data items and Entities

Casual User - Casual visitor who browses the website shall be able to view and search for recipes.
Registered User - Users who are registered can view and browse like a casual user but also has access to add recipes.
Admin - User who has elevated permissions, able to edit and delete recipes.
Privacy Policy - Shall denote policies about user's data and why, what, and where the data is stored.
Profile - User customization, preferences, and other data shall be saved here.
Recipes - Recipes will be stored in a table with the name of the food as well as the ingredients to make the food.
Favorites from us - This is where popular/recommended recipes will be displayed.
Recipe Directions - Instructions for the recipes step by step.
Ingredients - A list of ingredients and other names that can be added.

Pantry- Ingredients the user has to use to cook any recipe.
Ingredient Type - Groups ingredients into certain types like vegetables, meats, dairy, dairy alternatives, etc.
Likes - Counts of likes for each recipe respectively.
Favorite - Users can mark recipes and save them as favorites.
Comments - Each recipe can be commented on.
Cuisine- The type of food, i.e Chinese, Mongolian, American, Pies, Cookies, etc. A recipe can be multiple food types.
Calorie Count- How much calorie a recipe might have.

4. Functional Requirement

Casual Users:

1. A casual user can find recipes without registering.
2. A casual user can check out ingredients.
3. A casual user can check price of ingredients.
4. A casual user can enter ingredients to find recipes.
5. A casual user can check budget recipes.

Registered Users:

6. A registered user can get recipe from another registered user.
7. A registered user can upload recipes.
8. A registered user can maintain pantry.
9. A registered user can list ingredients to buy them.
10. A registered user will be able to manage their own profile.

Website and Admins:

11. An administrator can update delete recipes.
12. An administrator can update delete users.
13. The application will have pages for recipes, this page will feature steps, cuisine, meal type, general type, serving size, ingredients needed vs ingredients available, recipe rating (in terms of likes), calorie count and nutritional information. We will also have a comments section on each recipe page.
14. The application shall have pages for ingredients, type and availability in pantry, also the list of stores from where order can be made, calorie count and nutritional information.
15. The application shall have a recommendations Tab where we will store recommended recipes.
16. New users shall be able to register.
17. A privacy policy shall be displayed to new users upon registration, and it shall be available under user settings.
18. Registered users shall be given the option to upload their photo on their profile.

19. Registered users shall be given the option to set default preferences such as their price range, pantry and favorite recipes.
20. A free text search box shall be displayed to all users which shall accept search terms regarding recipes, a button would be placed on this search bar to open ingredient search.
21. The default display in the free text search box shall be “Enter recipes”, with a button to switch to ingredients Search parameters shall be price/store location of ingredients or cooking time/cuisine for recipes.
22. All (registered and casual) users shall be allowed to filter search results by type of prices ‘low to high’, ‘high to low ‘and/or user.
23. All users shall be able to view a list of stores for an ingredient to buy with store rating and distance from pin code.
24. There will be a sidebar to search from ingredients (need to review search feature if this needs to be added), you can type, and it will return you suggestions or you can find them ordered categorically in dairy, vegetables, fruits etc.
25. An ‘add to favorites’ option shall be displayed so that registered users can save recipes or ingredients to this list.
26. A help section shall be displayed to all users with details regarding how the application can be used.

5. Non-Functional Requirement

Security:

1. Login or creating account shall be a requirement for all user who wishes to save the information
2. The username will not be available if another user has already created the account with the same name
3. There will not be any restrictions and limits for a password.
4. The user will be stayed logged in unless he/she refresh the page.

Performance:

1. The load time should not take more than 1 min.
2. The search shall be executed in the background.
3. Query shall be executed in the background.

Availability:

1. The server shall be up for as long as admin wish it to be.
2. The server will be down for no more than 1 day if there is any maintenance.
3. The server shall not crash for any client-side reasons.

Recovery:

1. If there is any event where server crash, recover time shall not take longer than 1 day.
2. In the case of server failure, there should be through revision for the server log

Data and privacy:

1. Data will be backed up every week.
2. The administrator can view and look at any account data, including username and password.
3. There will not be any image uploads for the profile picture.
4. All passwords will be encrypted with PBKDF2.

Conformance with Coding Standards:

1. The coding style will be professional and easy to read.
2. Only working code will be pushed or submitted.
3. All the code will be tested before submitting.
4. Any error will be stored in the log and be reviewed.
5. Any error shall be handled in a way that does not affect the functionality of the site.
6. This site shall be tested and debugged thoroughly before final submission.

Compatibility:

1. The site shall be compatible with the Internet Explorer 10+.
2. The site shall be compatible with the Firefox 60.0+.
3. The site shall be compatible with the Chrome 65+.
4. The site shall be compatible with devices running Android 7.0+ with Firefox or Chrome as specified above.
5. The site shall be compatible with devices running IOS 11.4.1+ with Firefox or Chrome as specified above.

Product's Standard

1. The application will look casual and user-friendly.
2. The site will be simple to navigate and easy to be used by anyone.
3. The site will provide related or recommended products base on the current product.
4. The site will look the same across all different browser.

6. Competitive Analysis

Recipe Based	Supercook	Allrecipes	Epicurious	Cookable

Sites:				
Recommendations Tab (Recipes)	-	+	-	+
Recipe Instructions	+	+	+	+
Ingredients available to cook	+	-	-	+
Food Type	+	+	+	+
Feature to add personal recipes	-	-	-	+
Liked/ Saved / Favorites	-	+	-	+

Supercook offers an extensive variety of recipes on their site. With many features that allow easy access to these recipes by either searching for foods with a certain set of ingredients or even finding recipes to prepare with ingredients the user may suggest. However, Supercook lacks in features such as saving/favoriting recipes, adding personal recipes to their application, and navigating through what is their recommended recipe portion on their site. Allrecipes excels in what Supercook lacks by offering a variety of different recipes as well as links to recommended recipes. They also offer an option to save and personalize your homepage with recipes you have favorited. However, Allrecipes much like its competitor lacks the ability to add personal recipes to the application which is what we aim to solve. Finally, Epicurious was a site that had more articles and blogs than actual recipes. Our application will try to include all the features listed above along with the feature of allowing registered users to add personal recipes to our database.

7. High-level system architecture and technologies used:

Server Host: AWS t2.micro 1vCPU 1 GB RAM
Operating System: Ubuntu v16.04 Server
Web Server: NGINX 1.15.8
Server-Side language: Javascript
Additional Technologies:

Front-end library: React v16.8.3, Material-UI v3.9.2

Web Framework: Node.js v10.15.1, Express.js v4.16.4
Database: MySQL v2.16.0

1. Application shall be developed using a pre-approved set of Software development tools.
2. Application Front-end user interface will be developed using the React javascript framework.
3. Application will have an intuitive interface that consists of individually rendered and updated components in React.
4. Application's Front-end will also be developed using the React library called material.ui
5. Application's Back-end will be handled using javascript libraries node.js and express.js.
6. Application's data will be stored in a database using MySQL version 2.16.0.
7. Application shall communicate between front and back-end using GET and POST calls.
8. Application port routing will be handled using NGINX version 1.15.8
9. Application shall use a Ubuntu version 16.04 server and will be hosted and deployed on an AWS t2.micro EC2 instance containing 1vCPU and 1GB RAM.
10. Application shall be viewable in standard web browsers, and shall render correctly on all major browsers: Firefox, Safari, Chrome, Internet Explorer
11. User privacy shall be protected and all privacy policies will be communicated to users.

8. Team

Trevor Sampson: Team Lead, Back-end/Front-end Developer
Thanh Dip: Back-end Lead, Database Manager
Tony Rodriguez: Front-end Lead
Pyae Naing: Back-end Developer, Github Master
Shivam Rai: Front-end Developer
Surabhi Chavan: Front-end Developer

9. Checklist

- Team found a time slot to meet outside of class
  (ON TRACK)
- Github Master chosen
  (DONE)
- Team decided and agreed together on using the listed SW tools and deployment server
  (DONE)
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
  (ON TRACK)

- Team lead ensured that all team members read the final M1 and agree/ understand it before submission
  (DONE)
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
  (DONE)