Group 2: FoodWeb

Deliverable 3: Analysis

Nazhem Jones, Zachary DeHaven, Tyler Lockwood, Dylan Anderson, Ethan Green, Eduardo Martinez

System Description

The system we are developing is an app, called "foodWeb", which will simplify searching and sharing recipes. The problem of the inefficient organization of recipes negatively affects those looking to plan their meals or cook; the impact of which is inefficient time spent looking for recipes and money spent on unnecessary ingredient.

For those who are actively looking for new **recipes**, FoodWeb is a meal organization app that allows the user to find, organize, and share **recipes**. Unlike cookbooks, Google, or social media, our product focuses on an ever-expanding library of recipes as the main content source the user interacts with.

Our app will allow **users** to *search*, save/favorite **recipes**, download and upload **recipes**, and register **user** profiles in a timely manner. The **user** can search via text/or *category* and find a **collection** of relevant **recipes**. The **user** can save **recipes** by selecting a save option that will store them in their **saved list**, and display them in a list format. They are then able to manage their **saved list** and organize them.

A user can upload their own **recipes** for other **users** to view and use. When uploading, the **user** will be able to upload a *title* for their **recipe**, a *category*, an **ingredient** list, a written cooking *procedure*, *images*, and a *video* of their recipe. The uploaded **recipe** will be linked directly to the **user** who created it. Once the upload is made it will be visible in the **recipe library**.

A **highlights** feed will be shown to the user derived from the **recipe library** based on their *preferences*.

When a user chooses to save a recipe it is stored under their **saved list** and will be accessible to them only. When a **user** selects their **saved list** they will be able to view their saved **recipes** and organize them in different folders to their preference. The user will be able to explore **recipes** through the home page. From there they can navigate to publishing a recipe to the main feed where other users can view them. The **recipe library** will be a large database of **recipes** that users can constantly add to. The **recipe library** can be filtered through user *preferences*.

With the ability to search and save **recipes** the user will be able to organize their **recipes** by day, *week*, *month*, and *meal time* in their **planner**.

The **user** will be able to track and manage their **ingredients** in their **pantry**.

Moderators will be able to flag recipes, flag **user** accounts, remove *flags*, and remove already flagged **recipes** from the **recipe library**. A moderator is a user with additional *permissions*. Admins will be able to remove **user** accounts, create **moderators**, and change the *permissions* of **moderators**. An admin is a user with specialized *permissions*.

Model

User

+ registerAccount()

Main Feed

+ field: Type

+ searchByName()
+ searchByIngredient()
+ searchByCategory()

Moderator

- permissions

- flagRecipe()
- flagAccount()
- unflagRecipe()
- removeFlaggedRecipe()

Saved List

+ addRecipe()
+ removeRecipe()

Recipe

+ name
+ author
+ category
+ description
+ procedure
+ video
+ flag

+ editName()
+ editIngredients()
+ editInstructions()
+ editDescription()

Ingredient

+ name
+ amount
+ type

+ updateIngredientAmt()

Pantry

+ checkIngredients()
+ addIngredient()
+ removeIngredient()

Highlights Feed

+ category preferences
+ popular recipes

- checkPopularity()
+ updatePreferences()

Registered User

+ username
- password
+ bio
+ picture
+ preferences

- changePassword()
- changeUserName()
- changePicture()
- editBio()
- publishRecipe()

Meal Planner

+ week
+ day
+ breakfast recipe
+ lunch recipe
+ dinner recipe

- addMeal()
- removeMeal()
- editMeal()

Admin

- permissions

- changePermissions()
- addModerator()
- removeAccount()
- unflag()

<extends>
<includes>