GrumpyChef

TEAM “MASTER CHEF”

SUBIKSHA VAIDHYANATHAN, KYLE COX, EMMANUEL GUTIERREZ RIVERA

Definitions

Priority (P) – 1 to 10, 1 being lowest.

Volatility (V) – 1 to 10, 1 being lowest.

Functional (F) – i.e. what should the system do.

Non-functional (NF) – i.e. how well does the system work.

Standalone (S) – i.e. does not depend on another requirement.

Dependent (D) – i.e. depends on another requirement.

Time (T) – measured in weeks.

Requirement 1: User Accounts

The customer will be able to sign up for an account with a username and password that is stored securely. Once logging into their account, the user will have their own recipes, pantry, and other user data. This will be displayed to them, can be edited or changed, and these changes will save to their account through our Firebase database.

Format: name, lastname, username, password (hashed), pantryitems, userrecipes

Example: John,Doe,JohnDoe1234,######,[cheese,bread,salt,pepper],[grilled cheese]

Requirement 1: User Accounts

Priority: 10

Volatility: 5

Functional or Non-functional: F

Standalone or Dependent: D

T: 2-3 weeks

Developer: Emmanuel Gutierrez Rivera

Tester: Entire team

Requirement 1 Test Cases

1. Test with an empty user.
2. Test with creating new, valid users.
3. Test with creating account with a weak password (length, special characters, etc.).
4. Test creating an account with an existing username.
5. Test with trying to log into an existing account with wrong password/username.
6. Test with logging into an account that already exists and see if user data persists.
7. Test if the user stays logs in when navigating the website.

Requirement 2: Pantry

The user will be able to have a "pantry" where they can add certain ingredients and groceries that they have within their house. The pantry interacts with the API of a major grocery store to conveniently show images of the item. The user will hopefully be able to identify how much they have left of a certain item – for example, if they have 5 eggs left in a cartridge. The user is able to add their own images as well if needed, so they could identify where in their pantry an item is.

Format: Username, [(Item1, Picture1, ItemCount1, Unit1) (Item2, Picture2, ItemCount2, Unit2), ...]

Example: JohnDoe1234, [(Eggs,EGG_IMG,11,individual), (Bread,BREAD_IMG,2, loaves), (Milk,MILK_IMG,2,gallons)]

Requirement 2: Pantry

Priority: 9

Volatility: 4

Functional or Non-functional: F

Standalone or Dependent: D, depends on the user system being completed

T: 2-weeks for primary function, 1 or 2-weeks for API connection/bigger functions

Developer: Kyle Cox

Tester: Entire team

Requirement 2 Test Cases

1. Test adding an empty ingredient.
2. Test adding an invalid ingredient.
3. Test adding an ingredient that is not in the API.
4. Test removing an ingredient that is not in the pantry.
5. Test removing more of an ingredient that is currently in the pantry.

Requirement 3: Recipe Sharing

The user will be able to create recipes, listing the ingredients and instructions to make the recipe and, also set timers that shows on how long each step takes to be completed. The user will be able to share these recipes and take others’ shared recipes. This system will integrate with the pantry to identify the ingredients the user already has, and if there are any missing items from the user's pantry, those will get added to the “cart” so that the user can buy the missing ingredient.

Format: RecipeOwner (username), Name, Picture, Instructions, [Ingredients()]

Example: JohnDoe1234, Grilled Cheese, GRILLEDCHEESE_IMG, <Instructions>, [(Cheese,CHEESE_IMG,1,slice), (Bread,BREAD_IMG,2,slices)]

Requirement 3: Recipe Sharing

Priority: 6

Volatility: 9

Functional or Non-functional: F

Standalone or Dependent: D as it depends on both the user system AND the pantry system

T: 3-weeks

Developer: Subiksha Vaidhyanathan

Tester: Entire team

Requirement 3 Test Cases

1. Test creating a new recipe.
2. Test creating an invalid recipe with only step (no ingredients).
3. Test creating an invalid recipe with no name and no instructions (only ingredients).
4. Test creating an invalid recipe (no ingredients or no name).
5. Test creating a duplicate of a recipe.
6. Test sharing and/or receiving recipes that are invalid.