

Software Engineering Software Requirements Specification (SRS) Document

Ingredient Intel

May – 1 – 2024

V.1.0.0

Joseph Cramer, Justin Evans, Pradhy Kothapalli

By adding our names to this paper we acknowledge that the Academic Integrity Policy governs our academic activities. WE HAVE ABIDED BY THE UNCG ACADEMIC POLICY ON THIS ASSIGNMENT.

Table of Contents

1. Introduction.....	3
1.1. Purpose.....	3
1.2. Document Conventions.....	3
1.3. Definitions, Acronyms, and Abbreviations.....	3
1.4. Intended Audience.....	3
1.5. Project Scope.....	3
1.6. Technology Challenges.....	4
1.7. References.....	4
2. General Description.....	5
2.1. Product Features.....	5
2.2. User Class and Characteristics.....	5
2.3. Operating Environment.....	5
2.4. Constraints.....	5
2.5. Assumptions and Dependencies.....	5
3. Functional Requirements.....	6
3.1. Primary.....	6
3.2. Secondary.....	6
3.3. Use-Case Model.....	6
3.3.1. Use-Case Model Diagram.....	6
3.3.2.1. Actor: Admin (All).....	7
3.3.2.2. Actor: End User (Joseph).....	7
3.3.2.3. Actor: Company (Justin).....	8
3.3.2.4. Actor: Researcher (Pradhy).....	8
3.3.3. Use-Case Model Scenarios.....	8
3.3.3.1. Actor: Admin (All).....	8
3.3.3.2. Actor: End user (Joseph).....	9
3.3.3.3. Actor: Research Institution (Pradhy).....	9
3.3.3.4. Company(Justin).....	10
4. Technical Requirements.....	11
4.1. Interface Requirements.....	11
4.1.1. User Interfaces.....	11
4.1.2. Hardware Interfaces.....	11
4.1.3. Communications Interfaces.....	11
4.1.4. Software Interfaces.....	11

5.	Non-Functional Requirements	12
5.1.	Performance Requirements	12
5.2.	Safety Requirements	12
5.3.	Security Requirements	12
5.4.	Software Quality Attributes	12
5.4.1.	Availability	12
5.4.2.	Correctness.....	12
5.4.3.	Maintainability.....	12
5.4.4.	Usability.....	12
5.4.5.	Portability.....	12
5.5.	Process Requirements	12
5.5.1.	Development Process Used.....	12
5.5.2.	Time Constraints.....	13
5.5.3.	Cost and Delivery Date.....	13
5.6.	Other Requirements.....	13
6.	Design Documents	14
6.1.	Software Architecture	14
6.2.	High-Level Database Schema	14
6.3.	Software Design	15
6.3.1.	State Machine Diagram: Company (Justin Evans)	15
6.3.2.	State Machine Diagram: Researcher (Justin Evans).....	16
6.3.3.	State Machine Diagram: Customer (Justin Evans)	16
6.4.	UML Class Diagram	17
7.	Scenario.....	17
7.1.	Brief Written Scenario with Screenshots	17

1. Introduction

1.1. Purpose

The purpose of ingredient intel is to reduce the difficulty in reading, understanding, and extracting information from ingredient lists on food products. By doing this we additionally hope to make consumers more knowledgeable about what they eat and more healthy because of that.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the client-side and developer-view requirements for the Ingredient Intel web and android App. In it we will detail the requirements for a successful client experience and the requirements on the back end and for the developer-view in order for the service to run smoothly and require little technical knowledge to use. These requirements will include a description of the different user types and their levels of access and services provided to them. They will also include a system wide description of the software, database, and framework requirements and the metrics to determine software success such as performance, safety, and functionality.

1.3. Definitions, Acronyms, and Abbreviations

PostgreSQL	Open-source relational database management system.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
Django	A web framework that will be used with python to direct web traffic, interact with our database, and generally control website operations.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
VS Code	Our development environment where our code will be written for the website, backend, and database, and the interactions between them.
Python	An open source scripting language that will be used to write our application.
API	Application Programming Interface. We will use one to allow developer interaction with our database and will sync our database with the FDA's API for recalls, updates on ingredients, etc.

1.4. Intended Audience

The intended audience for the whole of this SRD document includes our development team, the UNCG CS department, the US FDA and FDC, and other interested computer scientists. The functional requirements and general description are intended for any users with companies and researchers as the main target.

1.5. Project Scope

The goal of the software is to provide a simple user interface to access a robust database which will provide a wide range of information on ingredients and products for companies, researchers, and normal users. This will benefit the goals of the government in providing transparency for safety information of ingredients reducing operational costs and difficulty of regulating food safety. Additionally, it will also serve as a service for consumers helping them stay informed about safety risks of ingredients, recalls of food items, and the inclusion of certain ingredients that may be allergens or high risk for certain individuals.

Itemized the benefits are:

- For Business
 - Increased customer satisfaction
 - Reduced negative health impacts of products
 - Ease of compliance with regulation and transparency with the government to obviate potential issues
- For Government
 - Reduced regulatory costs
 - Increased regulatory effectiveness
 - Increased consumer awareness about products and ingredients
 - Fast reacting and simple system for communication with consumers and companies
- For Consumers
 - Time saving when shopping for dietary restrictions
 - Reduced confusion with product ingredients, health effects, and recalls
 - Increased ability for people with dietary restrictions to try new products
- For Researchers
 - Increased ability for new findings to be communicated to consumers
 - Simplified systematic process for updating food safety information
 - Simple access to previous research about food ingredients
 - Simple access to product database which would allow for better data for correlation or impact studies

1.6. Technology Challenges

There will not be any challenges on the hardware technology side of this product. Some limitations may be put in place to increase access for people have limited access to modern cell phones.

1.7. References

Alred, F., Brusaw, C., and Oliu, W. (2003). Handbook of Technical Writing (7th ed.). Boston: Bedford/St. Martin's.

FDA API. Open.fda.gov. (n.d.). <https://open.fda.gov/apis/>

FDC API. <https://fdc.nal.usda.gov>

Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

Bootstrap style guides and docs: <https://getbootstrap.com/>

Django Documentation: <https://www.djangoproject.com/>

2. General Description

2.1. Product Features

Ingredient Intel, a revolutionary web app, empowers you to make informed food choices. Scan any product's barcode to access clear ingredient breakdowns, potential allergens, and health effects. Set dietary preferences and get alerts for unwanted ingredients. Companies benefit too, with easy ingredient management, recall communication, and transparency tools. Research institutions also play a role with access to contribute scientific notes, keeping consumers informed about the latest health findings. With the database connected to the FDC's api the transition of company products to the IngredientIntel database is just one click away. Future plans include linking the FDA's API to automatically flag recalled products.

- Barcode scanning feature to get product info and nutritional facts
- All user feature searching database for item by ingredients, companies, and products
- Researcher feature to add/modify/delete scientific notes on ingredients
- Company feature add/modify/delete products
- All user feature changing settings including adding flags for ingredients and changing the color mode
- Admin feature account and database management system to easily fix issues

Some of the products features will include barcode scanning capabilities and search bar implementation for looking into ingredients. Another feature will be the ability for researchers to change ingredients disclaimers on unwanted ingredients as new research comes out. Companies will also be able to add their products to database with their ingredients included.

2.2. User Class and Characteristics

Our application is intended to be accessible to anyone with access to the internet. Beyond that only basic internet operation skills and the ability to read English are needed.

For company and research users our application only requires the additional knowledge of information required for verification of identity like a company id or research credentials.

For the admin user more advanced computer skill may be needed to use the tools provided effectively to fix issues with users and databases.

2.3. Operating Environment

Ingredient Intel is operational only as a web application for the time being with testing on the chromium browser. Future plans include implementation of an app and support for other browsers.

2.4. Constraints

The biggest challenge of making this software will be generating the ingredients database and scaling as businesses add products. Additionally getting widespread adoption with companies will be time taking and require legislative action.

2.5. Assumptions and Dependencies

The software will be dependent upon Django to create and execute the MVC in VSCode, software will also be dependent on the FDC's api for product information importation, and on PostgreSQL open source database. All software dependancies are open source with long term support expected.

For hosting we rely on amazon web services to host our service.

3. Functional Requirements

3.1. Primary

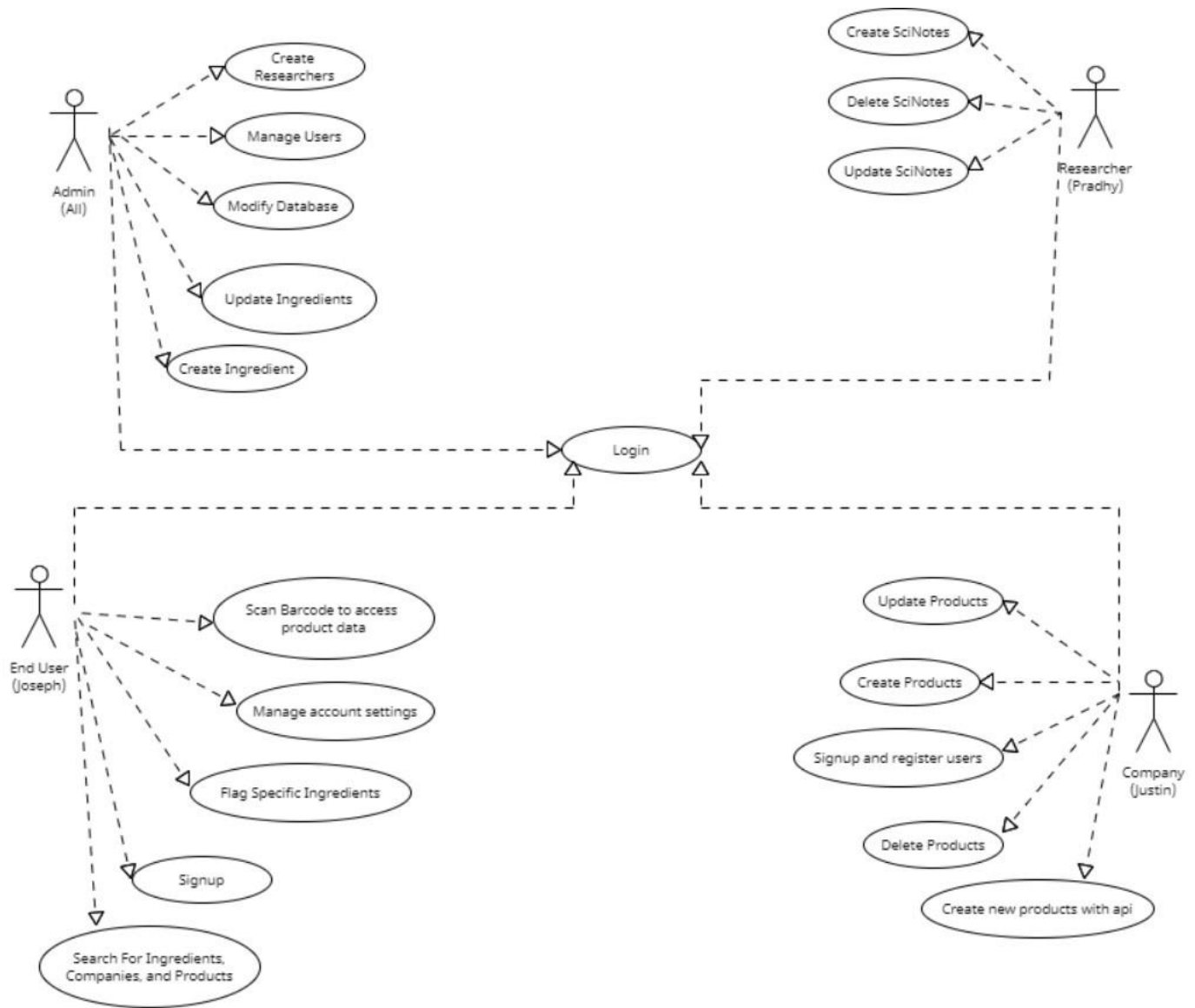
- FR0: The system should be able to scan barcodes accurately using images from an average camera. It should then be able to decode the information and retrieve the details associated with the product.
- FR1: The system should maintain a database of ingredients, and support functionality for adding, updating and deleting information as needed. The database of ingredients includes names, categories, potential health effects and allergen information.
- FR2: The system should be able to send alerts and notifications regarding product recalls, and other relevant updates. It should also allow users to customize their alerts based on dietary restrictions, allergens, etc.

3.2. Secondary

- Develop tools to categorize ingredients into groups to make it easier for users to find.
- Integrate with external APIs to enhance accuracy and send alert notifications to users promptly.
- Ensure there is an error handling mechanism to detect issues related to barcode scanning, DB operations, etc.

3.3. Use-Case Model

3.3.1. Use-Case Model Diagram



3.3.2. Use-Case Model Descriptions

3.3.2.1. Actor: Admin (All)

- **Modify database infrastructure:** super user has the ability to delete, modify, or create anything within the database including managing users and user groups.
- **Create Research Users:** Super users will be responsible for approving researchers for the research role

3.3.2.2. Actor: End User (Joseph)

- **Scan Barcode:** The user can access product information in the DB by scanning a barcode.
- **Receive Alerts:** If there is a recall or allergen detected on a specific product, then the user gets an alert notifying them with more information.
- **Search Database:** By using a simple search feature within the app or the website users can learn more about products or ingredients.
- **Create Account:** Users will be able to sign up and create credentials for the service allowing them to log in and access additional features.

- **Personalize Account:** Users with an account will be able to personalize their account by changing the color scheme of the website and by adding ingredients to flag when searching or viewing products.

3.3.2.3. Actor: Company (Justin)

- **Modify products:** If there is a change in a product the company can simply update the product in the database.
- **Communicate recall:** If the company identifies a product that needs to be recalled, the can communicate that with the users through the warnings in the product data.
- **Create products:** The company will be able to create new products manually in the dashboard.
- **Delete products:** The company will be able to delete any of their products from the company dashboard.
- **Import products:** The company will be able to import any currently registered products from the FDC's database.

3.3.2.4. Actor: Researcher (Pradhy)

- **Publish research notes:** Researchers can publish scientific notes regarding specific ingredient. For example: Potential health effects and other concerns.
- **Modify and Delete scientific notes:** Researchers are able to modify any details of a scientific note they have created and delete them if necessary.

3.3.3. Use-Case Model Scenarios

3.3.3.1. Actor: Admin (All)

- **Use-Case Name:** Modify Database
 - **Initial Assumption:** The admin has a valid staff account.
 - **Normal:** The super user will log into the database and correct any errors or issues with the data.
 - **What Can Go Wrong:** Admin forgetting credentials or having them stolen for malicious purposes.
 - **Other Activities:** Data can be copied, deleted, modified. Users can be removed, created, or modified and permissions granted or taken away.
 - **System State on Completion:** The database is updated and any issues are resolved on the back end and also changes propagate and can be viewed by the end user.
- **Use-Case Name:** Create Research User
 - **Initial Assumption:** The Admin has access to the staff role in the database and access to the admin page.
 - **Normal:** The admin logs into the admin page and selects the user account to modify, they then select the research permissions and add them to the users account.
 - **What Can Go Wrong:** Incorrectly entered passwords could cause setup to fail.
 - **Other Activities:** The superuser may also delete research API keys by searching for the name and using the delete button, confirming with password.

- **System State on Completion:** The profile of the researcher or research institution will now have access to the research dashboard and functionality.

3.3.3.2. Actor: End user (Joseph)

- **Use-Case Name:** Scanning barcodes
 - **Initial Assumption:** The user opens the web page and scans the barcode.
 - **Normal:** The app gets the product info and navigates the user to that page.
 - **What Can Go Wrong:** Bad lighting can cause the barcode to not be scanned properly, the barcode formatting may not be supported, or the product may not be in the database. In these cases a proper alert is returned.
 - **Other Activities:** N/A
 - **System State on Completion:** The app successfully retrieves the information from the database and displays the product info.
- **Use-Case Name:** Advanced Search Options
 - **Initial Assumption:** The user created an account and is logged in
 - **Normal:** The user selects any ingredients they want to flag and saves the changes in their settings.
 - **What Can Go Wrong:** Duplicate or mislabeled ingredients in the database may cause for flags to act unreliably.
 - **Other Activities:** The settings also allow for the user to change the color mode of the webpage to suit preferences.
 - **System State on Completion:** The user's preferences are saved and applied to their account successfully and when searching or viewing products they are alerted of any flagged ingredients.

3.3.3.3. Actor: Research Institution (Pradhy)

- **Use-Case Name:** Create new research notes
 - **Initial Assumption:** The researcher signs in and navigates to the research dashboard.
 - **Normal:** The researcher selects create a new scientific note and enters relevant info to create new note.
 - **What Can Go Wrong:** The form is filled out incorrectly or the research study itself has inaccurate or outdated information. Additionally, any links used as citations may reference pages that are no longer active.
 - **Other Activities:** The researcher can view, modify, and delete their existing notes.
 - **System State on Completion:** The researcher successfully accesses/modifies the research study and the information is updated in the database.
- **Use-Case Name:** Modify and delete old research notes
 - **Initial Assumption:** The researcher signs in and navigates to the research dashboard.
 - **Normal:** The researcher searches in the search bar for the relevant scientific note.
 - **What Can Go Wrong:** The form is filled out incorrectly and additionally if there are connection issues there may occasionally be errors causing duplicate actions and resulting error handling (redirects with incorrect alerts).

- **System State on Completion:** The researcher successfully accesses/modifies the research study and the information is updated in the database.

3.3.3.4. Company(Justin)

- **Use-Case Name:** Update product information
 - **Initial Assumption:** An employee at the company accesses the web platform and updates the ingredient information.
 - **Normal:** The employee logs in, and updates the product as needed.
 - **What Can Go Wrong:** The company employee enters incorrect/misleading information and uploads it to the database.
 - **Other Activities:** From the dashboard the user can create or delete products.
 - **System State on Completion:** The product information is successfully updated in the database.
- **Use-Case Name:** Register New Company
 - **Initial Assumption:** A registered user with a business is authenticated in the web app.
 - **Normal:** The user navigates to the company page and fills out all the details for the company they are registering.
 - **What Can Go Wrong:** The company data is incorrect or misleading.
 - **System State on Completion:** The new company is added to the database and the user is able to navigate to that company dashboard.
- **Use-Case Name:** Import Products
 - **Initial Assumption:** A user with access to a registered company is logged in.
 - **Normal:** The employee logs in, and navigates to the company dashboard and clicks the import products button.
 - **What Can Go Wrong:** There is an error with the api request
 - **System State on Completion:** The product information from the FDC is added to the companies products.
- **Use-Case Name:** Communicate ingredient or product recall
 - **Initial Assumption:** The company representative logs in to the Ingredient Intel platform.
 - **Normal:** The representative navigates to the company dashboard, selects the relevant products and updates the warning field to include the recall information.
 - **What Can Go Wrong:** Users overlook the warning data.
 - **System State on Completion:** The recall notification is successfully added to the database and communicated to users.

4. Technical Requirements

4.1. Interface Requirements

4.1.1. User Interfaces

The user will connect via login, after that the user can either type in ingredients to look up in our database via a search bar, or select a file so they can scan the bar code of an item, bringing them to its nutrition facts, recall info, etc.

4.1.2. Hardware Interfaces

In all instances the software will need internet to function. The device has to be able to fully interact with our webpage. In order to use the barcode scanning feature users will need to have access to a functioning camera.

4.1.3. Communications Interfaces

The software will need to be communicating with the internet, as well as PostgreSQL as the local database, HTTP, and lastly the open FDC API that we are using for product information.

4.1.4. Software Interfaces

On the backend we will be using PostgreSQL for our database, and we will be using Django with python to connect our front end and back end along with hosting and controlling our website including using some functionality for designing the frontend.

5. Non-Functional Requirements

5.1. Performance Requirements

- NFR0(R): The global database should not use more than 1 TB of memory.
- NFR1(R): Users will be able to find and use the user settings functionalities within 20 minutes of accessing the site.
- NFR2(R): New users will be able to easily scan a barcode and get nutrition facts in less than 5 minutes.
- NFR3(R): Experienced users can scan a bar code or look up items by ingredients in less than 1 minute.

5.2. Safety Requirements

Research studies listed for specific ingredients will include disclaimers when necessary to prevent consumer panic.

5.3. Security Requirements

- NFR4(R): All user data will be encrypted and stored in a AWS VPC with restricted access.
- NFR5(R): The system will have a sign in feature to verify users and password validation guaranteeing passwords meet a certain standard.
- NFR6(R): Database write access will only be available to verified users through specific forms.

5.4. Software Quality Attributes

5.4.1. Availability

The app will be free to use, meaning that anyone can use the app as they see fit. The app will only be available on Android for the time being but a limited version may also be accessed from the web.

5.4.2. Correctness

In general the onus of correctness falls on the companies and researchers. By restricting access or with fines companies or researchers will be required to provide accurate information. Users will be able to report any potential inaccuracies that can then be manually checked for accuracy.

5.4.3. Maintainability

Researchers and companies will update information about ingredients and products respectively. Our team would handle any bug fixes or updates needed along with managing data storage and service availability.

5.4.4. Usability

Software will maintain reasonable loading and response times and will include features for accessibility like high contrast or magnification.

5.4.5. Portability

The application will be available on android, making it very portable and easy to use so long as there is access to the internet which is necessary for accessing the database.

5.5. Process Requirements

5.5.1. Development Process Used

We are utilizing a swift development process.

5.5.2. Time Constraints

The main time constraint is our delivery date, as well as certain milestones we must reach along the way. Specifically:

- Feb-27-2024 Product prototype completion
- Apr-30-2024 Project completion

5.5.3. Cost and Delivery Date

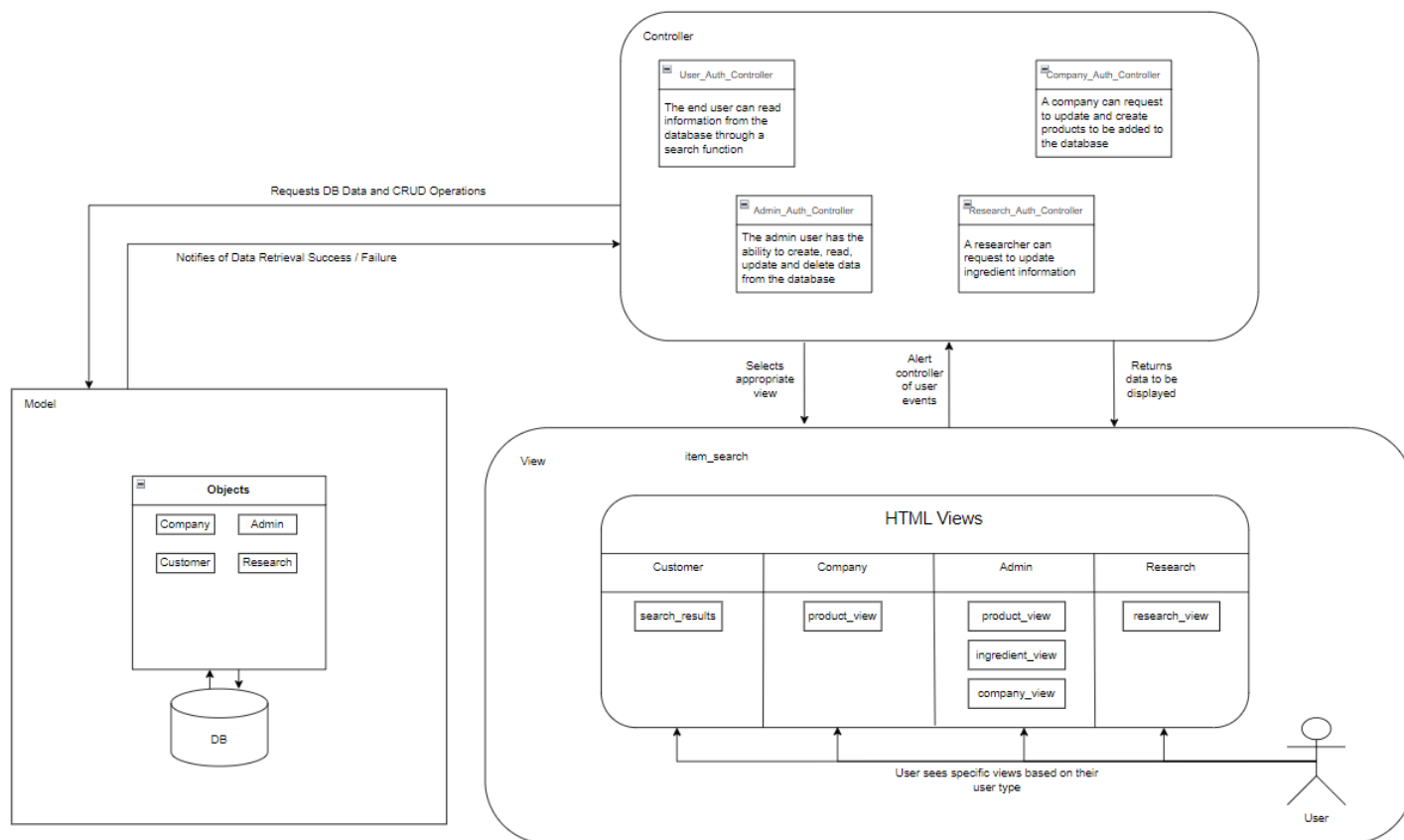
The delivery date is planned for later April 2024. Expected cost is \$15,000 in work time, and a hosting cost that will vary based off server costs changing rapidly currently due to AI development but we estimate a hosting cost of \$20,000/year when running at full capacity. Maintenance costs should be fairly low and will be contained within the purview of the existing FDA contractors.

5.6. Other Requirements

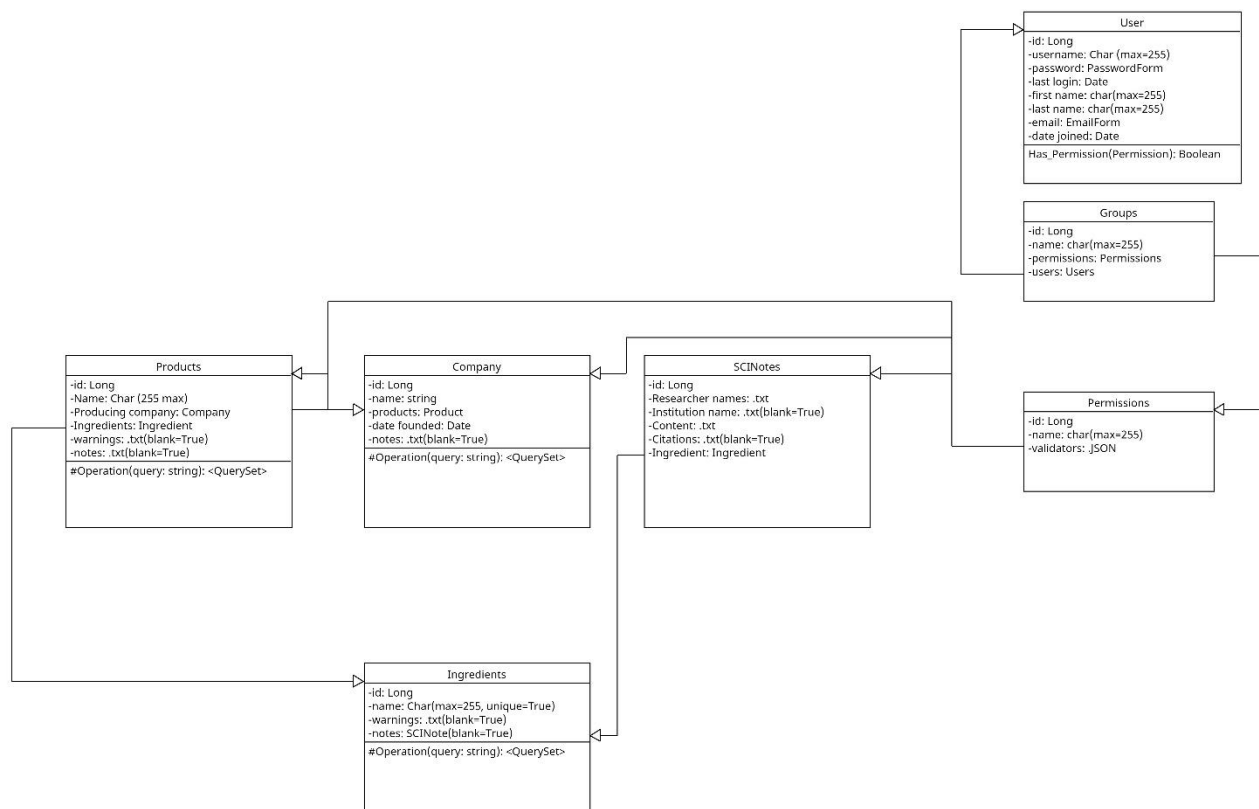
Post product development integration with FDA services and additional transition work to be done after approval by FDA.

6. Design Documents

6.1. Software Architecture



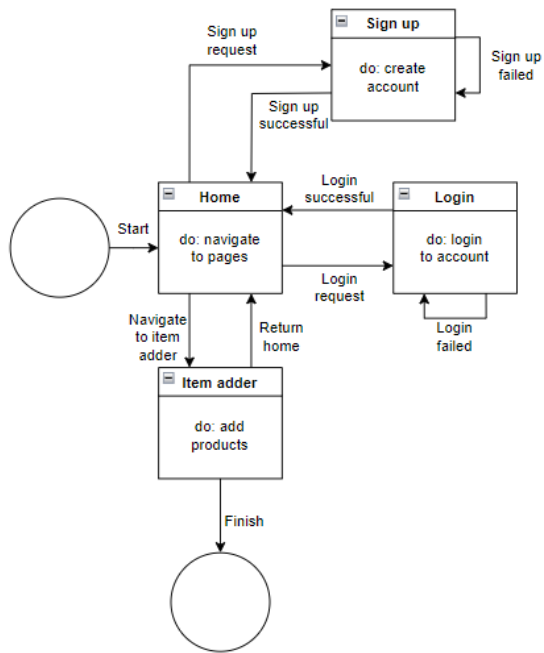
6.2. High-Level Database Schema



6.3. Software Design

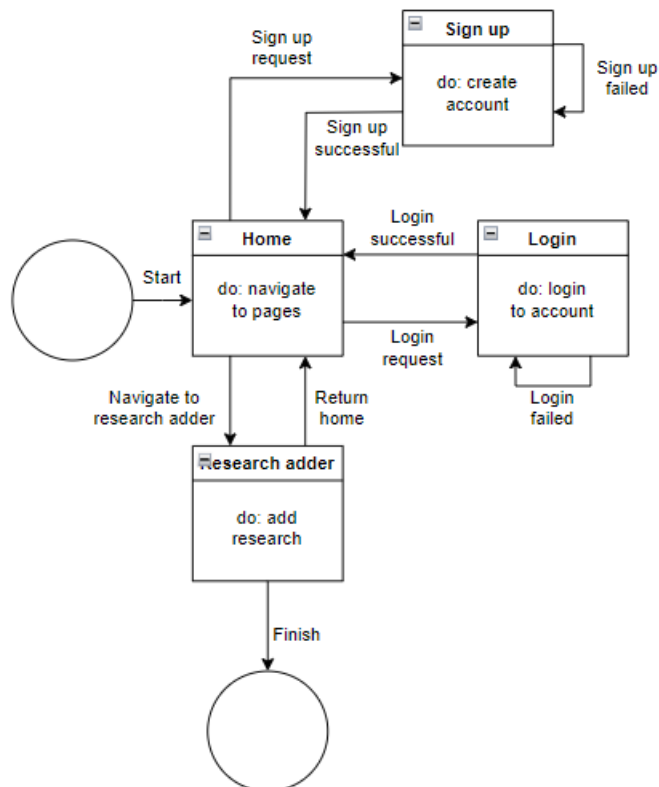
6.3.1. State Machine Diagram: Company (Justin Evans)

Company State Diagram



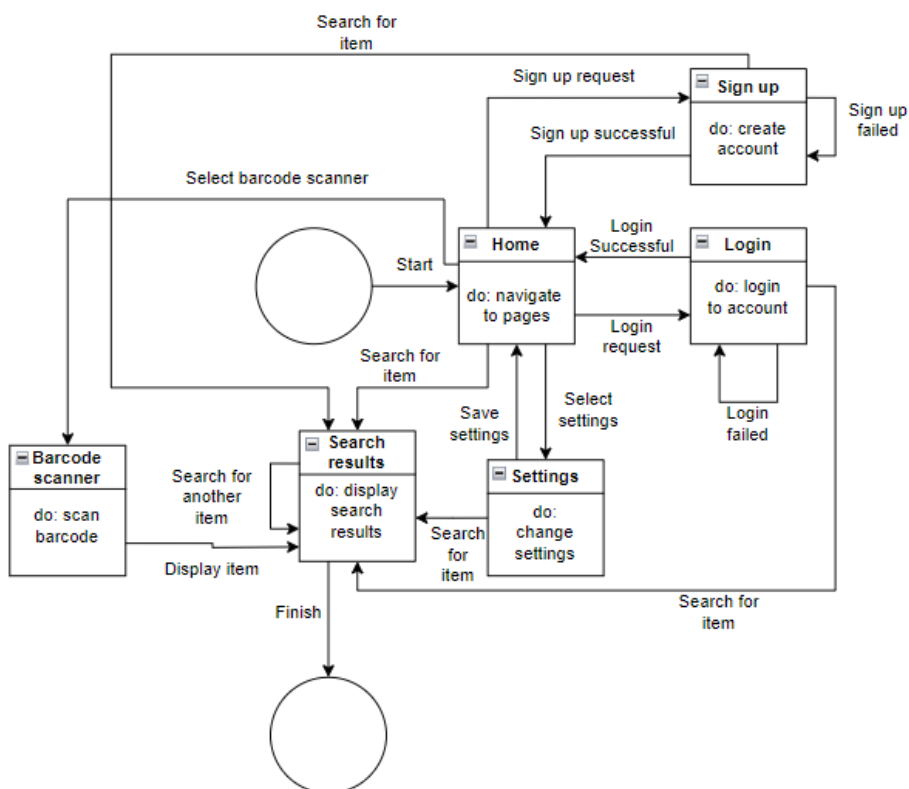
6.3.2. State Machine Diagram: Researcher (Justin Evans)

Researcher State Diagram



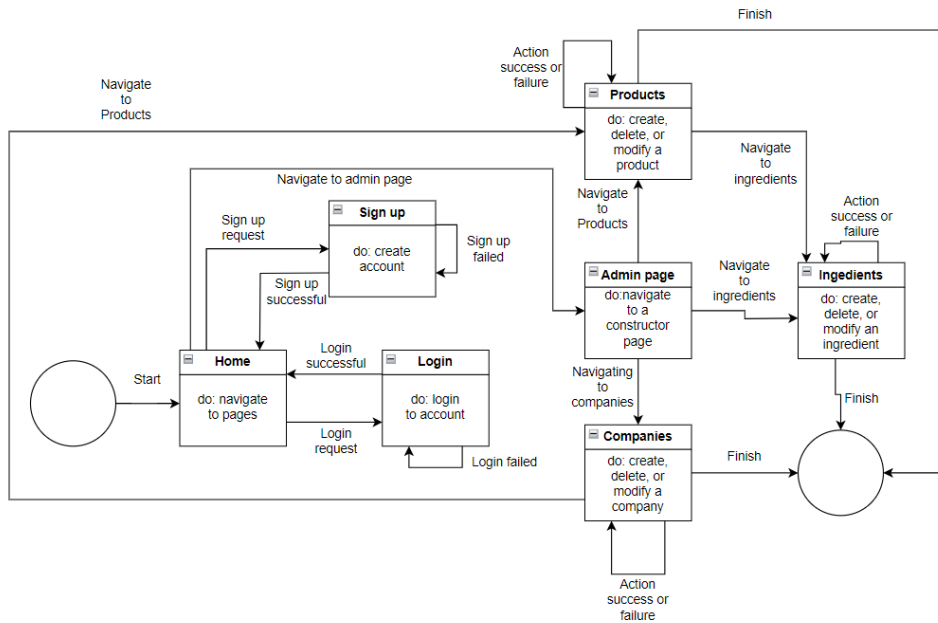
6.3.3. State Machine Diagram: Customer (Justin Evans)

Customer State Diagram

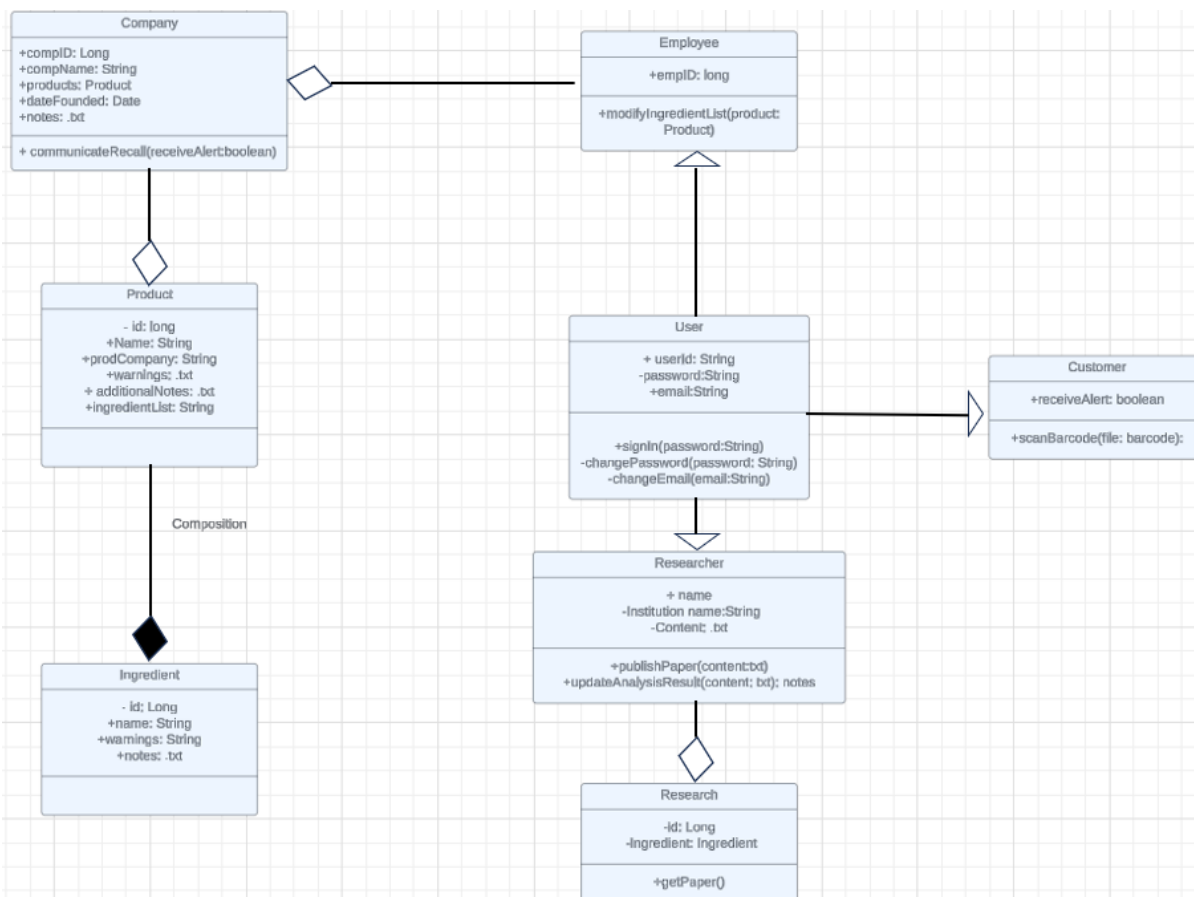


6.4.4. State Machine Diagram: Admin (Justin Evans)

Admin State Diagram



6.4. UML Class Diagram



7. Scenario

7.1. Brief Written Scenario with Screenshots

Admin: Login, modify database (create, delete, modify items), manage users, create research user

1. Admin logs in and navigates to admin page

- Admin selects database table and modifies ingredient database by creating an ingredient, modifying an ingredient, and deleting an ingredient.
- Admin selects user table and modifies user database to create a new user, to delete a user, and to convert a user account into a research user.
- Admin logs out

Company: Login, create new company, create new product, modify product, delete product, import products from FDC, and logout

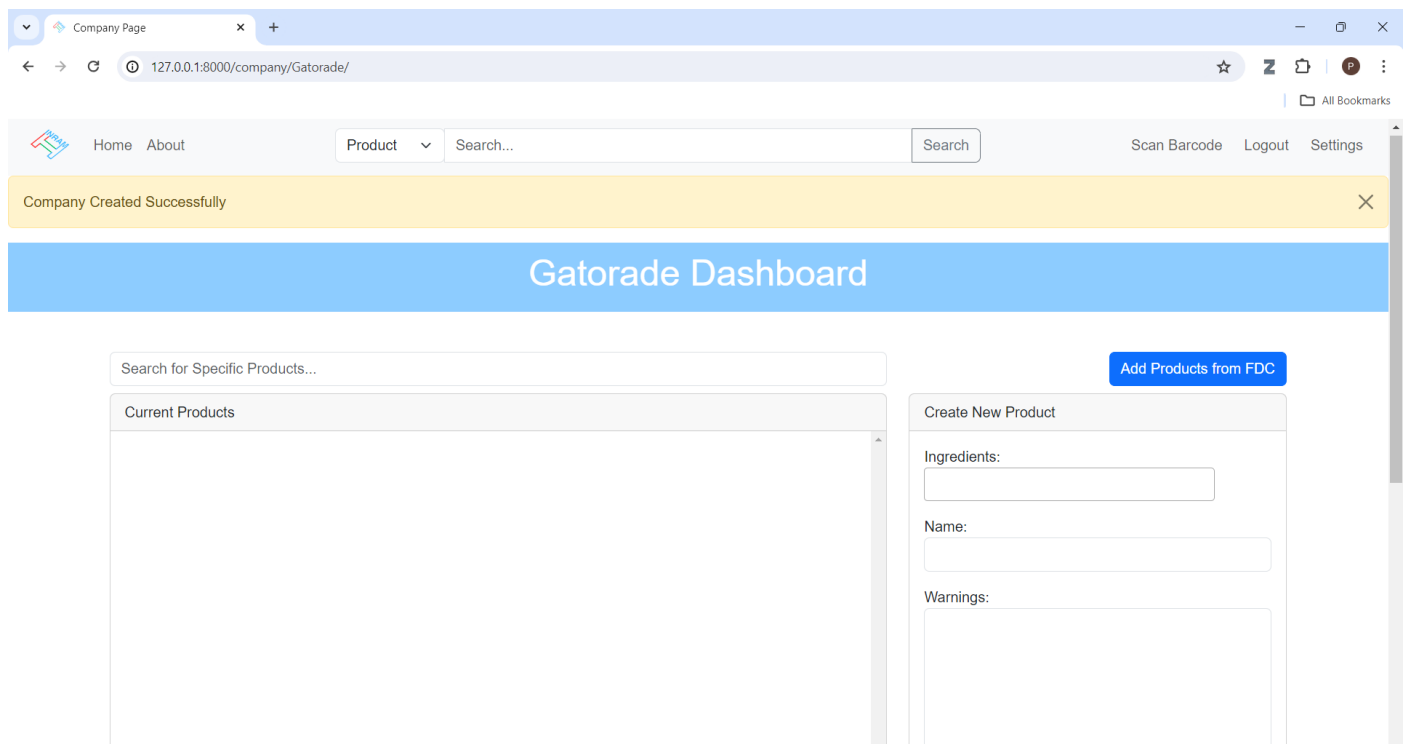
- Verified user logs in to user account and navigates to company page
- User fills out form to create a new company
- User navigates to company page and creates a product, modifies the product, and deletes the product
- User navigates to company dashboard and imports products from FDC api
- User logs out

Researcher:

- Researcher logs in and navigates to research page
- Researcher adds scientific note, modifies scientific note, and deletes scientific note
- Researcher searches through scientific notes then logs out

End User:

- New user accesses homepage and searches database for a product
- New user navigates to signup and creates a new account, then logs in
- Authenticated user changes account settings for color mode and adds flags for ingredients they are allergic to.
- Authenticated user searches database with flags and examine product with flags
- User navigates to barcode page and scans barcode to view item then user logs out



This is an image take after the company "Gatorade" was created, using the Add Products from FDC feature, The current products list will be populated by products in the FDC database

Company Page x +

127.0.0.1:8000/company/Gatorade/

Home About Product Search... Search Scan Barcode Logout Settings

Items added

Gatorade Dashboard

Search for Specific Products...

[Add Products from FDC](#)

Current Products	
GATORADE GATORADE PRIME ORANGE 4Z/6 48P	Update Delete
GATORADE LIME 28Z	Update Delete
GATORADE ORANGE 24Z	Update Delete
G2 THIRST QUENCHER GATORADE	Update Delete
GATORADE GRAPE 12Z8PK/1	Update Delete
GATORADE ICED PUNCH 12Z4PK	Update

Create New Product

Ingredients:

Name:

Warnings:

As you can see, the products were added to the dashboard and the company will be able to update make changes to their products as they need to.