University College Dublin
Ireland's Global University

University College Dublin

COMP3030J Software Engineering Project 2023-2024

Group_14_user_documentation

Author:
Ruotong Sun, Shizheng Wang, Huiyang Liu, Xinran Liu, Te Qi, Juncheng Zhang

TA:
Yishu Lai

A Course related document submission

April 19, 2024

Contents

Abstract

This document mainly describes the pages, page functions and implementation steps that have been implemented in the current project. Each time a new page is deployed, the progress will be continuously updated.

1 The main structure of the project

Front-end structure:
Page Components: The front end of a project consists of multiple page components, each of which is responsible for rendering the content of a particular page.
Common components: Common components are used to display duplicate content on multiple pages, such as navigation bars, footers, etc Router: Use React Router or router in Next.js to manage navigation between pages.

Back-end structure:
Server-side code: Back-end code is responsible for handling client requests and interacting with the database to get or store data.
API interface: The back end of the project provides a set of RESTful API interfaces for data exchange and communication with the front end.

Data storage:
Databases: The project uses databases to store and manage user data, application data, and so on. The database choice can be a relational database (such as MySQL, PostgreSQL) or a NoSQL database (such as MongoDB) (the database has not been selected yet).

Other components and tools:
State management: Use state management tools such as Redux, MobX, or the Context API to manage the state of your application.
Style management: Use CSS, Sass, or stylas-components to manage the style of an application.

Deployment and testing:
Deployment strategy: Choose an appropriate deployment strategy, such as containerizing your application with Docker and deploying it to a cloud platform (such as AWS, Azure) or on your own servers.
Test strategy: Write unit tests and integration tests to ensure that all parts of the project are working properly and that there are no potential bugs or problems.

2 Login and registration page

2.1 Page display and function introduction

The following pages are displayed:

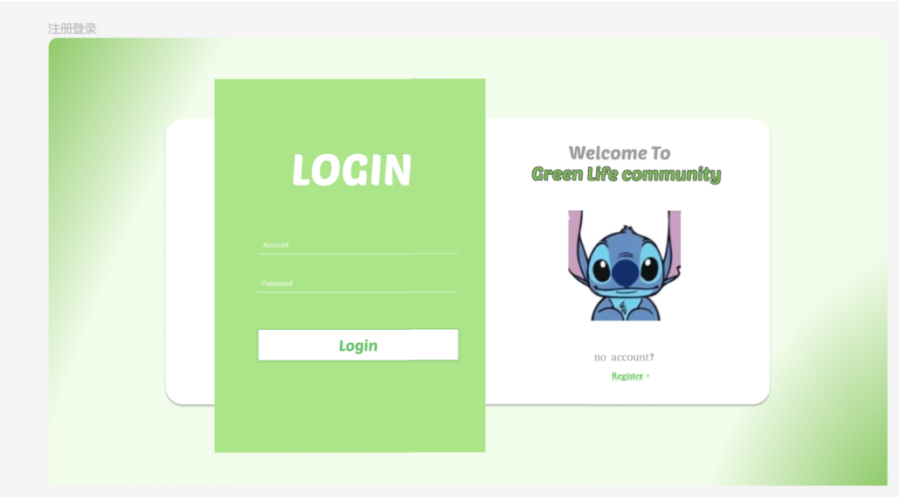


Figure 1: Login page

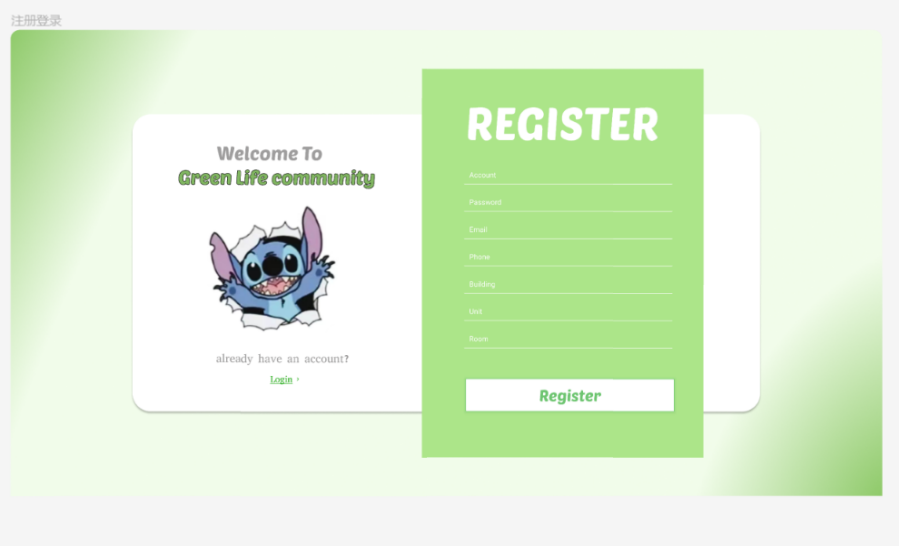


Figure 2: register page

The main function of the login page is that the user enters the account and password to log in to the platform. If you do not have an account, click Register to go to the registration page to register.

The main function of the registration page is to enter the account, password, email, telephone, address, unit and room number to complete the registration (note: only the community owner can register).

2.2 implementation step

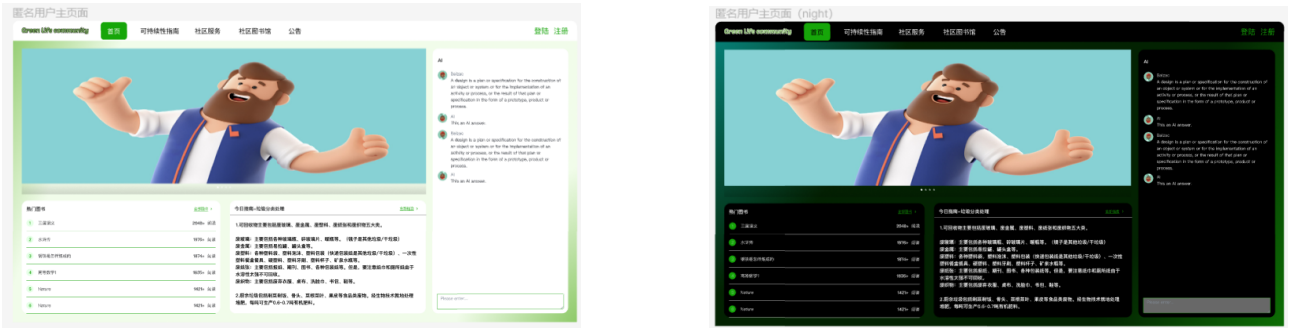
Login page implementation steps: Create the login page UI: Design the layout of the login page, including elements such as input fields, login buttons, and registration links. Create an interface for the login page using the react framework. Set up form validation: Add the necessary authentication rules in the input field, such as required fields, password length, and so on. Use JavaScript or a front-end framework to implement form validation logic to ensure that the account and password entered by the user are in the correct format. Write the logon logic: Create back-end interfaces or functions that handle login requests. Verify that the account and password entered by the user match the records in the database. The login success or failure status is returned, and the corresponding page jump or prompt is displayed according to the result. Add registration link: Add a registration link on the login page, so that users can jump to the registration page to register. When you link to the registration page, you can use URL parameters or other methods to pass related information, such as a link to the login page after successful registration. Registration page implementation steps: Create the registration page UI: Design the layout of the registration page, including elements such as input fields, registration buttons, and return to login links. Also uses the react framework Set up form validation: Add the necessary authentication rules in the input field, such as required fields, password length, email format, etc. Use JavaScript or a front-end framework to implement form validation logic to ensure the information entered by the user The format is correct.

Write the registration logic: Create back-end interfaces or functions that handle registration requests. Verify that the information entered by the user meets the requirements, such as whether the account exists, whether the mailbox has been registered, etc. The information filled in by the user is stored in the database, and the registration status is returned. Limit registration rights: Limit the ability for only community owners to sign up for an account, as required. Verification logic can be added to the registration page to ensure that only community owners can successfully register an account. Add back to login link: Add a return to login link on the registration page, so that users can easily return to the login page for login operations.

3 Platform home page

3.1 Page display and function introduction

The following pages are displayed:



(a) daytime (b) night

Figure 3: Anonymous user home page is displayed

This is the page used by anonymous users (and non-registered users). On the home page, you can see some relevant text information about the platform, as well as the functions that non-registered users can see and use, such as switching platform night mode. Click the upper right corner to log in to | to register, you can switch to the login or registration page for related operations.

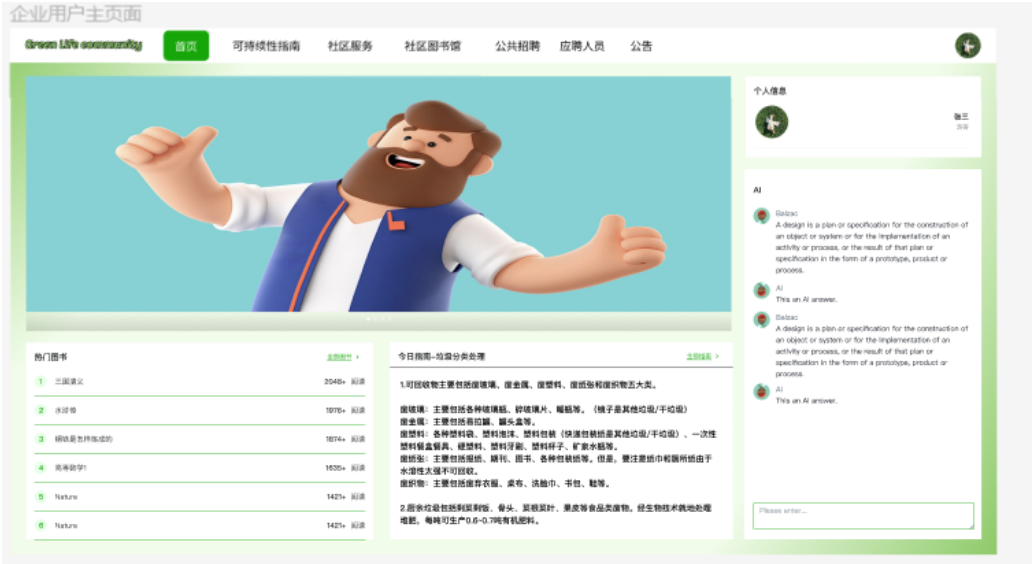


Figure 4: Home page for community owner users

This is the main page used by the owners of the community after logging in. The functions that can be used include clicking to jump to the sustainability guide page, community service page (to apply for some service content such as housekeeping service), community electronic library page (to read some books online), public recruitment page (to fill in the resume to apply for a job), and viewing announcements. There are also some auxiliary functions such as online AI-assisted consultation function, viewing resources related to the community and so on.

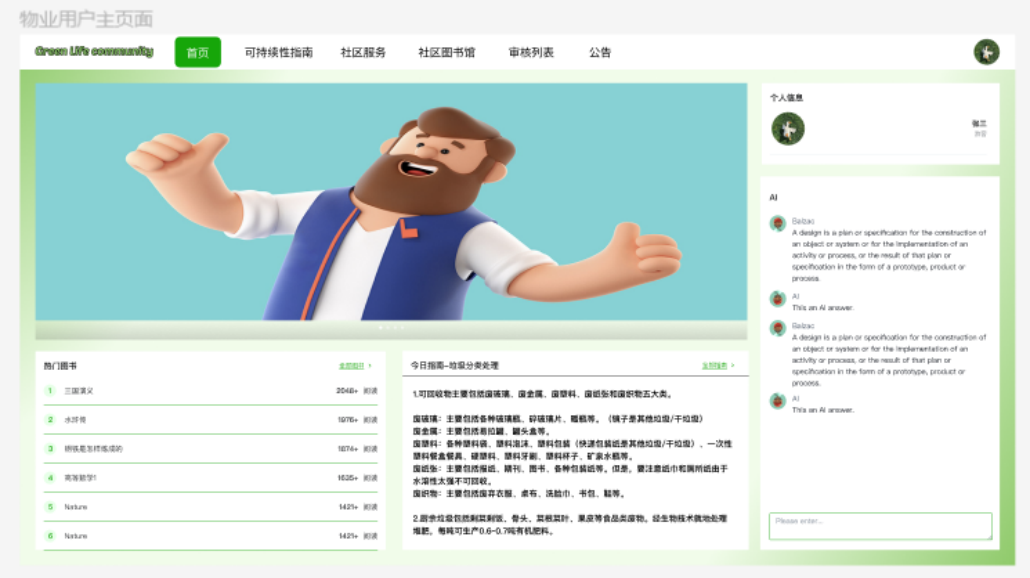


Figure 5: Residential property user homepages

This is the main page used by the residential property login. Compared with the home page of the residential property owner, there is no function of recruiting and viewing the recruited candidates, but the property has some management functions such as updating the residential information such as announcements.

3.2 implementation step

Community owner User homepage implementation steps:

Design Home page layout:
Use the Ant Design (Antd) component library to design the layout of the owner user's home page, including the header navigation bar, sidebar menu, and main content area.

Routing convention:
Use React Router or Next.js framework to design the routing convention of the home page of the owner user to ensure that the owner user can jump to various functional pages smoothly.

Status code:
Define appropriate HTTP status codes to handle page loading, request responses, and so on.

Use of plug-ins and AI interfaces:
Integrated SWR + Axios request library to achieve page data acquisition and display, improve page performance and user experience. Use the G2Antv chart display library to display visual charts of community data and provide users with intuitive data presentation. AI interface call (this function has not been implemented, please look forward to it).

Status Management:
Use Zustand State Management Library to manage page state and data to ensure consistency and reliability of page data.

Example page operation:
Add navigation links to the home page to make it easy for users to access the various functional pages of the community, such as the sustainability guide page, the Community services page, the community e-library page, the open recruitment page, and to view announcements.
Display real-time data and statistical information of the community on the home page, such as the number of community population, the number of community services, the number of announcements, etc., so that owners can understand the operation of the community.
Add function buttons or links on the home page so that owners can perform related actions, such as applying for community services, checking community announcements, submitting job applications, etc.

4 other pages

Although the front-end page has been built, the functions have not been fully implemented, resulting in incomplete pages and continuous updates...

The following only shows and explains some functions:

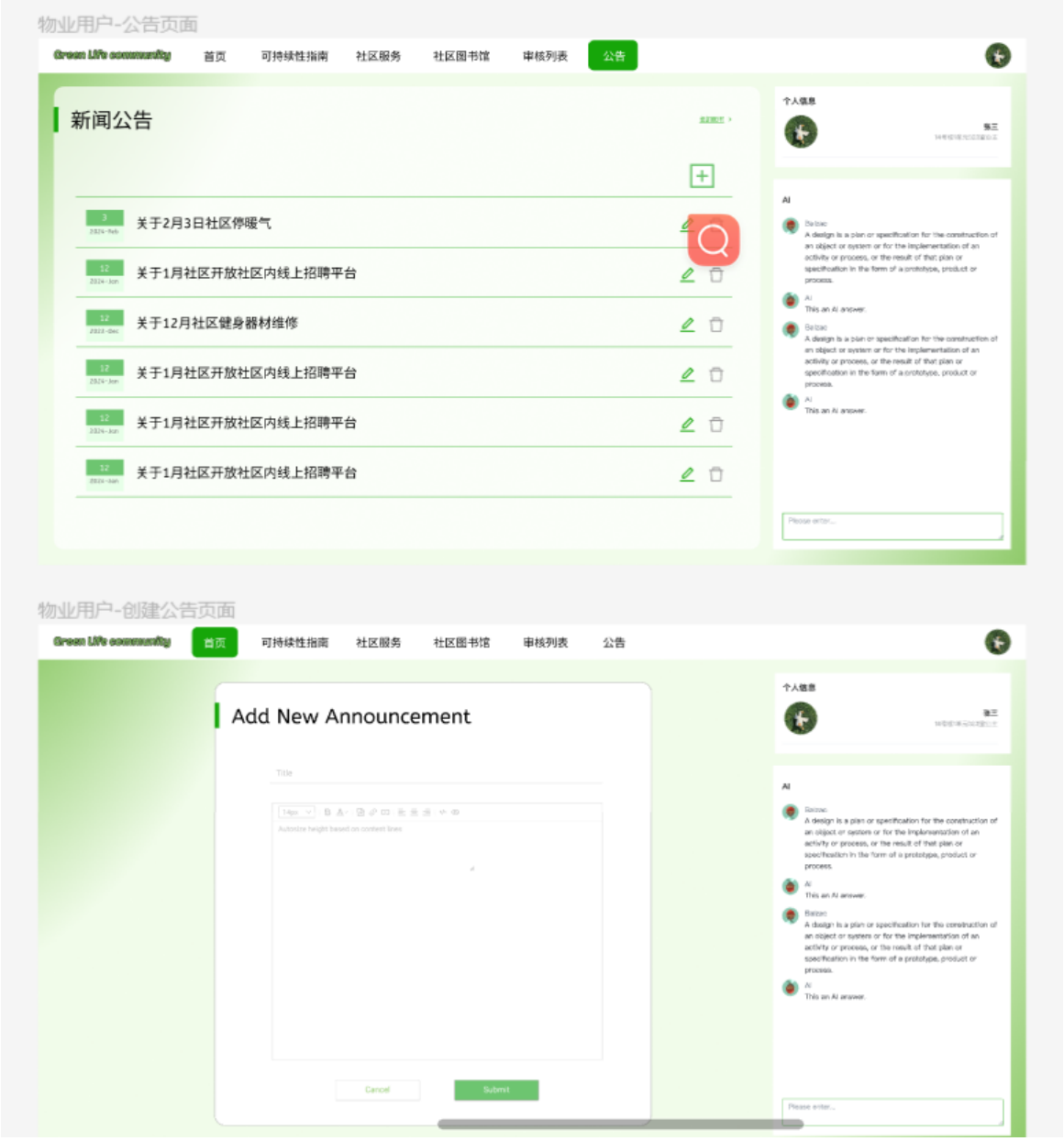


Figure 6: Community announcement page

On this page, owners can receive updates from the community and view all announcements in the history, and pop up reminders to owners after each update.
The property can update the announcement, delete, edit.

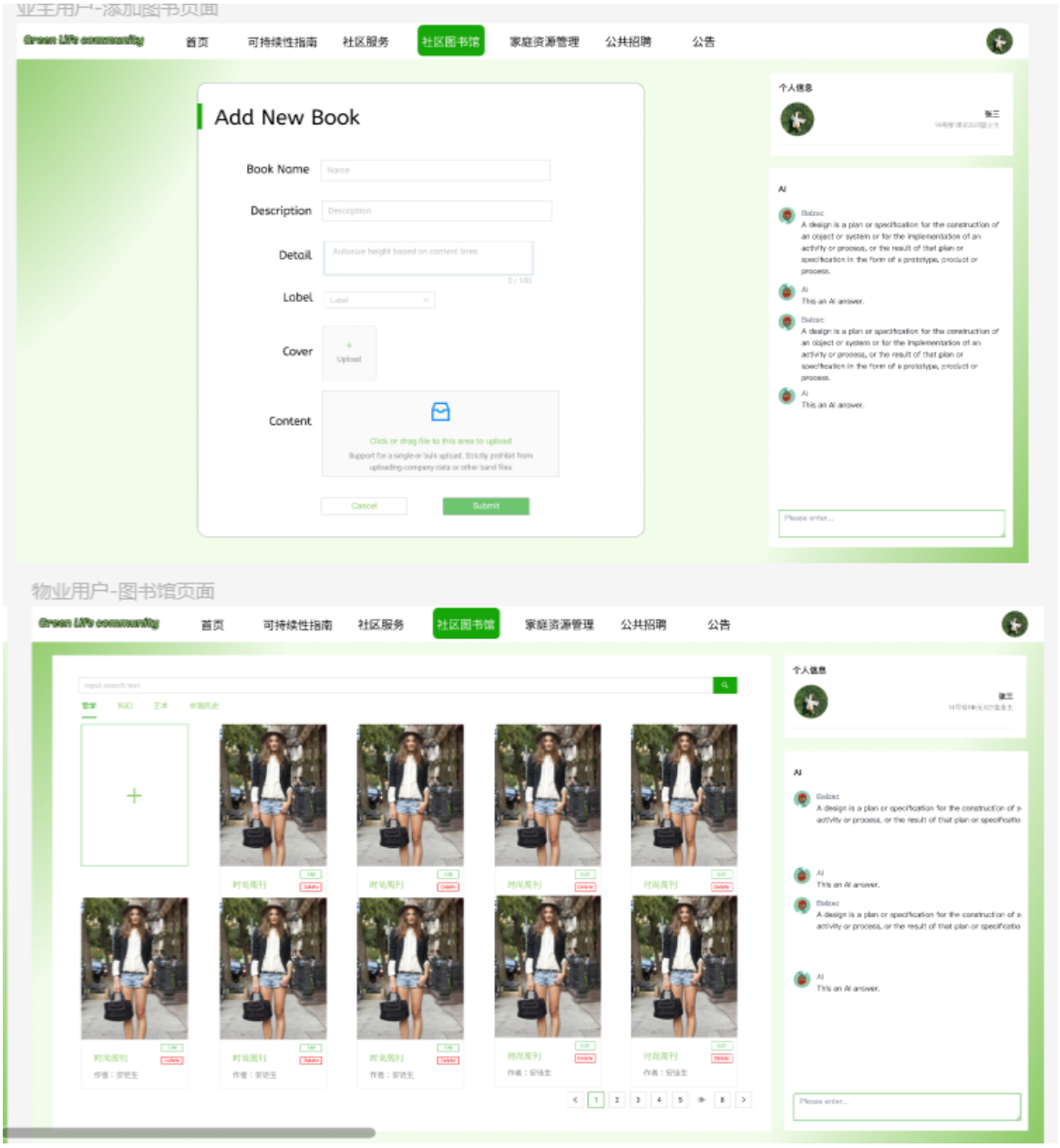


Figure 7: Electronic library page

This page allows owners to browse online books, and owners can add or remove books.