# CSC 431 Smart Food Storage Software Requirements Specification (SRS)

**Team 04**

|  |  |
| --- | --- |
| Angela Wang | Scrum Master |
| Hill Yu | Developer |
| Alex Luce | Developer |
| <Member Name> | <Role> |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1.0 | 2/18/2022 | Angela Wang | first draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[1.](#_3dy6vkm) System Requirements 6

[1.1](#_1t3h5sf) Functional Requirements 6

[1.1.1](#_4d34og8) Display a Food information

1.1.2 Recipe searching

[1.2](#_2s8eyo1) Non-Functional Requirements 6

[1.2.1 Retrieving information about existing items quickly](#_17dp8vu)

1.2.2 Scrapes Recipe Sites and Displays Recipes Quickly

[2.](#_3rdcrjn) System Constraints 7

[2.1](#_26in1rg) Tool Constraints 7

[2.1.1](#_lnxbz9) iOS Application Constraints 7

[2.2](#_35nkun2) Language Constraints 7

[2.2.1](#_1ksv4uv) iOS Application Language 7

[2.3](#_44sinio) Platform Constraints 7

[2.3.1](#_2jxsxqh) Apple Mobile Store Compliance 7

[2.4](#_z337ya) Hardware Constraints 7

[2.4.1](#_3j2qqm3) Compatible Smart Food Storage hardware required

[2.4.2](#_3j2qqm3) Computer / smart mobile device required 7

[2.5](#_1y810tw) Network Constraints 7

[2.5.1](#_4i7ojhp) Stable wireless network 8

[2.6](#_2xcytpi) Deployment Constraints 8

[2.6.1](#_1ci93xb) Apple Mobile Store Acceptance 8

[2.7](#_3whwml4) Transition & Support Constraints 8

[2.7.1](#_2bn6wsx) Hardware support constraints 8

[2.8](#_qsh70q) Budget & Schedule Constraints 8

[2.8.1](#_3as4poj) Requirement Title 8

[3.](#_2p2csry) Requirements Modeling 10

[3.1.1](#_147n2zr) Modeling 10

[4.](#_3o7alnk) Evolutionary Requirements 11

[4.1](#_23ckvvd) Functional Requirements 11

[4.1.1](#_ihv636) Requirement Title 11

[4.2](#_32hioqz) Non-Functional Requirements 11

[4.2.1](#_1hmsyys) Requirement Title 11

# Table of Tables

# Table of Figures

[Figure 1 15](#_Toc96294545)

### System Requirements

#### Functional Requirements

*< List all functional requirements in the following example format >*

##### Retrieve and Display Stored Food Information

|  |  |
| --- | --- |
| Title | Displaying stored food information |
| Description | The AI technology in the Smart Fridge needs to correctly identify what food is in the fridge and how much and their expiration date to display on the app. |
| Priority | 0 |
| Precondition(s) | User needs to have food in the Smart Fridge |
| Basic Flow | * User puts their groceries in the fridge * Fridge identifies what each grocery is through its camera or by user’s manual keyboard input * Fridge identifies the quantity based on weights in the specific location of the fridge * Fridge sends and updates the information to the connected app on the user’s phone |
| Postconditions(s) | App correctly displays and stores all the new items in the *Refrigerator View* and also the *Grocery List* view. |
| Use Case Diagram | User and System |

##### 

##### View Refrigerator/View Grocery List

|  |  |
| --- | --- |
| Title | View Refrigerator/View Grocery List |
| Description | This is the main page of the app, which displays the user’s current fridge contents. |
| Priority | 0 |
| Precondition(s) | Open the app |
| Basic Flow | * User opens app * User views currents items in the fridge |
| Postconditions(s) | Users can click into items to view food information or can generate recipes. |
| Use Case Diagram | User and System |

##### 

##### Manual Input

|  |  |
| --- | --- |
| Title | Manual Input |
| Description | users can manually input items if needed |
| Priority | 5 |
| Precondition(s) | Open the app |
| Basic Flow | * User opens app * User clicks on ‘edit’ icon to change any of the food items |
| Postconditions(s) | The Refrigerator View/Grocery List View correctly updates and depicts food items based on the manual input from the user. |
| Use Case Diagram | User and System |

##### 

##### Get More Information On Specific Food Item

|  |  |
| --- | --- |
| Title | Get more information on specific food item |
| Description | Each item needs to display details on that item per user request. The details consist of name, quantity, and expiration date. |
| Priority | 1 |
| Precondition(s) | User needs to have food in the Smart Fridge |
| Basic Flow | * User has items in the Smart Fridge and is updated in the app * User clicks on the item in the app * popup fills the screen and the user can see the name, quantity, and expiration date. * User clicks anywhere outside of the popup to close it |
| Postconditions(s) | App correctly displays a popup consisting of the details of a specific food item |
| Use Case Diagram | User and System |

##### Generate Recipes

|  |  |
| --- | --- |
| Title | Search for Recipes from External Sources |
| Description | The app searches for recipes by accessing other websites, priority given to *Food Network* and recipe sites listed on our “credible” list. |
| Priority | 0 |
| Precondition(s) | User has inputted their food items |
| Basic Flow | * User input food items * App compares food items with ingredients of recipes online * Returns recipes with a ratings icon that indicated the average rating from other users |
| Postconditions(s) | App correctly displays 3-4 recipes which the user can click into and look at the details. When a user clicks ‘finish’ with a recipe, indicating that they are done creating food, it automatically updates the quantity of food. |
| Use Case Diagram | Dialogue between user and system |

##### 

##### Refresh Recipes

|  |  |
| --- | --- |
| Title | Refresh recipes |
| Description | Refresh recipe list to display 3-4 additional recipes. The first to display are the most highly rated and from most credible sources, with additional recipes being less rated. |
| Priority | 5 |
| Precondition(s) | User has inputted their food items |
| Basic Flow | * User input food items * App compares food items with ingredients of recipes online * Returns recipes * User clicks ‘more recipes’ * Displays 3-4 more recipes |
| Postconditions(s) | App correctly displays 3-4 recipes which the user can click into and look at the details. When a user clicks ‘finish’ with a recipe, indicating that they are done creating food, it automatically updates the quantity of food. |
| Use Case Diagram | Dialogue between user and system |

##### Substitute Missing Ingredients

|  |  |
| --- | --- |
| Title | Substitute missing ingredients |
| Description | If a user has almost all of the ingredients except one or two, the app will generate popular substitutions for the missing ingredients based on food items in the fridge, and will also identify and suggest which ingredients can be left out. |
| Priority | 2 |
| Precondition(s) | User has inputted their food items |
| Basic Flow | * Recipes almost available but missing one or two ingredients are highlighted in yellow * User clicks recipe highlighted in yellow and check the ingredient list for what is missing * Missing ingredient is highlighted in red * user clicks missing ingredient * System searches in a database for substitution items * a popup displays a substitution item or if it can be left out * User clicks accept * The original ingredient in the recipe is crossed out and shows the new ingredient (or is just crossed out to show not to include) |
| Postconditions(s) | App correctly displays a new recipe that is modified with a substitution ingredient or the exclusion of the missing ingredient |
| Use Case Diagram | Dialogue between user and system |

##### Rate Recipes

|  |  |
| --- | --- |
| Title | Rating Systems |
| Description | Users can give ratings to recipes |
| Priority | 5 |
| Precondition(s) | User has generated recipes from their food items and are browsing for recipes |
| Basic Flow | * User input food items * App compares food items with ingredients of recipes online and returns recipes * When clicks into a recipe to view it * User clicks ‘rate me’ * Popup displays and asks for user to input a rating * User clicks x stars/ 5 stars * User clicks ‘submit’ |
| Postconditions(s) | App recalculates the average rating and displays according to the user rating. |
| Use Case Diagram | Dialogue between user and system |

##### 

#### 

#### Non-Functional Requirements

*< List all non-functional requirements in the following example format >*

##### Retrieving information about existing items quickly

|  |  |
| --- | --- |
| Title | Retrieving information about existing items quickly |
| Description | The app needs to display information about certain item(s) upon the user's request. This information includes a detailed name, quantity and expiration date. |
| Priority | 0 |
| Applicable FR(s) | Display a food item and its quantity |

##### 

##### Generate Recipes Time

|  |  |
| --- | --- |
| Title | Generate Recipes Time |
| Description | Because the app is looking for recipes from many different external sources, this process can be very slow (and also determined by network requirements.) We want this process to be as quick as possible, less than 1000ms. |
| Priority | 0 |
| Applicable FR(s) | Search for Recipes from External Sources |

##### 

##### Maintenance Time

|  |  |
| --- | --- |
| Title | Maintenance Time |
| Description | To ensure that the app is working properly, maintenance will be conducted every month. The down time for the application will be no longer than an hour. |
| Priority | 2 |
| Applicable FR(s) | N/A |

##### 

##### Food Item Storage

|  |  |
| --- | --- |
| Title | Food Item Storage |
| Description | The database must be able to handle more than 300 food items from the user. |
| Priority | 0 |
| Applicable FR(s) | Retrieve and Display Stored Food Information |

### System Constraints

#### Tool Constraints

*< List all tool constraints in the following example format >*

##### iOS Application Constraints

|  |  |
| --- | --- |
| Title | iOS Application Constraints |
| Description | Application for iOS and will be designed in XCode for optimal UX design and device specific testing/debugging. |
| Priority | 3 |

#### Language Constraints

*< List all language constraints in the following example format >*

##### iOS Application Language

|  |  |
| --- | --- |
| Title | iOS Application Language |
| Description | The application will use Swift, the native language for Apple mobile devices. Swift will minimize time spent developing/debugging when used in conjunction with XCode. |
| Priority | 1 |

#### Platform Constraints

*< List all platform constraints in the following example format >*

##### Apple Mobile Store Compliance

|  |  |
| --- | --- |
| Title | Apple Mobile Store Compliance |
| Description | The Apple iOS store is strict on design and functionality. Font, color palette, and other design considerations must be compliant with Apple standards for the app to be listed. |
| Priority | 1 |

#### Hardware Constraints

*< List all hardware constraints in the following example format >*

##### Compatible Smart Food Storage hardware required

|  |  |
| --- | --- |
| Title | Certified compatible Smart food storage hardware required |
| Description | The software application is only compatible with our own Smart Food Storage hardware, and also other certified OEM partners’ smart food storage hardwares. |
| Priority | 0 |

##### Computer / smart mobile device required

|  |  |
| --- | --- |
| Title | Computer / smart mobile device required |
| Description | The software can only be accessed on our website or the mobile application( compatible with both android and iOS). Therefore, all users need to have either a computer running Windows/Mac OS or a mobile device running Android/iOS. |
| Priority | 0 |

#### Network Constraints

##### Stable wireless network

|  |  |
| --- | --- |
| Title | Stable wireless network |
| Description | A stable wireless network is required for the system to access online recipes and databases. Also, the Smart Food Storage needs to be connected to a stable wireless network. |
| Priority | 0 |

#### Deployment Constraints

*< List all deployment constraints in the following example format >*

##### Apple Mobile Store Acceptance

|  |  |
| --- | --- |
| Title | Apple Mobile Store Acceptance |
| Description | The application must have been accepted to be listed in the Apple store. Without this users will not be able to effortlessly download the app and is a barrier to mass adoption. |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Transition & Support Constraints

*< List all transition & support constraints in the following example format >*

##### Hardware Support Constraints

|  |  |
| --- | --- |
| Title | Hardware support constraints |
| Description | Some troubleshooting processes involving the hardware will require a hardware technician onsite to debug the system. |
| Priority | 3 |

##### Food Items can Only be Displayed in App from Manual Input or *Smart Food Storage* System

|  |  |
| --- | --- |
| Title | Food Items can Only be Displayed in App from Manual Input or *Smart Food Storage* System |
| Description | For items to be displayed in the app, there needs to be food items detected first. The app cannot display food unless it is given food items. |
| Priority | 0 |

#### Budget & Schedule Constraints

*< List all budget & schedule constraints in the following example format >*

##### Budget of $1000 and Future Investments

|  |  |
| --- | --- |
| Title | Budget of $1000 and future investments |
| Description | The app itself can be developed from the core developers with no compensation, however, the technology required to develop the *Smart Food Storage* itself requires compensation towards electrical engineers and material to create the fridge. |
| Priority | 0 |

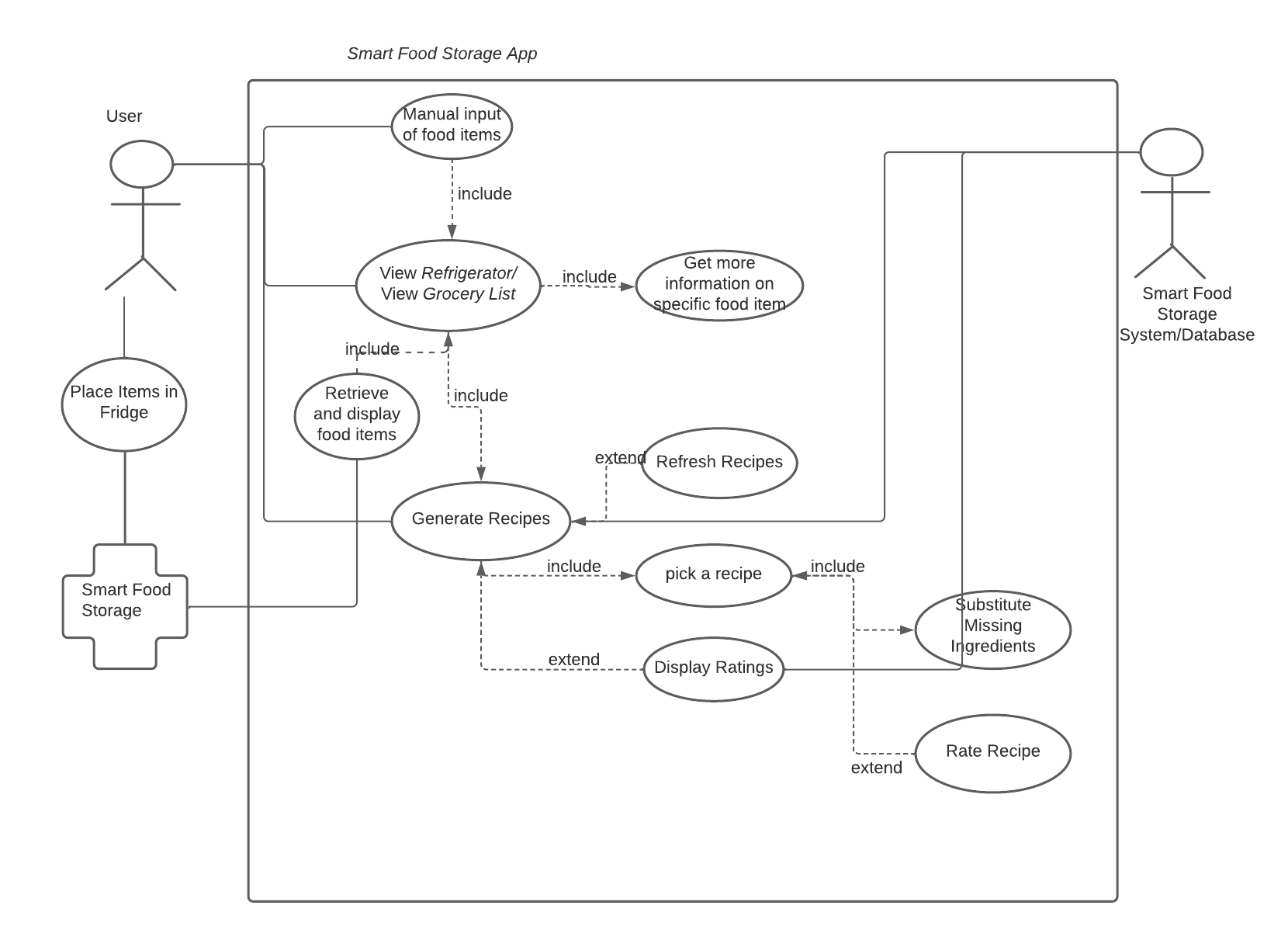
##### Be Finished by May 13th 2022

|  |  |
| --- | --- |
| Title | Be finished by May 13th 2022 |
| Description | The app must be finished before May 13th 2022 as developers will be graduating. |
| Priority | 0 |

### Requirements Modeling

*< List all Use-case diagrams for the functional requirements in the following format>*

##### Modeling



Figure

<https://lucid.app/lucidchart/1dadb44b-da58-4123-8086-9542d69f1619/edit?invitationId=inv_717a9939-08fe-467f-a456-25ee1dcba02f>

### Evolutionary Requirements

#### Functional Requirements

##### Android Port

|  |  |
| --- | --- |
| Title | Android Port |
| Description | The app will first be created for iOS devices but ported to Android to be accessible to more users. |
| Priority | 3 |
| Precondition(s) | Successful iOS deployment |
| Postconditions(s) | Almost entirely integrated version of *Smart Food Storage* to Android devices |
| Use Case Diagram | None |