PL8
Design Document

Sundeep Bath, Kyle Heilman, Travis Kovacic,
Austin Senseman, Rajith Weerasinghe, Kevin Zhou

Purpose:

Nutrition can be difficult for those who aren't skilled in cooking, especially those doing so on a budget and with dietary restrictions. There are a few services already in existence that suggest recipes based on what foods you already have. However, none of them suggest extra ingredients to make even more recipes or allow diabetics and others with specific diets to monitor the levels of certain nutrients that could cause them harm. PL8 will present users with recipes based on their individual needs.

Functional Requirements:

- **Users can see what recipes they can make based on**
    - As a user, I want to have new recipes suggested to me based on what I already have so that I can have a variety of foods
    - As a user, I want to have shopping items suggested to me which allow me to make as many new dishes as possible
    - As a user, I want an easy and convenient way to add and remove ingredients from my virtual pantry
    - As a user, I want to have shopping items suggested to me which allow me to make as many new dishes as possible
- **Users can filter and sort recipes based on specific criterion**
    - As a user, I want to be able to be able to filter results so that certain ingredients never come up in results
    - As a user, I want to be able to filter based on type of meal
- **Users should be able to see additional information when they look at a specific recipe**
    - As a user, I want to see suggestions for other recipes when I view a recipe
    - As a user, I want to see an option that lets me print the recipe or share it via email or social media
    - As a user, I want to see an option that lets me save this recipe the next time I visit
    - As a user, I want to see advice from users on recipes so that I can know what to expect when attempting a recipe
    - As a user, I want to see the nutritional information of a recipe compared to a tangible value, ie "Your recipe has as many calories as 3 Hershey's Chocolate Bars"
    - As a user, I want to know how long a certain recipe takes to prepare

- **Users should be able to contribute to the growth and well-being of the site**
    - As a user, I want to be able to submit my own recipes
    - As an owner I want my application to not be vandalized (report system for vandals posting not-recipes)
    - As a user I want to be able to post comments on recipes and suggest alternatives with different nutritional content

Non-Functional Requirements:

Security

- As a user I want my login credentials as well as my preferences and everything else to be secure.
- As a user I want my dietary restrictions to be private and unviewable by other users.
- As a developer, I want my security to be robust with minimal effects on performance.

Server

- As a user I want to be able to access saved recipes and my shopping list offline.
- As a developer, I want server downtime to be minimal (less than 2 hours per month) or preferably, non existent.
- As a developer, I would like the server to be able to handle thousands of users at a time.

Design

- As a developer, I would like the design to be intuitive and minimalistic.

Performance

- As a developer, I would like the search algorithms to be both fast and effective.
- As a developer, I would like the different screens to load fast.

Design Outline:

(a) Outline your design decisions (for example client-server model), identify the components of your system, and describe the purpose of each component.
(b) Describe the interactions between individual system components.
(c) Include at least one UML diagram that clearly shows high-level structure of your system.

Program Model:

PL8 will use a client-server model. The web client will interact with the server, which will interact with the database. The details of which are listed below.

1. **Web Client**
    a. The web client will be the main interface of the system.
    b. Users will be able to retrieve and send data using the client
2. **Server**
    a. The server will accept requests from the client and send information to the database.
    b. The server will handle certain algorithmic calculations
    c. The server will query the database in order to retrieve recipes and nutrition.
    d. Validates user requests.
3. **Database**
    a. The database will store user accounts, recipes, and recipe information.

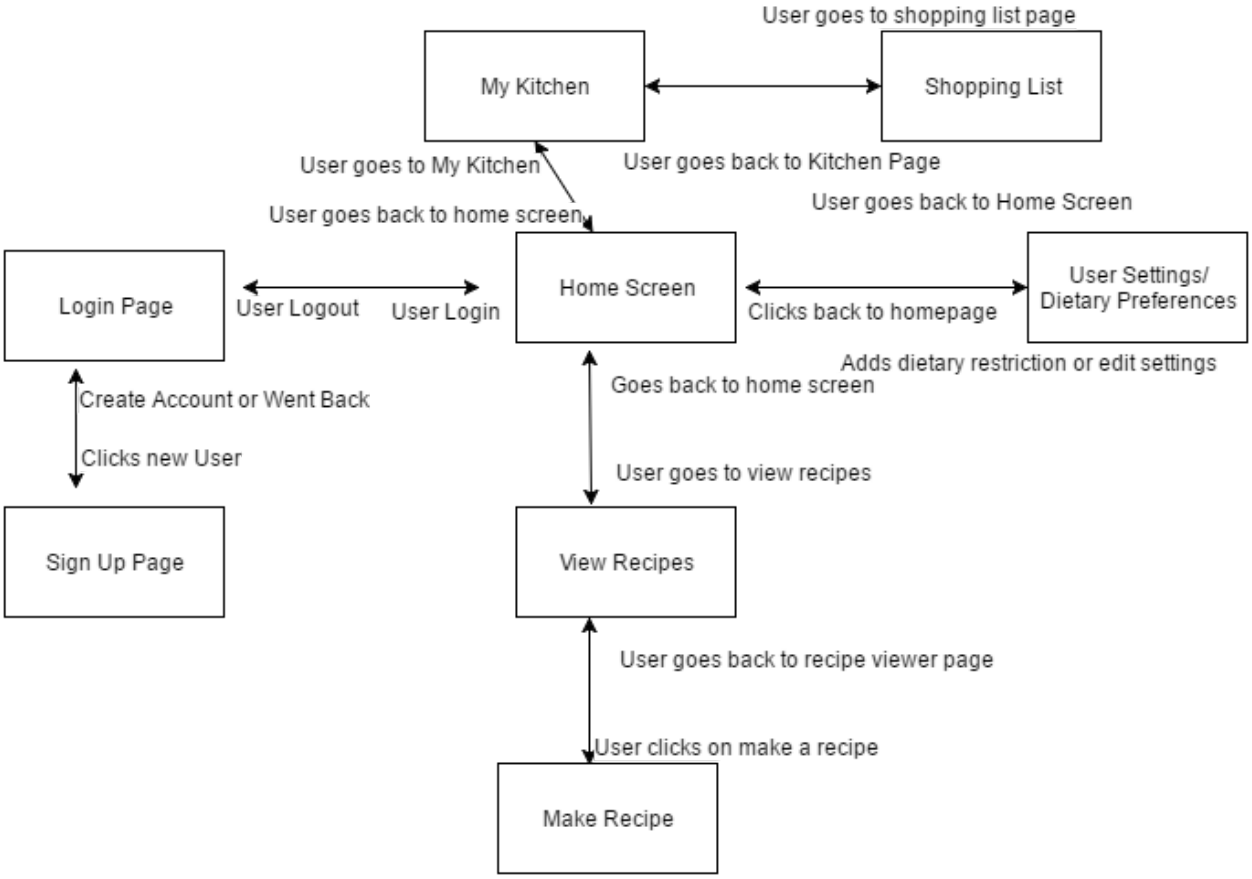
Site Navigation:

Naturally, the website will have multiple pages, each with their own purpose and set of tasks. Each page will have the necessary information.

- Login/Signup
    - Allows users to create an account or to log into their existing account
- View recipes
    - A list of recipes that can be made with the given ingredients
    - Suggestions for additional ingredients
- Make recipe
    - Detailed view of recipe including ingredients, steps, nutritional info, etc
- Shopping List
    - List of items to buy later
- My Kitchen
    - List of ingredients which can be used in a recipe

- User Settings
- Dietary Preferences
    - Additional filters for recipes

The following diagram shows the behavior of the app as it moves from one page to another.



Design Issues:

Functional Issues:

What actions need a user login to perform?

- Option 1: View any recipe
- Option 2: View nutrition information
- **Option 3: Save settings**
- **Option 4: Vote, comment, or submit alternative for a recipe**
- **Option 5: Create recipe**

We want to find a balance between displaying information to all, and not over-saturating or inflating the community features. Therefore, we are restricting the options for the selected above to logged in users only.

Should users be able to vote on recipes without having an account?

- Option 1: Yes
- **Option 2: No**

We decided that users must create an account in order to vote on recipes. This would help prevent vote manipulation

Should login be handled via a third party or by PL8

- **Option 1: Handle by PL8**
- **Option 2: Handle via Google**
- **Option 3: Handle via Facebook**

Users will have the option to login via their Facebook or Google account, which offers convenience, or via PL8, which has the advantage of privacy. The majority of modern websites have this type of system.

Should we enforce password restrictions(need a number or special character)?

- **Option 1: No**
- Option 2: Yes, need a capital letter, number, and special character
- Option 3: Yes, only need a capital letter and number

Password restrictions require users to often times add new characters to their passwords that may cause them to forget their login information.

Where should we display the nutritional info?

- Option 1: In the recipe
- Option 2: In each ingredient

- **Option 3: Both; Total in the recipe, and individual nutritional info in each ingredient**

Users may find it helpful to know what each individual ingredient contains in case they wish to remove an ingredient. The full recipe nutrition will also likely be useful to the user.

Non-Functional Issues:

What architecture will PL8 use?

- **Option 1: Client Server**
- Option 2: Unified Architecture

Dividing the architecture into two parts, frontend and backend, will allow for easier collaboration as it helps break down the task.

Where will the ingredient nutrition info be stored?

- Option 1: Calculated on-the fly through an API
- Option 2: PL8 Database
- **Option 3: Combination of Options 1 and 2; cache commonly searched**

Comprehensive and accurate nutrition information APIs are available, which will be utilized. In addition, common calls will be cached and stored on PL8's database. This combination of the two options allows for quick and accurate information to be used, while not taking up a significant amount of memory space.

What service should we use for our database?

- **Option 1: NoSQL**
- Option 2: MySQL
- Option 3: SQL

NoSQL is the database system automatically implemented by Google App Engine's Datastore system. NoSQL is easy to set up and is very scalable. Using the Datastore, we can easily create properties, entities, keys, etc. It also allows us to easily query and index the database.

How to manage the database effectively for storing recipes?

- Option 1: Heap-based
- Option 2: Doubly linked list
- **Option 3: Combination of both**

Initially, we could use a doubly linked list to handle insertions and deletions in constant time. Then, after our data is collected, we could run it through a heap-based structure for sorting.

How should we handle scaling to different screen sizes?

- Option 1: Only worry about a few
- **Option 2: Bootstrap**
- Option 3: Account for mobile

Bootstrap allows scaling for all sizes. We will implement Bootstrap in our Angular templates in order for scaling on all web pages. This allows the website to be easily scaled on both desktop and mobile browsers.

CS 307
10
Team 21 Design Document

Design Details:

Classes:

User

- email
- userName
    - For comments and submissions
- password
- settings
    - Hold user settings, such as nutrition limits
- favoriteRecipes
    - Likely saved as a list of recipe IDs

Ingredient

- type
    - Is it a food? Drink?
- amount
    - How much is used?
- limits
    - What nutrients that could be limited are in this ingredient? Peanuts, sugars, etc
- nutrition
    - Nutrition information that the user would like to see e.g. calories, sugars, etc

Recipe

- ingredients [ ]
- nutritionInfo
- user
- directions
- mealType
    - Is it a main course? Drink? Etc
- idNumber
    - Unique identifier for recipe
- Comment

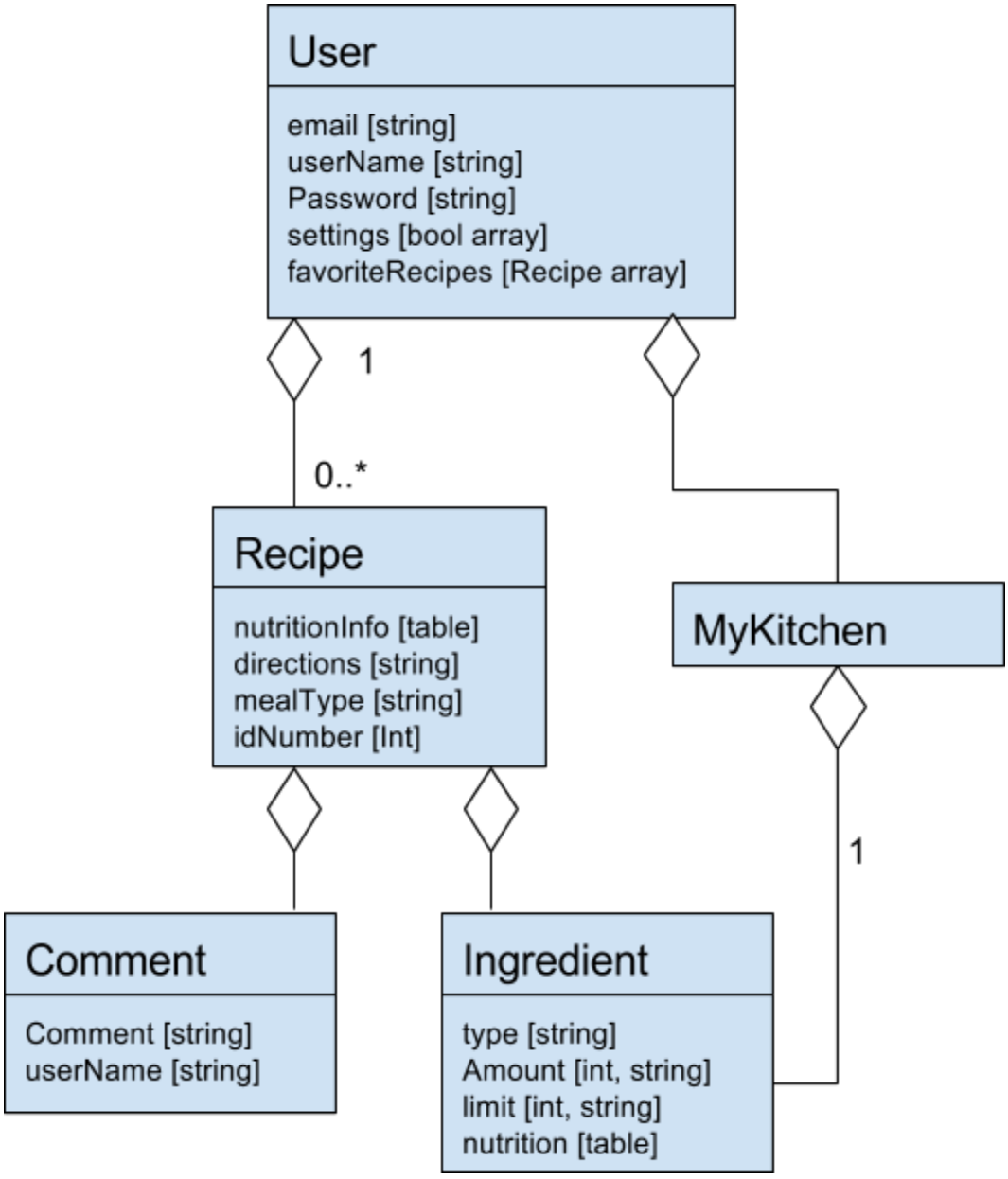
Comment

- User.userName
- commentString

MyKitchen

- Ingredients[]

Class Relations:



Framework Details:

- The design will be built using the Angular2 web framework on the Google App Engine platform.
- Components - Components control views. These are accompanied with view templates to tell Angular how to render the component. There will also be metadata within the component to tell Angular how to process the class.
- There will be a component controlling each web page. These components will be defined in a class specific to each page or view we want to have.