CS 3354.003 Software Engineering

Final Project Deliverable 2

CookEase

Pallavi Mekala, Varunika Ramadugu, Jacian Wynn, Yueying Ke, Fnu Mibu Kurian Sam

Changes in our Project Plan

Changing the name of an app can be a daunting task, but in the case of our app, it was a necessary step to ensure the success of our brand. We have changed our app's name from RecipeMaster to Cookease, and we believe that this change will make a significant difference in our user engagement and brand recognition.

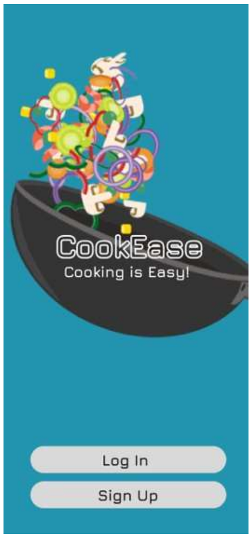
The reason for this change is simple. We wanted to make our app's name more memorable and easily recognizable to our target audience. The new name, Cookease, is a combination of the words "cook" and "ease," which we believe accurately reflects what our app does - it eases the stress of finding recipes.

When you say Cookease out loud, it sounds like "cookie," which is not only memorable but also creates an instant association with our app. This association will make it easier for users to remember our app's name and recommend it to others.

In today's digital world, branding is everything. We believe that the new name, Cookease, will help us establish a strong and recognizable brand identity, which will ultimately lead to increased user engagement and loyalty.

In conclusion, we are excited about the change from RecipeMaster to Cookease. We believe that the new name accurately reflects what our app does, is easy to remember, and will help us establish a strong brand identity. We are confident that this change will lead to increased user engagement and loyalty, and we look forward to seeing the positive impact it will have on our app's success.

New Demo Login Page using Figma



We used Figma to create the login page of our CookEase App.

Figma Link: https://www.figma.com/proto/feFL6WyHjF0tkCBB6X0dzo/CookEase-(3354)?node-id=1-2&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A2

Question 1 – Delegation of Tasks

Pallavi Mekala

- Worked on Q3.1 - Project Scheduling
- Helped with Q4 - Test plan for software
- Worked on references
- Worked on Question 5 - Comparison to other similar designs
- Worked on Question 6 - Conclusion
- Presentation Slides:
  - Title slide
  - Objective of project, introduction
  - Comparison with similar designs
  - Conclusion and future work

Varunika Ramadugu

- Worked on Q3.3 - Cost of Hardware Products
- Worked on references
- Helped with Question 5 - Comparison to other similar designs
- Presentation Slides:
  - Estimated Cost of Hardware Products
  - Cost, Effort, Pricing Estimation
  - Sequence Diagram
  - Architecture Pattern - MVC

Jacian Wynn

- Worked on Q3.4 - Cost of Software Products
- Worked on references
- Helped with Question 5 - Comparison to other similar designs
- Presentation Slides:
  - Non-functional requirements
  - Functional requirements
  - Estimated cost of software products

Yueying Ke

- Worked on Q3.5 - Cost of Personnel
- Worked on Q4 - Test plan for software
- Worked on references

- Helped with Question 5 - Comparison to other similar designs
- Worked on
- Presentation Slides:
    - Class diagram
    - Cost Estimation of Personnel
    - Testing Plan

Fnu Mibu Kurian Sam

- Worked on Q3.2 - Cost, Effort, and Pricing Estimation
- Worked on references
- Helped with Question 5 - Comparison to other similar designs
- Presentation Slides:
    - Use Case Diagram
    - User Interface Design
    - Project Timeline

Question 2 - Project Deliverable 1 Content

Question 1 – Final Project Description

Final Project Proposal

Title of Project: Recipe Master

Group Members: Pallavi Mekala, Varunika Ramadugu, Jacian Wynn, Yueying Ke (Dina Henry), Fnu Mibu Kurian Sam

Project Idea Description:

- Personalize recipes to ingredients in your pantry, or by meal preference.
- An app or website will be created to access the recipes.
- Ingredients will be categorized (by fruits, vegetables, meats, dairy, etc.)
- Users can input the ingredients they currently have, and the application will generate a customized recipe based on the ingredients available.
- Recipes will include nutritional information, serving size, cooking time.
- Users can post their own recipes.
- Users can select recipes through filters - by different cuisines, by meal of the day, by sweet/savory, by dietary preference (e.g. vegan, vegetarian, pescatarian, non-vegetarian.)
- Alternative ingredients options will be available for any substitution
- Each recipe will have a comment function and will include both video instructions and written instructions.
- Allows to connect with friends.

Motivation Description:

- Resolve the daily problem about what to cook.
- Reduce food waste.
- Less stress over cooking.
- Stay healthy.
- Save time.
- Make friends and socialize.

Tasks:

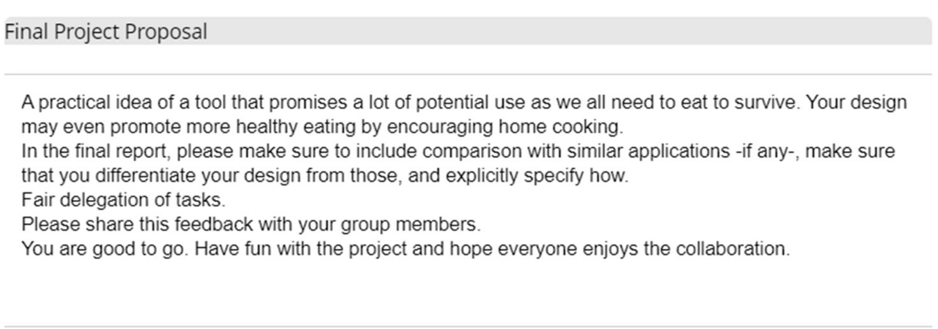
1. Design a blueprint using Figma.
2. Research recipe websites, note down ingredients and categorize them.
3. List of cuisines
4. IEEE Citations
5. Work on diagrams: class diagrams, case diagrams, sequence diagrams, architectural design and select type software process model, list functional and non-functional requirements, etc.
6. Evaluate cost
7. Come up with a project schedule
8. Create pseudocode or plan before coding
9. Final presentation: PowerPoint

- **Pallavi Mekala** - Research recipes: Mediterranean, Middle Eastern, French, participate in designing blueprint Figma
- **Varunika Ramadugu** - Research recipes: Indian, Spanish, Thai, participate in designing blueprint Figma
- **Jacian Wynn** - Research recipes: Chinese and Carribean, participate in designing blueprint Figma
- **Yueying Ke (Dina Henry)** - Research recipes: American, Japanese, Brazilian, participate in designing blueprint Figma
- **Fnu Mibu Kurian Sam** - Research recipes: Italian and Greek, participate in designing blueprint Figma

Interest in scholar paper?

Yes.

Feedback:



Final Project Proposal

A practical idea of a tool that promises a lot of potential use as we all need to eat to survive. Your design may even promote more healthy eating by encouraging home cooking.
In the final report, please make sure to include comparison with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how.
Fair delegation of tasks.
Please share this feedback with your group members.
You are good to go. Have fun with the project and hope everyone enjoys the collaboration.

What we plan to do differently:

- The users can share their recipes to other users and the users can view the recipes unlimited number of times but when they share recipes on other social media platforms to non-users, the non-users can only view the recipes for up to 5 times since the recipe will be linked to our application. By making it a limited number of views for non-users, this will motivate them to also create an account and join the app and view more recipes and also upload their own recipe. This will help our app to promote the application and it will draw more attention to more people to join the app.
- There will be a dark mode/light mode switch feature to allow the users to personalize the app to their preference.
- There will also be a translator option, to accommodate non-English speakers and to target a wider audience.
- There will be a read-aloud button which will read out the recipe. Users will be able to save recipes to their profile.
- Users will be able to follow other users, to get updates when they post new recipes.

- If a user is not logged in, they will get to view 5 free recipes before it requires them to sign up, to encourage users to make an account.
- The user will be able to send a tip to the recipe owner, this will be set up via buymeacoffee.com, to encourage users to post recipes.

The ‘Buy Me a Coffee’ section will look like the following as shown in Figure 1 [1]:

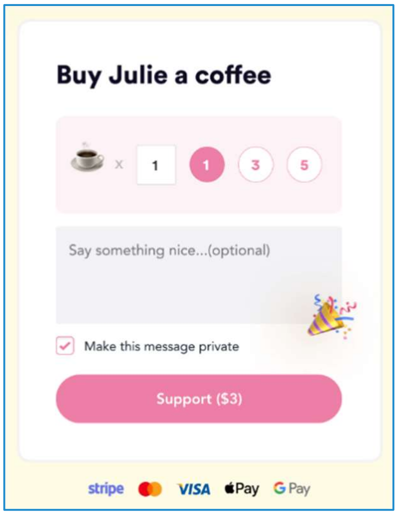


Figure 1: ‘Buy Me a Coffee’ Image [1]

- Comparison to Supercook [2]:

  Supercook Link: https://www.supercook.com/#/desktop [2]

  In comparison to the Supercook application [2], our app has additional social functions such as commenting on other users’ recipes and collecting recipes. In our application Recipe Master, the users will get a chance to view other recipes posted by other users and make comments and suggestions. This way, the users can connect with other users.

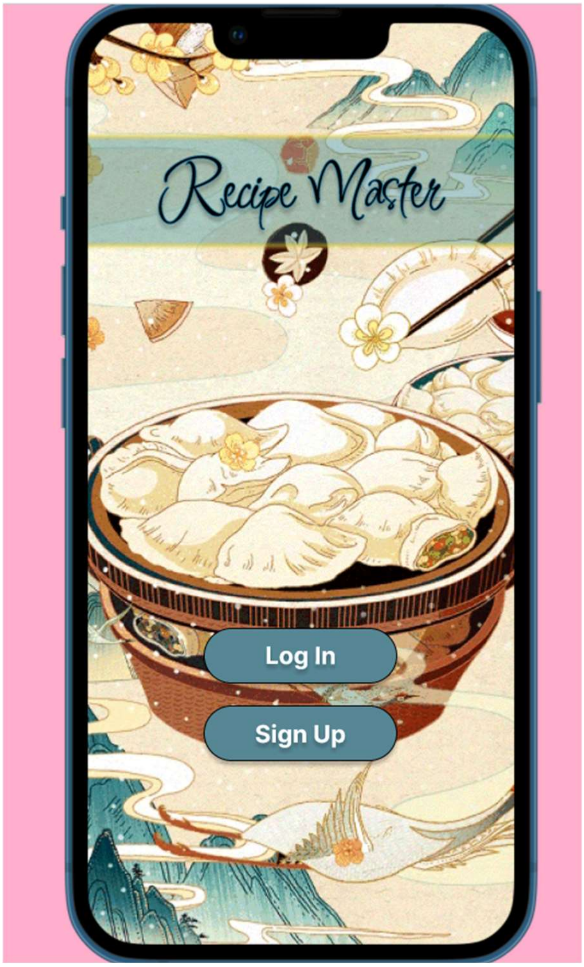
What we did:

- We used Trello to plan out the tasks and responsibilities and manage our project to complete the project components on time as shown in the screenshot below. This way we were able to keep track of all the tasks, responsibilities, and our progress and we were able to see what needed to be completed before a certain deadline. This helped to avoid confusion and helped the whole team to stay on track until completion of the Project Deliverable 1.

Trello Link: https://trello.com/b/Vofr3KqF/se-project-recipe-master

- We also used Figma to design the homepage of our Recipe Master App as an outline.

Figma design link: https://www.figma.com/proto/pdJ8s6bgneO7qdgN7L8PPY/Login-Page?node-id=13-3



- We also used Lucid Chart to create the diagrams [3].

  LucidChart Link: https://www.lucidchart.com/pages/ [3]

Question 2 - Setting up a GitHub repository

GitHub Link: https://github.com/pallavim222/RecipeMaster

Question 3 - Delegation of tasks

Pallavi Mekala

- Worked on Q6 Case Diagram
- Q2 - created GitHub Repository account
- Q2 1.3 - added all members and TA
- Q2 1.4 - created README file
- Helped and worked on Q2 - 1.5 Project Scope
- Helped and worked on Q1 Project description - project proposal, what we plan to do differently, comparison, what we did
- Participation in group discussions in sharing project ideas
- Helped in organizing project tasks in Trello
- Helped in making blueprint of login page using Figma
- Helped and worked on Q4 Software Process Model
- Helped and worked on Q9 Architectural design

Varunika Ramadugu

- Worked on Q7 Sequence Diagram
- Q2 - created GitHub Repository account
- Helped and worked on Q2 - 1.5 Project Scope
- Helped and worked on Q1 Project description - project proposal, what we plan to do differently, comparison, what we did
- Participation in group discussions in sharing project ideas
- Helped in organizing project tasks in Trello
- Helped in making blueprint of login page using Figma
- Helped and worked on Q4 Software Process Model
- Helped and worked on Q9 Architectural design

Jacian Wynn

- Worked on Q5A Functional Requirements
- Q2 - created GitHub Repository account
- Helped and worked on Q2 - 1.5 Project Scope
- Helped and worked on Q1 Project description - project proposal, what we plan to do differently, comparison, what we did
- Participated in group discussions in sharing project ideas
- Helped in organizing project tasks in Trello
- Helped in making blueprint of login page using Figma
- Helped and worked on Q4 Software Process Model
- Helped and worked on Q9 Architectural design

Yueying Ke

- Worked on Q8 Class Diagram
- Q2 - created GitHub Repository account
- Q2 1.4 - helped to add team names in README file
- Helped and worked on Q2 - 1.5 Project Scope
- Helped and worked on Q1 Project description - project proposal, what we plan to do differently, comparison, what we did
- Participation in group discussions in sharing project ideas
- Helped in organizing project tasks in Trello
- Helped in making blueprint of login page using Figma
- Helped and worked on Q4 Software Process Model
- Helped and worked on Q9 Architectural design

Fnu Mibu Kurian Sam

- Worked on Q5B Non-Functional Requirements
- Q2 - created GitHub Repository account
- Helped and worked on Q2 - 1.5 Project Scope
- Helped and worked on Q1 Project description - project proposal, what we plan to do differently, comparison, what we did
- Participation in group discussions in sharing project ideas
- Helped in organizing project tasks in Trello
- Helped in making blueprint of login page using Figma
- Helped and worked on Q4 Software Process Model
- Helped and worked on Q9 Architectural design

Question 4 - Which software process model is employed in the project and why

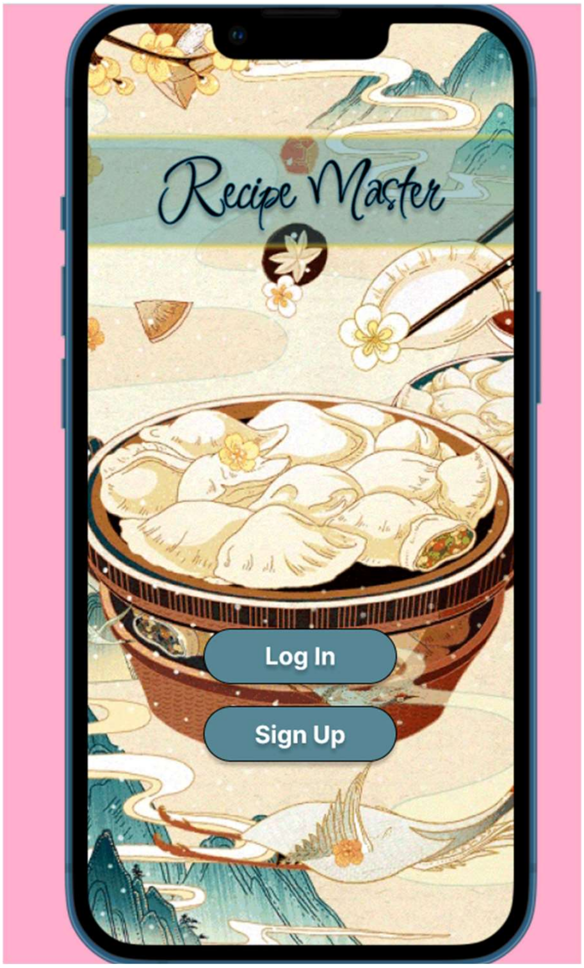
The software process model that we chose to implement in our project is the prototyping model; it allows us to present a simple design quickly such as a layout, so the customers and stakeholders can have a better expectation of how the application will look like early on and provide quick feedback. This model involves creating multiple prototypes of the software, refining them through user feedback, and iterating until a final version is produced.

Here are the steps in the prototyping software process model employed in the development of this recipe app:

- Requirements gathering: The development team should gather the requirements for the recipe app from potential users, stakeholders, and other sources (e.g. information on the features that users want, the types of recipes they would like to see, and the dietary restrictions they may have).
- Initial prototype: Based on the requirements gathered, the development team should create an initial prototype of the recipe app. This prototype should include the core features that users have requested, such as the ability to search for recipes by ingredients or dietary preferences.
- User testing: The initial prototype should be tested by a group of users who fit the target demographic for the recipe app. Users should be asked to provide feedback on the app's usability, functionality, and overall user experience.
- Refinement: Based on the feedback gathered from users, the development team should refine the initial prototype to address any issues that were identified. This may involve adding new features, simplifying existing features, or making other changes to improve the user experience.
- Iteration: The refined prototype should be tested again with a new group of users, and the feedback gathered should be used to further refine the app. This process should be repeated until a final version of the recipe app is produced that meets the needs and expectations of its users.

For instance, we utilized Figma to create a simple design of the homepage of our Recipe Master application as shown below.

Figma design link: https://www.figma.com/proto/pdJ8s6bgneO7qdgN7L8PPY/Login-Page?node-id=13-3



Question 5 - Software Requirements

A) Functional requirements

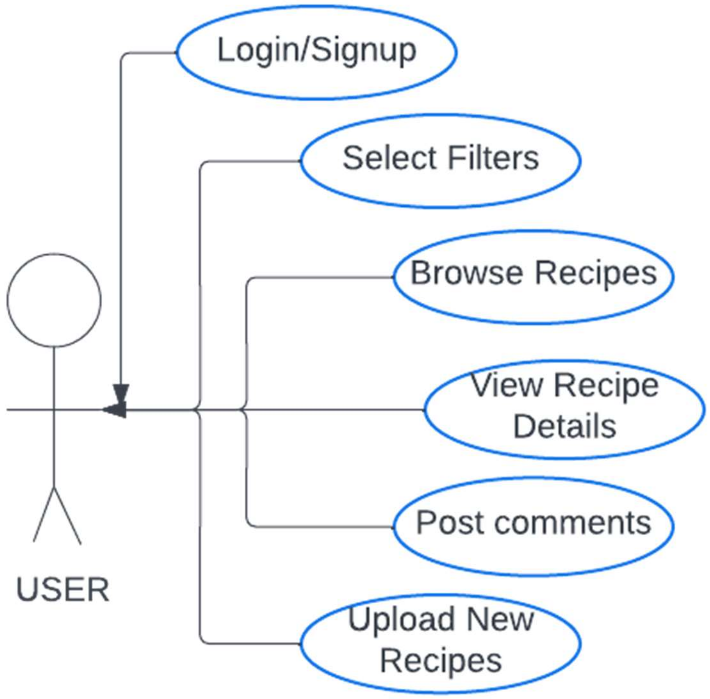
- Personalization of recipes based on pantry ingredients or meal preference.
- Creation of an app or website to access the recipes.
- Categorization of ingredients by fruits, vegetables, meats, dairy, etc.
- Input of available ingredients by the user and generation of customized recipe based on the ingredients available.
- Provision of nutritional information, serving size, and cooking time for each recipe.
- Posting of user-generated recipes.
- Selection of recipes through filters - by different cuisines, by meal of the day, by sweet/savory, by dietary preference (e.g., vegan, vegetarian, pescatarian, non-vegetarian).
- Availability of alternative ingredients for substitution.
- Inclusion of a comment function for each recipe.
- Provision of both video and written instructions for each recipe.
- Ability to connect with friends.

B) Non-functional requirements

- Usability requirements: User interface must be intuitive and easy to navigate. The app must be accessible and usable on a range of devices and platforms and handle multiple users and sessions at the same time.
- Performance requirements: The app must respond quickly with minimal loading time to user requests, handle multiple users simultaneously and heavy traffic during peak usage periods.
- Space requirements: Store many recipes and user data and handle large file sizes for images and videos.
- Dependability requirements: Reliable and available at all times and has backup and recovery capabilities in case of data loss or system failure.
- Security requirements: User data must be securely stored and protected from unauthorized access. Implement secure authentication and authorization mechanisms and be able to detect and prevent malicious activity (e.g., hacking).
- Environmental requirements: Designed to minimize its environmental impact, such as by using energy-efficient technologies
- Operational requirements: Easy to install and maintain. Scalable and able to accommodate growth in users and data.
- Development requirements: Developed using reliable and secure software development practices. Able to integrate with other systems and services as needed.
- Regulatory requirements: Must comply with relevant laws and regulations, such as data protection and privacy laws

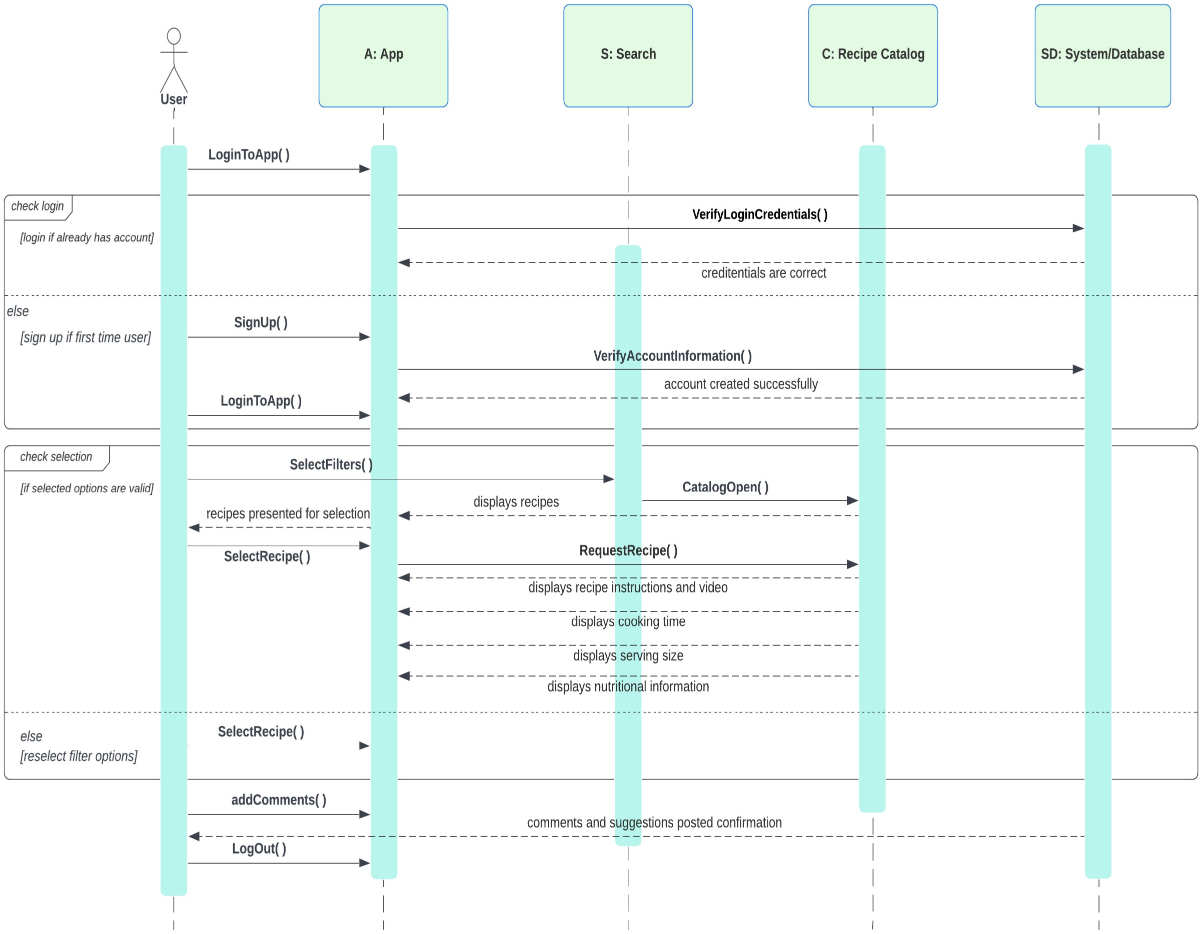
- Ethical requirements: Respects user privacy and confidentiality. Must not promote harmful or unethical cooking practices.
- Accounting requirements: Able to track user purchases and transactions.
- Safety/security requirements: Must not contain any harmful or dangerous recipes or instructions, nor expose users to any health or safety risks.

Question 6 - Use case diagram

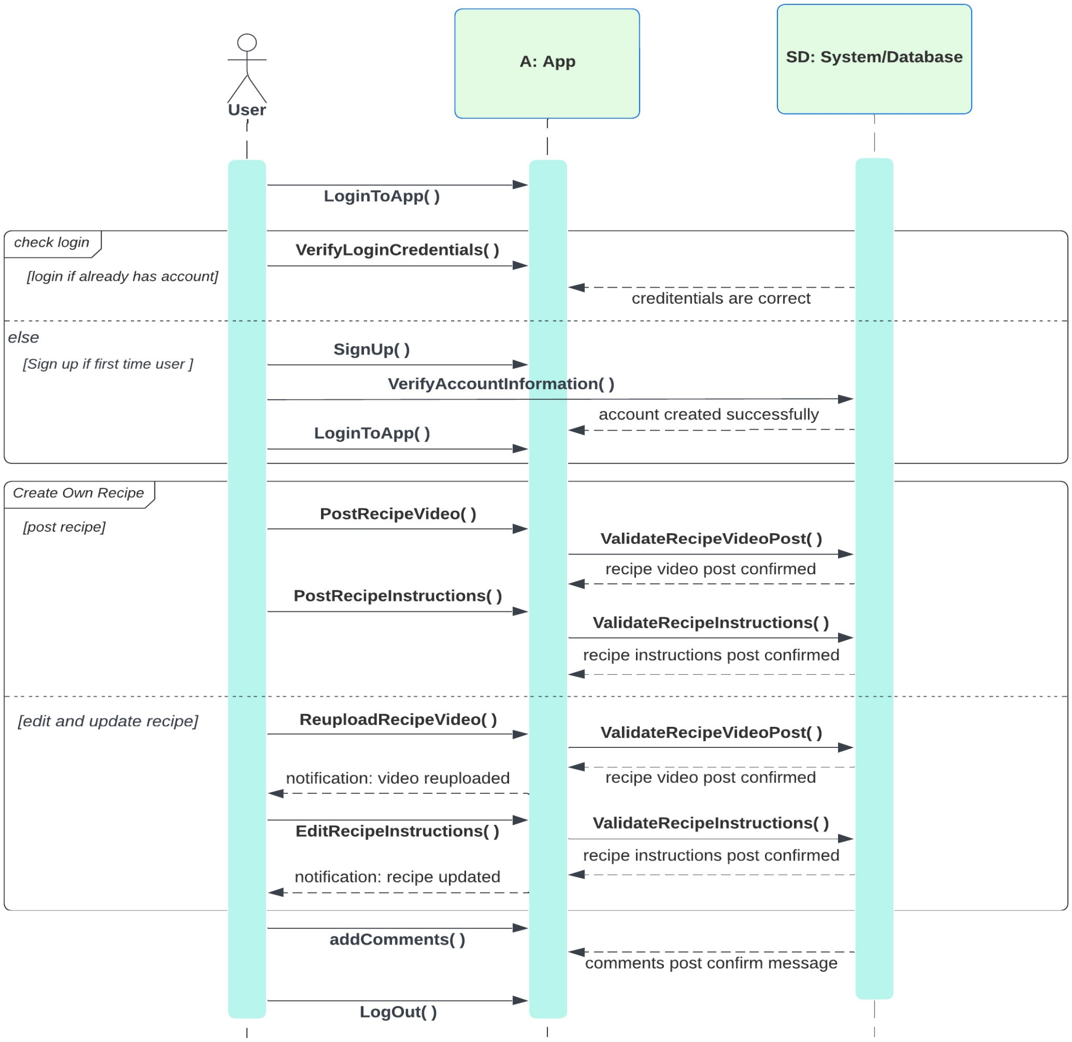


Question 7 - Sequence diagrams

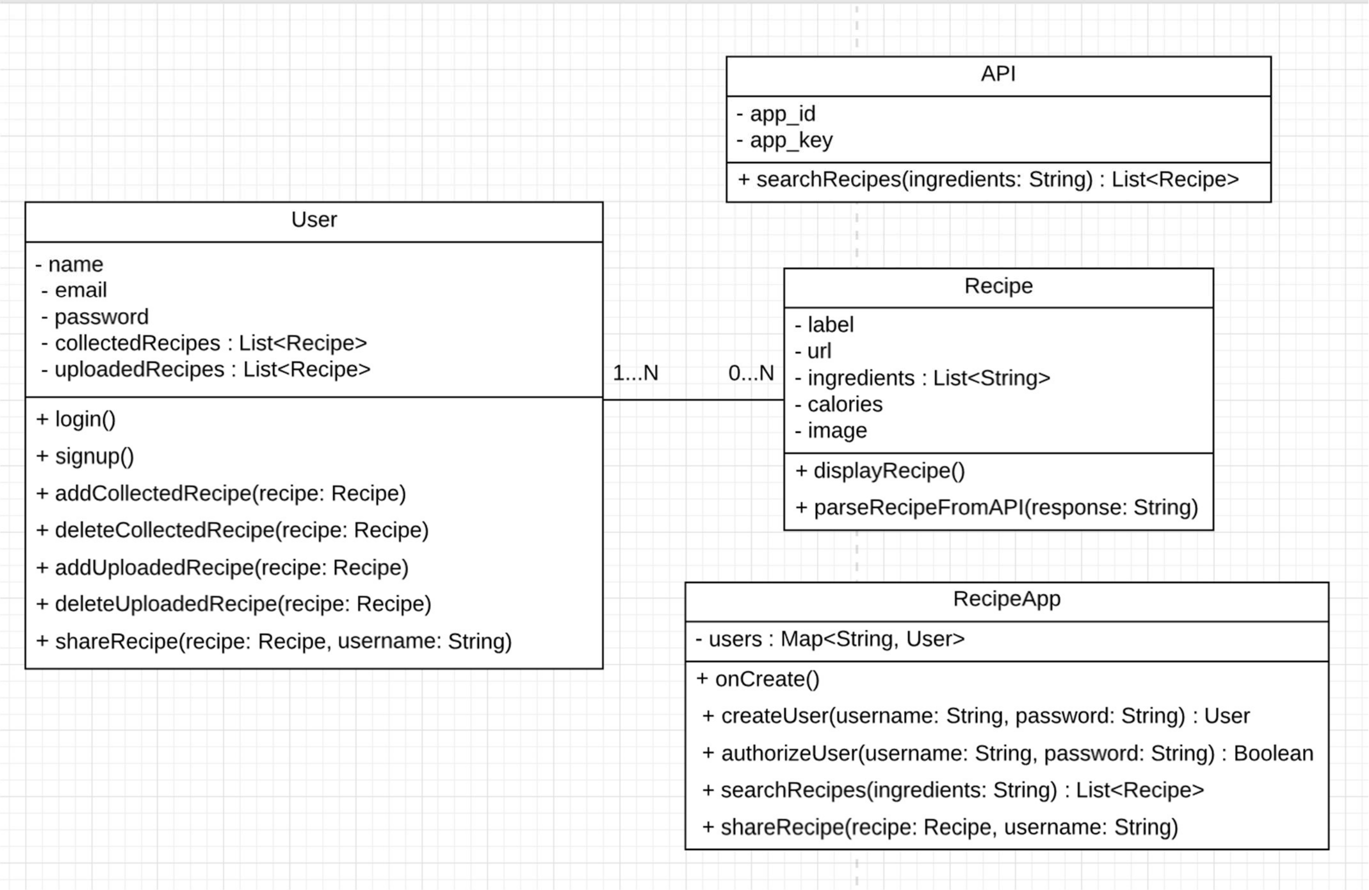
Sequence Diagram for generating recipes


User
A: App
S: Search
C: Recipe Catalog
SD: System/Database
LoginToApp( )
check login
[login if already has account]
VerifyLoginCredentials( )
creditentials are correct
else
[sign up if first time user]
SignUp( )
VerifyAccountInformation( )
account created successfully
LoginToApp( )
check selection
[if selected options are valid]
SelectFilters( )
CatalogOpen( )
displays recipes
recipes presented for selection
SelectRecipe( )
RequestRecipe( )
displays recipe instructions and video
displays cooking time
displays serving size
displays nutritional information
else
[reselect filter options]
SelectRecipe( )
addComments( )
comments and suggestions posted confirmation
LogOut( )

Sequence Diagram for posting own recipe


A: App
SD: System/Database
User
LoginToApp( )
check login
[login if already has account]
VerifyLoginCredentials( )
creditentials are correct
else
[Sign up if first time user]
SignUp( )
VerifyAccountInformation( )
account created successfully
LoginToApp( )
Create Own Recipe
[post recipe]
PostRecipeVideo( )
ValidateRecipeVideoPost( )
recipe video post confirmed
PostRecipeInstructions( )
ValidateRecipeInstructions( )
recipe instructions post confirmed
[edit and update recipe]
ReuploadRecipeVideo( )
ValidateRecipeVideoPost( )
notification: video reuploaded
recipe video post confirmed
EditRecipeInstructions( )
ValidateRecipeInstructions( )
recipe instructions post confirmed
notification: recipe updated
addComments( )
comments post confirm message
LogOut( )

Question 8 - Class diagram


API
- app_id
- app_key
+ searchRecipes(ingredients: String) : List<Recipe>
User
- name
- email
- password
- collectedRecipes : List<Recipe>
- uploadedRecipes : List<Recipe>
+ login()
+ signup()
+ addCollectedRecipe(recipe: Recipe)
+ deleteCollectedRecipe(recipe: Recipe)
+ addUploadedRecipe(recipe: Recipe)
+ deleteUploadedRecipe(recipe: Recipe)
+ shareRecipe(recipe: Recipe, username: String)
1...N
0...N
Recipe
- label
- url
- ingredients : List<String>
- calories
- image
+ displayRecipe()
+ parseRecipeFromAPI(response: String)
RecipeApp
- users : Map<String, User>
+ onCreate()
+ createUser(username: String, password: String) : User
+ authorizeUser(username: String, password: String) : Boolean
+ searchRecipes(ingredients: String) : List<Recipe>
+ shareRecipe(recipe: Recipe, username: String)

Question 9 - Architectural design: Model-View-Controller (MVC) pattern

One architectural design pattern that we chose for our project is the Model-View-Controller (MVC) pattern. We believe it is the most suitable pattern for the following reasons:

Model:

The Model represents the data and business logic of the Recipe Master application (e.g., nutritional information, serving size, cooking time, and ingredients). It will contain a database of recipes and their corresponding ingredients, nutritional information, serving size, and cooking time, as well as the logic to retrieve and filter them based on user input. The Model will also contain methods to add, delete, and update recipes in the database.

View:

The View represents the user interface of the Recipe Master application. It will provide a way for the user to input their desired ingredients and filter options, as well as displaying the resulting recipes to the user and the video tutorials. The view includes various filters available for selecting recipes, (e.g., cuisines, meal of the day, sweet/savory, and dietary preference) as well as provide the recipe information. The View will also be responsible for capturing user input and passing it to the Controller.

Controller:

The Controller acts as an intermediary between the Model and the View. It will receive user input from the View and pass it to the Model to retrieve and filter recipes. The Controller will also receive updates from the Model and pass them back to the View to be displayed to the user. The controller handles user input for selecting filters, inputting available ingredients, and posting recipes, including substitution options. It combines both the model and view part in which the user chooses the ingredients and filters and will generate recipes based on the input.

Therefore, the MVC pattern would be the most suitable architecture pattern for our project; it's well designed for user interfaces and promotes separation of concerns, making it a good choice for an app that focuses on presenting information to the user, recipe customization, user-generated content, and social networking.

Question 3.1 - Project Scheduling

Start Date: May 1, 2023

End Date: July 31, 2023

Justification:

- This project is a web application, which typically takes around 3 months to develop, test, and deploy.
- May 1, 2023, is a Monday which allows for some buffer time before the start of the project in case of any unforeseen delays or issues.
- July 31, 2023, is also a Monday, which provides some buffer time for testing and debugging before the project is deployed.

Weekends will not be included in this schedule as it is assumed that the development team will not be working on the weekends.

This will give 66 full working days for the project. The number of working hours per day for the project will be assumed to be 8 hours, the standard workday for most software development teams. This gives 528 working hours for the project.

Note that this is just an estimation of the project schedule, and actual schedules may vary based on factors such as the project's complexity and any unforeseen issues that may arise during the development process.

Question 3.2 - Cost, Effort, Pricing Estimation

Function Point

We chose Function Point algorithmic estimation technique because of the input of ingredients and output of recipes.

User inputs = 10

For example, if the user enters ten ingredients based on fruits, vegetables, dairy, spices, etc.

User outputs = 1

For example, for ten ingredients, it can generate one list of recipes.

User Queries = 4

1. Search
2. filtering
3. generating recipes
4. data exports: sharing recipe instructions and recipe videos

Data files = 5

1. User data information
2. recipe information
3. comments and reviews
4. search history
5. user preferences

External interfaces = 6

1. social media usage for sharing
2. ‘Buy Me a Coffee’ [1] external website
3. Email service for notifications
4. Cloud storage
5. Recipe edamam API [5]
6. Embed YouTube link of recipe

Average Complexity Level

We chose average complexity level because of user input of ingredients and output of recipes being generated.

	Functional Category	Count	Complexity			Count x Complexity
			Simple	Average	Complex	
1	**Number of user input**	10	3	4	6	40
2	**Number of user output**	1	4	5	7	5
3	**Number of user queries**	4	3	4	6	16
4	**Number of data files and relational tables**	5	7	10	15	50
5	**Number of external interfaces**	6	5	7	10	42
					GFP	**153**

PC calculation

Processing Complexity (PC) = 2 + 2 + 1 + 5 + 4 + 5 + 1 + 4 + 3 + 3 + 4 + 0 + 0 + 5 = 39

14 questions from Ch 23 Project Planning Slides [1]:

Questions [4]:

Q1 – "Does the system require reliable backup and recovery? [4]"

PC = 2 – moderate, because the backup and recovery are mostly needed just for managing the user's data.

Q2 – "Are data communications required? [4]"

PC = 2 – moderate, data communications required but they will be only needed for a section of the software such as the user getting notifications whenever they get comments or when there are successfully uploaded recipes, etc.

Q3 – "Are there distributed processing functions? [4]"

PC = 1 – incidental, there are not many distributed processing functions in our software.

Q4 – "Is performance critical? [4]"

PC = 5 – essential, the performances are critical to generate recipes and videos efficiently

Q5 – "Will the system run in an existing, heavily utilized operational environment? [4]"

PC = 4 – significant, since our app runs in a mobile app.

Q6 – “Does the system require online data entry? [4]”

PC = 5 – essential, because the system needs online data entries such as the user inputs ingredients to generate recipes.

Q7 – “Does the online data entry require the input transaction to be built over multiple screens or operations? [4]”

PC = 1 – incidental, because there is only one external website that is connected to our app which is the “Buy Me a Coffee [1]” website.

Q8 – “Are the master files updated online? [4]”

PC = 4 – significant, because the master files that are updates is the source code of our app which is accessible to be updated and reused in GitHub therefore master files are updated online.

Q9 – “Are the inputs, outputs, files, or inquiries complex? [4]”

PC = 3 – average, because the complexity level is average therefore the inputs, outputs, files, and inquiries are not that complex.

Q10 – “Is the internal processing complex? [4]”

PC = 3 – average, because the internal processing is not that complex.

Q11 – “Is the code designed to be reusable? [4]”

PC = 4 – significant, because the source code is reusable for making advancements and updates to our app.

Q12 – “Are conversion and installation included in the design? [4]”

PC = 0 – no influence, because there is no conversion and installation in our app’s design.

Q13 – “Is the system designed for multiple installations in different organizations? [4]”

PC = 0 – no influence, because there are no multiple installations.

Q14 – “Is the application designed to facilitate change and ease of use by the user? [4]”

PC = 5 – essential, because the goal of our app is to make it easier for users to make recipes fast and easy.

GFP = 153 FP

PCA = 0.65 + 0.01 (2 + 2 + 1 + 5 + 4 + 5 + 1 + 4 + 3 + 3 + 4 + 0 + 0 + 5) = **1.04**

FP = GFP x PCA = 153 x 1.04 = 159.12 FP

E = FP / productivity = 159.12 / 30 ≈ 5.304 ≈ 6 person-weeks

Productivity of development and maintenance teams = **30**

Project Duration = D = E / team size = 6 / 4 = 1.5 ≈ 2 weeks

For instance, based on Team size of 4 developers.

Question 3.3 - Cost Estimation of Hardware Products

For our CookEase app, the cost estimation of hardware products includes:

- Memory – Random Access Memory (RAM) [6], [7] = 16 GB = $108 [8]
- Network – T-Mobile [9] = $50 per month [9]
- PC for app development (5 HP laptops) [6], [7] = 5 laptops x $330 [10] = $1,650
- Processing Unit [6], [7]
    - Central Processing Unit (CPU) = Intel Core i5 [6], [7], [11] = $319.13 [11]
    - Storage [6], [7]
        - Hard disk drive (HDD) = 50TB (Amazon S3 Simple Storage) [12] = $0.023 per GB [12]
            - $0.023 per GB x 50,000 GB = $1,150 per month

Total Cost Estimation = $108 + $50 + $1,650 + $319.13 + $1,150 = **$3,277.13 per month**

Question 3.4 - Cost Estimation of Software Products

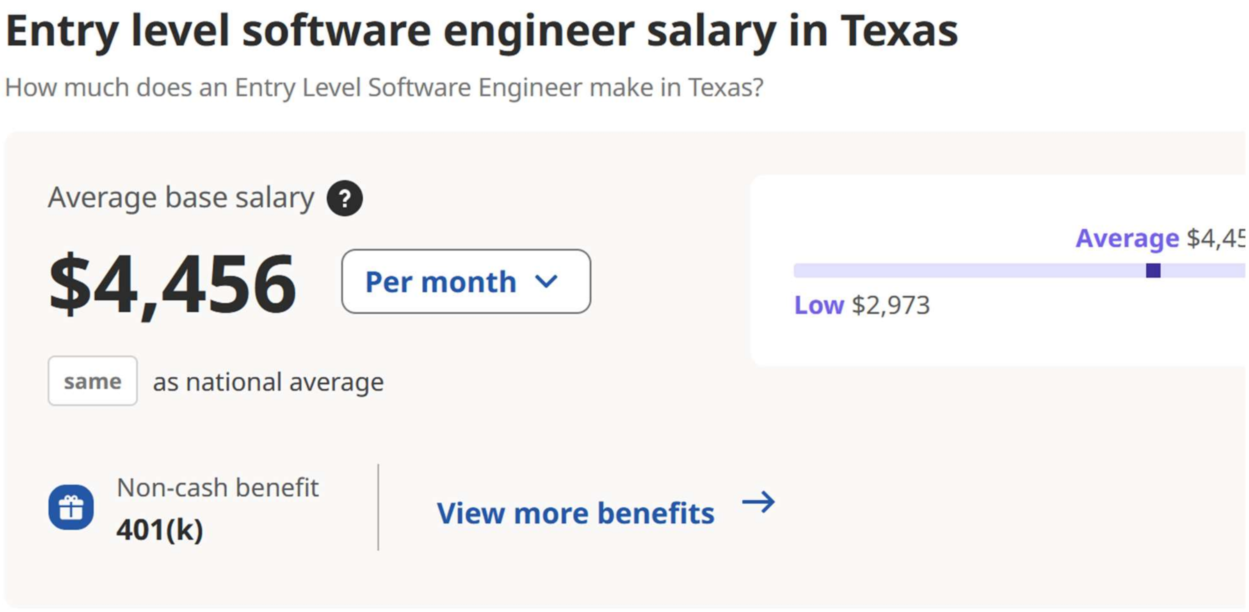
For our CookEase app, the cost estimation of software products includes:

- Front-end development tools = HTML, CSS, and JavaScript [13] = open-source and free
- Back-end development tools = Node.js, Ruby on Rails, and Django [14] = open-source and free
- Database management system = MySQL [15] = open-source and free
- API Integration tools = RESTful APIs and GraphQL [12], [16] = open-source and free
- Search engine software = Elasticsearch [17] = open-source and free
- Video hosting and streaming service = YouTube [18] = free
- Social media integration tools = Facebook and Twitter APIs [19], [20] = free to use, but with limits and restrictions
- Recipe management and publishing software = Cookpad and TastyIgniter [21], [22] = free
- Analytics and reporting tools = Google Analytics and Mixpanel [23], [24] = free
- Buy Me a Coffee [25] = free

Total Cost Estimation = **$0 per month**

Question 3.3 - Cost Estimation of Personnel

According to Indeed, the average monthly cost of an entry level software engineer is $4,456.



Entry level software engineer salary in Texas

How much does an Entry Level Software Engineer make in Texas?

Average base salary ?

$4,456 Per month

same as national average

Average $4,45

Low $2,973

Non-cash benefit
401(k)

View more benefits →

According to our previous effort estimates,

Effort (person/month) = 159.12 / 30

Effort (person/month) = 5.3 person-months

There are 5 members of in the team, so the total effort is:

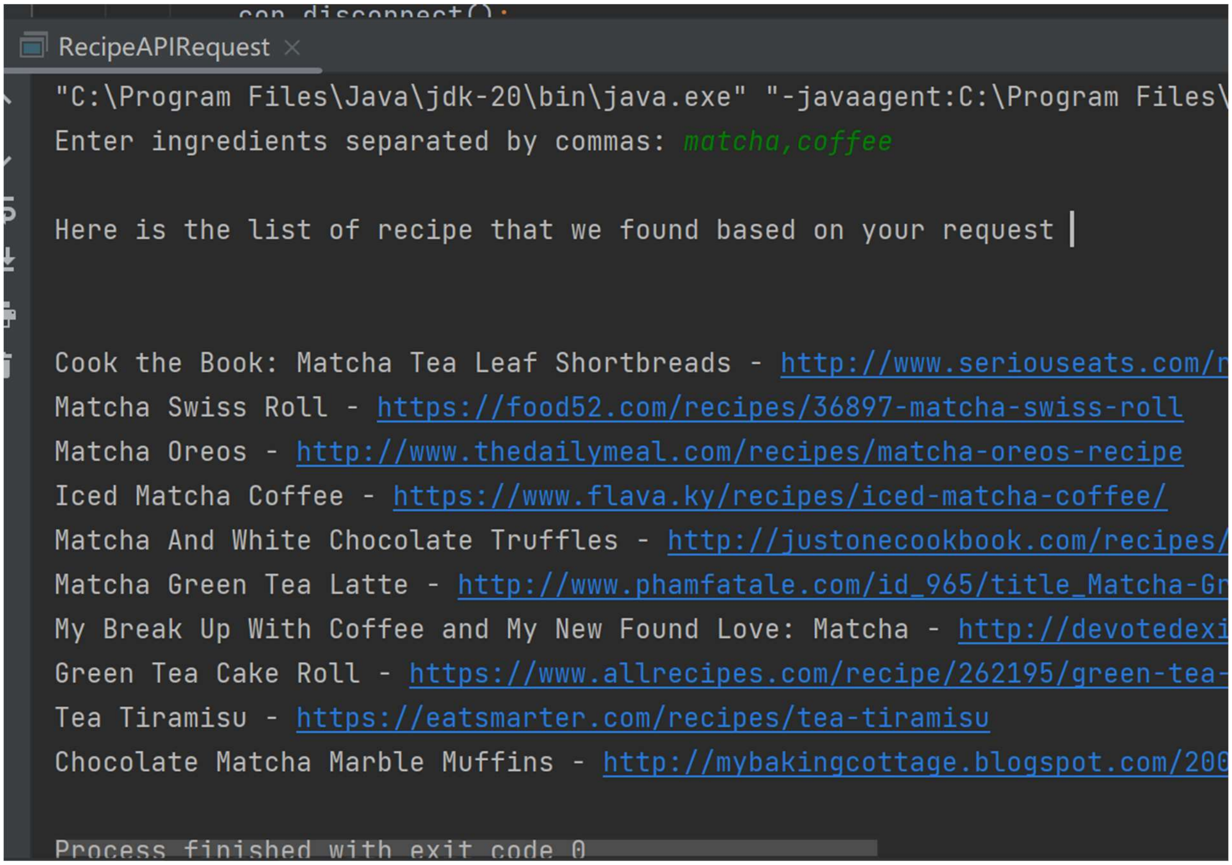
Total effort (in person/month) = 5.3 x 5 = 26.5 person/month

Total Cost = Total effort (in person-months) x Cost per person/month = 26.5 x $4,456 =$118084 [19]

Question 4 – Test plan for software

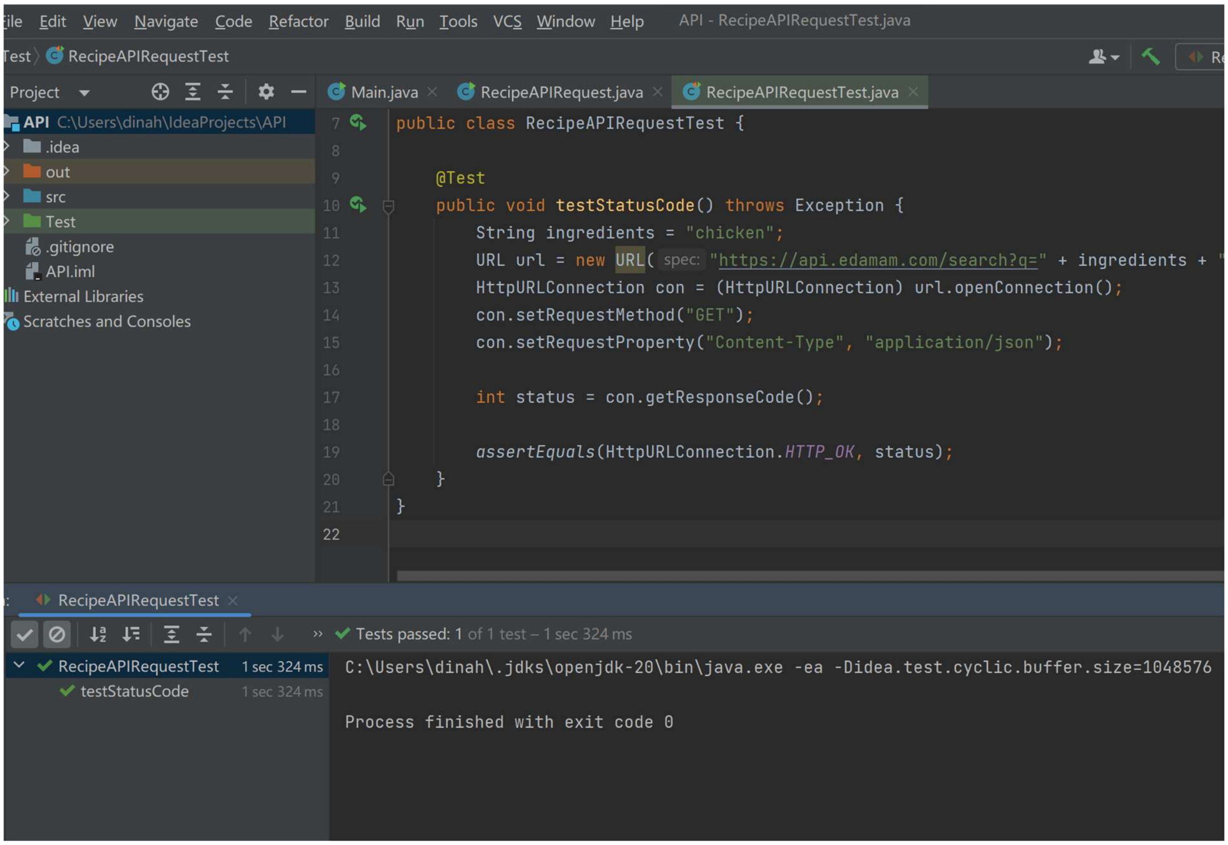
The class runs and uses the "GET" method to request calls from the Edamam API [5].

It then receives responses from the API and returns a list of recipes.



```
con.disconnect();
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\
Enter ingredients separated by commas: matcha,coffee

Here is the list of recipe that we found based on your request 



Cook the Book: Matcha Tea Leaf Shortbreads - http://www.seriouseats.com/r
Matcha Swiss Roll - https://food52.com/recipes/36897-matcha-swiss-roll
Matcha Oreos - http://www.thedailymeal.com/recipes/matcha-oreos-recipe
Iced Matcha Coffee - https://www.flava.ky/recipes/iced-matcha-coffee/
Matcha And White Chocolate Truffles - http://justonecookbook.com/recipes/
Matcha Green Tea Latte - http://www.phamfatale.com/id_965/title_Matcha-Gr
My Break Up With Coffee and My New Found Love: Matcha - http://devotedexi
Green Tea Cake Roll - https://www.allrecipes.com/recipe/262195/green-tea-
Tea Tiramisu - https://eatsmarter.com/recipes/tea-tiramisu
Chocolate Matcha Marble Muffins - http://mybakingcottage.blogspot.com/200

Process finished with exit code 0
```

The test shows that the “GET” request call from the Edamam API [5] was successful.


ile Edit View Navigate Code Refactor Build Run Tools VCS Window Help
API - RecipeAPIRequestTest.java
Test 〉 RecipeAPIRequestTest
Project
API C:\Users\dinah\IdeaProjects\API
.idea
out
src
Test
.gitignore
API.iml
External Libraries
Scratches and Consoles
Main.java
RecipeAPIRequest.java
RecipeAPIRequestTest.java
RecipeAPIRequestTest

```
public class RecipeAPIRequestTest {

    @Test
    public void testStatusCode() throws Exception {
        String ingredients = "chicken";
        URL url = new URL( spec: "https://api.edamam.com/search?q=" + ingredients + "
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("GET");
        con.setRequestProperty("Content-Type", "application/json");

        int status = con.getResponseCode();

        assertEquals(HttpURLConnection.HTTP_OK, status);
    }
}
```

✓ Tests passed: 1 of 1 test – 1 sec 324 ms

✓ RecipeAPIRequestTest 1 sec 324 ms
✓ testStatusCode 1 sec 324 ms

```
C:\Users\dinah\.jdks\openjdk-20\bin\java.exe -ea -Didea.test.cyclic.buffer.size=1048576

Process finished with exit code 0
```

Question 5 – Comparison to other similar designs

- Comparison to Supercook [2]:

  Supercook Link: https://www.supercook.com/#/desktop [2]

  In comparison to the Supercook application [2], our app has additional social functions such as commenting on other users' recipes and collecting recipes. In our application CookEase, the users will get a chance to view other recipes posted by other users and make comments and suggestions. This way, the users can connect with other users.

Question 6 – Conclusion

In conclusion, our team worked collaboratively and effectively on the development of the CookEase app, which generates recipes based on input of ingredients. We were able to follow our timeline and meet deadlines, which resulted in significant progress during the project.

However, one aspect that deviated from our original plan was the creation of a website for the project. Initially, we had planned to include a website as part of our project, but we underestimated the workload and encountered challenges that hindered us from completing it within the given timeframe. Despite our best efforts, we were not able to allocate enough resources and time to fully implement the website as we had originally intended.

We believe that this change was justifiable considering the limited resources and time constraints we faced. Instead of compromising the quality of our main project, the CookEase app, we made the decision to prioritize its development and focused on planning to ensure its functionality and usability.

Overall, we are satisfied with the outcome of our project and proud of the work we accomplished as a team. We have learned valuable lessons about project planning and resource allocation, and we acknowledge the importance of accurate workload estimation in future projects. Despite the deviation from our original plan, we were able to successfully complete the plan of the CookEase app, and we look forward to further improvements and future iterations based on user feedback.

References

[1] "Buy Me a Coffee," (Buy Me a Coffee), [Online]. Available: https://www.buymeacoffee.com/ (Accessed 22 Mar 2023).

[2] "Recipe search by ingredients you have at home," (SuperCook), [Online]. Available: https://www.supercook.com/#/desktop (Accessed 22 Mar 2023).

[3] "Lucidchart," (Lucidchart), [Online]. Available: https://www.lucidchart.com/pages/ (Accessed 22 Mar 2023).

[4] I. Sommerville, "Software Engineering," 10th ed., Boston, MA: Pearson, 2016.

[5] "Recipe Search API – documentation – edamam," (Edamam.com), [Online]. Available: https://developer.edamam.com/edamam-docs-recipe-api (Accessed: 13 April 2023).

[6] I. Ogunbiyi, "What Computer Specs Are Important for Mobile App Development?", "Waldo [Online]. (2022, September 13). Available: https://www.waldo.com/blog/what-computer-specs-are-important-for-mobile-app-development#:~:text=CPU%3A%20The%20lowest%20processor%20type (Accessed: 14 April 2023).

[7] "How much does it cost to make an app for business in 2023," (Addevice), [Online]. Available: https://www.addevice.io/blog/how-much-does-it-cost-to-build-a-mobile-app (Accessed: 14 April 2023).

[8] "April 2023 Server Memory Prices," (Memory.net), [Online]. Available: https://memory.net/memory-prices/ (Accessed: 20 April 2023).

[9] "Small Business Internet," (T-Mobile), [Online]. Available: https://www.t-mobile.com/business/solutions/business-internet-services/small-business-internet?&cmpid=TFB_PG_P_TFBBIZINT_43700072663373187_651782844427&gad=1&gclid=EAIaIQobChMI2PqZvcy2_gIVjhTUAR0svwJJEAAYASAAEgJDt_D_BwE&gclsrc=aw.ds (Accessed: 20 April 2023).

[10] "HP 15-ef2723od Laptop," (OfficeDepot), [Online]. Available: https://www.officedepot.com/a/products/1676428/HP-15-ef2723od-Laptop-156-Screen/ (Accessed: 20 April 2023).

[11] "Intel i5-13600K Desktop Processor," (Amazon), [Online]. Available: https://www.amazon.com/Intel-i5-13600K-Desktop-Processor-P-cores/dp/B0BCDR9M33/?th=1 (Accessed: 20 April 2023).

[12] "Amazon S3 Pricing," (AWS.amazon), [Online]. Available: https://aws.amazon.com/s3/pricing/ (Accessed: 20 April 2023).

[13] "HTML & CSS Classes Cost," (NobleDesktop), [Online]. Available: https://www.nobledesktop.com/learn/html-css/cost (Accessed: 20 April 2023).

[14] R. Schulte, "Rails, Django, and Node.js: A Primer on Web Frameworks," (Medium), [Online]. Available: https://medium.com/@schulte.robert/rails-django-and-node-js-a-primer-on-web-frameworks-9c734dc1bd2c (Accessed: 20 April 2023).

[15] "MySQL Pricing," (GetApp), [Online]. Available: https://www.getapp.com/it-management-software/a/mysql/pricing/ (Accessed: 20 April 2023).

[16] "Plans and Pricing," (Hasura), [Online]. Available: https://hasura.io/docs/latest/hasura-cloud/plans/ (Accessed: 20 April 2023).

[17] "Elastic Pricing FAQ," (Elastic). [Online]. Available: https://www.elastic.co/pricing/faq (Accessed: 20 April 2023).

[18] "5 Reasons YouTube Premium is Worth Every Penny," (the ascent), [Online]. Available: https://www.fool.com/the-ascent/personal-finance/articles/5-reasons-youtube-premium-is-worth-every-penny/ (Accessed: 20 April 2023).

[19] "Entry Level software engineer salary in Texas," (Indeed), [Online]. Available: https://www.indeed.com/career/entry-level-software-engineer/salaries/TX (Accessed: 20 April 2023).

[20] "Twitter's New API Access Charges Could Price Many Apps and Researchers Out of Their Projects," (SocialMediaToday), [Online]. Available: https://www.socialmediatoday.com/news/twitters-new-api-access-charges-could-price-many-apps-and-researchers-out/644775/ (Accessed: 20 April 2023).

[21] "Cookpad Terms of Service," (Cookpad), [Online]. Available: https://cookpad.com/us/terms (Accessed: 20 April 2023).

[22] "TastyIgniter Pricing, Features, Reviews and Alternatives," (GetApp), [Online]. Available: https://www.getapp.com/all-software/a/tastyigniter/ (Accessed: 20 April 2023).

[23] "Google Analytics," (Google grketing Platform), [Online]. Available: https://marketingplatform.google.com/about/analytics/ (Accessed: 20 April 2023).

[24] "Mixpanel Pricing," (Mixpanel), [Online]. Available: https://mixpanel.com/pricing/ (Accessed: 20 April 2023).

[25] "Frequently Asked Questions," (Help.BuyMeACoffee.com), [Online]. Available: https://help.buymeacoffee.com/en/articles/4539170-frequently-asked-questions (Accessed: 20 April 2023).