COOKBOOK

By
Aaron Amha,
Alex Foucher,
Michael Dempsey.
Sarah Sharroufna

Description Of The Project

Inspired by the most successful social media ever, Facebook, our group sought to provide foodies and general culinary enthusiasts a network that allows all of the limitless and creative ideas had in the kitchen, to come together on one platform. On our interface, registered users have their own personalized profiles which consist of their contributions to the network, as well as recipes that have captivated their interests. Users are allowed to interact with other recipes on the platform with comments which fosters a sense of community.

THE TOOLS USED BY OUR GROUP:

PROJECT TRACKING: OUTLOOK

Outlook constantly reminding us throughout the week about Varsha's weekly project meetings, as well as team updates to the github repository, gives it…

Ranking: ★★★★★

VERSION CONTROL SYSTEM (VCS) REPOSITORY: GITHUB

Despite its prominence in this industry, our project's Github Repository presented a plethora of merge errors. Code pulled for some members would be fully functional while others had issues.

Ranking: ★★★½☆

THE TOOLS USED BY OUR GROUP:

DATABASE: POSTGRESQL

PostgreSQL

Granted our Database Configuration and Connection, as well as pg-promise libraries and create.sql all either say .env.POSTGRES or use PostgreSQL syntax, we can say without a doubt that our database was POSTGRESQL. Issues with this Database were mainly its complexity and lack of efficiency in handling BLOBs or (Binary Large Objects). A BLOB, for those unfamiliar, is the conversion of picture/video attachments to binary for storage purposes. With an application that deals largely with picture uploads, this inefficiency is limiting… Otherwise, not too bad…

Rating: ★★★★☆

THE TOOLS USED BY OUR GROUP:

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE): VS CODE

Visual Studio Code

Quick, reliable, free…

Good ole VS Code

Ranking:

★★★★★

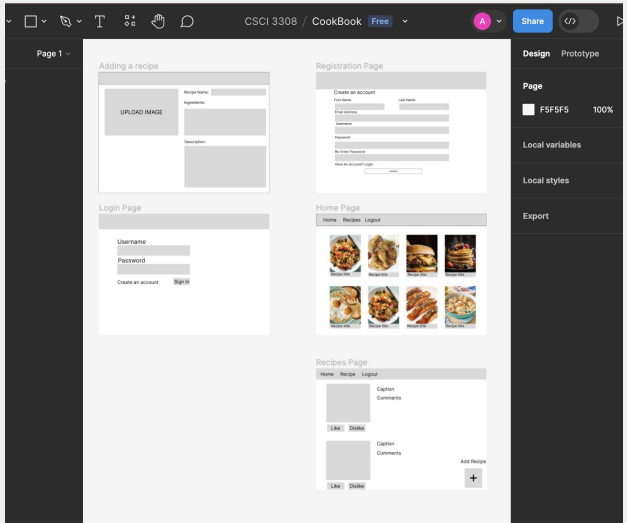
THE TOOLS USED BY OUR GROUP:

UI TOOLS (WIREFRAMES): FIGMA

Although it was only used during the initial development/planning phase to get an idea of the necessary pages and features, it set a major precedent for formulating of the UI. The initial plans made on FIGMA certainly had a major influence on the outcome.

Ranking:

★★★★★



THE TOOLS USED BY OUR GROUP:

DEPLOYMENT ENVIRONMENTS:

CLOUD DEPLOYMENT: AZURE

A phenomenal source of Iaas, Paas, and SaaS cloud services, Microsoft Azure is a solid cloud services environment for deployment reasons. Downside, wayy too many settings to figure out during initialization without Lab #13.

Rating: ★★★★☆

CONTAINER DEPLOYMENT: DOCKER

No issues at all, a vital part in development as it provides access to easily accessible isolated environments. Efficient containerization.

Rating: ★★★★★

THE TOOLS USED BY OUR GROUP:

APPLICATION SERVER: NODEJS

Node.JS is highly optimal in handling an immense amount of libraries improving the software development process. Frequent async/await usage!

Ranking: ★★★★★

EXTERNAL APIS: SPOONACULAR

This API was close to perfect for our project as it not only provided a sufficient amount of recipes, but it even provided extra functionalities such as recipe searching and summarizing. The only recordable issue would be the API Rate Limit :(.

Ranking: ★★★★½

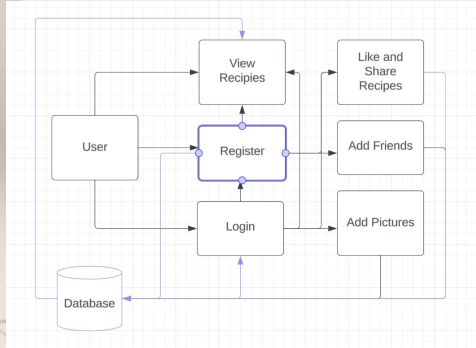
THE TOOLS USED BY OUR GROUP:

TESTING TOOLS: CHAI HTTP - NODE.JS

Chai HTTP provides an interface for live integration testing of the API's. The syntax behind it is easy to understand and implement, however, this process adds more seconds on Docker-Compose database/container initialization

Ranking: ★★★★½

FRAMEWORK: USER - DIAGRAM



We created our user diagram early on in the initial development phase of the project.

METHODOLOGY

AGILE SCRUM FRAMEWORK

Prioritized list of features

Based on duration of task, importance to project, reliance of other tasks

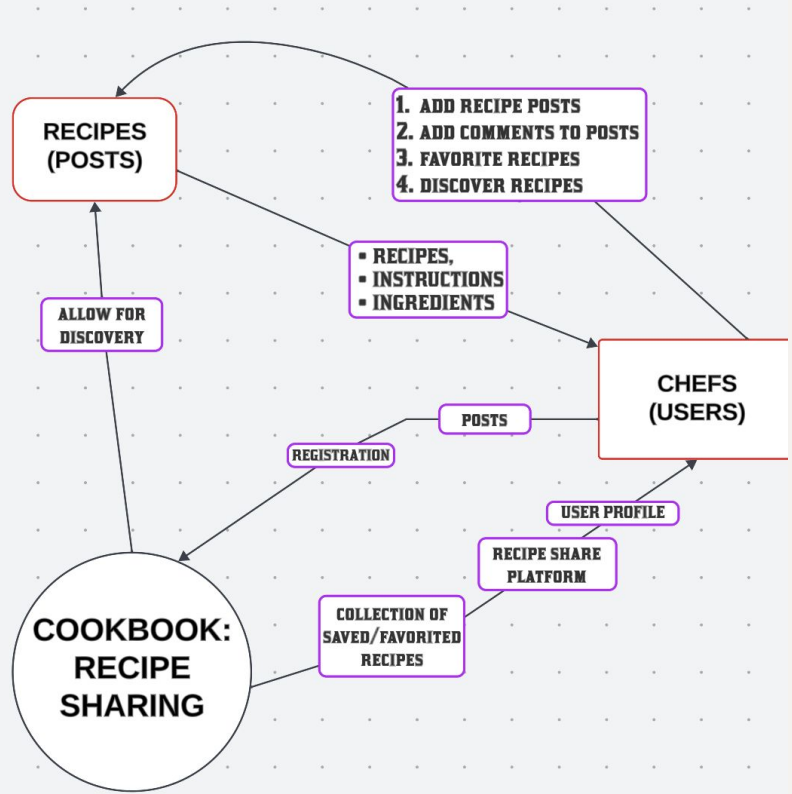
Team members were assigned different tasks

Scrum master

Ranking: ★★★½

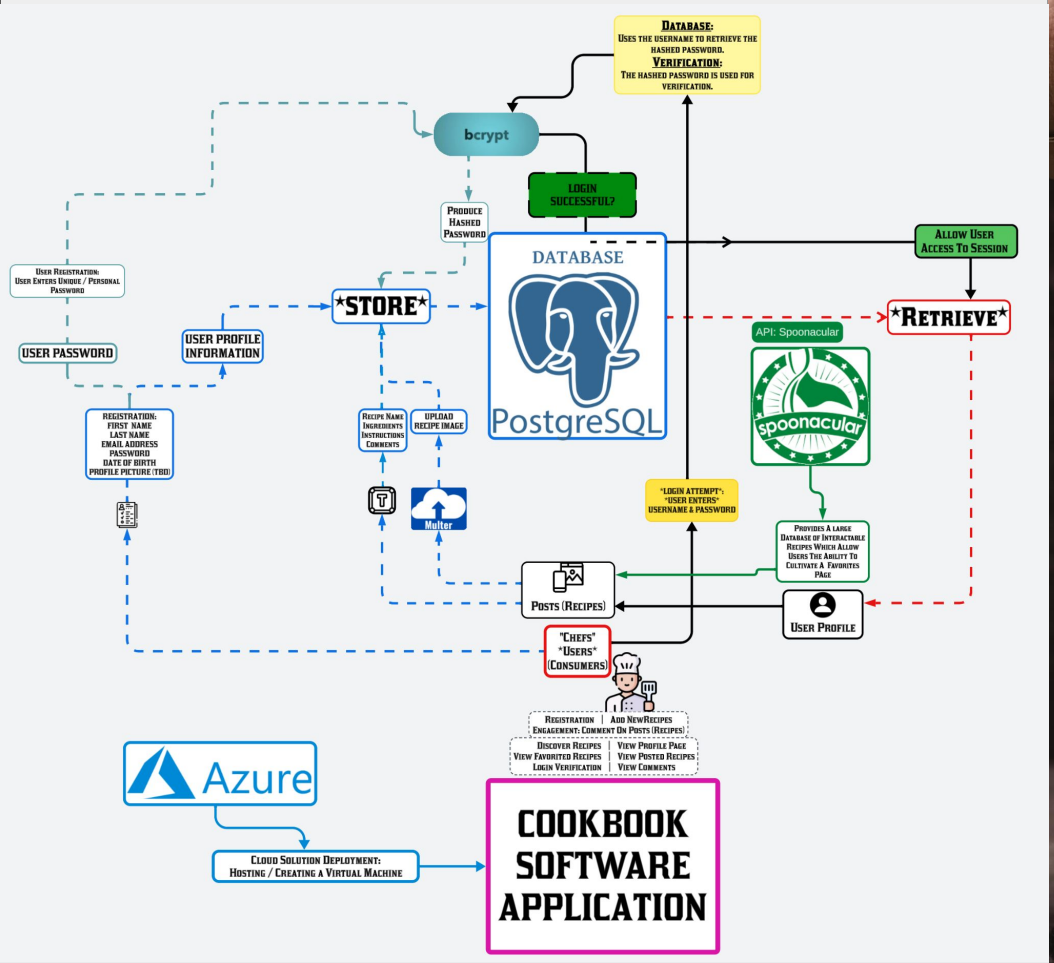
SYSTEM ARCHITECTURE DIAGRAMS

CONTEXT DIAGRAM



Contribute: https://lucid.app/join Enter Join ID: 2W4 4MD

HIGH SOLUTION ARCHITECTURE DIAGRAM



CHALLENGES FACED

- One challenge we faced was merge conflicts. Everyone had their own work in different branches and more times than not, there were lots of errors that arose.
  - We resolved most merge conflicts by merging our branches together prior to pushing to main
  - We used the Merge Editor in VS Code to look at all the differences
  - Merge conflicts caused certain tasks to take longer than expected, which lowered the scope of the project
- Another challenge we faced involved tasks being dependent on other tasks.
  - For example, in order to have a profile page, the user database would need to be completed.

CHALLENGES FACED (CONT)

- We faced the challenge of delegating tasks. At the beginning, people were working on the same things, which led to us replacing code.
    - This caused tasks to take longer than expected
    - We overcame this challenge by assigning tasks and explicitly stating what we were going to work on