SSH – Fridge Contents Tracker

Engineering Design Review

Author: Thomas Nourish - 2555422
Date: 31 October 2024
Status: Complete

Introduction

Many students find that cooking can be expensive, mainly due to the cost of individual ingredients. Additionally, cooking for just one person can lead to a lot of waste, causing potential harm to the environment. To address these challenges, it is important to optimise the use of ingredients. Student households who use SSH have a selection of products available to them, one of which is the SSH Camera. The SSH Camera can provide information about the contents of the fridge, including the quantities, to the residents. It can often be difficult to find recipes using only certain ingredients, especially given the busy lifestyle of students, while maintaining a healthy and varied diet.

We propose building a fridge contents tracker using the SSH Camera, making it easier for students to plan meals while saving money on new ingredients. The tracker will determine what ingredients are available in the fridge, including quantities, and use this information to suggest possible recipes using those ingredients. This plan will require a database of recipes, and it will be able to make suggestions where most ingredients are available and only some need to be purchased. We plan to add a feature where the user can input the best before date of each product when it is placed in the fridge, with the plan to suggest recipes that will use the ingredients soonest to go off. Another feature of this tracker would be allowing users to suggest new recipes to be added to the recipe database, hopefully further decreasing waste. This tracker aligns with the SSH's goal to ease the busy schedule of students without making any permanent modifications to the accommodation.

Goals and non-goals

- **Goal:** Use the data provided by the SSH camera to ensure that 50% less ingredients are wasted in student households through recipe suggestions.
- **Goal:** To allow users to add their own recipe suggestions to the recipe database. This is hoped to further reduce wasted ingredients.
- **Non-goal:** Allow the user to buy ingredients through the program.
- **Non-goal:** Automatic recipe creation, instead the program aims to match the ingredients available with pre-made recipes in the database table.

Design Overview

The new fridge contents tracker page will primarily consist of a table with a row for each recipe which uses at least one of the ingredients that the SSH Camera has determined is in the fridge. Each row will contain information about the recipe, including the name, a description and information about the ingredients used:

- The total number of ingredients needed for the recipe.

- The number of ingredients needed for the recipe which are already available in the fridge.
- The estimated cost of the remaining ingredients which are not available in the fridge.

Each row of recipes will, by default, be sorted by the estimated cost of the remaining ingredients, with the option for the user to sort the recipes by either the number of ingredients or the number of ingredients already available. The user will be able to click on the name of any recipe and this will bring them to a pop-up page where they are given the step-by-step instructions.

The information on the contents of the fridge will be automatically updated by the SSH Camera when something is placed in or removed from the fridge. The estimated cost of ingredients will be the sum of the estimated cost of each ingredient which must be purchased. These estimated prices will be updated weekly with the average price of the ingredient across the leading supermarkets in the UK, if the user believes that the prices are incorrect or outdated, they will have the option to manually refresh the pricing.

In the Backend, there will be a collection of database tables storing all the data required. These are demonstrated in the table below:

Table	**Relevant fields**	**Relevance**
`recipes`	`recipe_id`, `recipe_name`, `recipe_instructions`	Stores each recipe (represented by recipe_id) with the name of the recipe and instructions.
`ingredients`	`ingredient_id`, `ingredient_name`, `quantity_available`, `cost_per_kg`	Stores a list of all ingredients (ingredient_id) that are included in any recipe in the recipes table or are in the fridge with the name, the quantity available in the fridge, and the cost per kg.
`recipe_ingredients`	`recipe_id`, `ingredient_id`, `quantity_needed`, `total_cost`	Using foreign keys, this table links each recipe to the list of ingredients needed for the given recipe, including the quantity needed and the total cost of the ingredients that are not available in the fridge.
`categories`	`category_id`, `category_name`	This table is used to store the various categories of recipes, such as the meal it is (breakfast, lunch, …) or dietary requirements (vegan, gluten free, …)
`recipe_categories`	`recipe_id`, `category_id`	This table links the recipes to the categories that it is a part of. We need this table as a recipe could come under multiple categories.

There will be a many-to-many relationship between recipes and ingredients, this is because a recipe will have multiple ingredients, and an ingredient can be used in multiple recipes.

SSH already has the table with information about the contents of the fridge. This is currently displayed to the user through either the SSH App or through the SSH Console Table. We plan to use this table to carry out some basic calculations to figure out if enough of each ingredient is available for the recipes in the list and if not then how much will need to be spent on the new ingredients.

To add recipes for to the recipes database, the user will be able to scroll to the bottom of the recipes page and press a "new recipe" button, this will create a pop-up page with fields for the recipe name, ingredients, the category (if it applies) and the step-by-step instructions. This will add this recipe to the database locally, allowing the user to repeat this recipe with ease in the future.

We also plan to add filters at the top of the recipes page to allow the user to filter the recipes by categories, for example, if a student is vegan, they can choose to only have vegan recipes shown in the list.

Alternatives

User inputted price estimates:

- This was a potential idea for the pricing of ingredients, where the user would have to manually input the prices of ingredients in the recipes.
- Pros:
    - Users can tailor prices to their own spending, for example, if someone shops at Aldi, they will likely pay less for some ingredients than compared to other supermarkets.
    - Minimal maintenance needed on the prices.
    - Less dependant on external data sources.
- Cons:
    - May require users to input data regularly.
    - May result in inaccurate prices should the user not update it frequently, for example over the summer holidays.

Recipe importing from Web URLs:

- This would allow users to find recipes online and easily import them through a simple copy-paste action.
- Pros:
    - Convenient for users who will need less data entry effort to add their own recipes.
    - Encourages a more diverse selection of recipes from external sources.
- Cons:
    - This would require very complex programming and maintenance to parse and format the recipes correctly.
    - Dependant on external sites for content, could be formatted inconsistently or incorrectly written.

Card-based layout of recipes:

- Display an initial page with cards displaying filters, which on clicking leads to a new page of recipes containing images which come under the initial filters.
- Pros:
    - Provides a visually engaging interface.
    - Easier for users to get an idea of what their dish should look like when completed.
- Cons:
    - Limited space for displaying details about the recipe.
    - Users may not see recipes where they can use more of the ingredients that are already available to them, potentially creating more wasted food.

Milestones

Milestone 1: Begin implementing the backend by setting up the basic database tables with the various primary and foreign keys and linking the required fields. Using SQL we should be able to run some basic tests to ensure that the databases are correctly set up.

Milestone 2: Integrate the SSH Camera's information into the database and ensure that it automatically updates whenever a change is made to the contents of the fridge.

Milestone 3: Develop the cost estimation tool that calculates the expected price of the remaining ingredients using average prices from a database. We can further develop this if required.

Milestone 4: Allow the design team to come up with a design for the new page, containing fields for all required information about each recipe.

Milestone 5: Allow the UI team to create the frontend based on the design from Milestone 4.

Milestone 6: Run intensive testing through a beta release to select users until enough feedback is collected. If required, make tweaks to the program.

*Milestone 7: R*elease the product to all users of the SSH Camera.

Dependencies

- *UI team:* design the recipes page with plenty of space for all information that may be required about a recipe by a glance. Update the SSH Cloud homepage to ensure that the new tool can be accessed correctly.
- *Database team:* Create all the required tables and ensure that all keys and links are carried out correctly. Fill the databases with the required data, such as the provided recipes and the cost of ingredients.
- *Quality assurance team:* ensure that customers are happy with the quality of the new product throughout the beta and after the full release.
- *Legal Team:* ensure that the new product does not break any of the rules in the user agreement of SSH.

Cost

We do not expect a large increase in operating costs as a result of the release of the new feature. The main cost of running will operate over the SSH Cloud when the user places or removes something from the fridge. If this performs poorly then we can implement a less frequent update to the contents of the fridge.

Privacy and security concerns

There is no new data about the users that is inputted into this new feature. As most of the backend happens locally, there is no concern about any new weakness or potential for a leak/breach. The only place that could be a cause for concern would be if we made the feature of adding new recipes into a feature where you can upload recipes for all users of SSH to see. This would require much more moderation as someone could fill in the fields with anything, not only information about a recipe. However, since this is currently not a planned feature there is no extra concern security wise.

Risks

Risk	**Mitigation**
Students will not use the new recipe suggestion tool.	The new feature could be advertised on the SSH Cloud and the SSH App to all users who own an SSH Camera.
Taking or adding ingredients to the fridge while cooking a recipe could confuse the program and could provide incorrect amounts or costs in the recipe	We could implement a feature to only update the contents of the fridge every so often rather than every time the fridge is opened.
Reshuffling the homepage of the SSH cloud to ensure this feature is reachable could make it more difficult to locate other features of SSH.	If this becomes a problem, we can think of a new design for the SSH cloud homepage.

Supporting material

Intentionally left blank