Fridge Manager: Requirements, Vision, and Scope

Product Vision Statement

Motivation / Opportunity

Our goal is to implement an android app to solve the problem of not knowing what food is currently at home and when it expires. In addition, our app should solve the problem of a missing overview of the spendings and food consumption. The advantage of our product over pre-existing apps is that users are able to scan the grocery bill with their smartphone and the app will provide them with a list of all foods the users purchased and automatically add them to their local food database.

Problem Statement

The problem of	Reduce the food waste caused by expired food
affects	Consumers looking to reduce waste and maintain an overview of spendings on groceries without having to spend time on tracking grocery purchases.
The impact of which is	Solves the problem of food waste by reminding users of the expiry date for each product. An example case would be meat that spoils quickly and is expensive which might have been forgotten by the user.
A successful solution would be	A program which gives you an overview of your groceries currently at home, in the fridge or pantry and then provides information on nutritional value and sets reminders for the expiry dates of the said items.

Product Position Statements

For	All consumers who make regular grocery purchases.
Who	Leave food in the fridge/pantry without being aware of whether it is still in good state.
Our System	Android app
That	Automatically scans receipts, adds food to virtual fridge manager, and reminds the user of food about to expire.
Unlike	Other fridge manager apps like fresh box, fridge pal, best before or grocery hero.
Our Product	Allows users to scan receipts and automatically adds the food products and provides updates to expiry date

	information on their current state.

Users: The user of our app is everybody who wants to have a clear overview over his spendings and the currently food stock.
Feature List:
- Ability to take a picture of grocery receipts and recognize and analyze text
- Search through database for expiry date of food item and input groceries onto list and display expiry date through UI
- Visual interface of fridge; allows user to add and remove groceries from fridge
- Ability to remove items and manually add items
- Categorize items by physical location they might be placed (items that might go in the fridge, freezer, or pantry) or food type (fish, poultry, beef, vegetables, etc.)
- Ability to track weekly/monthly spendings on food

Constraints: The app has to be running on major Android devices and versions with in-built camera recognition features.
Scope and Limitations: The app doesn't provide any recipe suggestions for the food in the food stock or suggestions which food should be bought. It also does not provide the ability to scan barcodes and provide information about the item.
Assumptions and Dependencies: We expect that the text recognition service, Google Mobile Vision, allow us to scan grocery bills and output text, including blocks, lines and words.

Use Cases

1. **Analyze and Read Text from Receipt**
    a. Primary Actor : End user
    b. Stakeholders : End user
    c. Precondition : The app must be running on an android-device with camera features
    d. Postcondition : Items on the receipt will be recorded on the App and will be displayed to the user
    e. Main Success Scenario :
        1. User takes a picture of the receipt
        2. App receives the image and sends it to text recognition service, Google Mobile Vision
        3. Google Mobile Vision processes the image to retrieve a series of text on the image
        4. App receives the series of text within the receipt and identifies texts that are related to food items
        5. App displays the list of food items it was able to identify from the receipt
        6. User confirms that recongnized items are linked with the right food items.
    f. Extensions and Alternative Flows :
        1. Image was unreadable
            1. App prompts the user to make a picture again or manually enter items
        2. Text was unrecognized
            1. App prompts the user which items it couldn't recongnize.

        2. User has to enter the corresponding food manually.
        3. App saves the llink between unknown item and entered name
    3. Corresponding food item already exists in the local food database
        1. App alerts the user of the duplicate
        2. App prompts the user to enter another food item or tag numerical extension to the food (eg. apple(2))
  g. Open Issues : Compatibility with cameras on different phones

2. **Search expiry date database for expiry date**
    a. Primary Actor : App
    b. Stakeholders : End user
    c. Precondition : The database works, and contains expiry date for each food item
    d. Postcondition : Gets the expiry date for a certain food item
    e. Main Success Scenario :
        1. App access the expiry data database
        2. App searches through the database for expiry date corresponding to specified food item
        3. App retrieves the expiry date from database and mapps it to specifed food
        4. User confirms that the found expiry dates are correct
    f. Extensions and Alternative Flows :
        1. Finds the wrong expiry date for the food item
            1. User enters exipry date manually

3. **Manually Add Food to Food Item Database**
    a. Primary Actor : End user
    b. Stakeholders : End user
    c. Precondition : Food item database is intact and accessible
    d. Postcondition : Food item is added into food item database, and can be viewed by the end user through the interface
    e. Main Success Scenario :
        1. User selects a food from dropdown list or manually enters a food name
        2. App verifies the name so that it contains valid characters
        3. App adds the food item to the food item database
        4. App retrieves expiry date of the food item if exists, and links it with the food
        5. App refreshes its user interface so the new changes to the database is visible to user
    f. Extensions and Alternative Flows
        1. The food item already exists in the food item database
            1. App increases the quantity of the food item
            2. App retrieves expiry date if exists, and links it with the food

4. **Manually enter expiry date of food item**
    a. Primary Actor : End user
    b. Stakeholders : End user
    c. Precondition : Food item is already added to fridge database
    d. Postcondition : The corresponding expiry date for the food item is added

e. Main Success Scenario :
    1. User selects a food from drop downlist.
    2. User manually enter the expiry date for that food item.
    3. App verifies that the date is a valid day
    4. App links the entered expiry date with the food item
    5. App adds the expiry date specified from the user to the corresponding food, and the updates new change on the user interface

f. Extensions and Alternative Flows :
    1. The expiry date is in the past
        1. App alerts the user of past expiry date
        2. App prompts the user to allow automatic expiry date reset (expiry countdown from present date) or have the user input new expiry date
    2. Expiry date is unknown by user and the app
        1. Food is assigned with “Unknown exipry”
        2. Food is additionally stored in “unknown exipry date products”
        3. The User can enter this list via the front menu

5. **Remove Food from Food Item Database**
    a. Primary Actor : End user
    b. Stakeholders : End user
    c. Precondition : Food item database is functional,accessible and not empty
    d. Postcondition : Food item gets removed from the database along with all its relevant information
    e. Main Success Scenario :
        1. User enters the foodstock
        2. User removes a food item listed in the foodstock
        3. App removes the food item from the database
    f. Extensions and Alternative Flows :
        1. User accidentally removes the wrong food
            1. User can press the Undo button to undo last action

6. **Alert user of food about to expire**
    a. Primary Actor : App
    b. Stakeholders : End user
    c. Precondition : A food item is about to reach its expiry date in 3 days.
    d. Postcondition : User receives alert that certain food item is about to expire
    e. Main Success Scenario :
        1. App finds in the database a food item that expires in 3 days.
        2. App displays a push message to the user.
        3. The user confirms the message with the OK button.
    f. Extensions and Alternative Flows
        1. The smartphone was off, when the message should have been displayed and the food already expired.
            1. The user gets informed when the food item expired once the phone is turned back on

Non-functional Requirements

Performance requirements:

- Finishes processing each photo and adding food items under 2 seconds
- Any additional buttons (eg. Undo) should have an instantaneous-like (0.1s) response time

Software Quality Attributes:

- Ease of use over ease of learning
- Reliability, availability, scalability, and recoverability (RASR) due to database usage

Security Requirement:

- Personal information from receipts (store location/credit card information) processing should be kept private to users only

User Demographics

End Users: Only Android users will have access to this mobile app. They are expected to have experience with an android UI. They aren't expected to have any programming skills.