



**CSC 431**

**WeCook**

## **Software Requirements Specification (SRS)**

**Team 04**

Avani Choudhary	Lead Designer
Alessandro D'Urso	Architecture Lead
Daniel Moran	Scrum Master and Planning Lead

# Version History

Version	Date	Author(s)	Change Comments
1.1	2/23/21	Daniel Moran, Alessandro D'Urso, Avani Choudhary	first draft
1.2	3/9/21	Daniel Moran, Alessandro D'Urso, Avani Choudhary	second draft
1.3	4/13/21	Daniel Moran, Alessandro D'Urso, Avani Choudhary	third draft
2.0	5/02/21	Daniel Moran, Alessandro D'Urso, Avani Choudhary	final draft

# Table of Contents

1.	System Requirements	5
1.1	Functional Requirements	5
1.1.1	Sign-up for account	5
1.1.2	Log-in to existing account	5
1.1.3	Account & Settings Modification	5
1.1.4	List Cookbooks	6
1.1.5	View Recipe	6
1.1.6	View Recipe	6
1.1.7	Search Page	7
1.1.8	Explore Page	7
1.1.9	Liking Recipes	7
1.1.10	Saving Recipes	8
1.1.11	Commenting functionality	8
1.1.12	Messaging functionality	8
1.1.13	Notification Badges	8
1.1.14	Password Recovery	9
1.2	Non-Functional Requirements	9
1.2.1	Database input sanitization	9
1.2.2	Message encryption	9
1.2.3	Automatic password reset requests	9
1.2.4	Search result time	10
1.2.5	Uptime	10
1.2.6	Cookbook Storage	10
2.	System Constraints	11
2.1	Tool Constraints	11
2.1.1	Mobile Application Framework Constraint	11
2.2	Language Constraints	11
2.2.1	Backend Framework	11
2.3	Platform Constraints	11
2.3.1	Mobile Application Platform	11
2.4	Hardware Constraints	11
2.4.1	Storage Constraints	11
2.4.2	Computation Constraints	12
2.5	Network Constraints	12
2.5.1	Access Database	12
2.5.2	Uploading	12
2.5.3	Messaging	12
2.6	Deployment Constraints	12
2.6.1	Amazon Lightsail Deployment	12
2.7	Transition & Support Constraints	13
2.7.1	End of Life	13
2.8	Budget & Schedule Constraints	13
2.8.1	Time Constraints	13
2.8.2	Budget Constraints	13
3.	Requirements Modeling	14
3.1	Primary Use Case	14

3.2	WeCook Application System	15
3.1.1	Class Diagram	16
4.	Evolutionary Requirements	17
4.1	Website Application	17
4.2	Elderly Accessibility mode	17

# Table of Tables

1.	System Requirements	5
	1 . Sign-up for account	5
	2. Log-in to existing account	5
	3. Account & Settings Modification	5
	4. List Cookbooks	6
	5. View Recipe	6
	6. View Recipe	6
	7. Search Page	7
	8. Explore Page	7
	9. Liking Recipes	7
	10. Saving Recipes	8
	11. Commenting functionality	8
	12. Messaging functionality	8
	13. Notification Badges	8
	14. Password Recovery	9
	15. Database input sanitization	9
	16. Message encryption	9
	17. Automatic password reset requests	9
	18. Search result time	10
	19. Uptime	10
	20. Cookbook Storage	10
2.	System Constraints	11
	21. Mobile Application Framework Constraint	11
	22. Backend Framework	11
	23. Mobile Application Platform	11
	24. Storage Constraints	11
	25. Computation Constraints	12
	26. Access Database	12
	27. Uploading	12
	28. Messaging	12
	29. Amazon Lightsail Deployment	12
	30. End of Life	13
	31. Time Constraints	13
	32. Budget Constraints	13
3.	Use Case Diagram	
	33. Primary Use Case	14
4.	Evolutionary Requirements	17
	34. Website Application	17
	35. Elderly Accessibility mode	17

# Table of Figures

<u>2.1</u> WeCook Application System	14
<u>3.3</u> Structural Design	16

# 1. System Requirements

## 1.1 Functional Requirements

### 1.1.1 Sign-up for account

Table 1: Sign-up for account

Title	Sign-up
Description	Form that allows users to create an account
Priority	0
Precondition(s)	Open up the application after downloading
Basic Flow	User clicks create account Form opens, user enters details including name, username, email and password User submits information and the account is created and saved to database User is directed to login page
Postconditions(s)	User account is created and they are directed to login, can access all features of the app
Use Case Diagram	Figure 1

### 1.1.2 Log-in to existing account

Table 2: Log-in to existing account

Title	Log-in
Description	Existing users must be able to sign into their accounts
Priority	0
Precondition(s)	User should have already signed up for an account
Basic Flow	User enters username/email User enters password User presses "Sign-in" and validation occurs
Postconditions(s)	User is redirected to the home page Repeat if incorrect username/password
Use Case Diagram	Figure 1

### 1.1.3 Account & Settings modification

Table 3: Account & Settings modification

Title	Account & Settings
Description	A user must be able to edit their account and application settings
Priority	0
Precondition(s)	User should have already signed into the application, and clicked account page button from home page
Basic Flow	User can modify account and application settings

Postconditions(s)	User account information is modified in backend database, user settings are stored locally
Use Case Diagram	Figure 1

## 1.1.4 List Cookbooks

Table 4: List Cookbooks

Title	Cookbook List
Description	A user must be able to view a list of their cookbooks, and select them interactively
Priority	0
Precondition(s)	User should have signed in to the application and clicked "Cookbooks" icon to be redirected
Basic Flow	User can modify, create or remove new or pre-existing cookbooks
Postconditions(s)	New Recipes added/removed Cookbook settings changed Cookbook availability changed
Use Case Diagram	Figure 1

## 1.1.5 View Cookbook

Table 5: View Cookbook

Title	Cookbook View
Description	A user must be able to individually view public or private cookbooks they are members of
Priority	3
Precondition(s)	User should have already signed in to the application and selected a Cookbook from the Cookbook List
Basic Flow	User can add/delete/edit recipe If user owns Cookbook, they can change public/private If user owns Cookbook, they can invite users to Cookbook If user owns Cookbook, they can change permissions User can add/remove members from a private cookbook
Postconditions(s)	New Recipes added/removed Cookbook settings changed Cookbook availability changed
Use Case Diagram	Figure 1

## 1.1.6 View Recipe

Table 6: View Recipe

Title	Recipe View
Description	A user must be able to view an individual recipe using the recipe icon in their cookbook next to a particular entry
Priority	1
Precondition(s)	User should have already signed in to the application and selected a recipe from a public or private Cookbook
Basic Flow	User can create, edit or modify new or preexisting recipes within Cookbook functionality

Postconditions(s)	New Recipes added/removed Recipe modified Recipe published and saved to database
Use Case Diagram	Figure 1

## 1.1.7 Search

Table 7: Search

Title	Search Page
Description	A user must be able to search for their recipes, as well as view recipes similar to those they already like, or have viewed recently
Priority	2
Precondition(s)	Sign into application, navigate to search page from home page
Basic Flow	User clicks on search bar Enters a few words based on what recipe/cookbook/user they are looking for Application provides results based on what user enters/public Cookbook availability
Postconditions(s)	Recipes saved in search are recorded by database, search history is also saved by database
Use Case Diagram	Figure 1

## 1.1.8 Explore

Table 8: Explore

Title	Explore Page
Description	A user must be able to search for their recipes from a selection of categories.
Priority	2
Precondition(s)	Sign into application, navigate to explore page from home page
Basic Flow	User clicks on one of the categories provided by the page, including types of cuisines and meals Given the category they click on, they are provided with a feed of recipes and Cookbooks saved under that category
Postconditions(s)	Recipes liked are recorded by database
Use Case Diagram	Figure 1

## 1.1.9 Liking Recipes

Table 9: Liking Recipes

Title	Like
Description	A user must be able to like a recipe
Priority	4
Precondition(s)	Sign into application, navigate to recipe
Basic Flow	User selects like icon underneath recipe



	Creator is notified that their recipe has been liked Liked recipe is placed at a higher rank in future searches User may unlike a recipe
Postconditions(s)	Recipes are given higher ranking in the cookbook search algorithm.
Use Case Diagram	-

### 1.1.10 Saving Recipes

Table 10: Saving Recipes

Title	Save
Description	A user must be able to save a recipe
Priority	4
Precondition(s)	Sign into application, navigate to recipe
Basic Flow	Saving a Recipe User selects save icon below the recipe Saved recipe is added to the User's "Saved Recipes" cookbook User may unsave a recipe
Postconditions(s)	Recipes are added to "saved recipes" cookbook, or to the database for retrieval by user if saved
Use Case Diagram	-

### 1.1.11 Commenting Functionality

Table 11: Commenting Functionality

Title	Comment
Description	A user must be able to comment on recipes
Priority	5
Precondition(s)	Sign into application, navigate to recipe
Basic Flow	<ul style="list-style-type: none"> <li>- User selects "add a comment"</li> <li>- User writes comment</li> <li>- Creator of recipe is notified</li> <li>- Creator, or other users may reply to individual comments</li> <li>- Commenting may be turned off by the creator</li> </ul>
Postconditions(s)	Comment is added to the recipe
Use Case Diagram	-

### 1.1.12 Messaging Functionality

Table 12: Messaging Functionality

Title	Messaging
Description	A user must be able to send private messages to another user
Priority	4
Precondition(s)	Select recipe, or "friend" and select messaging icon
Basic Flow	<ul style="list-style-type: none"> <li>- User selects messaging icon</li> <li>- Message is sent/delivered to recipient</li> <li>- Recipient is notified</li> <li>- Recipient may reply</li> </ul>

	- Recipients, if not already friends, may reject a messaging request
Postconditions(s)	Message thread is created or updated
Use Case Diagram	-

### 1.1.13 Notification Badges

Table 13: Notification Badges

Title	Notifications
Description	A user must be notified when their recipes are liked, cookbooks are followed, or comments/messages are received
Priority	4
Precondition(s)	User must have an account
Basic Flow	<ul style="list-style-type: none"> <li>- One of above actions (like, message, comment, friend request) occurs</li> <li>- User is notified</li> <li>- User may respond appropriately</li> </ul>
Postconditions(s)	User is notified
Use Case Diagram	-

### 1.1.14 Password Recovery

Table 14: Password Recovery

Title	Password Recovery
Description	A user must be able to recover their password, should the forget it or have their account compromised (who hacks a cookbook app? Seriously.)
Priority	0
Precondition(s)	User must have an account
Basic Flow	<ul style="list-style-type: none"> <li>- User has forgotten password, or account has been locked for security reasons</li> <li>- User submits a request for a password reset</li> <li>- User receives email with password reset instructions</li> <li>- User is redirected to an in-app password reset tool from unique link</li> </ul>
Postconditions(s)	User password is modified, password is updated in database
Use Case Diagram	-

## 1.2 Non-Functional Requirements

### 1.2.1 Database input sanitization

Table 15: Database input sanitization

Title	Database input sanitization
Description	Any time inputs are entered into database, they must be sanitized to prevent errors and attacks
Priority	0

Applicable FR(s)	1.1.1(,3,5,6,7,8)
------------------	-------------------

## 1.2.2 Message encryption

Table 16: Message encryption

Title	Message encryption
Description	Messages between users must be encrypted. This will prevent bad actors from intercepting messages
Priority	3
Applicable FR(s)	1.1.10

## 1.2.3 Automatic password reset requests

Table 17: Automatic password reset requests

Title	Automatic password reset requests
Description	Emails for password resets must be sent automatically
Priority	0
Applicable FR(s)	1.1.2, 1.1.12

## 1.2.4 Search result time

Table 18: Search result time

Title	Search result time
Description	Search results must be returned in less than 500ms
Priority	0
Applicable FR(s)	1.1.2, 1.1.12

## 1.2.5 Uptime

Table 19: Uptime

Title	Uptime
Description	The application must be down for no longer than 15 minutes each month
Priority	3
Applicable FR(s)	N/A

## 1.2.6 Cookbook Storage

Table 20: Cookbook storage

Title	Cookbook Storage
Description	Database must be able to support dynamic storage resource allocation to adequately store the users' data
Priority	0
Applicable FR(s)	N/A

## 2. System Constraints

### 2.1 Tool Constraints

#### 2.1.1 Mobile Application Framework Constraint

Table 21: Mobile Application Framework Constraint

Title	Mobile Application Framework Constraint
Description	We will be using Apache Cordova as the framework for the application. This allows for the creator to wrap a "web application" into a mobile app
Priority	Mandatory: 0

### 2.2 Language Constraints

#### 2.2.1 Backend Framework

Table 22: Backend Framework

Title	Backend Framework
Description	As we are using Apache Cordova, HTML5, CSS3 and JavaScript are a requirement. The entire backend framework will be written in these languages
Priority	Mandatory: 0

### 2.3 Platform Constraints

#### 2.3.1 Mobile Application Platform

Table 23: Mobile Application Platform

Title	Mobile Application Platform
Description	Apache Cordova is platform independent and allows for cross-platform mobile application creation.
Priority	Lowest: 5

### 2.4 Hardware Constraints

#### 2.4.1 Storage Constraints

Table 24: Storage Constraints

Title	Storage Constraints
Description	Will be set by Amazon Lightsail, but user information will be saved in the database, so this is of minimal importance.
Priority	Lowest: 5

## 2.4.2 Computation Constraints

Table 25: Computation Constraints

Title	Computation Constraints
Description	Set by Amazon Lightsail, sufficient for the purposes of this version of the application and for this course. Simple computations are necessary for this application, such as the ranking of search results.
Priority	Medium: 3

## 2.5 Network Constraints

### 2.5.1 Access Database

Table 26: Access Database

Title	Access Database
Description	Must have access to and query the MySQL database in order to get account, user, and recipe information.
Priority	Mandatory: 0

### 2.5.2 Uploading

Table 27: Uploading

Title	Uploading
Description	Users must be able to upload and share recipes to their accounts, which must be saved to the MySQL database
Priority	Mandatory: 0

### 2.5.3 Messaging

Table 28: Messaging

Title	Messaging
Description	Users must be able to send and receive messages, and share posts
Priority	Mandatory: 0

## 2.6 Deployment Constraints

### 2.6.1 Amazon Lightsail Deployment

Table 29: Amazon Lightsail Deployment

Title	Amazon Lightsail Deployment
-------	-----------------------------

Description	This mobile application will be deployed by Lightsail. Lightsail provides a free month of usage which will be sufficient for the purposes of this course.
Priority	Medium: 3

## 2.7 Transition & Support Constraints

### 2.7.1 End of Life

Table 30: End of Life

Title	End of Life
Description	As this project will end at the termination of this course in May, another team will have to take over in order to develop the mobile application.
Priority	Lowest: 5

## 2.8 Budget & Schedule Constraints

### 2.8.1 Time Constraints

Table 31: Time Constraints

Title	Time Constraints
Description	Application must be completely designed before the end of the semester (May 2021). A final project packet with a Requirements Document and Design Document must be completed by then.
Priority	Mandatory: 0

### 2.8.2 Budget Constraints

Table 32: Budget Constraints

Title	Budget Constraints
Description	There are no funds available for this project, therefore there is no budget.
Priority	Lowest: 5

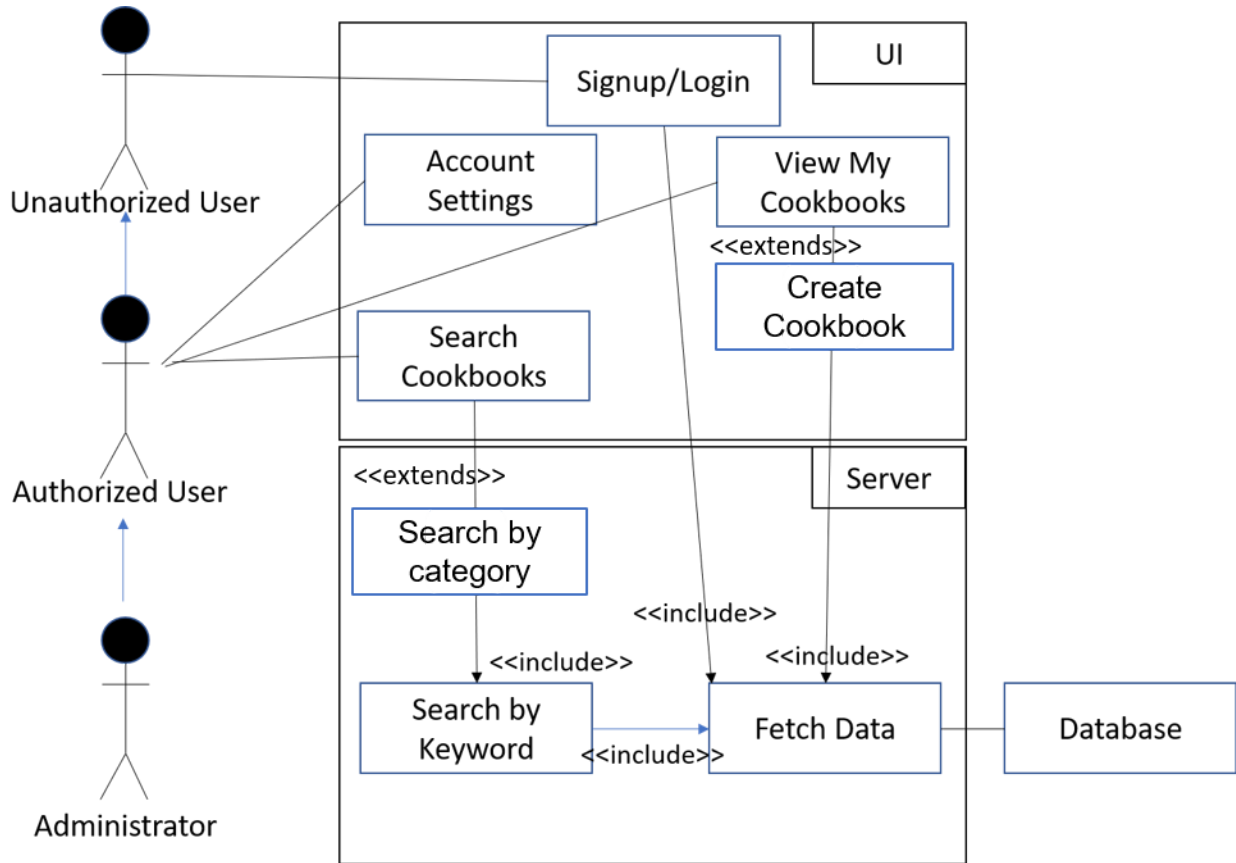
## 3. Requirements Modeling

### 3.1 Primary Use Case (create and view cookbook)

Table 33: Primary Use Case

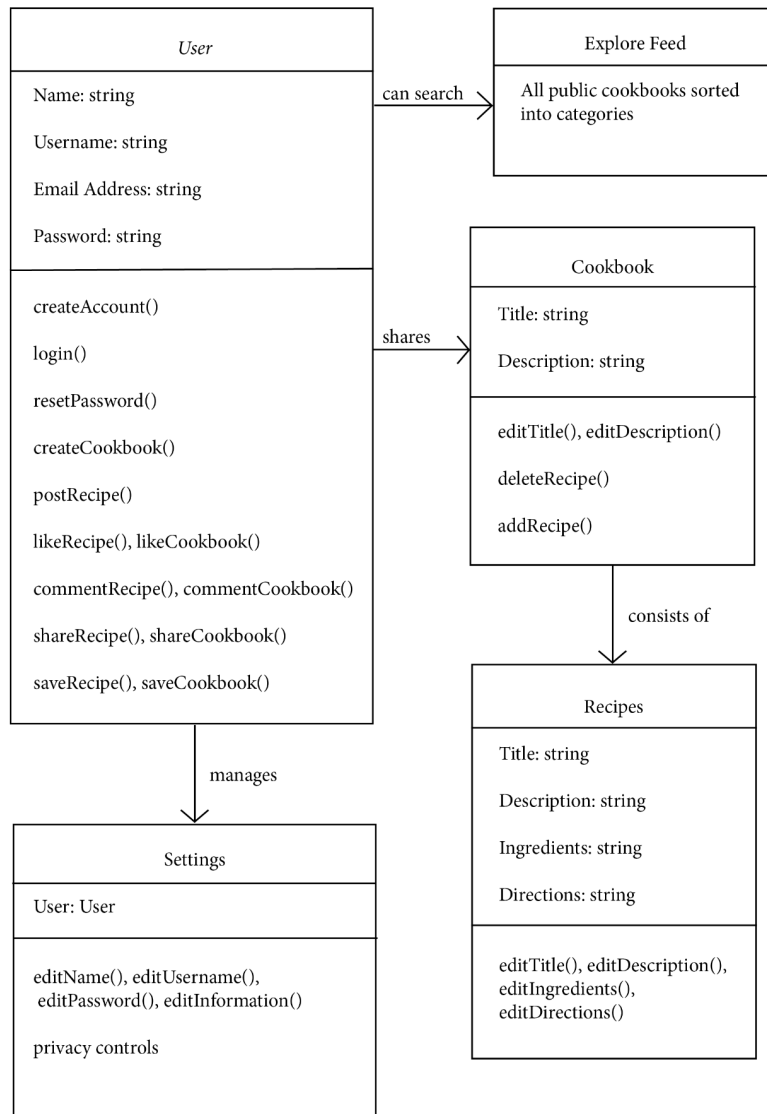
Title	WeCook create and view cookbook
Description	This the primary use case for the WeCook App
Actors	The Administrator, the Authorized User, and the Unauthorized user
Trigger	This use case is initiated when the User starts the app.
Precondition(s)	The user has downloaded the app onto their device via the app store.
Basic Flow	<ol style="list-style-type: none"><li>1. If the user is unauthorized, the user selects Log in or Sign up</li><li>2. Once authorized, the user can view their cookbooks by clicking the "view my cookbooks" button</li><li>3. From cookbook list, the user can create a cookbook by clicking "create a cookbook"</li><li>4. The user can add recipes to a cookbook, by clicking the "add a recipe" button.</li><li>5. The user can save, delete, or publish cookbooks by syncing cookbook data with WeCook database.</li></ol>
Exceptions	If the user does not have the proper authentication to use the app, they must create an account before using any of WeCook's features. If there are errors with signing up/logging in the user will be suggested to try again at another time.
Postcondition(s)	The user has saved, deleted, or published their cookbook, and the change has been synced to the central database
Special Requirements	App requires that the user is authorized before using any features. There is a 255 cookbook limit for authorized users.

## 3.2 WeCook Application System





### 3.3 Class Diagram



## 4. Evolutionary Requirements

### 4.1 Web application

Table 34: Web Application

Title	Web application
Description	The mobile application will be ported to a web application to allow non-mobile users to participate on WeCook.
Priority	5
Applicable FR(s)	None

### 4.2 Elderly accessibility mode

Table 35. Elderly accessibility mode

Title	Elderly accessibility mode
Description	The mobile application will be optimized for elderly users (increased font size, contrast)
Priority	5
Applicable FR(s)	Account and Settings modification