"As a [user] I want to [post my recipes] to a wider range of people."
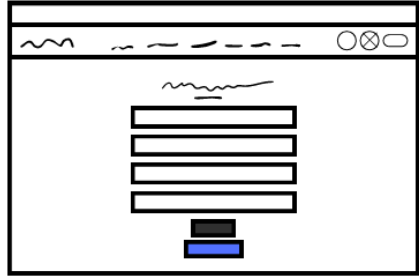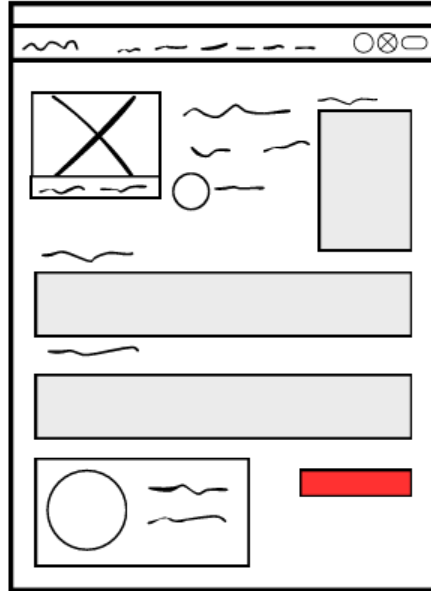
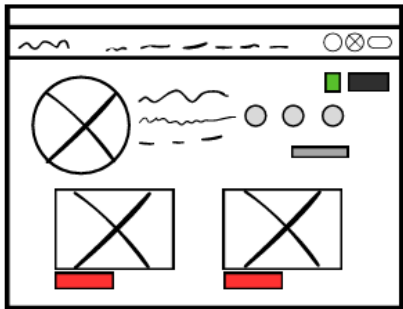>>Account creation/Login

>>Recipe post page with sections to fill

>>View profile

Use case scenarios

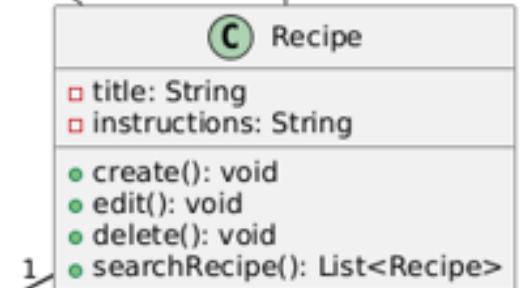**Register**

Main Actor: User (new user)
Steps:
1. User navigates to sign up page.
2. User enters account details, email, username and password.
3. System validates correct inputs.
4. New user account is created and directed to their profile.
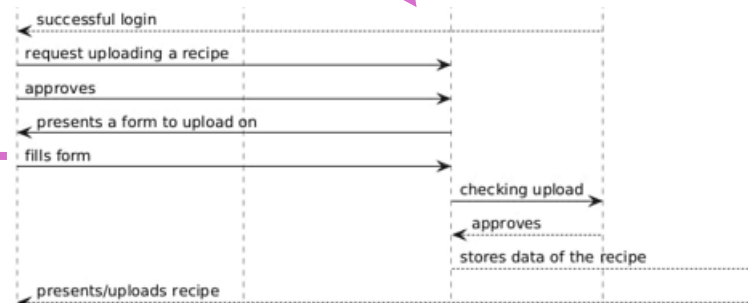
Post Recipe

Main Actor: User
Steps:
1. User navigate to profile.
2. Users select 'Add recipe'.
3. User is redirected to Recipe posting page to fill out each section.
4. Recipe is posted onto their profile and category page.

C Recipe
- title: String
- instructions: String
- create(): void
- edit(): void
- delete(): void
- searchRecipe(): List<Recipe>

Class diagram representation

Sequence diagram incorporation
- successful login
- request uploading a recipe
- approves
- presents a form to upload on
- fills form
- checking upload
- approves
- stores data of the recipe
- presents/uploads recipe

Structure mapping

Homepage
Recipe Swap
Recipe Explore
Cuisines
Recipe Listings

- User login or create their account.
- go to their profile and create a post. They must then fill in the required sections.
- They can click publish when done.
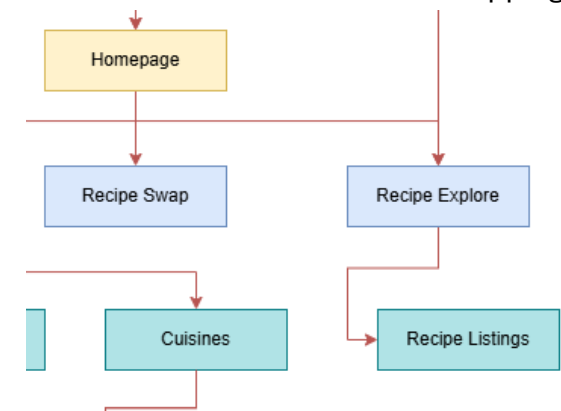
"As a [user] I want to [browse recipes] with a filtered approach."

>>Homepage view

Use case scenarios

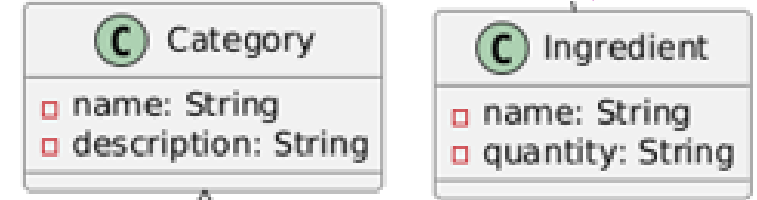**View Recipes**

Main Actor: User
Steps:
1. User navigates to Recipe page by selecting category.
2. User scrolls through recipes.
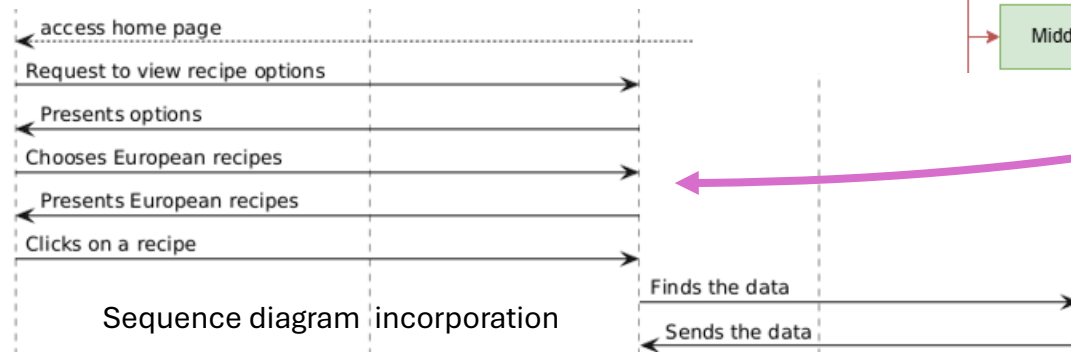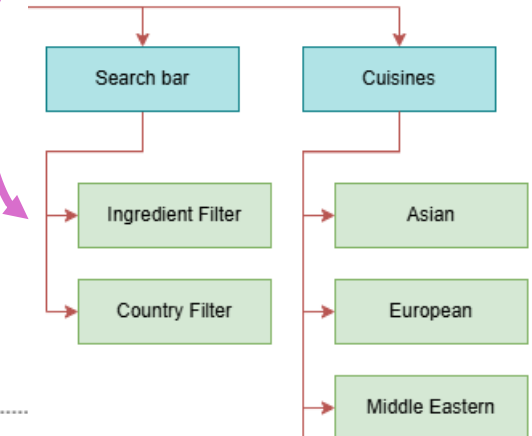3. User can click on recipe to view further details of chosen recipe.

Filter Recipes

Main Actor: User
Steps:
1. User navigates to Recipe page by selecting category.
2. User uses filtering option to select filters of choice (etc. dietary requirements of specified ingredients).
4. The page will only display the filtered down recipes.

C Category
- name: String
- description: String

C Ingredient
- name: String
- quantity: String

Class diagram representation

Structure mapping

Search bar | Cuisines

Ingredient Filter | Asian

Country Filter | European

Middle Eastern

>>Category view with filtering/search

- Users select a category.
- Users can further filter options via search bar or selection.

access home page
Request to view recipe options
Presents options
Chooses European recipes
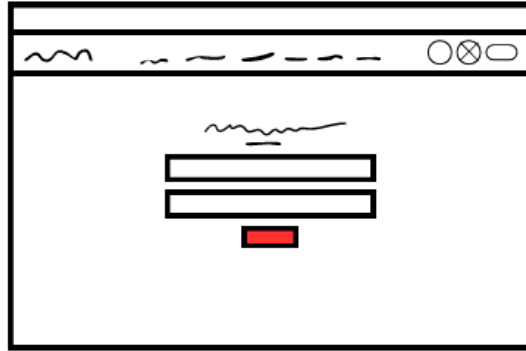Presents European recipes
Clicks on a recipe

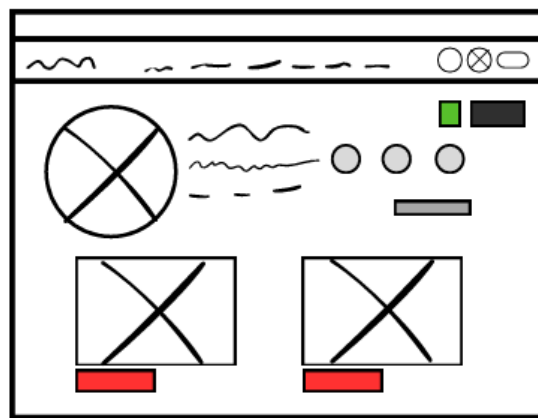Finds the data
Sends the data

Sequence diagram incorporation

"As a [user] I want to [view my profile] where I can [edit my details and posts]."

>>Login page

>>Recipe post page

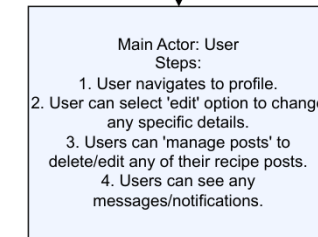>>Profile view

Use case scenarios

Login

Main Actor: User
Steps:
1. User navigates to login page.
2. User enters username and password.
3. System validates details and checks account exists.
4. User is directed to their profile.

Manage Profile

Main Actor: User
Steps:
1. User navigates to profile.
2. User can select 'edit' option to change any specific details.
3. Users can 'manage posts' to delete/edit any of their recipe posts.
4. Users can see any messages/notifications.

Class diagram representation

C User

- name: String
- email: String
- password: String

- register(): void
- login(): void
- swapRecipe(): void
- updateProfile(): void
- resetPassword(): void

Sequence diagram incorporation

- successful login
- request uploading a recipe
- approves
- presents a form to upload on
- fills form
- checking upload
- approves
- stores data of the recipe
- presents/uploads recipe

Activity diagram application

- Login Attempts i = 0
- Display Login Form
- User Enters Details
- Have Details Been Provided?
  - yes
  - no → Highlight Required Fields

- Users' login with their existing details or they can create an account.
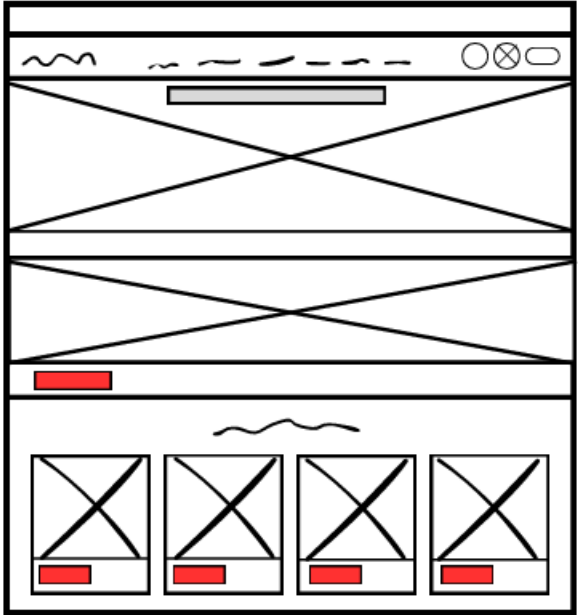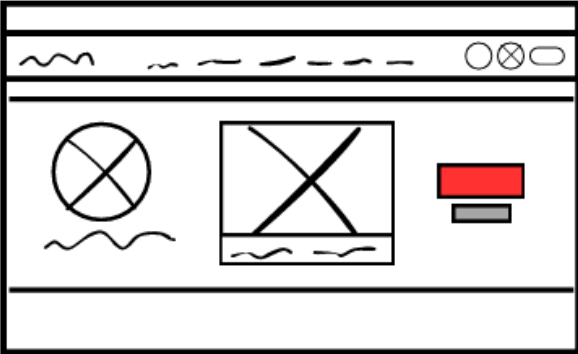- They navigate to their profile page to view current posts.
- They select a post they want to edit.
- All filled in sections are available for editing.
- They republish when done and can save and exit.

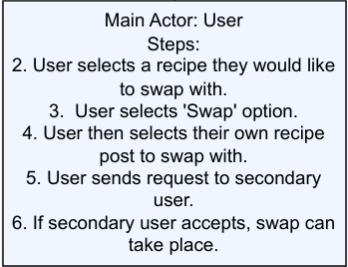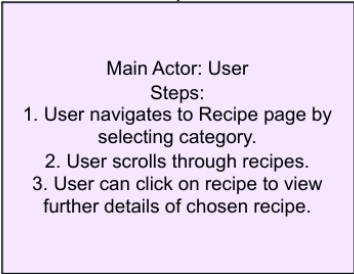# "As a [user] I want to [select a recipe] so that I can [swap my own with it]."
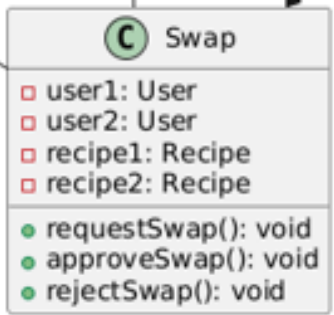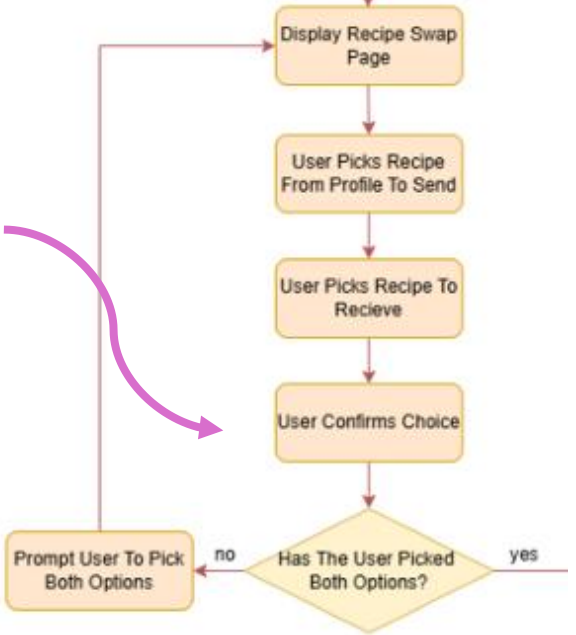
>>Recipe browsing

View Recipes

Swap Recipe

Class diagram representation

Swap
- user1: User
- user2: User
- recipe1: Recipe
- recipe2: Recipe

- requestSwap(): void
- approveSwap(): void
- rejectSwap(): void

Use case scenarios

Main Actor: User
Steps:
1. User navigates to Recipe page by selecting category.
2. User scrolls through recipes.
3. User can click on recipe to view further details of chosen recipe.

Main Actor: User
Steps:
2. User selects a recipe they would like to swap with.
3. User selects 'Swap' option.
4. User then selects their own recipe post to swap with.
5. User sends request to secondary user.
6. If secondary user accepts, swap can take place.

Activity diagram application

>>Swap requesting

Sequence diagram incorporation

successful login db->auth : verified successful

successful login

request recipe swap

approve request

Display Recipe Swap Page

User Picks Recipe From Profile To Send

User Picks Recipe To Recieve

User Confirms Choice

Prompt User To Pick Both Options   no   Has The User Picked Both Options?   yes

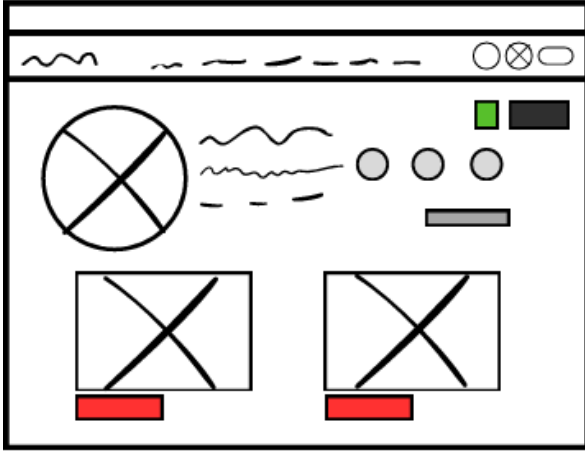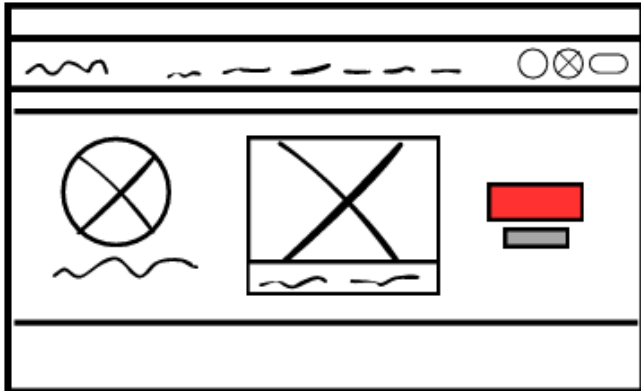- Users browse through recipes.
- They select the one they would like to swap with.
- They pick their own post that they will be exchanging.
- They send the request.

"As a [user] I want to [leave reviews] after trying out a recipe swap."

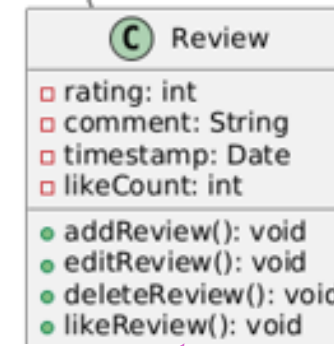>>Profile view

Manage Profile

Leave Review

Use case scenarios

Main Actor: User
Steps:
1. User navigates to profile.
2. User can select 'edit' option to change any specific details.
3. Users can 'manage posts' to delete/edit any of their recipe posts.
4. Users can see any messages/notifications.

Main Actor: User
Steps:
1. After successful swap completed, user can select 'leave a review' under swapped recipes post.
2. Pop up allows user to input feedback text.
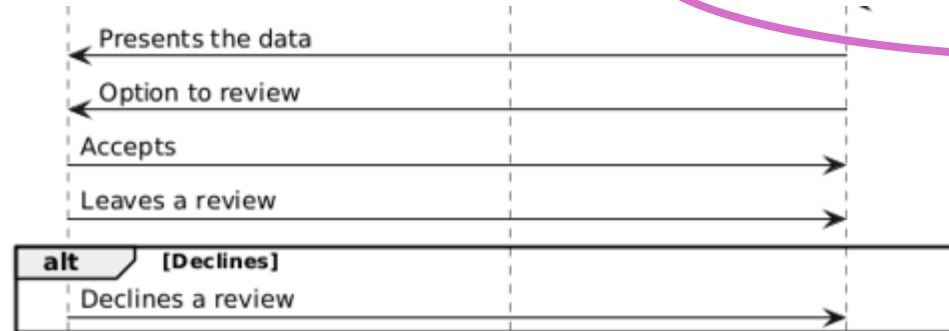3. User sends review and it is available on secondary users profile/post.

Class diagram representation

© Review
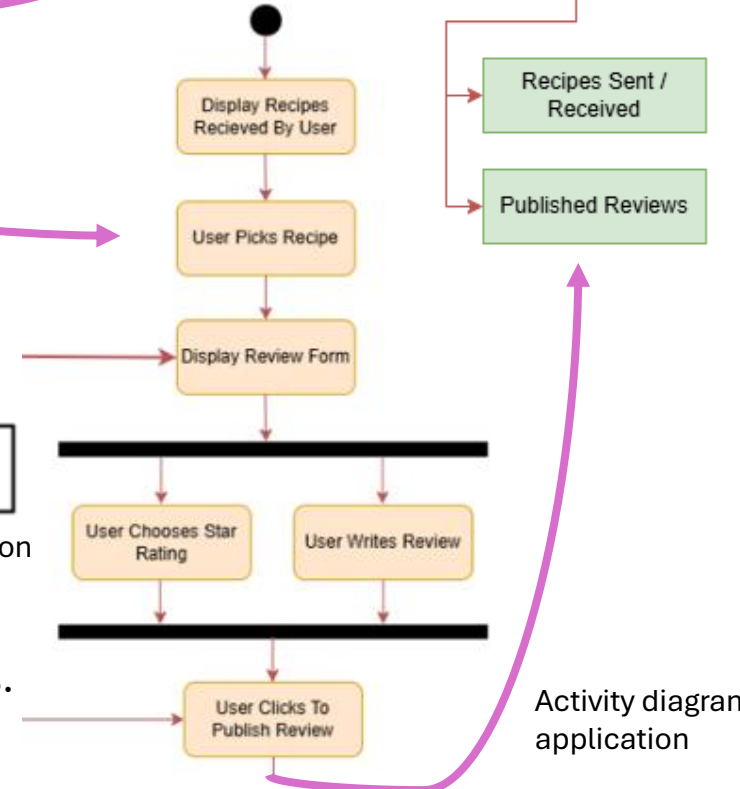- rating: int
- comment: String
- timestamp: Date
- likeCount: int

- addReview(): void
- editReview(): void
- deleteReview(): void
- likeReview(): void

Structure mapping

User Profile

Recipes Sent / Received

Published Reviews

>>Review recipe swap

Presents the data

Option to review

Accepts

Leaves a review

alt [Declines]
Declines a review

Sequence diagram incorporation

Display Recipes Recieved By User

User Picks Recipe

Display Review Form

User Chooses Star Rating
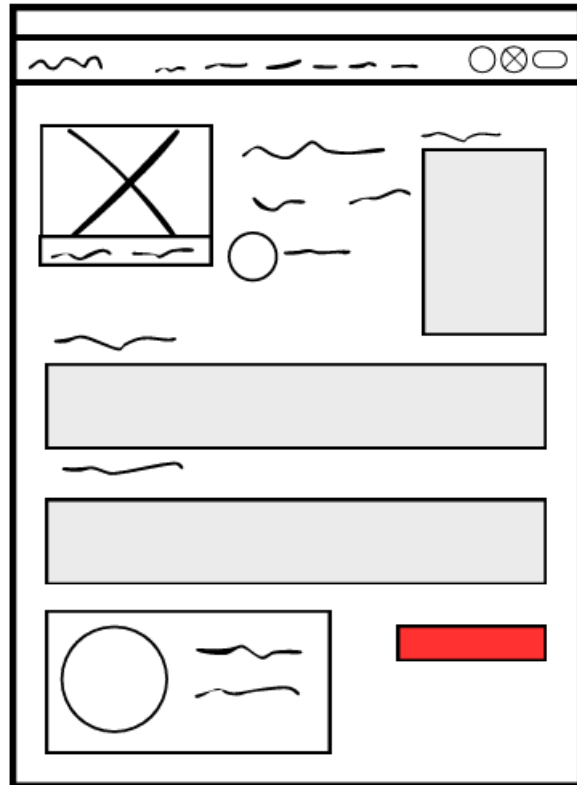
User Writes Review

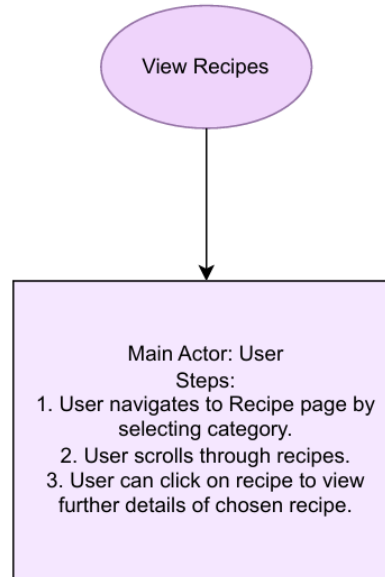User Clicks To Publish Review

Activity diagram application

- Users go to profile to view completed recipe swaps.
- Users select the recipe's review option.
- Users fill in review and publish.

"As a [user] I want to [easily navigate the instructions of a recipe] with titled sections."
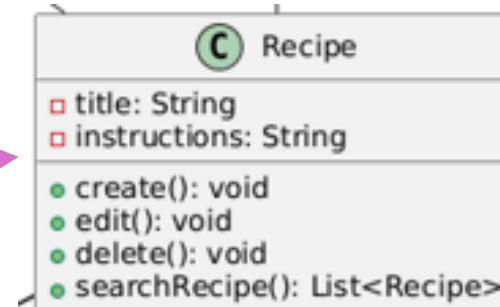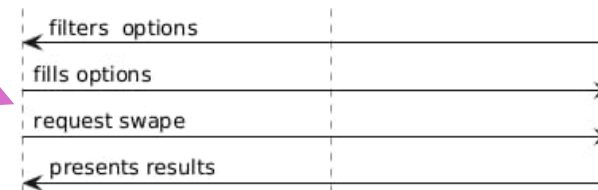
>>Recipe post

Use case scenario

View Recipes

Main Actor: User
Steps:
1. User navigates to Recipe page by selecting category.
2. User scrolls through recipes.
3. User can click on recipe to view further details of chosen recipe.

Class diagram representation

C Recipe

- title: String
- instructions: String

- create(): void
- edit(): void
- delete(): void
- searchRecipe(): List<Recipe>

Sequence diagram incorporation

filters options

fills options

request swape

presents results
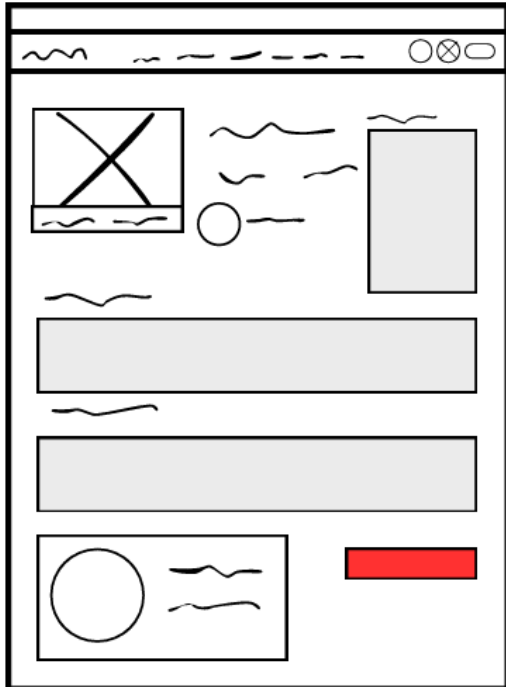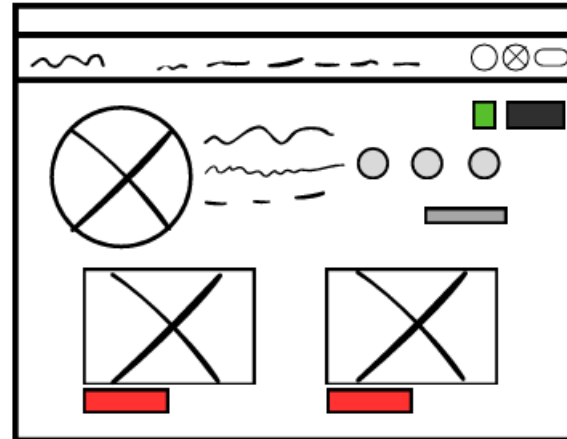
- Users can click on a recipe.
- They will be directed to the recipe's full post which has all details in specified sections.

"As a [user] I want to [view other's profiles] so that I can [network with them and follow more of their recipes]."
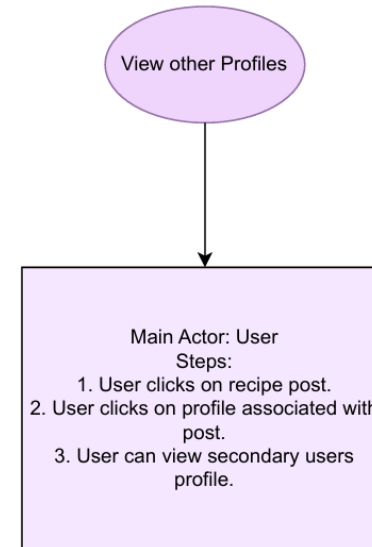
>>Recipe post, publisher below

>>Publisher profile

Use case scenario

View other Profiles

Main Actor: User
Steps:
1. User clicks on recipe post.
2. User clicks on profile associated with post.
3. User can view secondary users profile.

Clicks on a recipe

Finds the data

Sends the data

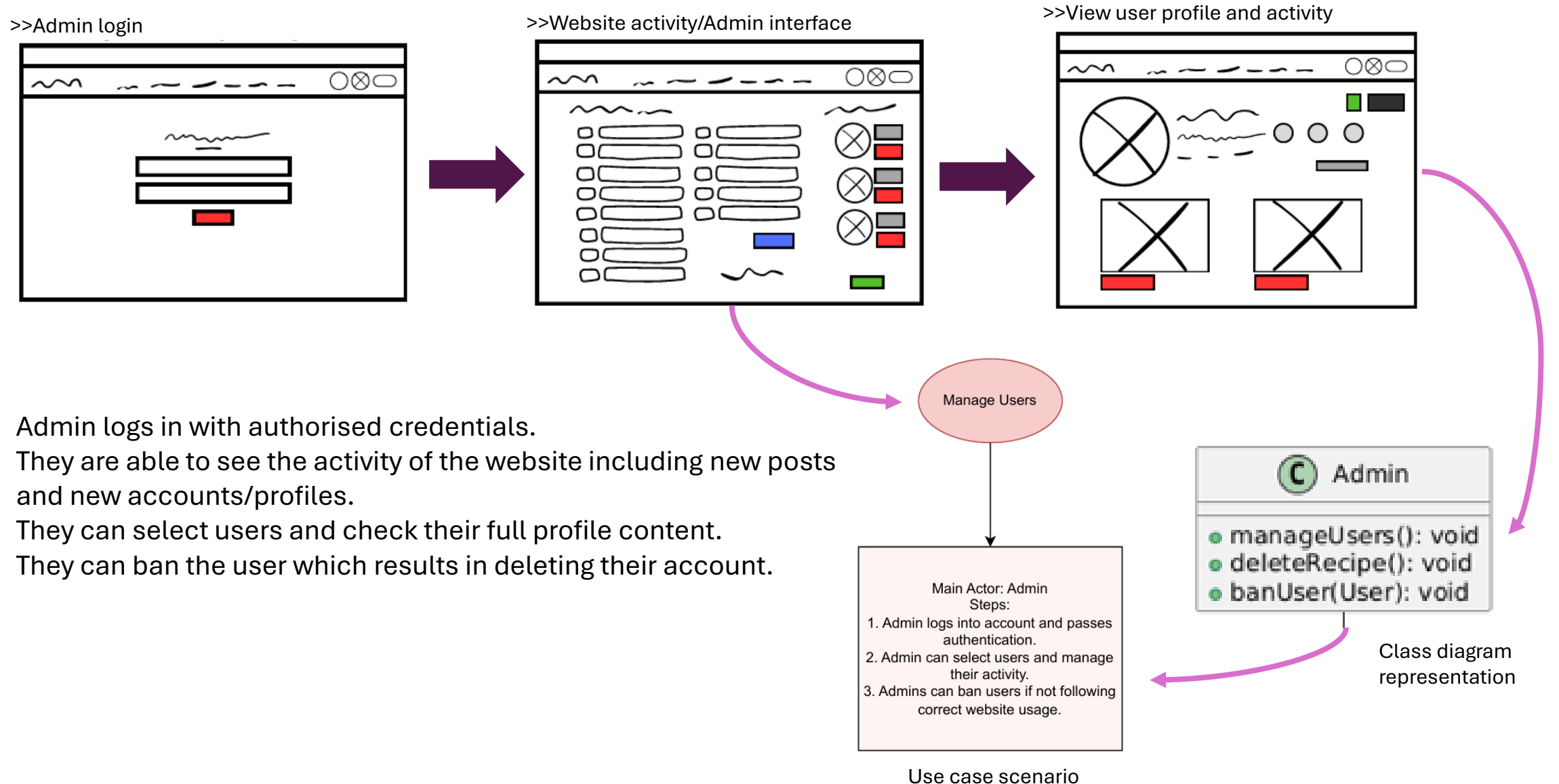Presents the data

Sequence diagram incorporation

- Users select a recipe and view full details.
- They scroll down and can see the 'publisher' of the recipe.
- They can click on their profile.
- This redirects them to the recipe publishers' profile.
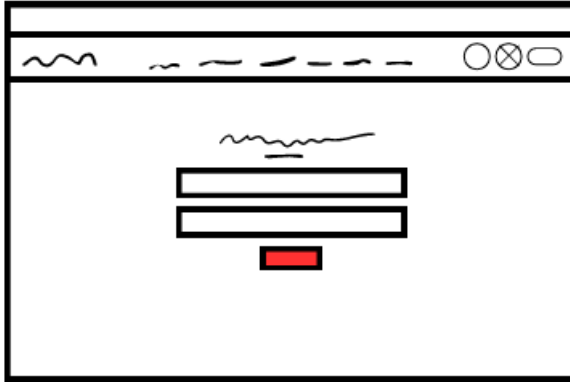- They can view details of them, including networking options.
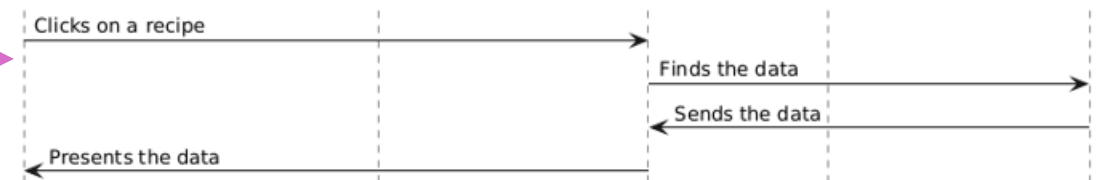
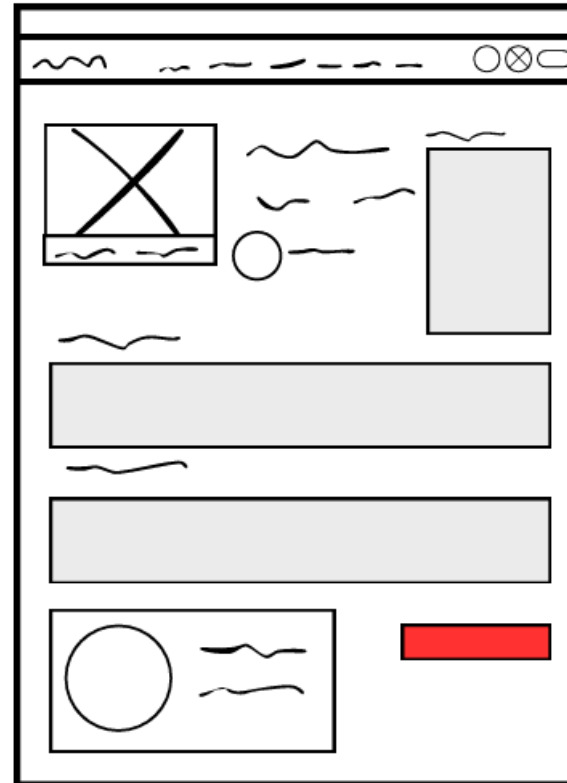# "As an [admin] I want to [manage users] so that I can [check their activity and ban any unauthorised users]."

>>Admin login

>>Website activity/Admin interface

>>View user profile and activity

- Admin logs in with authorised credentials.
- They are able to see the activity of the website including new posts and new accounts/profiles.
- They can select users and check their full profile content.
- They can ban the user which results in deleting their account.

Manage Users

Main Actor: Admin
Steps:
1. Admin logs into account and passes authentication.
2. Admin can select users and manage their activity.
3. Admins can ban users if not following correct website usage.

Use case scenario

C Admin

- manageUsers(): void
- deleteRecipe(): void
- banUser(User): void

Class diagram representation

"As an [admin] I want to [delete recipe posts] if they do not meet the guidelines."

Use case scenario
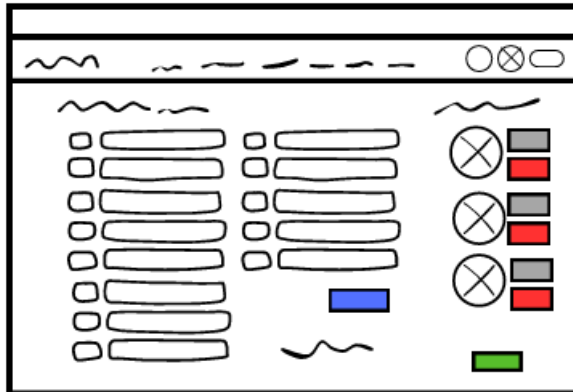
>>Admin login

>>Recipe post

>>Website activity/Admin interface

Delete Recipe Posts

Main Actor: Admin
Steps:
1. Admin navigates through recipe pages.
2. Admin selects a post.
3. Admin has option to 'delete post'.

© Admin
● manageUsers(): void
● deleteRecipe(): void
● banUser(User): void

Class diagram representation
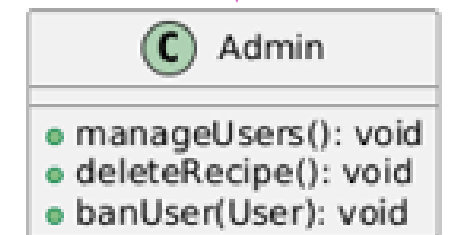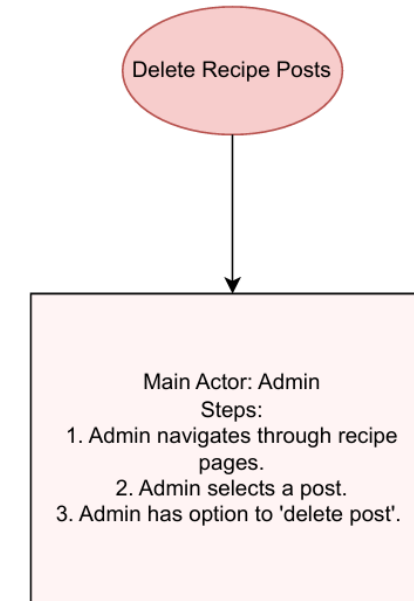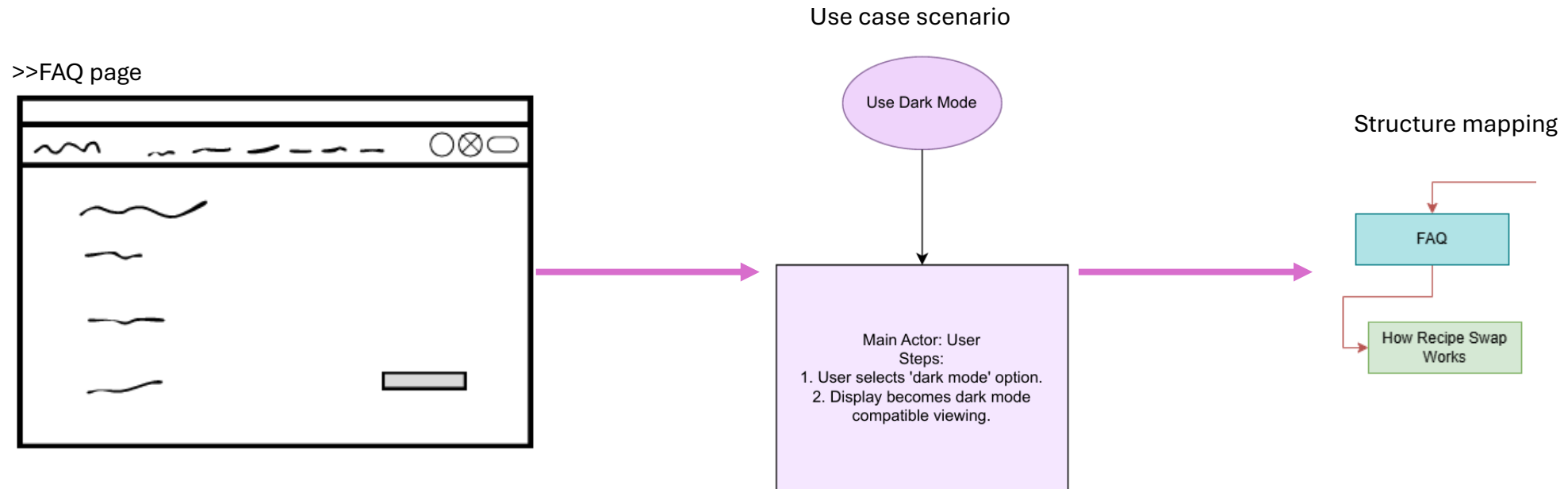
- Admin logs in with authorised credentials.
- They are able to see the activity of the website including all accounts.
- They can select users and check their posts.
- They can view the post and delete if it doesn't meet guidelines.

"As a [user] I want to [use a dark-mode option] to [decrease my eye strain]."

Use case scenario

>>FAQ page

Use Dark Mode

Structure mapping

FAQ

How Recipe Swap Works

Main Actor: User
Steps:
1. User selects 'dark mode' option.
2. Display becomes dark mode compatible viewing.

- Users can click the toggle on the top right for dark mode.
- If they are unsure, they can go to FAQs.
- Instructions will be found in the FAQs.