Clicking Cookies Design Document

Group:14

Members: Kevin Vu (vuk56.stu@uncc.edu), Forrest Wilkerson (fwilkers@uncc.edu)

Final Submission GitHub:
https://github.com/KevinVuUNCC/ITSC4155_MDSp24_Group14_Final_Submission

History of Creation Github:
https://github.com/KevinVuUNCC/ITSC4155_MDSp24_Group14

Project Overview

Clicking Cookies is a Recipe Tracking Mobile Application for all Culinary Individuals from Chefs, to Food Critics, to College Students looking to cook a quick meal for the night. Allowing culinary individuals to quickly store and share their favorite recipes with friends, family, and audiences. With the rise of social media and its boom of spreading information and food recipes, there has also been a lack of uniformity and accessibility when trying to properly find and store those recipes on platforms such as TikTok, YouTube, and Instagram. The problem Clicking Cookie looks to resolve is the time-consuming process of finding a recipe that meets your dietary needs instead of sifting through non-food-related Instagram posts and YouTube videos to find the right recipe for you. Clicking Cookies solves this problem by being the platform for a food-oriented experience and providing you tools to filter the recipes on the platform to find the right one for You.

Stakeholders (Culinary Individuals):
- Kitchen or House Chefs
- Food Critics
- Individuals with Dietary Restrictions
- College Students

Problem:
- In a world with many virtual outlets that display food and recipes. It can be a hassle going through food articles, YouTube-recommended recipe videos, and quick snack-making videos on TikTok. Our project looks to congregate all those recipes and make them easier to share among everybody.

How it Will Be Addressed:
- Our project provides a system where Recipe Seekers can add their recipes and find other recipes that interest them.
- Share recipes with others
- Accessible display of recipe information and instructions
- Ability to save recipes they like for later use.

Link to UserStories and Product Backlog:
https://docs.google.com/spreadsheets/d/167RE-UeZfcS8EwxxckGO36KOEpBOGREtwLQRITICWI8/edit?usp=sharing

UserStories/ProductBacklog

Green = Complete, Yellow = InProgress, Red = Incomplete(Might Not Be Able to Complete)

Feature 1	As a Recipe Seeker, I would like to be able to filter recipes by ingredients. So that I can find recipes that match my dietary restrictions	Acceptance Criteria, Recipe Seekers should be able to tag and filter recipes	COMPLETE
Feature 2	As a Recipe Seeker, I would like to be able to see a list of available recipes. So that I can find recipes faster and easier	Acceptance Criteria, Recipe Seekers should have a list/feed of recipes to view and choose from	COMPLETE
Feature 3	As a user, I would like to be able to upload recipes so I can keep track of my favorite recipes and share them	Acceptance criteria: Recipe Seekers should be able to upload recipes, preferably with a quality filter such as a rating system(?)	COMPLETE
Feature 4	As a Recipe Seekers, I want to be able to adjust the sizes of recipes so that I can make enough servings	Acceptance Criteria: Recipe Seekers should be able to input the number of servings they want to make and have the measurements adjusted accordingly	INCOMPLETE
Feature 5	As a Recipe Seeker, I would like to be able to store items I have in an account. So that I can more easily find recipes	Acceptance Criteria: Recipe Seekers should be able to store ingredients they have on their account, and add and remove them when necessary.	COMPLETE

Feature 6	As a Recipe Seeker, I want to customize my account so I can distinguish myself.	Acceptance Criteria: Recipe Seekers should be able to add personal information and pictures to their account.	COMPLETE
Feature 7	As a user, I would like to be able to rate a recipe I made so that I can share my opinion about the recipe with others	Acceptance Criteria: The user should be able to provide a rating out of 5 stars for a recipe, as well as a longer form text review if they want to	COMPLETE
Feature 8	As a user, I would like to be able to customize my recipe feed to certain recipes I like.	Acceptance Criteria: Users will provide information at account creation about what they like and would like to see more of. Information will be in some form of a checklist.	INCOMPLETE

Summary of UserStories and Product Backlog:
Our Mobile Application revolves around Recipes and Enabling a place where Recipe Seekers can quickly access them. The UserStories mainly prioritized in the Product Backlog were basic Storing and Sharing components. The Product Backlog items such as Review and Search can then be addressed.

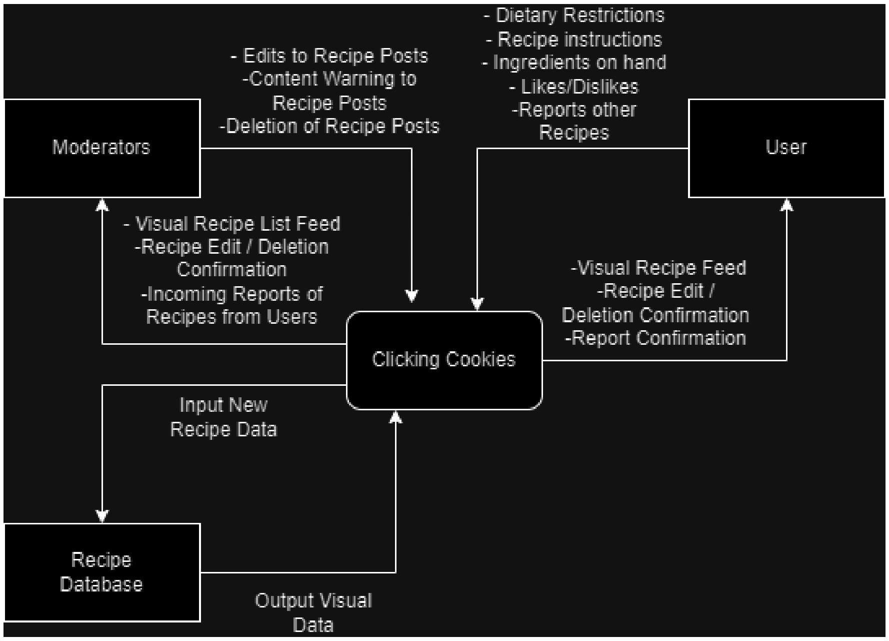

Figure 1.1 Context Diagram

Architectural Overview

The Software Architecture Style used in Clicking Cookies is “Client-Server” Architecture. In which the Clicking Cookies Mobile Application works as the service component that displays Information from the Firebase Platform and its own sub-services of Authentication, Cloud Storage, and Database Storage. Clicking Cookies being used as the “Client” side of the exchange to the Firebase “Server” Side. This allows for easy and quick communication between components..

The Major Architectural Components used for Clicking Cookies are the Mobile Application itself and the Database. The Mobile Application acting as the front of the Application, is developed in Android Studio: The Database being used for the backend is Firebase. Both work in tandem to store and share recipes in Clicking Cookies

Alternatives Components: Ideas for the Project mainly stemmed from the Database, due to the varying ways of storing and retrieving data and the varying skills and knowledge the team members had of each Database type. Alternative Databases that were tried were PHP/SQL, MongoDB, and Simple JsonAPI. PHP/SQL was eliminated due to team skills and knowledge of PHP/SQL being varied and work velocity for implementation being suboptimal. Similarly to PHP/SQL, MongoDB was eliminated as it took far more time and training to properly implement than it was being used for the Mobile Application. JSON API was eliminated due to not being the complexity of implementation and the lack of Compatibility when it came to updating the data to fit new FrontEnd Development.

Subsystems Architecture - Dependency Diagrams

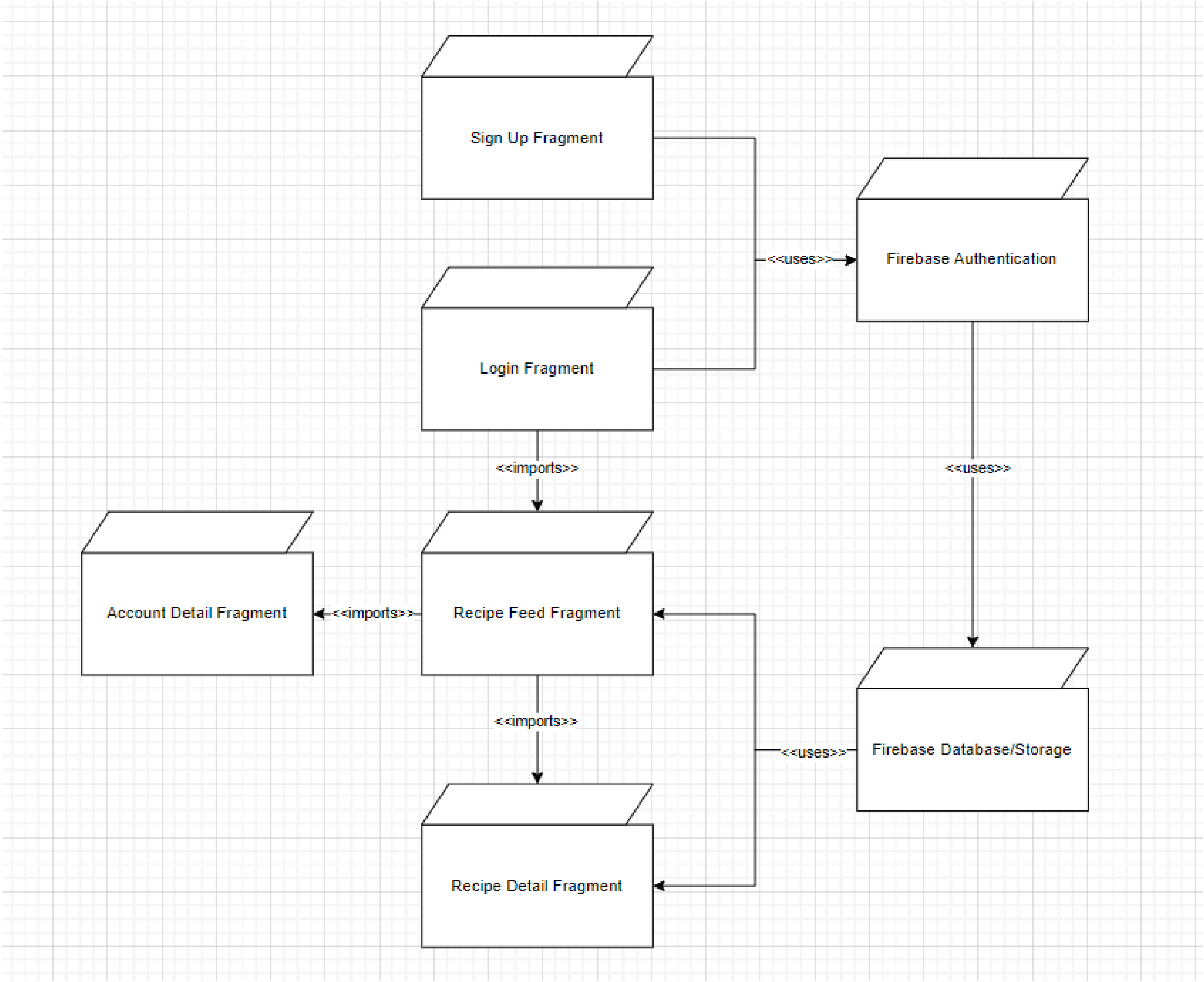

Figure 2.1 UML Package Diagram (General Workflow of the Application)

No deeper Subsystem Architecture was used for the Project, Just the Clicking Cookies acting as the Front End while Firebase is the back end.

Persistent Data Storage

Cookie Clicking Cookies has Persistent Data Storage through the use of Firebase. Firebase is a Data storage for User Accounts/Authentication and Basic Data Storage for the Recipes being Saved and Shared.

Global Control Flow

The Control Flow is Procedural, Creating a proper Recipe to be uploaded to the Firebase server requires every step and its subsequent steps to be completed before uploading to the server. There is no event loop coded within Clicking Cookies. No Time Dependencies are coded within Clicking Cookies. Clicking Cookies does not use Multithreading.

Static View Semantics and Quality

Clicking Cookies does not use classes or systems similar to classes to Visualize into a UML Class Diagram. Clicking Cookies does not pass information between ‘Fragments’ to even show relationships similarly shown in UML Class Diagrams, However, I can describe the responsibilities of each ‘Fragment’.

- SignUp Fragment: Works in Tandem with Firebase Authentication to create Accounts
- Login Fragment: Where Accounts are accessed and proper tokens and IDs are registered to display the right information for the account.
- Recipe Feed Fragment / Account Detail Fragment / Recipe Detail: Based on the token and ID of Firebase Authentication, Information is displayed. Tokens and IDs are NOT passed in from Login Fragment but go through Firebase Platform to verify information, the import is meant to describe the flow of the application as does the other import specified in (figure 2.1).

In summary, Fragments don’t pass information between other Fragments as Classes do. They rely on the overarching Database to pass and store data.

Dynamic View Semantics and Syntax

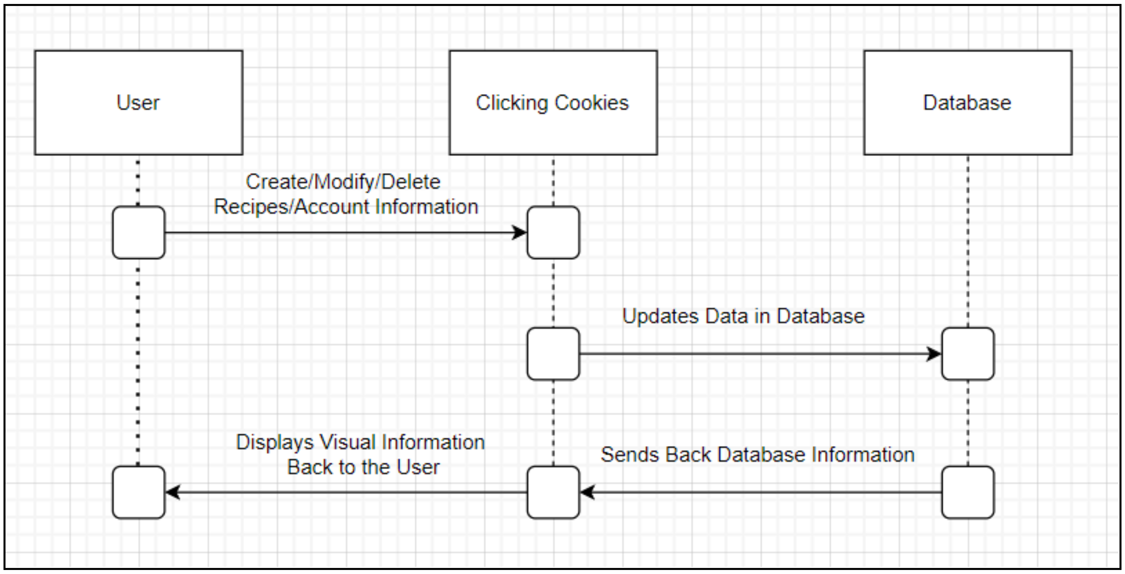
User
Clicking Cookies
Database
Create/Modify/Delete Recipes/Account Information
Updates Data in Database
Displays Visual Information Back to the User
Sends Back Database Information

Figure 3.2 UML Sequence Diagram (Based on the Deployment Diagram)

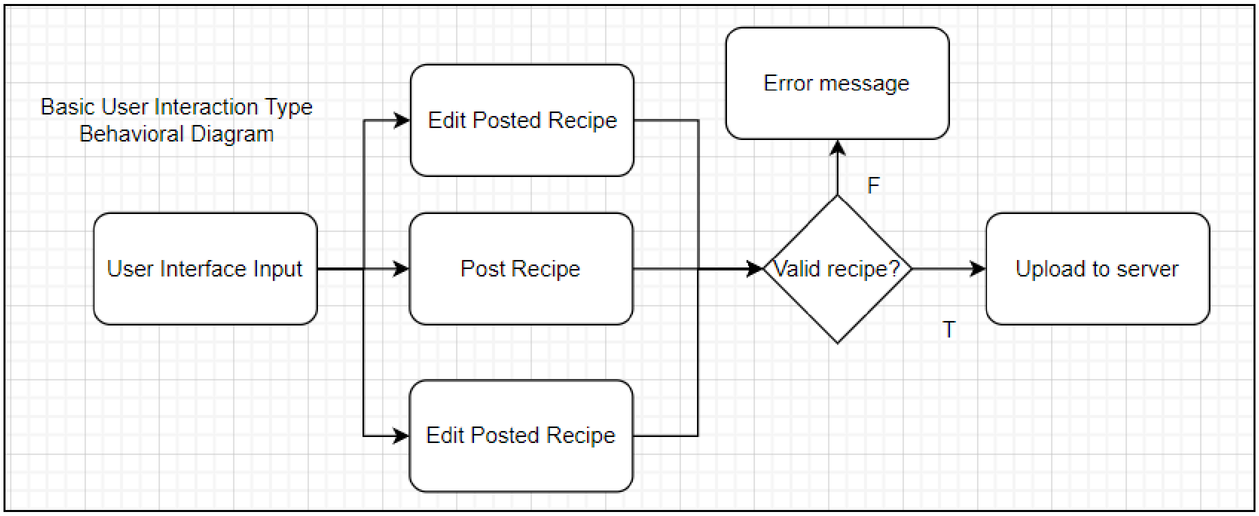

Figure 3.3 Behavior Diagram

Design Rational

The Design rationale of the Graphs is meant to communicate a simple, quick, and effective way of displaying, storing, and sharing information between the Clicking Cookies Application and the Firebase Platform. This is characterized by the main problem our Software Application looks to solve, to be able to provide Recipe Seekers the ability to have quick access to their recipes and to quickly share with others.