CS 3354.004 (Dr. Ebru Cankaya)

Software Engineering
Final Project Deliverable 1 Report

SCRAPPY MEALS

By:
Irfan Zobayed
Po-Yu Liu
Zunayed Siddiqui
Rohit Shenoy
Ryan Rahman
Steven Ta
Martin Lam

PART 1A: Final Project Draft Description

Project Description

The main goal is to develop a design for an application that allows a user to identify potential food items from a preset library of items typically utilized for cooking (Ex: tomatoes, cucumbers, onions) and input the specified quantity of these items in sequential order. Once the user inputs one item, a list of meals that can be developed concurrently will be shown. As the user begins to input more items, meals become more specified and the list becomes smaller until the desired number of food items are inputted. Alongside these quantified food item inputs, the user will have the option to select: the difficulty of making a meal, time to make a meal, ethnicity of a meal, spice level of a meal, “feed how many” input of a meal, and other potential ideas. All meal ideas will come from a library where users can also input meals with proper criteria.

Project Motivation

The majority of students are typically hungry, lazy, and often buy supplies that go unused because they are short on time. Expanding from a student’s environment, a lot of households have leftover food items, such as eggs, milk, or vegetables, that will go bad if it is not used. The goal of this app is to allow users to put in all these potential ingredients and see what they can make with what they have, or even potentially buy what they are missing to make a meal of their choice. This application will allow users to become chefs and promote healthy lifestyle choices and less wasted food.

Real World Applications

This application can be utilized by essentially anyone who has access to food items and a kitchen. The target audience is college students and people who don't have the time to always prepare a well thought out meal yet seeks originality and variety in their daily meals with all the items they have in their pantries, which helps avoid fast food options.

Similar Software Comparisons

This application is comparable to many applications that are already out there. SuperCook is an application that essentially does the same thing as Scrappy Meals in that the user can input what ingredients that have, and the software spits out recipes using those ingredients. However, Scrappy Meals has filters that the user may use that will only show recipes that suitable with how many people they need to feed, how much time they have to cook, and what methods of cooking they have available.

Feedback Response

According to our initial feedback from our proposal, Dr. Cankaya believed our project was a very interesting idea. She accepted our description and delegations of the tasks. All she wanted was for us to add a section about similar software models that already do a similar task, compare the similarities and differences between the two processes, and provide a statement as to why our software is unique compared to others, as depicted by the section above.

PART 1B: Setting Up a GitHub Repository

Team Name: SCRAPPY MEALS

Team Member Github ID's:

- Irfan Zobayed *(created repository)*
    - SurfinIrfin
    - imz160030@utdallas.edu
- Po-Yu Liu *(uploaded README)*
    - pxl170130
    - pxl170130@utdallas.edu
- Ryan Rahman *(uploaded project_scope.docx)*
    - rahmannoodles98
    - rahmanr98@gmail.com
- Steven Ta
    - sxt168530
    - sxt168530@utdallas.edu
- Martin Lam *(uploaded project_scope.pdf)*
    - yayap12
    - martinlam0505@gmail.com
- Zunayed Siddiqui
    - zunayedsiddiqui
    - zis170030@utdallas.edu
- Rohit Shenoy
    - rohitkshenoy
    - rohit.shenoy@utdallas.edu

Repository Link: https://github.com/SurfinIrfin/3354-ScrappyMeals.git

PART 2: Delegation of Tasks Amongst Group

Each individual below will be in charge of these specific tasks for the project (subject to change):

- Irfan:
  - Recognize a library of grocery items that the user can select
  - Organize Github and finalize description (1A, 1B, 2)
- Steven:
  - List out potential options to help narrow meal search results
  - Creating a case diagram (5)
- Martin:
  - Create an option for users to input their own meals ideas into the meal library
  - Creating a sequence diagram (6)
- Po-Yu:
  - Identify a library that has meals based on the food items library utilized
  - Defined requirements (4)
  - Committed README file
- Ryan:
  - Construct a user interface that is adaptable and friendly for a target audience
  - Defined software process model (3)
  - Committed project_scope file
- Rohit:
  - Determine how many options will suffice and simplify the user experience
  - Defined architectural design (8)
- Zunayed:
  - Develop tests for app functions & organize GitHub
  - Designed sequence diagram (7)

PART 3: Our Software Process Model

The software process model that was employed was the waterfall model. The waterfall model was chosen as we’ve understood all the requirements and they were unlikely to change throughout the process of development. In addition, the waterfall model allows for a schedule with deadlines for each stage of development, so the software can be built on time. The waterfall model also allows for more tasks to be arranged with ease, which was helpful as different aspects of the software was worked on by different people, and the validation at each level made sure that the software that was being properly developed and integrated, which was checking off each of the required elements.

This is demonstrated in our software as it was broken up into five sub-categories: ingredients library, meal results, search results filter, user-modification, and other. Each of these sub-categories had different tasks that had to be completed and was well-documented.

PART 4: Software Requirements

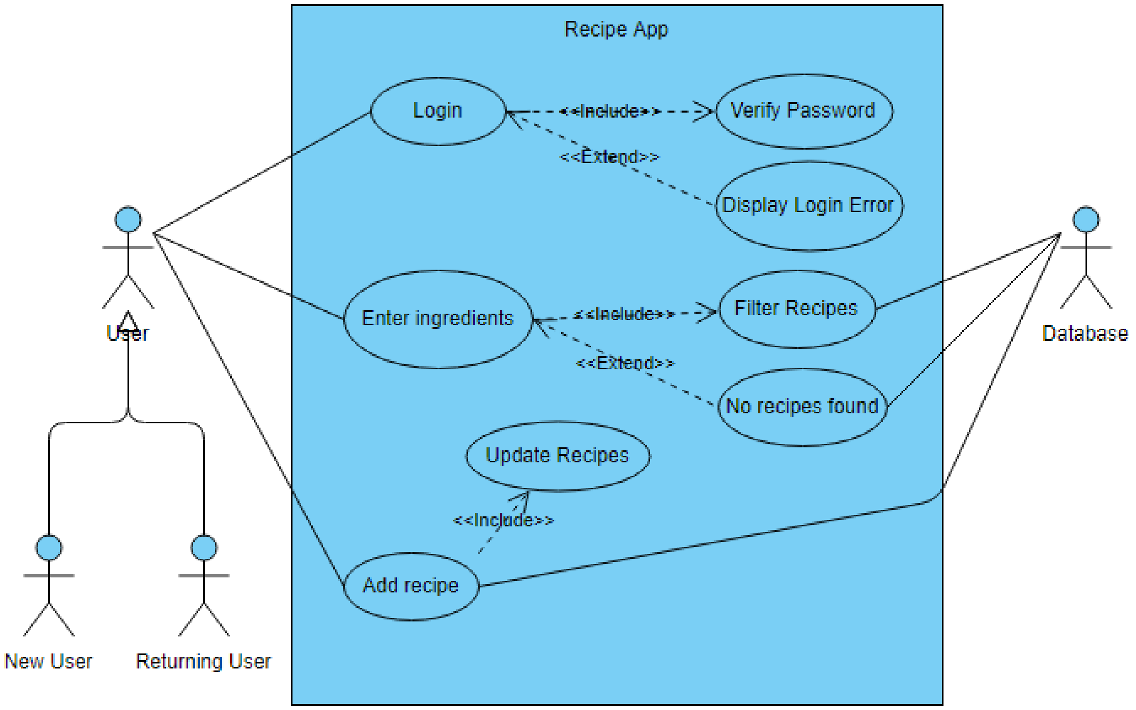
Functional Requirements

i. Allow user input for ingredients
ii. Back-end filtering of ingredient information
iii. Hold database of recipes
iv. Connect the recipe database to filter for ingredients, to search for a recipe
v. Output recipe options for the user

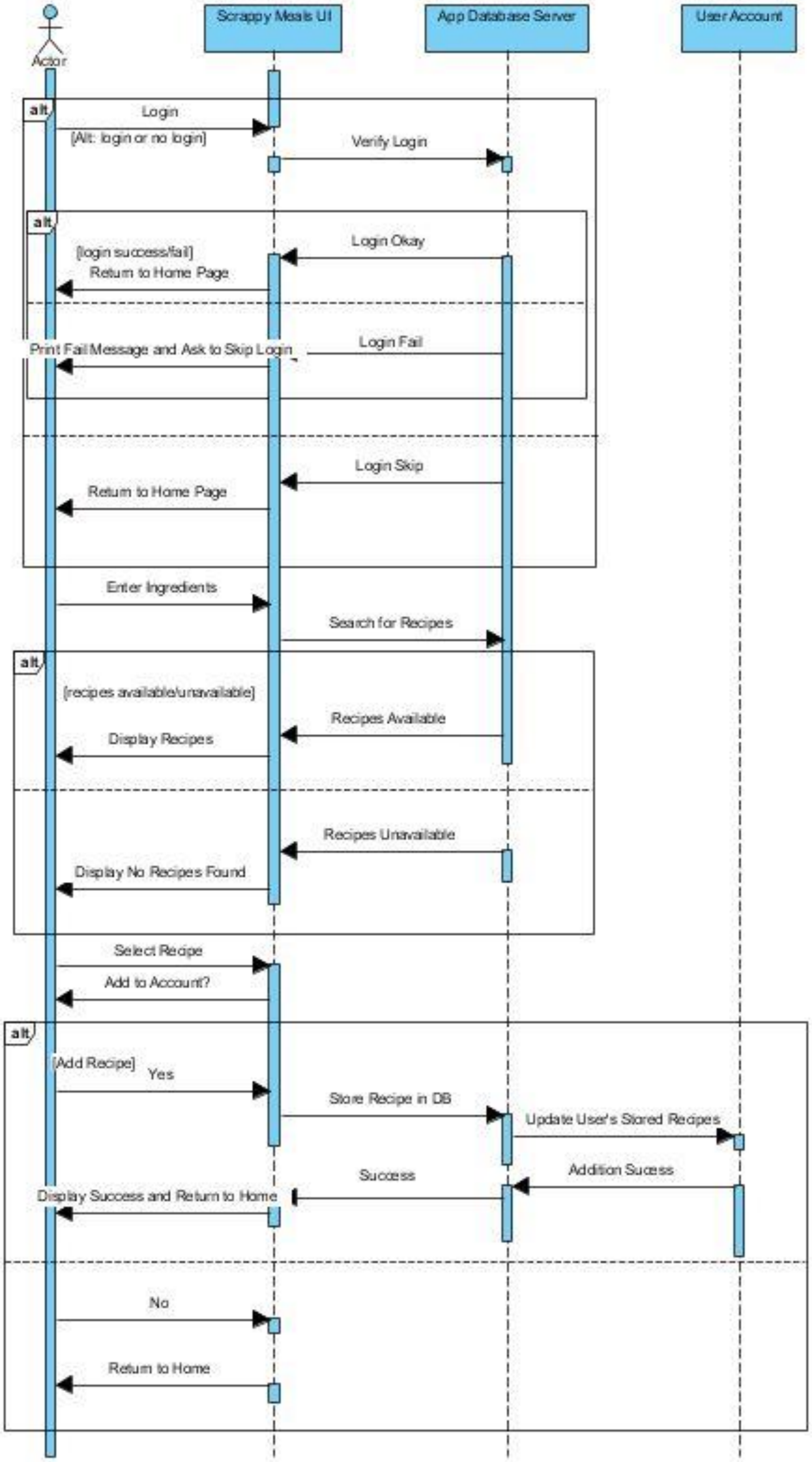
Nonfunctional Requirements

vi. Delete duplicate ingredients
vii. Complete filtering of ingredients and output recipe within 5 seconds
viii. Provide at least one alternative recipe even if ingredients don't match
ix. Allow organizations to login if they want to add their own recipes
x. Ensure the recipe database complies with Intellectual Property laws

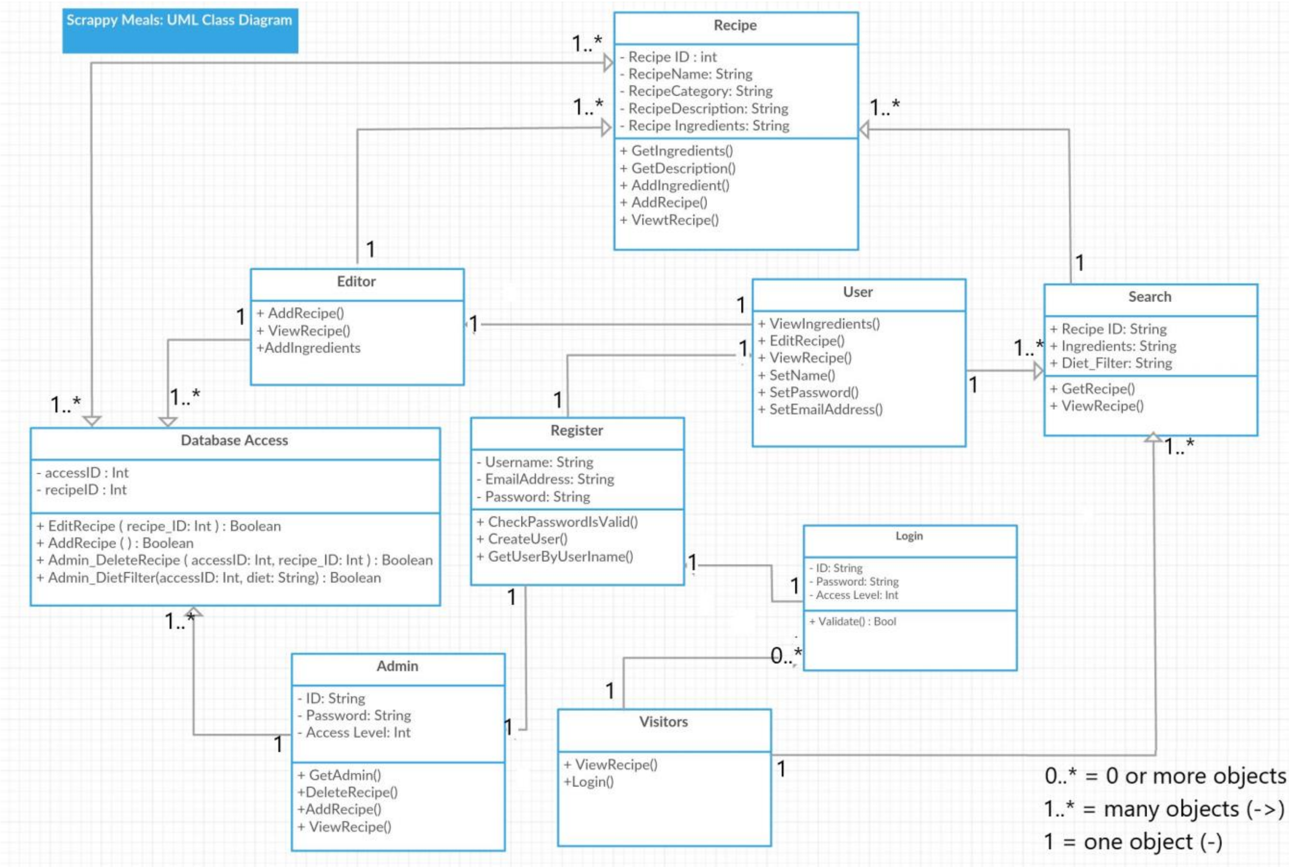
PART 5: Case Diagram



PART 6: Sequence Diagram



PART 7: Class Diagram



PART 8: Architectural Design

Our project requires the user to enter the ingredients they have in their house one by one and it will provide a list of meals the user can make after each ingredient is entered (so the list will get shorter as the ingredients are updated). In a pipe and filter system, the data in a system is organized so that each processing component (filter) is discrete and carries out one type of data transformation. The data flows (as in a pipe) from one component to another for processing, which is exactly how the data in our system will flow. The input data goes through a pipe and our database is filtered to find the matching recipes. The normal problems of agreeing upon standards for data transfer and parsing aren’t too disadvantageous in this project because the parsing and data transfer is pretty simple.

Furthermore, the general use case for pipe and filter system is for data processing applications (both batch- and transaction-based) where inputs are processed in separate stages to generate related outputs. This workflow style matches the structure of many business processes, which is what our project would eventually be - a business that provides meal recommendations. Finally, the evolution by adding transformations is also straightforward, which would help us as we’ll constantly need to evolve the database.