



MARCH 30, 2024

DESIGN DOC

DISHYDISHES


GROUP 17

JIE ZHANG

ZIRUI HE

KAICHENG XU

SHJUREN XU



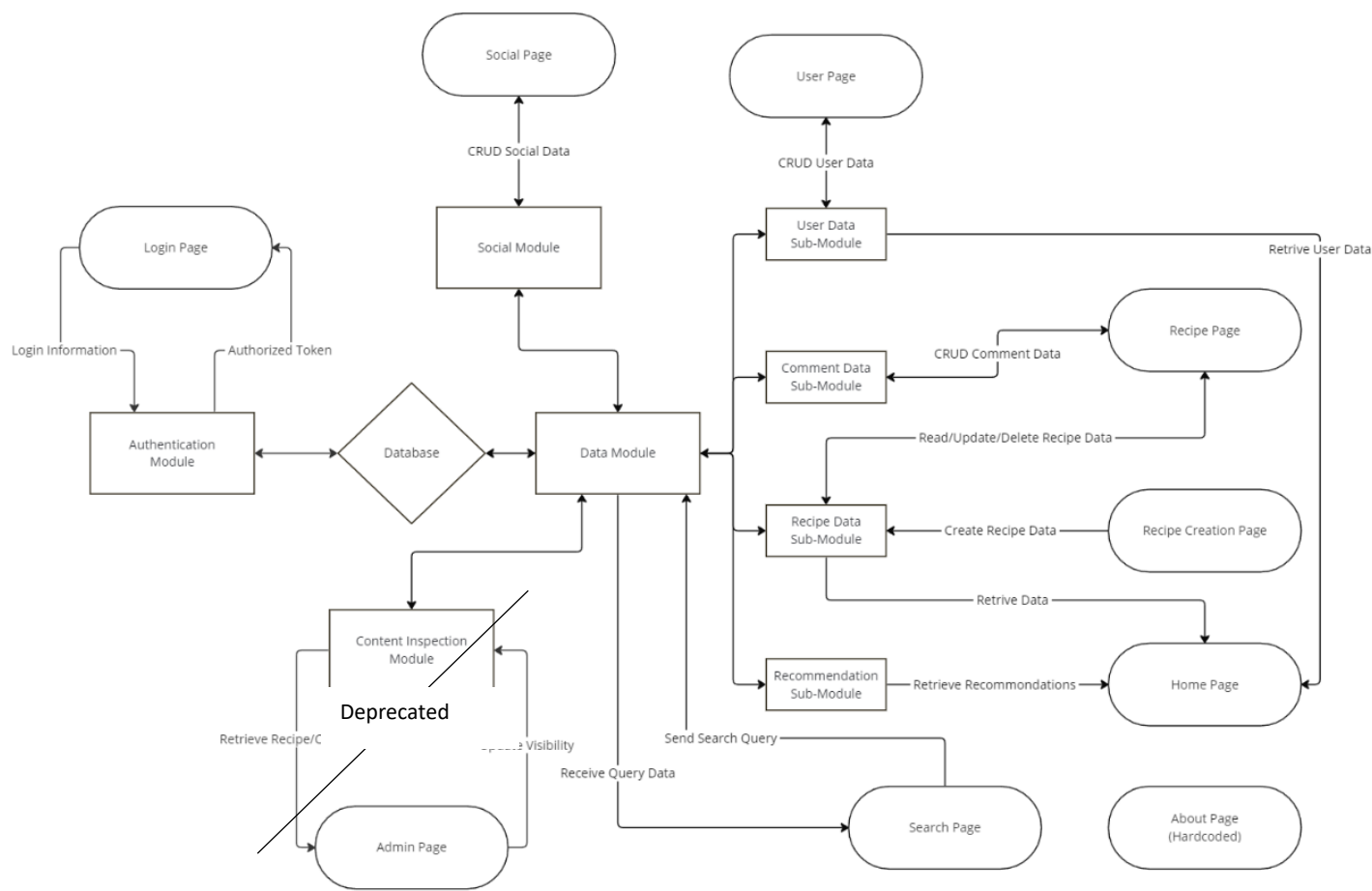
Revision History

Version	Date	Notes
0	Jan 24, 2024	Initial design document
1	Mar 30, 2024	Final Update based on actual progress

1. Purpose Statement.

Dishydishes helps users find good recipes and share their experience in cooking. This project aims to deliver a smooth, intuitive user experience, along with appealing, modern aesthetic design.

2. Component diagram: Components and their relationships.



3. Relationship between components and requirements

Authentication module

Functional Requirements:

Users should be able to register and login to access personalized content.

Usability and Humanity Requirements:

The registration and login interface should be user-friendly. Then buttons should be easy to find and perform the functions indicated by the buttons. If an error occurs, there should be clear messages and guidance for the next steps.

Security and Privacy:

Encryption should be added to protect users' password. Users should be able to control the visibility of their profiles and information.

Site information module: user, recipe, comment

Functional Requirements:

This module should store user information, recipes, and comments, keeping synchronized with the database.

Usability and Humanity Requirements:

The navigation should be intuitive to find recipes, comments under recipes and view user profiles. Accessibility features should be added, like text resizing and color contrast for readability.

Cultural Requirements:

For some cultural cuisines and cooking styles, special logo should be provided. The website should accept different languages like French, Chinese, and Indian.

Content inspection module

Functional Requirements:

The website should allow users to manage the visibility of recipes and comments and notifies users about changes in their content status.

Security and Privacy:

Ensures that user-generated content follows the privacy guidelines set by users and the website. Sensitive information can be filtered by users.

Social module

Functional Requirements

This module should facilitate user interactions through features like following other users, sharing recipes, and participating in forums or discussion groups, allowing users to rate, comment on, and share recipes. Users should be able to see the new posts of the uploader they followed. Sharing posts to other social media platforms should be allowed. Notification should be provided about new followers, comments, or likes on their recipes and posts.

Usability and Humanity Requirements

The website should provide easy navigation and interaction, ensuring that users of all ages and tech abilities can engage with the community. We tailor recommendations and feeds based on user preferences and previous interactions.

Cultural Requirements

The website should support various cultural perspectives and encourage the sharing of different culinary traditions.

Legal

This website should ensure user-generated content adheres to legal standards and community guidelines. We respect copyright laws, especially in user-submitted recipes and images.

Overall Considerations

Look and Feel Requirements:

We want a consistent, modern aesthetic across all pages. We want a responsive design for a variety of devices to ensure a seamless experience on desktops, tablets and smartphones.

Performance and Speed Requirements:

Pages should load quickly, especially for recipe browsing and searching. Information updates should be near real-time, with changes observed within seconds. Backend processing for user registration, login and content management should be efficient.

4. Component Description

Authentication module

a. Normal behavior:

- Users can register, log in, and log out.
- The system securely handles user credentials.
- Provides feedback on successful or failed authentication attempts.

b. API (input/outputs with emphasis on data structures being exchanged between components).

- Inputs: TLS encrypted login data
- Outputs: Bcrypt hashed token
- Data Structures: String

- c. Implementation (Include details: classes, methods, and their relationships).
 - Classes: UserController, UserService, UserRepo
 - Methods: login, createToken, validateToken
 - Relationships: UserController handles post request and call UserService, UserService handles calculations such as verify login information, create and validate token, UserRepo is the bridge between the database and the backend.
- d. Potential undesired behaviours.
 - Unauthorized access due to weak password standards.
 - Token Leakage
 - Failure to log out users properly, leading to security risks.

Data module

- a. Normal behavior:
 - Displays user information, recipes, and comments.
 - Allows users to search and filter recipes.
 - Users can post, edit, and delete their recipes and comments.
- b. API (input/outputs with emphasis on data structures being exchanged between components).
 - Inputs: user id, recipe id, comment id, search query
 - Outputs: user data, recipe data, comment data, search results
 - Data Structures: string, User object, recipe object, comment object
- c. Implementation (Include details: classes, methods, and their relationships).
 - Classes: Controller classes, service classes, repository classes for user, recipe and comment
 - Methods: createUser, createRecipe, createComment, getUserById, getRecipeById, getCommentById, getUserByUsername, searchUserByAlias, searchRecipeByName
 - Relationships: Each controller class handles https request and call corresponding service class, which handles calculations. Repository classes are the bridges between the database and the backend.
- d. Potential undesired behaviours.
 - Incorrect or incomplete recipe information.
 - Inefficient search algorithm.
 - Inability to update or delete user-generated content.

~~Content inspection module~~

- a. ~~Normal behavior:~~
 - ~~• Filter newly posted recipes and comments.~~
 - ~~• Send suspicious posts to admin page.~~
 - ~~• Site administrators can hide or approve posts during inspections.~~

b. API (input/outputs with emphasis on data structures being exchanged between components).

- Inputs: hide/approve status
- Outputs: suspicious posts
- Data Structures: boolean, recipe object, comment object

c. Implementation (Include details: classes, methods, and their relationships).

- Classes: RecipeService, CommentService
- Methods: filter, markSuspicious, updateVisibility
- Relationships: RecipeService and CommentService simultaneously create new post and apply filter to its content, mark suspicious content and wait for approval.

d. Potential undesired behaviours.

- Faulty alarm by filter deviation.
- Inefficient filter algorithm.

Social module

a. Normal behavior:

- Push new posts posted by followed creators.
- More pending

b. API (input/outputs with emphasis on data structures being exchanged between components).

- Inputs: user id
- Outputs: new posts posted by followed creators.
- Data Structures: string, recipe object

c. Implementation (Include details: classes, methods, and their relationships).

- Classes: UserService
- Methods: getFollowedNewPosts

d. Potential undesired behaviours.

- Delayed Updates.
- Broken Links or Content.

5. User interface

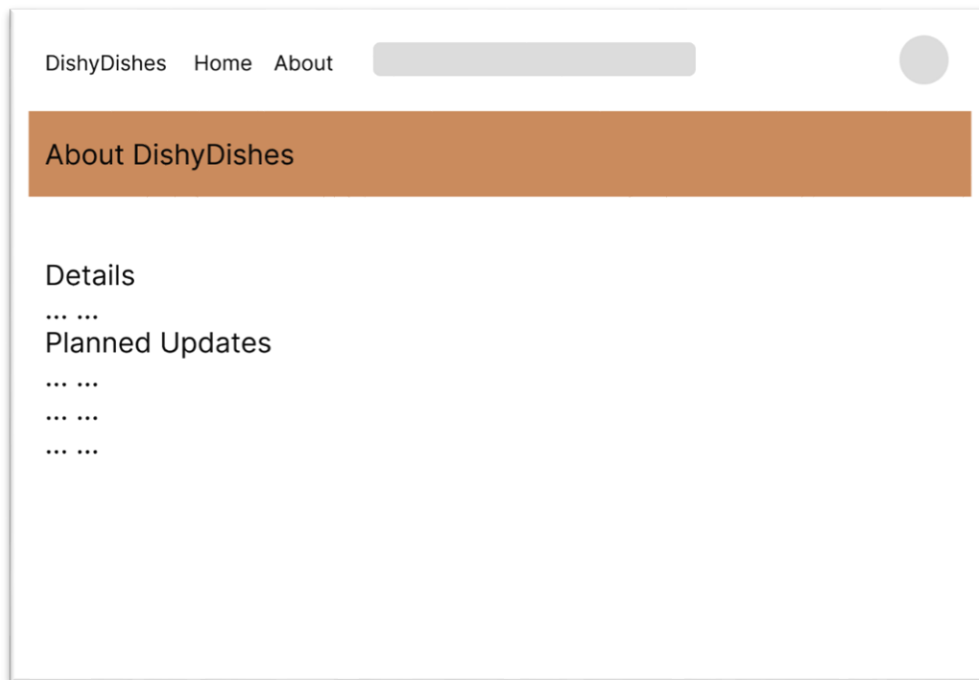
For the whole design, we chose a light brown palette because it reminds people of steak, barbecue, chestnut, and other warm and delicious foods. Also, low-saturation pink, blue, and green colors are used on the buttons, which can easily remind people of some certain kinds of food like dessert, and vegetables. It is more in line with the platform we wanted to build for all kinds of people to share recipes.

For font, we decide to use a font family of 'Avenir', 'Helvetica', 'Arial', and 'sans-serif'. A user agent will iterate through the prioritized list until matching an available one.

Specific UI designs are attached in appendix with figma charts.

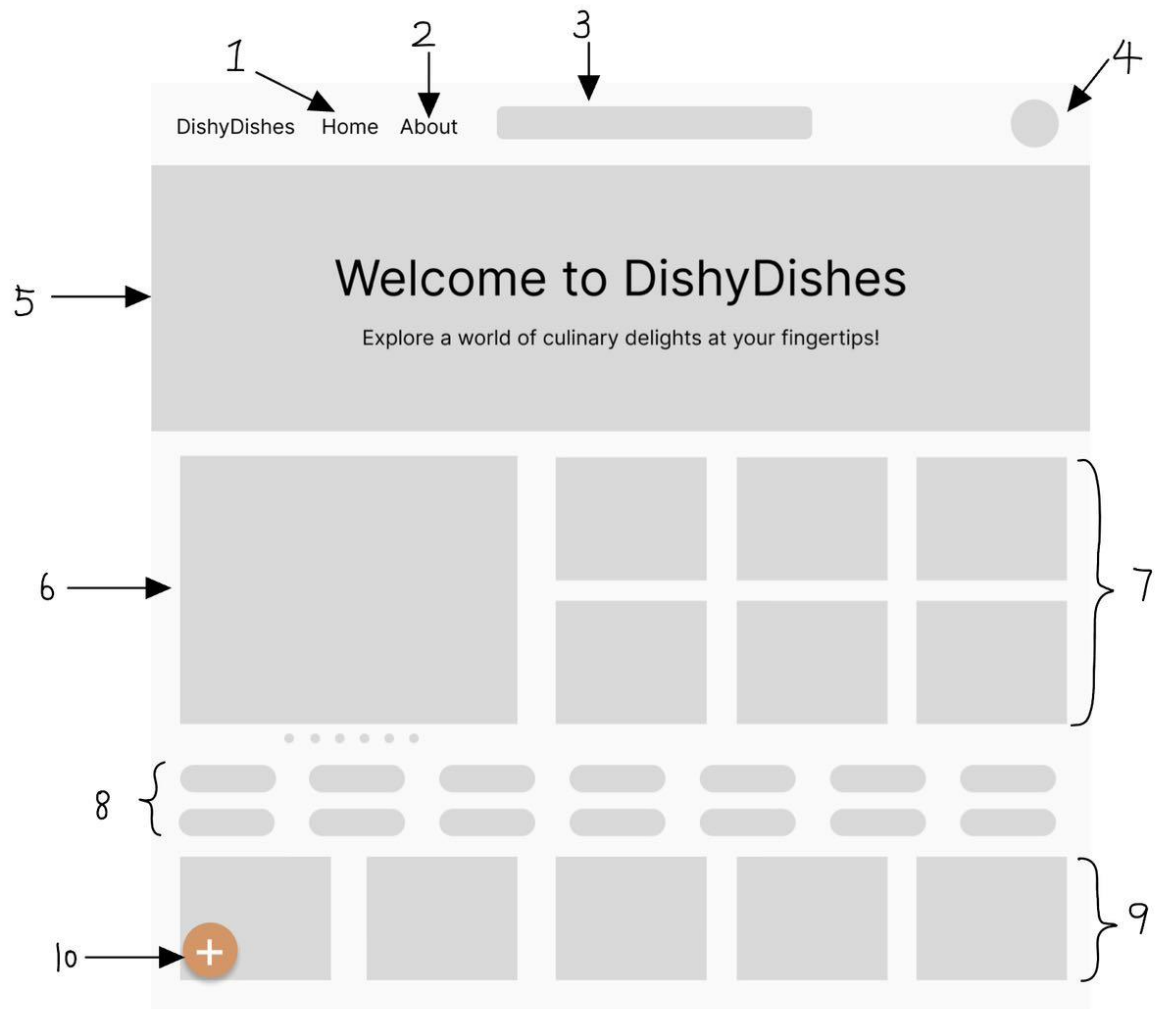
Appendix

About Page: where we leave announcements to users.

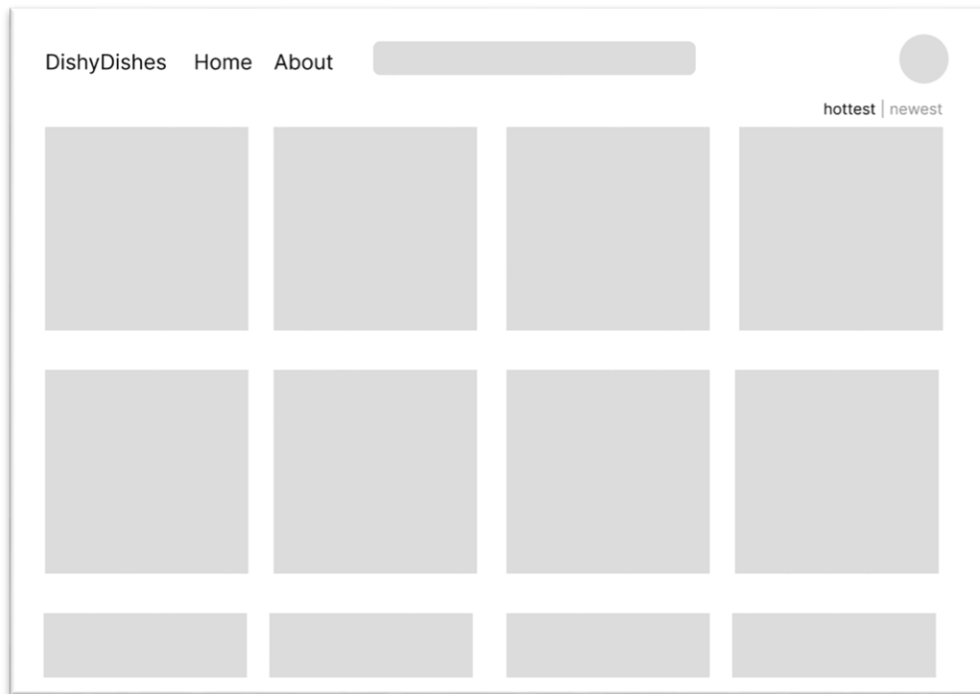


Home Page:

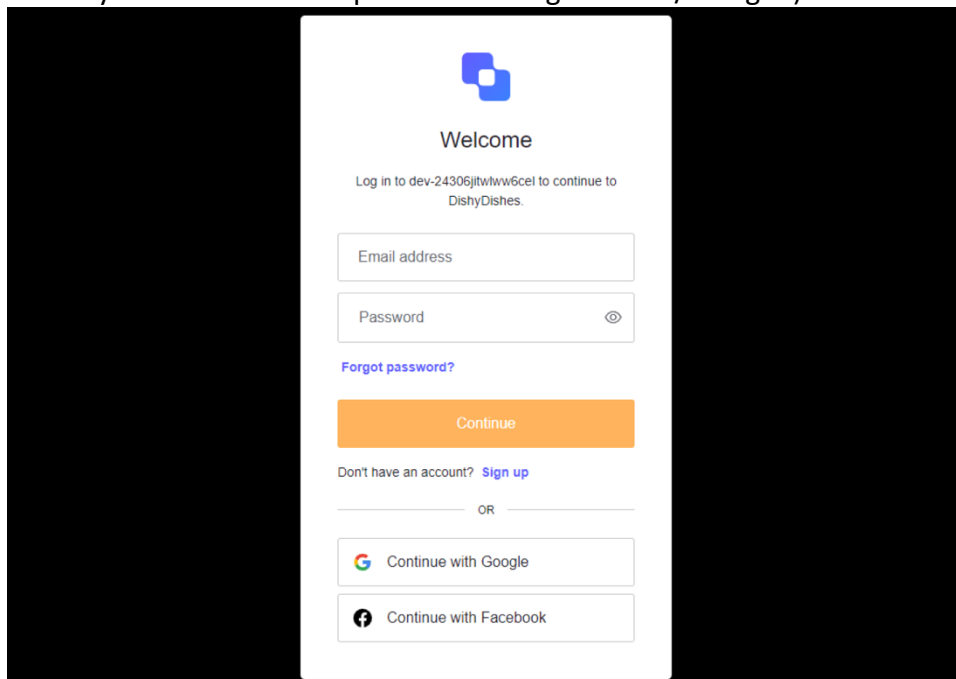
1. A link to the Home page;
2. A link to the About page;
3. A search bar; The search results will show in Search results page;
4. It was a button shown as "Login" before the user logs in; by clicking it, user will go to Login page; After login, this button will become a user profile photo, also a link to the user page;
5. A food photo with welcome message on top, this photo is most likely to be brown palette, but can be changed during holidays;
6. A carousel to list 6 most liked posts in a week;
7. A show dashboard to show 6 random posts posted in 24 hours;
8. Category buttons with links to Search page;
9. 5 most recent recipes posted by user's followees;
10. A '+' button with link to Recipe Creation page.



Search Results Page: where the results of the search query get displayed by most liked/recent.



Login Page: where user can sign up and/or log in. (Our login page temporarily skipped auth0 and rely on username and password to register and/or login.)



User Page: where a user's personal preference, recipe posts, favourite recipes, comments, followers, and followees shows.

DishyDishes

Home

About

Name

Preferences

FOLLOW

Followers:

Following:

MY POST

FAVOURITE

COMMENTS

Admin Page: Recipes and comments that are detected as substandard or reported will appear on this page and will be tested by the administrator to see if they should be visible or not.

ID	NAME	PASS	FAIL

ID	COMMENTS	PASS	FAIL

Recipe Creation Page: where users can create their recipe by providing information including recipe name, description, ingredients & corresponding amount, related tags, and step-by-step instructions.

[DishyDishes](#) [Home](#) [About](#)

Enter your recipe name

Discription

Ingredients

-	-	Remove
-	-	Add

Steps

1. step1

Click to upload an image

Remove

Add

Related Tags

x

Add

Submit Recipe

Recipe View Page: where users can view & bookmark a recipe. Some other information such as author, postdate, and how many people bookmarked this recipe. Also, there is a section for users to see and leave comments.

recipe name

Discription

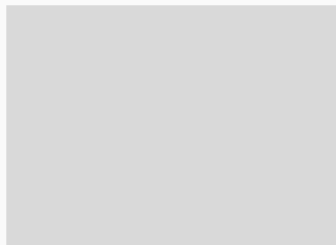
This is a discription.

Ingredients

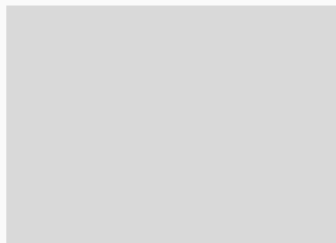
-	-
-	-
-	-

Steps

1. step1



2. step2



Comments 789

hottest | newest



User1
dd/mm/yyyy

456



This is a comment.