Lab 2 – LivelyShelfs Product Specification

Jaylen Wheeler, Roman Pena, Asia Gillispie, Colin Thoe,

Andrew Benham, Michael Nguyen, Chase Murray, Hadeel Qaiymah

Old Dominion University

CS411W

Dr. Sarah Hosni

April 18, 2025

Version 2

Table of Contents

List of Tables

[This page intentionally left blank]

3 Product Requirements

3.1 Functional Requirements

3.1.1 **User Interfaces** (O: Wheeler)

The system shall provide separate pages that can be accessed by the user. These pages shall allow users to complete multiple objectives.

3.1.1.1 User Authentication

The system shall provide user interfaces for login and account registration. The system shall allow existing users to access their account after opening the LivelyShelfs application. The system shall allow new users to create an account after clicking on the dedicated registration button on the login page.

3.1.1.1.1 **Login Page** (O: Wheeler)

Upon launch of the LivelyShelfs application, the user will be directed to the login page which shall include the following requirements that coincide with the Login algorithm (3.1.2.1):

- **LP1**: The welcome box shall include a greeting that includes the LivelyShelfs name.
- **LP2**: The login box shall include the ability to login to an existing account
- **LP3**: The “Register Now” button below the login box shall redirect the user to the registration page (3.1.1.1.2)
- **LP4**: The screen shall display a login form that will take input from the user.
- **LP5**: The login form shall contain input fields for the username and password.
- **LP6**: The system shall submit the form data upon selection of the “Login” button.

- **LP7:** The system shall display an authentication error message when the form receives a username and password combination that is not within the database.
- **LP8**: The system shall redirect the user to the dashboard after the login form receives a username and password combination that is within the database (3.1.1.2).

3.1.1.1.2 Registration Page (O: Wheeler)

Upon clicking on the “Register Now” button at the bottom of the login page, the user will be directed to the registration page which shall include the following requirements that coincide with the registration algorithm (3.1.2.2):

- **RP1:** The screen shall display a registration form that will take input from the user.
- **RP2:** The registration form shall contain input fields for the first name, last name, email, username, password, and a confirmation password.
- **RP3:** The system shall display an error message when the form receives a preexisting username within the database.
- **RP4:** The system shall display an error message when the form receives a password and confirmation password that are not the same.
- **RP5:** The system shall store the account credentials in the database after the registration form receives a valid password combination and unique username.
- **RP6:** The system shall redirect the user to the login page after successfully registering an account by clicking the “Register” button (3.1.1.1.1).

3.1.1.2 Dashboard

Upon redirection from the login page after clicking the “Login” button or after clicking the “Dashboard” button in the navigation bar, the user will shall be able to view detailed graphs

regarding their food usage that coincide with Predictive Waste and Trend Analysis algorithms (3.1.1.6, 3.1.2.7, & 3.1.2.8).

3.1.1.2.1 **Elements** (O: Wheeler)

- The screen shall display a greeting that utilizes the user's first and last name at the top of the page.
- The screen shall display the email that is attached to the user's account under the greeting.
- The screen shall display a logout button at the bottom of the screen.
- The system shall logout the user and redirect them to the login page upon selecting the logout button (3.1.1.1.1).

3.1.1.2.2 **Trend Graphs** (O: Murray)

The system shall allow users to view trend graphs related to their food usage and food waste data. The following trend graphs will be found on the user's dashboard:

- Spoilage statistics will be recorded each month and displayed as a line graph showcasing the user's yearly food wastage.
- Food item wastage will be recorded and displayed to the user as a pie chart showcasing the top five most wasted food items.

3.1.1.3 Inventory

(O: Gillispie)

The system shall display the user's inventory page upon clicking on the "inventory" button in the navigation bar. The inventory page will allow the user to access their calendar and list view along with the ability to add and remove food items into their inventory.

3.1.1.3.1 Calendar / List

CL1: (O: Gillispie)

The system shall display a calendar and list view element within the center of the Inventory page under the calendar heading.

- The calendar view shall always be the default view when loading the page.
- At the top of the calendar and list view element is the toolbar:
    - The left arrow button on the left side of the tool will navigate to the previous month each time it is clicked.
    - The right arrow button to the left of the left arrow button will navigate to the next month each time it is clicked.
    - The today button to the right of the right arrow button will display the current month view when it is clicked.
    - The name of the month and the year of the month's view that user is currently viewing will display as text in the middle of the tool bar.
    - On the far right of the tool bar is the month and list buttons. The month button on the left will show the calendar view upon being clicked and the list button on the right will show the list view upon being clicked.
- The calendar and list view shall display the same inventory information with different formatting.
    - The calendar view will display a calendar grid where food item information is formatted as events within the grid cells. Only the name of the food item expiring will be displayed as an event. Hovering the mouse over an event will display the food item's name and information.

- The list view will display food item information in a list format. Only food items with expiration dates within a month's view will appear on the list.

CL2: (O: Gillispie)

The system shall display the quantity of each food item within the calendar and list view where the food item's information is displayed.

- Within the calendar view, the quantity will be displayed next to the associated food item which are color coded based on the item's time remaining till the expiration date within the calendar grid.
- Within the list view, the quantity will be displayed next to the associated food item that are listed which are color coded based on the time remaining till the food item's expiration date.

The quantity is based on the data that was logged or updated in the inventory.

CL3: (O: Pena)

The system shall track the expiration status of food determined by the use by date when entering an item in inventory or a shelf life estimated by the system depending on the type of food (e.g. fresh produce).

- Upon selecting a food item in the calendar or list view via clicking on or hovering over the item, the system shall display the food status which will be either fresh or spoiled. If the food item's status is fresh the time until the food item is spoiled will also be displayed.
- The system shall send a text prompt notification if a good item is expiring within the week. The notification will be visible in the user's notification drawer (3.1.1.5.1) component visible via button click on the dashboard and inventory pages.

CL4: (O: Pena)

The system shall color code food items in the calendar and list depending on how much time remains until it is spoiled.

- The system shall color items red if they are close to expiration (within 1-3 days).
- The system shall color items yellow if they expire soon (within 7-4 days).
- The system shall color items grey and struck-through if they are expired.
- The system shall color all other food items that are fresh green.

3.1.1.3.2 Forms

F1: Addition (O: Nguyen)

The system shall allow users to add food items from their inventory by specifying the item name, quantity, and purchase date.

F2: Removal (O: Nguyen)

The system shall allow users to remove food items from their inventory by selecting the item from a displayed list of current inventory items. After selecting the item, users can specify the quantity to be removed. The system will then update the inventory accordingly.

- When users click the "Consumed/Spoiled Form" button, this will open a "Consumed/Spoiled Form," enabling the user to manually deduct quantities for items that are consumed.
- Users will select the food item and quantity where the system shall provide an option to mark items as expired, allowing users to remove them from the inventory.

F3: Consumption / Spoilage (O: Wheeler)

- The screen shall display a consumption / spoilage form that will take input from the user after clicking the “Consumption / Spoilage” form button displayed beside the calendar (3.1.1.3.1).
- The consumption / spoilage form will have a drop down menu with selectable items as input.
- The system shall allow users to select either consumed or spoiled as a checked box.
- The system shall update the food item within the database after the form has been submitted by clicking the “Submit” button at the bottom of the form.

3.1.1.4 Shelf Friends

Upon clicking the “Shelf Friends” button in the navigation bar, the user shall be redirected to the Shelf Friends page that will allow them to manage their Shelf Friends and shareable food (3.1.1.6).

3.1.1.4.1 **Friend Management** (O: Wheeler)

- **FM1:** The screen shall display the user’s Shelf Friends in a list at the top of the page with a message and remove button beside each entry**.**
- **FM2:** Upon selection of the message button, the user shall be redirected to a dedicated messaging page between themselves and the selected Shelf Friend (3.1.1.4.2).
- **FM3:** Upon selection of the remove button, the system shall remove the stored connection between the user and the selected Shelf Friend.
- **FM4:** The screen shall display a friend addition form that will take input from the user.

- **FM5:** The friend addition form shall contain an input field for the friend ID.
- **FM6:** The system shall display an error message if a friend ID that is not present in the database is submitted to the form.

3.1.1.4.2 Messaging (O: Wheeler)

- **M1:** The screen shall display a message box that will allow the user to message a Shelf Friend.
- **M2:** The system shall display previous messages above the message input field at the bottom of the message box.
- **M3:** Upon submission of the message through pressing the “Send” button, the message shall be stored in the database**.**

3.1.1.4.3 Sharing

(O: Wheeler)

- **S1:** The system shall provide the ability to mark items as shareable by clicking the “Share” button beside a displayed food item on the page.
- **S2:** The system shall provide the ability to unmark items as shareable by removing the shareable status from an item after clicking the “Cancel” button beside a displayed food item.

(O: Benham)

- **S3:** The system will allow the user to make requests for food items that their Shelf Friends have set to shareable by pressing the button next to the Shelf Friends food item.

- **S4:** The system shall transfer ownership of the shared items to the receiving shelf-friend's inventory when the user presses the accept button next to a Shelf Friends request for that specific food item.

3.1.1.5 **Notifications** (O: Gillispie)

3.1.1.5.1 The system shall display a notifications button that is labeled "← Notifications" within the Dashboard, Inventory, Shelf Friends, and Settings pages. The notifications button will be located at the top of the page slightly to the left.

3.1.1.5.2 The system shall display the user's notifications drawer upon clicking the notifications button and the drawer will do the following:

- **N1:** The notification drawer shall slide in from the right at a size of less than half of the width of the page.
- **N2:** When the notification drawer slides in from the right, the left side that is not the drawer will darken.
- **N3:** The user shall only be able to interact with the notification drawer while it is displayed and will not be able to interact with darken elements on the page.
- **N4:** When the notification drawer is being displayed, clicking on any part of the page other than the drawer's scroll bars will close the drawer and make all elements interactable again.

3.1.1.5.3 The system shall display the user's notifications within the notifications drawer and the notifications will be divided into three sections: Inventory notifications, Shelf Friends notifications, and Other notifications.

- **N5:** The system shall display Inventory notifications listed within the top box with the scroll wheels labeled “Inventory”. The Inventory notifications displayed are based on notifications related to the user’s inventory and are sourced from the following:
  - When a user’s food items are nearing their expiration date.
  - When a user adds a food item to their inventory that they tend to spoil (3.1.2.5.1).
  - When a user adds an item that they already have in their inventory.
- **N6:** The system shall display Shelf Friends notifications listed within the middle box with the scroll wheels labeled “Shelf Friends”. The Shelf Friends notifications are based off notifications related to the user’s Shelf Friends and are sourced from the following:
  - When another user messages the user that they want to share a food item.
  - When a user’s sharable food item is being requested by another user.
  - When another user accepts the user’s sharable food item request.
- **N7:** The system shall display Other notifications listed within the bottom box with the scroll wheels labeled “Other”. The Other notifications displayed are sourced from the following:
  - When a user receives their incentive (3.1.2.3).
  - When a user successfully changes account info in Settings (3.1.1.7).
- **N8:** The system shall update the Inventory notifications and Shelf Friends notifications displayed upon refreshing the page or navigating to another page.

3.1.1.6 **Navigation Bar** (O: Pena)

3.1.1.6.1 The system shall provide a sidebar with a link for each LivelyShelfs functionality: the dashboard (3.1.1.2 Dashboard), calendar and list (3.1.1.3.1 Calendar/List), Shelf Friends (3.1.1.4 Shelf Friends), and settings (3.1.17 Settings).

3.1.1.6.2 The system shall display a button for each major component with its own page. Upon clicking on the button, the user will be redirected to the corresponding page. For example, a button labeled “Inventory” (3.1.1.3 Inventory) will take the user to the inventory page.

3.1.1.7 Settings (O: Benham)

3.1.1.7.1 The system shall allow the user to update their information. It is a button on the Navigation bar that when pressed will take the user to the settings page which will allow them to change their name and email.

- **ST1:** Then submit a new email in the new email textbox and then apply the change by pressing the change button next to the text.
- **ST2:** When the user confirms the new email, the email will be updated in the database.
- **ST3:** The system will allow the user to change their first name by typing their new name in the change name textbox and then applying the change by pressing the apply button next to the textbox.
- **ST4:** The system will allow the user to change their last name by typing their new name in the change name textbox and then applying the change by pressing the apply button next to the textbox.

- **ST5:** When the user confirms the new last or first name, or both, the name will be updated in the database.

3.1.1.7.2 Notification Frequency

- **NF1:** The system shall allow the user to adjust the number of notifications that the user receives.
- **NF2:** The system shall allow the user to set the number of notifications they receive to none.
- **NF3:** The system shall allow the user to set the number of notifications that is displayed to the user to all notifications the app sends**.**

3.1.1.7.3 Account Management

- The system shall allow the user to delete their account**.**
- When the user deletes their account, all information pertaining to the user shall be deleted from the database, except for food analytical data.

3.1.2 **Algorithms** (O: Thoe)

3.1.2.1 **Login** (O: Wheeler)

- The system shall accept a login if the following conditions are met:
    - The submitted username exists within the database
    - The submitted password corresponds to the submitted username within the database

3.1.2.2 **Registration** (O: Wheeler)

- The system shall accept registration data if the following conditions are met:
    - The submitted username does not exist within the database

- The submitted password matches the submitted confirmation password

3.1.2.3 **Spoilage Data** (O: Thoe)

3.1.2.3.1 The system shall calculate and store the expiration date of an inputted food item's purchase date by adding the expiration period of the item to the purchase date to get the expiration date.

3.1.2.4 **Recommendations** (O: Thoe)

3.1.2.4.1 The system shall generate recipe suggestions by cross referencing the user's current inventory and popular recipe items, prioritizing inventory items near expiration.

3.1.2.4.2 The system shall analyze past user preferences and food usage patterns to refine recommendations.

3.1.2.5 **Incentives** (O: Thoe)

3.1.2.5.1 The system shall track the period over which the user has had an account.

3.1.2.5.2 The system shall provide visual feedback when an incentive is earned in the form of a notification.

3.1.2.5.3 The system shall send a notification for the first tier of incentives when the user has had an account for three months.

3.1.2.5.4 The system shall send a notification for the second tier of incentives when the user has had an account for one year.

3.1.2.6 **Notifications** (O: Thoe)

3.1.2.6.1 The system shall generate proactive notifications for users based on spoilage risks and recommended consumption timelines.

3.1.2.6.2 The system shall provide reminders for frequently wasted items, encouraging timely consumption.

3.1.2.6.3 The system shall send alerts when food-sharing opportunities arise with Shelf Friends.

3.1.2.7 Predictive Waste Analysis (O: Thoe)

3.1.2.7.1 The system shall analyze food usage trends before the user purchases items by calculating the most wasted items in the user's inventory, examining the frequency at which the user purchases the items, and then sending a notification to inform the user if it is likely to go to waste.

3.1.2.8 Trend Analysis (O: Thoe)

3.1.2.8.1 The system shall analyze long-term food usage trends by tracking the frequency at which items are wasted or consumed and then comparing that data to previous tracking periods.

3.1.2.8.2 The system shall calculate the most wasted and consumed items to use in pie charts and line graphs of the user's inventory habits.

3.1.2.9 Inventory Management (O: Thoe)

3.1.2.9.1 The system shall allow users to use manual input or camera input to add and remove food items within their inventory.

3.2 Performance Requirements (O: Nguyen)

3.2.1 The calendar shall not take longer than 20 seconds to update and display food items that were just manually inputted into the inventory.

3.2.2 The navigation across different pages will be updated and displayed, taking no longer than 3 seconds when clicked upon.

3.3 Assumptions and Restraints (O: Nguyen)

Table 1: Effects of Assumptions and Constraints on Requirements

Condition	Type	Effect on Requirements
Only entries containing valid food items will be accepted as legitimate data within the system.	Assumption	Reduced parameters for acceptable food inserted into data storage.
The database will be hosted on the local system where it will be accessed.	Restraint	The system will not require online connection to cloud storage as it is a local database.

3.4 Non-functional Requirements

3.4.1 Database Management and Security

3.4.1.1 Database (O: Thoe)

3.4.1.1.1 The data stored in the LivelyShelfs database will be encrypted.

3.4.1.1.2 Inventory items will be accessible only to the associated account.

3.4.1.2 Authentication (O: Thoe)

3.4.1.1.3 Accounts will be secured with a username and password.

3.4.2 Risk Management (O: Pena)

3.4.2.1.1 JSON web tokens are used to retrieve user specific data and will expire after 2 hours.

3.4.2.1.2 If users log into their account using a device separate from the one, they created the account with, they will have to authorize the device using email verification via the email attached to their account.