Project Milestone 4

Team #: 13-05
Team Name: Fruit Leather

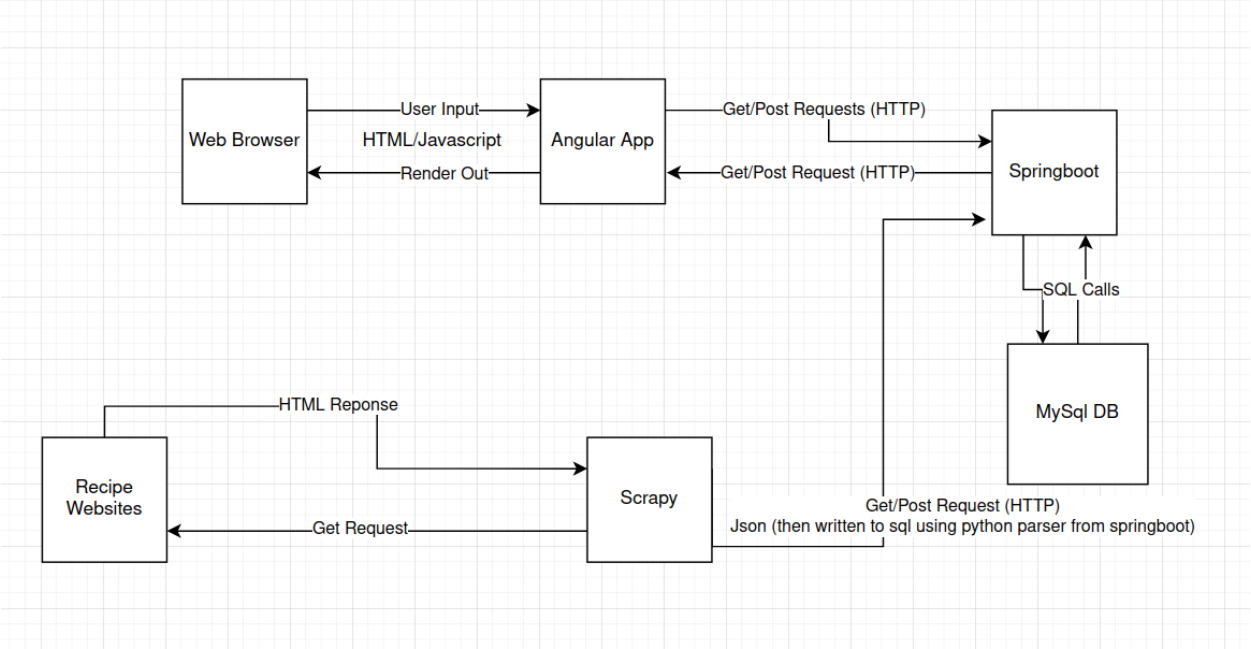
Features:

- Register and create an account
    - User uses email and password to create an account which is tied to their favorite recipes and sources
- Log In
    - User can log in to access their saved recipes and ingredients
- Enter Ingredients
    - User enters ingredients to generate a list of recipes
- Generate recipes
    - Pull list of recipes from web scraping database that are based on users ingredients
- Save favorites
    - Users can save their favorite recipes for future reference
- Share recipes
    - User can share recipes with friends
- Featured recipe page
    - Featured recipes, should be implemented on home page

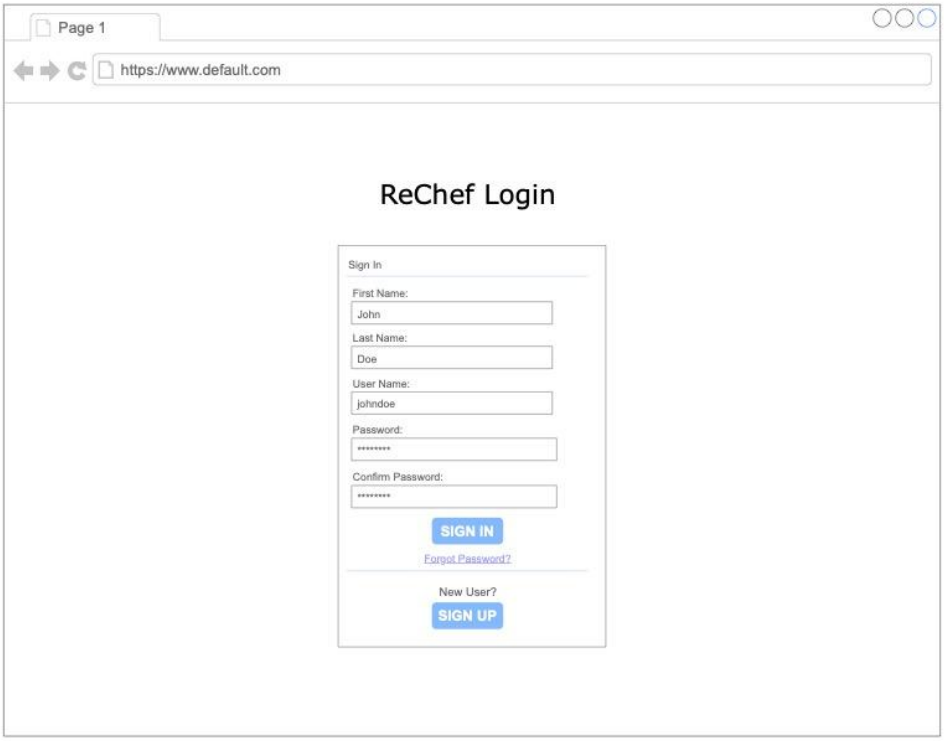
Application Architecture on next page…….. (formatting got messed up)

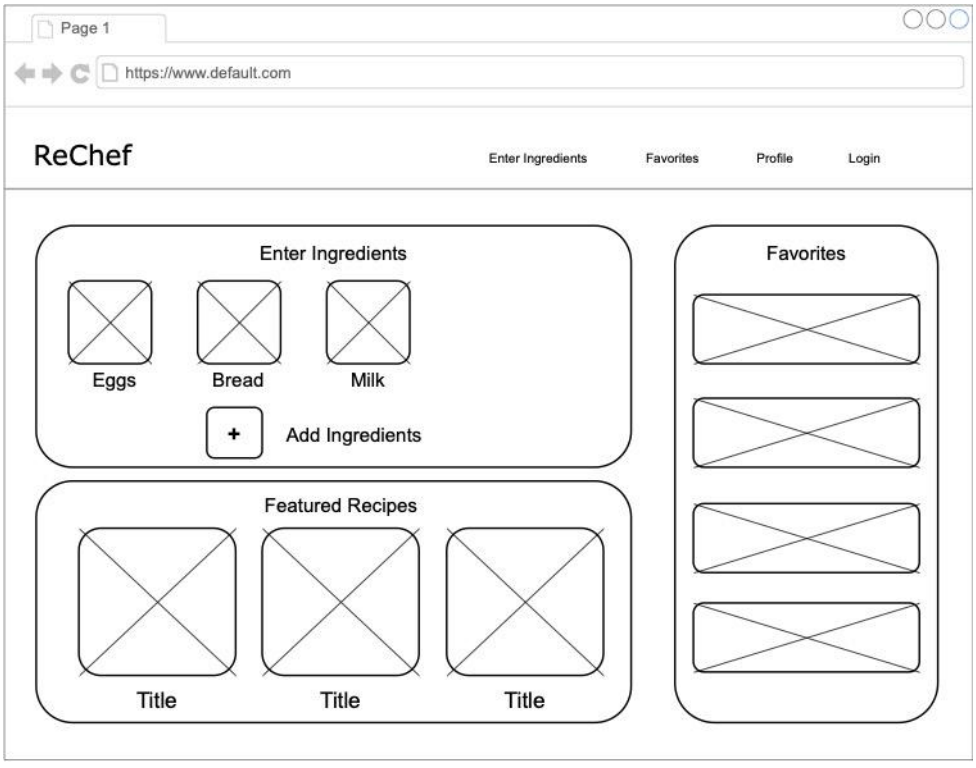
Application Architecture:

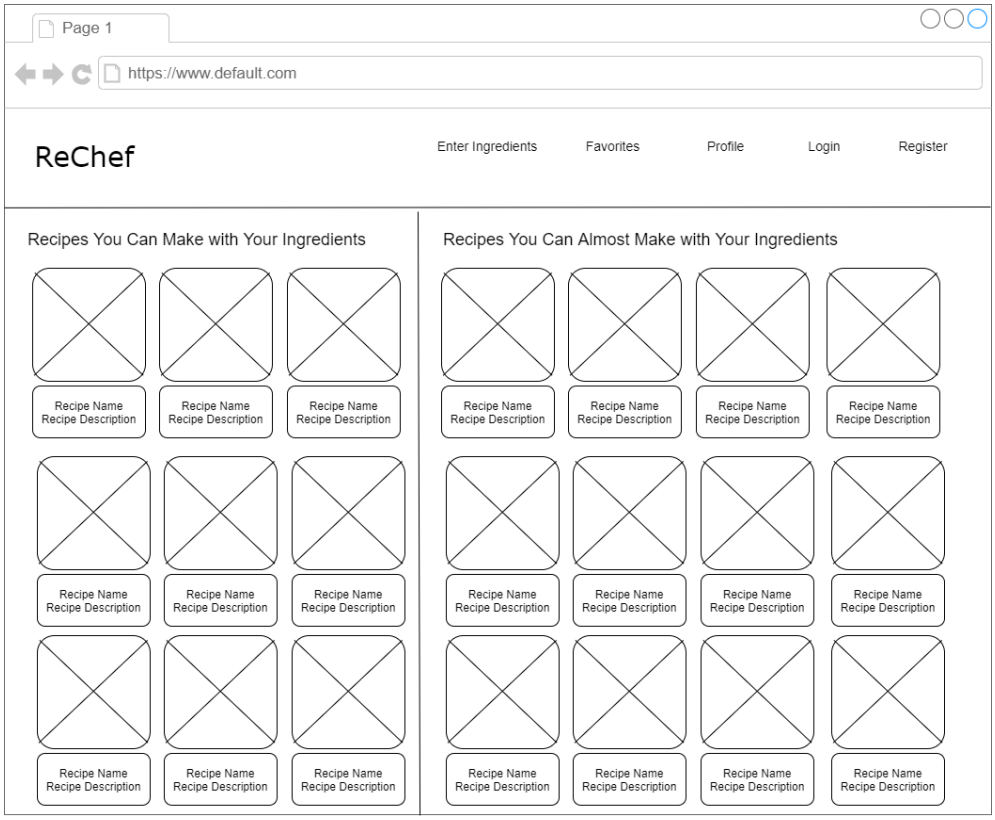
- The Springboot app, Angular app, and Mysql instance will be on one of our machines
- The python app that will run scrapy will be on the same local machine and run once a week, sending recipe info to the springboot app which will place that info into the db
- We currently only plan to have this as a local website



Front End Wireframes:



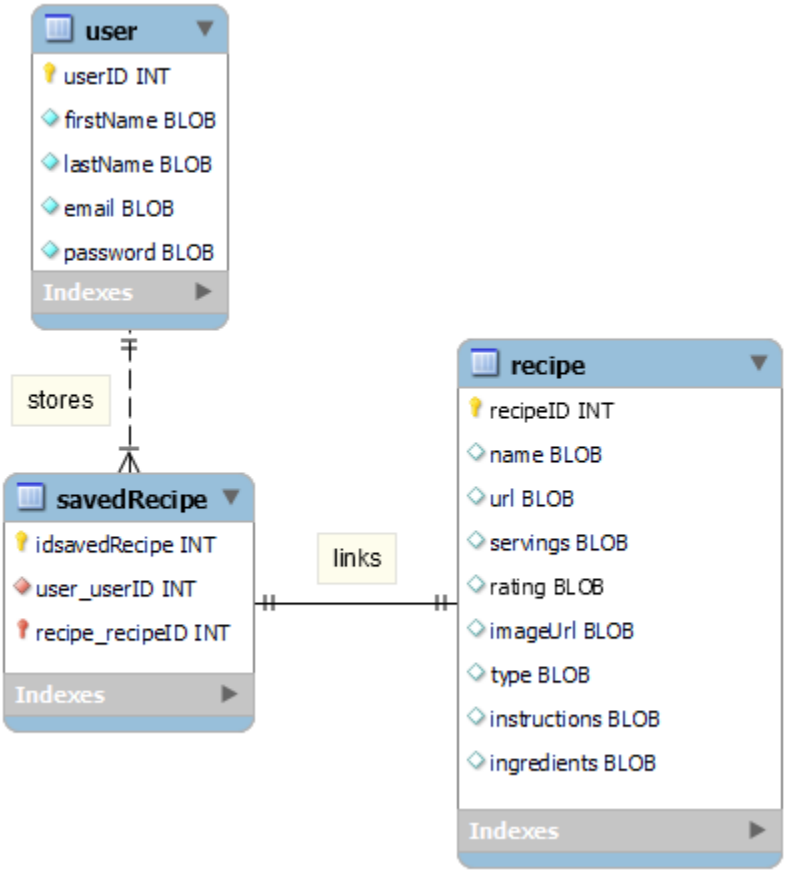




Web Service Design: N/A since we do not use any APIs

Database Design:

- We will be using MySQL for our Database server.



Challenges and Risks

- Challenge: Potential SQL injection problems (even with parameterized usage of JPA) or malformed input into the get/post requests (type handling)
- Challenge: Parsing recipe information from the webscrape and placing data into SQL database so that it can be efficiently queried. For example in our current interaction, the ingredients are given in the json as an array, but each ingredient string also contains quantity information - this may make querying more difficult, but parsing out the quantity info is also quite hard)
- Risk: We have been working on the angular/springboot stuff and the webscraping stuff in an isolated way. We know we need to integrate them sooner or later so that the springboot app handles all db activity, but we are sure there will be unforeseen problems in automating the webscraping separately and integrating it with Springboot.
- Risk Mitigation / Backup:

    - Addition of security framework which ensures safe and cleaned inputs
    - Manual/community implementation of recipe information and data instead of web scraping in case automated app and parsing doesn't work
    - Just shutting down the front end of the site when the webscraping is taking place and having the webscraper app write directly to the database

Individual Contributions:

Carson

- Wrote script to transfer recipes from json recipes file into the database.
- Edited the data model and database generation code.
- https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-013-05/commit/2ad96389d22674450c9aad06922351842eee0ff0

Md

- Updated scraper so already scraped recipes don't get scraped again
- Uploaded a new scraper for allrecipes.com
- https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-013-05/commit/f2b2d60d384c9b1d2b5a969078c36dad6bf06594

Ian M.

- Worked on back end features
- Coordinating enter ingredients with new SQL data table
- https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-013-05/commit/75ac1fd08cd537eed7a66b96744d06ececfdb8cc

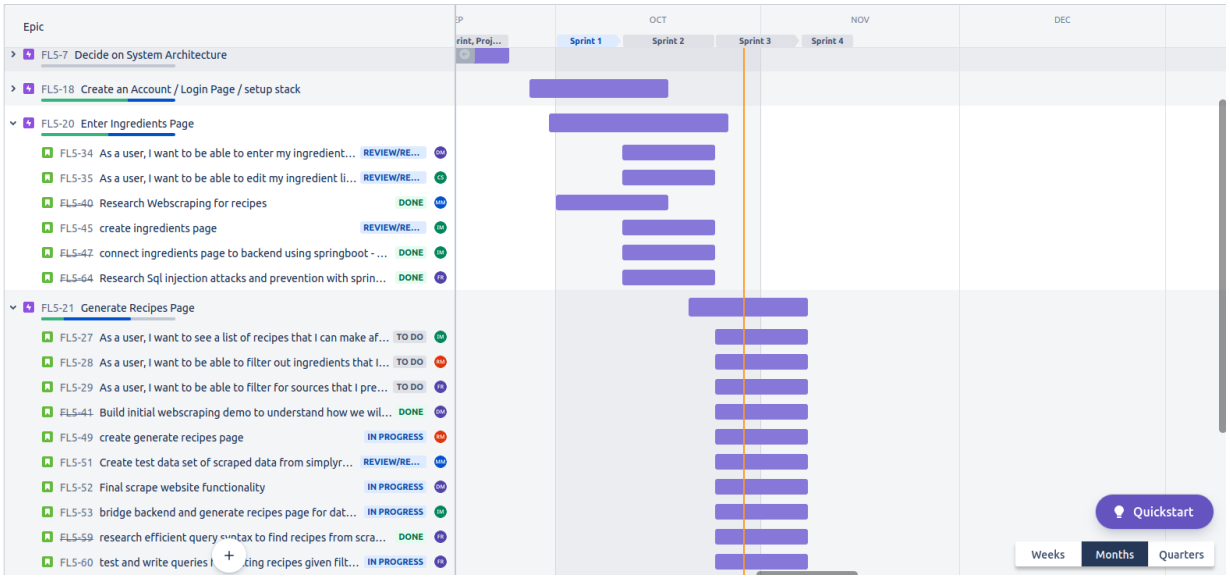
Ferin Von Reich

- Researched and implemented parameterized springboot queries to prevent sql injection
- Researched fuzzy queries for getting recipes from the db given mispellings or other small differences; currently writing those queries
- https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-013-05/commit/39db949a1be6bbe52b578a33991f442c5e69011d

Roadmap

Share Export

FR CS IM DM +3 Status category Today View settings



Projects / Fruit Leather-13-5

Roadmap

Share Export

FR CS IM DM +3 Status category Today View settings