**PASSERELLES NUMÉRIQUES VIETNAM**



**Advanced web project report**

Dream Apartments Finder



Supervising teachers: Mr. Vo Cong Dinh

Mr. Hoang Nhu Vinh

Group: H H T D B

Class: PNV24A/PNV24B

Subject: Advanced Web Project

*Da Nang, July 07, 2023*

**Group Report**

**Advanced Web Project**

**Subject Title:** Advanced Web Project

**Team Name:** HHTDB

**Submission Date:** 07/07/2023

**Team Information**

|  |  |
| --- | --- |
| **Full Name** | **Email** |
| Phan Thi Thu Huong | huong.phan24@student.passerellesnumeriques.org |
| Tran Quoc Huu | huu.tran24@student.passerellesnumeriques.org |
| Phan Tran Thi Anh Thu | thu.phan24@student.passerellesnumeriques.org |
| Ho Van Di | di.ho24@student.passerellesnumeriques.org |
| Ho Thi Bich | bich.ho24@student.passerellesnumeriques.org |

**Table of content**

**[1. User’s requirements 4](#_Toc15488)**

**[2. Design Document 5](#_Toc29177)**

*[2.1. Site map](#_Toc8251)* [5](#_Toc8251)

*[2.2. Prototype designed with Figma](#_Toc7522)* [5](#_Toc7522)

*[2.3. Entity Relationship Diagram (ERD)](#_Toc29276)* [8](#_Toc29276)

*[2.4. API endpoints](#_Toc8595)* [8](#_Toc8595)

**[3. Implementation 11](#_Toc3884)**

*[3.1. Github repository images](#_Toc917)* [11](#_Toc917)

*[3.2. Jira board images](#_Toc8689)* [12](#_Toc8689)

*[3.3. Images of the final application](#_Toc31368)* [14](#_Toc31368)

**[4. Conclusion 16](#_Toc8817)**

*[4.1. What went well](#_Toc28571)* [16](#_Toc28571)

*[4.2. What did not go well](#_Toc27545)* [17](#_Toc27545)

*[4.3. Lessons learned and further improvements](#_Toc30366)* [17](#_Toc30366)

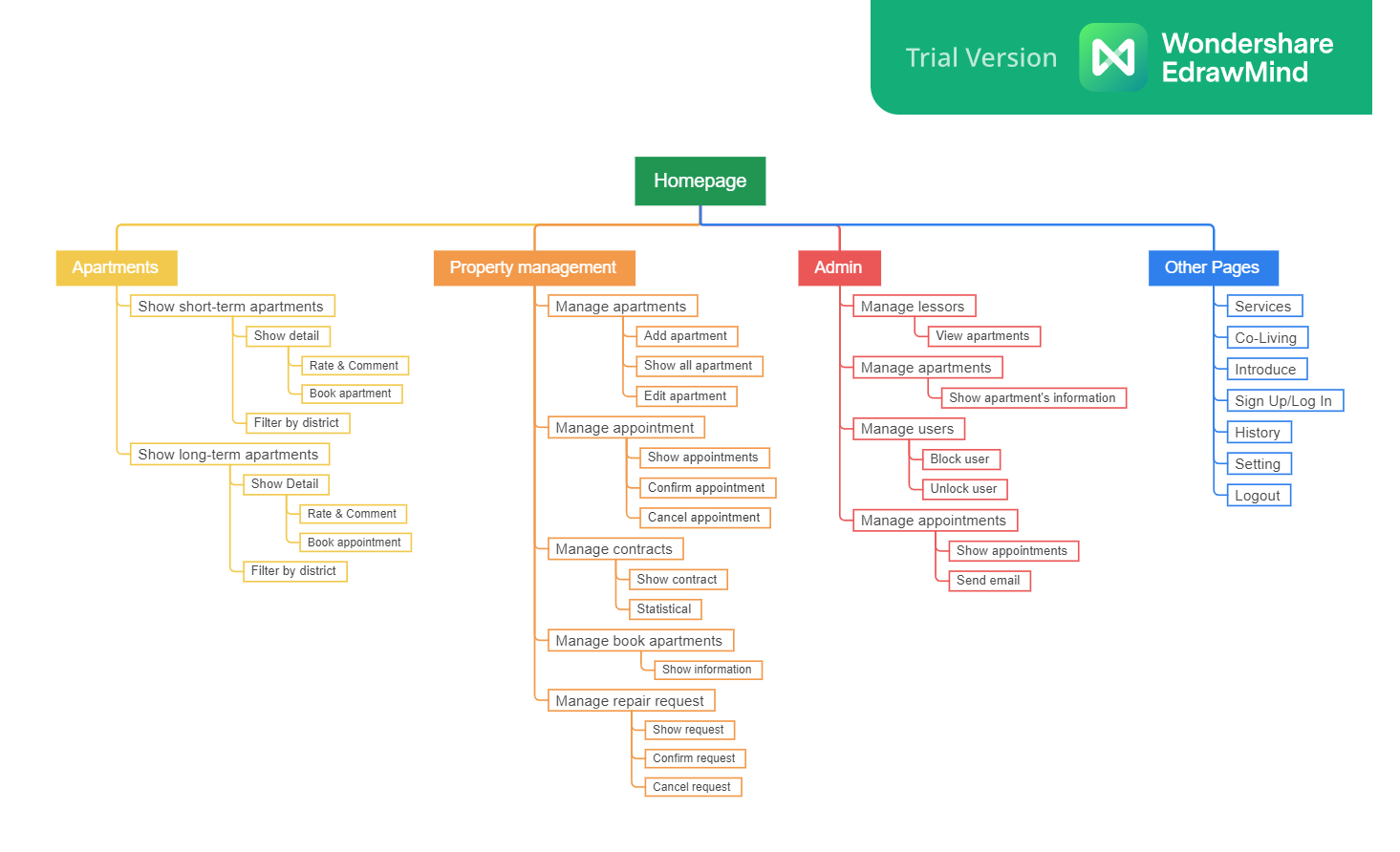
1. **User’s requirements**

*Table 1 – User’s requirements*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **PRIORITY** | **AS A** | **I NEED TO** | **SO THAT I CAN** |
| 1 | High | As a lessor | Add & edit my apartment with its many images | View, list & manage my apartments |
| 2 | High | As a lessor | View & confirm the appointment to see long-term apartments | Send confirmation notice to renter |
| 3 | High | As a lessor | Add & view the contracts | Control & report turnover |
| 4 | High | As a lessor | View short-term booking information | Control & report turnover |
| 5 | Medium | As a lessor | View & confirm apartment repair requests | Meet user needs |
| 6 | High | As a renter | See all apartments | Look at many options |
| 7 | Medium | As a renter | Filter apartments by district & by apartment type | Easily search for the desired apartments |
| 8 | High | As a renter | Register, log in & log out my account | Easy to use website |
| 9 | High | As a renter | Make a short-term reservation | Book a temporary apartment |
| 10 | High | As a renter | Book an appointment to see the apartment | Review the room’s quality & sign the contract |
| 11 | Medium | As a renter | Give a comment & rate star | Feedback about apartments |
| 12 | Medium | As a renter | Submit a request to report a damaged room | Request a repair |
| 13 | Low | As a renter | Edit my account information | Change my information when necessary |
| 14 | Medium | As a renter | View my history | Track booking status or make an appointment |
| 15 | Medium | As an  admin | Decentralization & account lock | Manage users |
| 16 | High | As an  admin | View apartments of each lessor | Easy to manage apartments |
| 17 | High | As an  admin | View & send appointment information to the relevant lessor and renter | Inform about the transaction guide between the two parties |
| 18 | High | As an  admin | View the number of contracts | Website revenue statistics |

1. **Design Document**
   1. ***Site map***

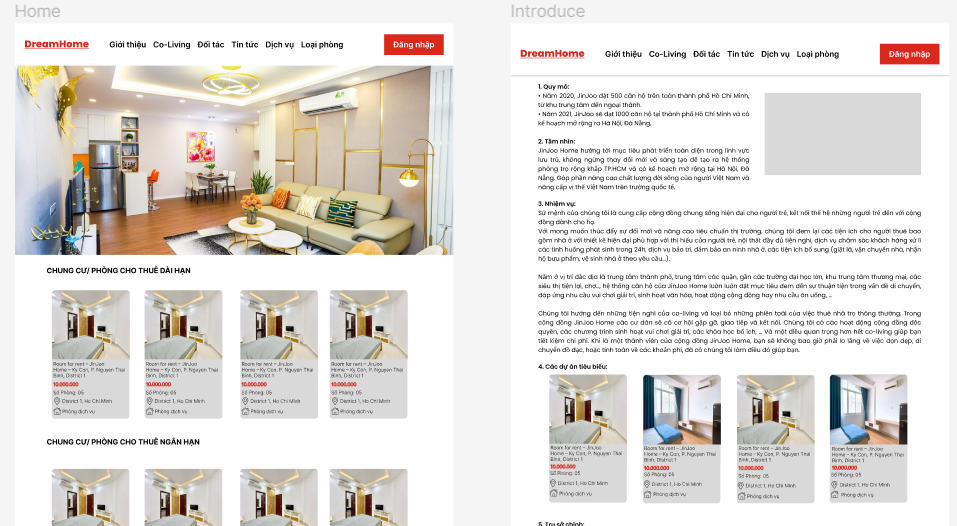
Dream Apartments is a website that allows lessor and renter to connect with each other to post apartments for rent and rent apartments. Our website is basically described as the following site map:

****

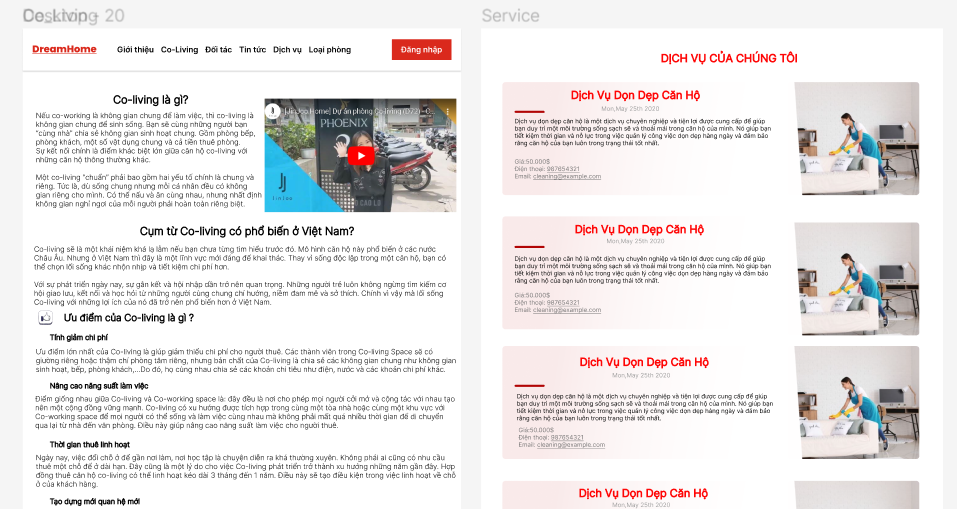
*Figure 2.1 – Site map of the project*

* 1. ***Prototype designed with Figma***

Besides the main tools, before actually starting the implementation of the website, we used Figma to design the basic framework of the website. And here are some pictures of our Figma:



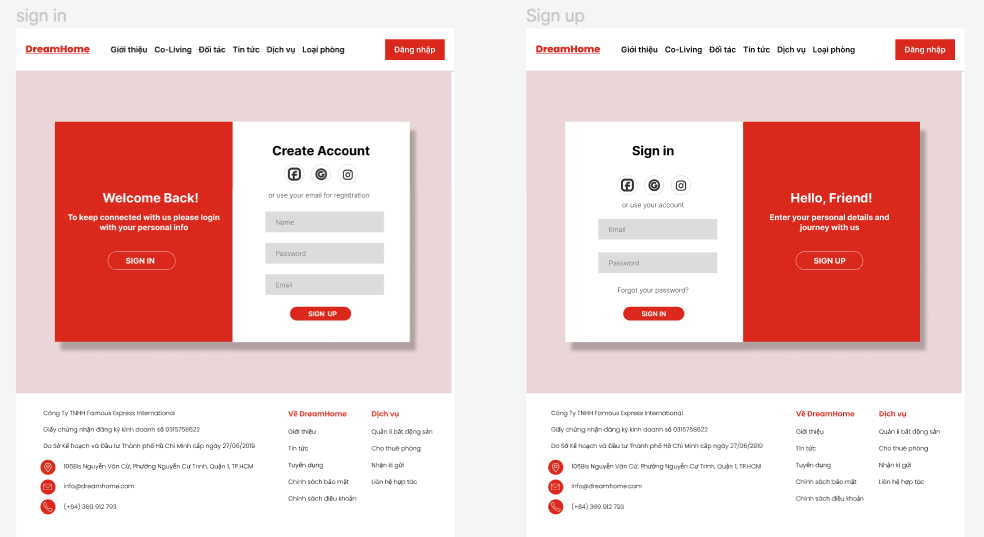
*Figure 2.2 – Prototype with Figma (1)*

**

*Figure 2.3 – Prototype with Figma (2)*

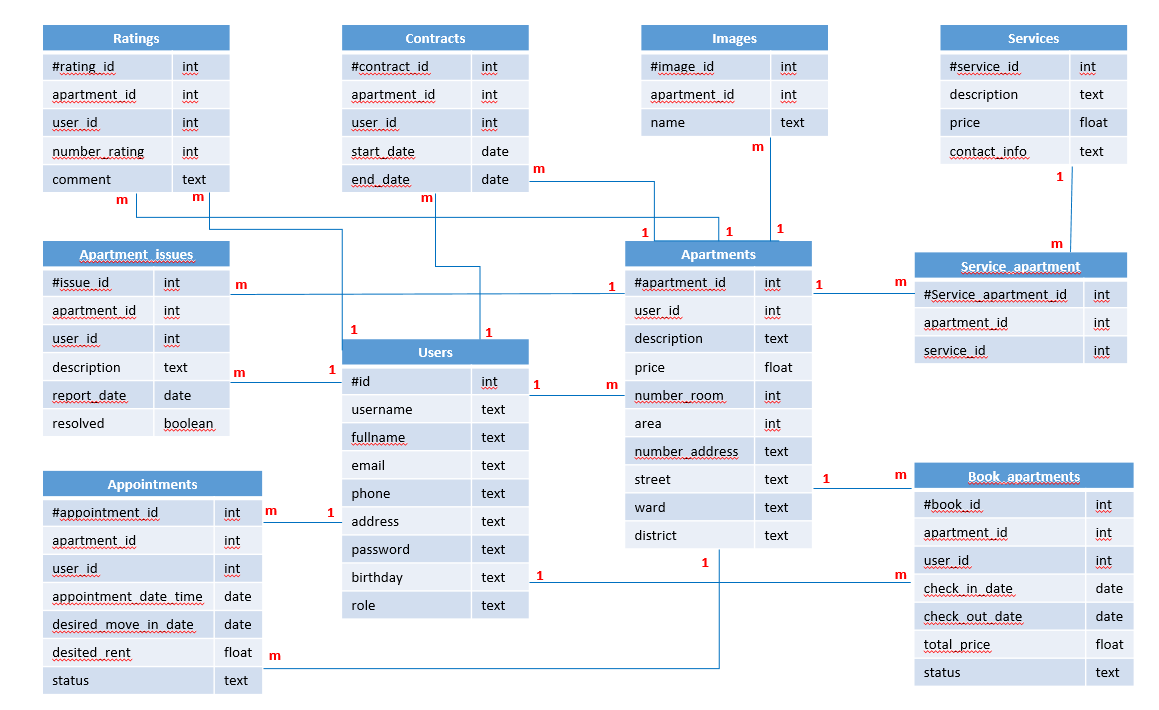
**

*Figure 2.4 – Prototype with Figma (3)*

**

*Figure 2.5 – Prototype with Figma (4)*

* 1. ***Entity Relationship Diagram (ERD)***



*Figure 2.6 – ERD of the project*

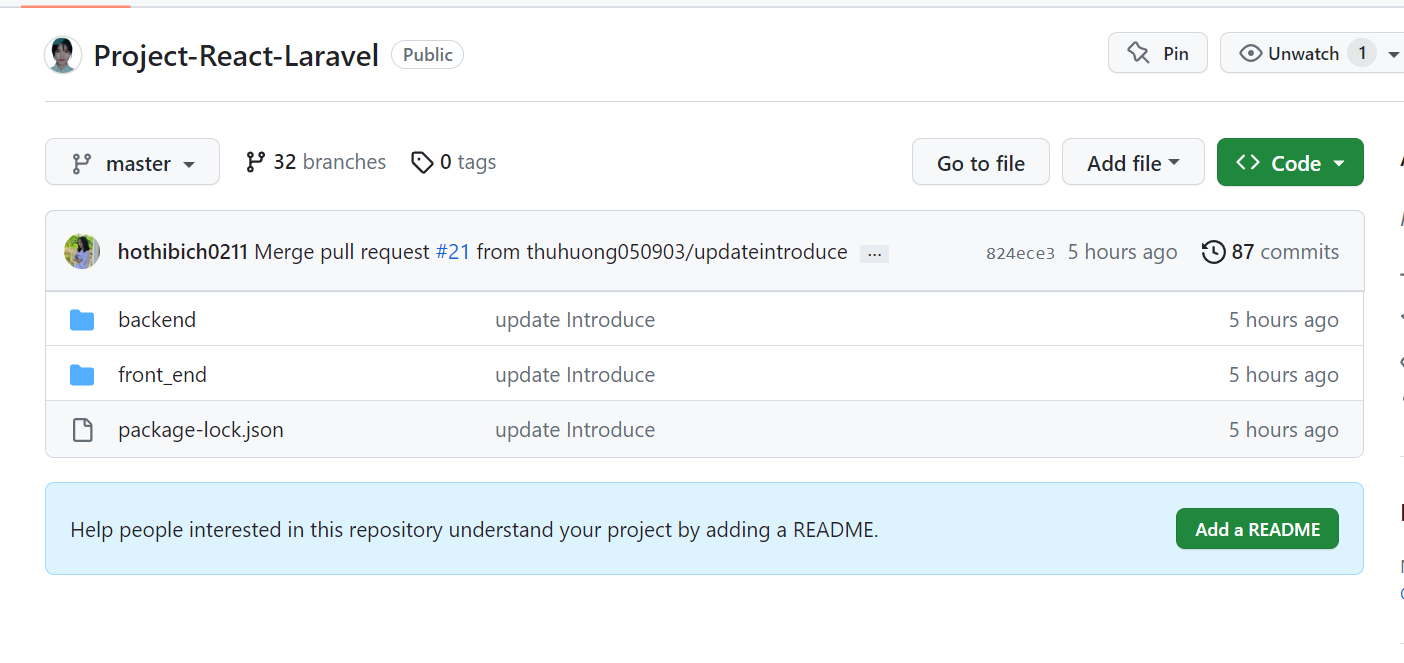
* 1. ***API endpoints***

*Table 2 – API endpoints of the project*

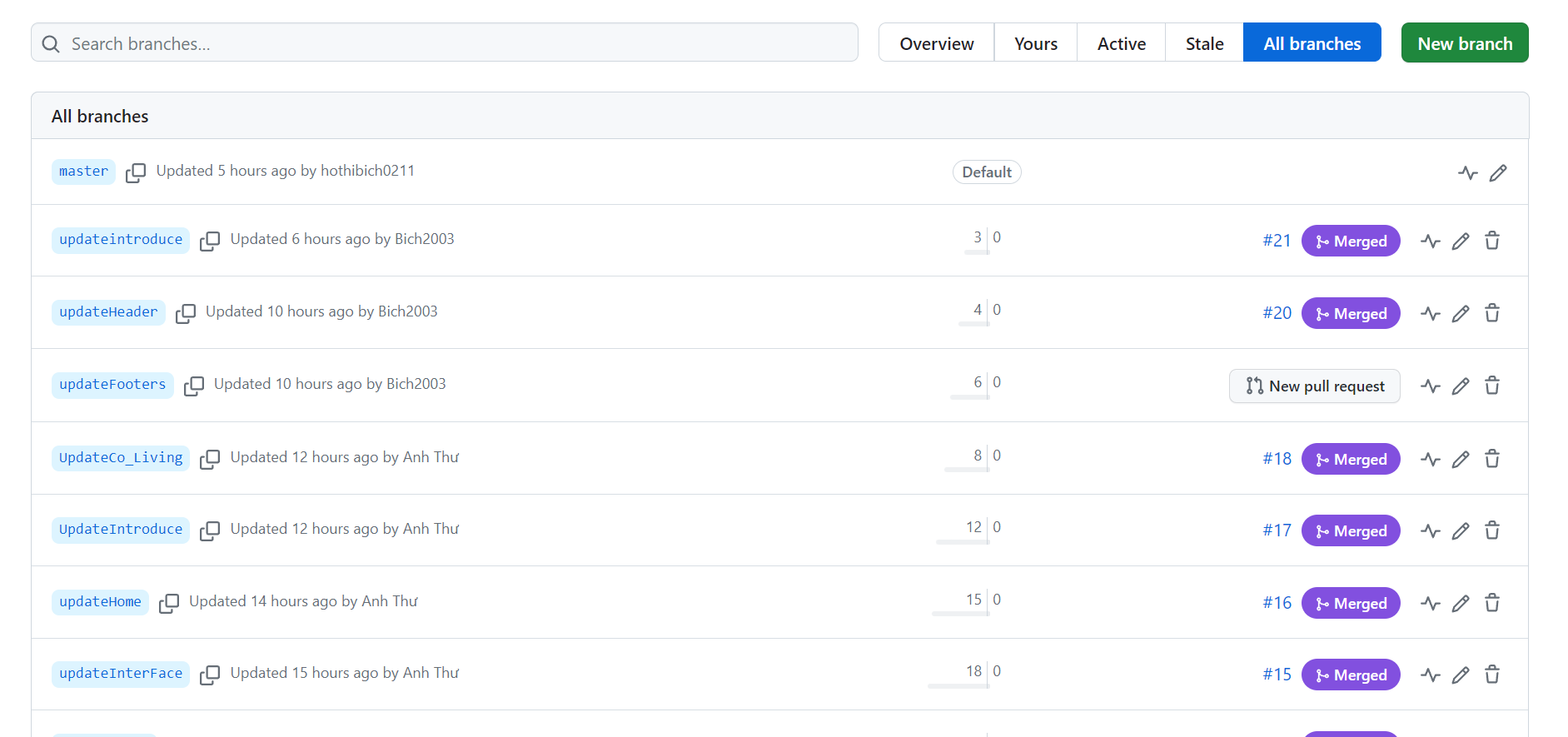
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Route** | **Description** | **Params** | **Return** |
| POST | /bookings | Book an apartment |  | error message or successful message |
| POST | /book-appointment | Book an appointment |  | error message or successful message |
| GET | /get-appointment/  {userId} | Get all appointments of 1 lessor | userId | error message or appointments information |
| PUT | /update-appointment/  {id} | Edit status appointment in the lessor page | id | error message or successful message |
| GET | /get-appointments | Get all appointment of all lessors |  | error message or appointments information |
| POST | /add-ratings | Rate & comment apartments |  | error message or successful message |
| GET | /get-ratings | Get number of ratings to display amount stars |  | error message or number of stars |
| GET | /rating-count/{id} | Get the number of ratings and comments on an apartment | id | error message or number of stars and comment of an apartment |
| GET | /get-contracts/  {userId} | Get the contract of each lessor | userId | error message or contracts of each lessor |
| GET | /get-contracts/  {ctr\_id} | Get the information of each contract | ctr\_id | error message or information of each contract |
| POST | /add-contracts | Create a contract |  | error message or successful message |
| PUT | /update-contracts/  {id} | Edit contract | id | error message or successful message |
| GET | /get-contract | Get all contract in admin page |  | error message or contracts |
| GET | /history-appointment/  {userId} | Get appointment history of each renter | userId | error message or appointment history |
| GET | /history-apartments/  {userId} | Get apartment history of each renter | userId | error message or apartment history |
| POST | /apartment-issues | Send request to repair apartments |  | error message or successful message |
| PUT | /update-user/{id} | Edit user’s information | id | error message or successful message |
| GET | /seederInfo/{userId} | Get seeder information | userId | error message or seeder information |
| GET | /get-user | Get all users |  | error message or list users |
| GET | /get-user/{id} | Get one user | id | error message or user’s information |
| POST | /login | Log in to the system |  | error message or token |
| POST | /register | Register an account |  | error message or send verification email to users |
| GET | /get-apartments-byLessorId/{userId} | Get all apartments of each lessor | userId | error message or list apartments |
| PUT | /update-apartment/  {apartId} | Edit one apartment | apartId | error message or successful message |
| POST | /add-apartment | Create a new apartment |  | error message or successful message |
| GET | /get-apartment/  {apartId} | Get detail information of each apartment | apartId | error message or detail information |
| GET | /get-apartment | Get all apartments |  | error message or list apartments |
| GET | /SeederApartment  Page/{id} | Get all apartments of each lessor | id | error message or list apartments |
| DELETE | /delete-photo/  {photoId} | Delete photo when edit apartment | photoId | error message or successful message |

1. **Implementation**
   1. ***Github repository images***

Besides the main tools, before actually starting the implementation of the website, we used Figma to design the basic framework of the website. And here are some pictures of our Figma:



*Figure 3.1 – Github repository*

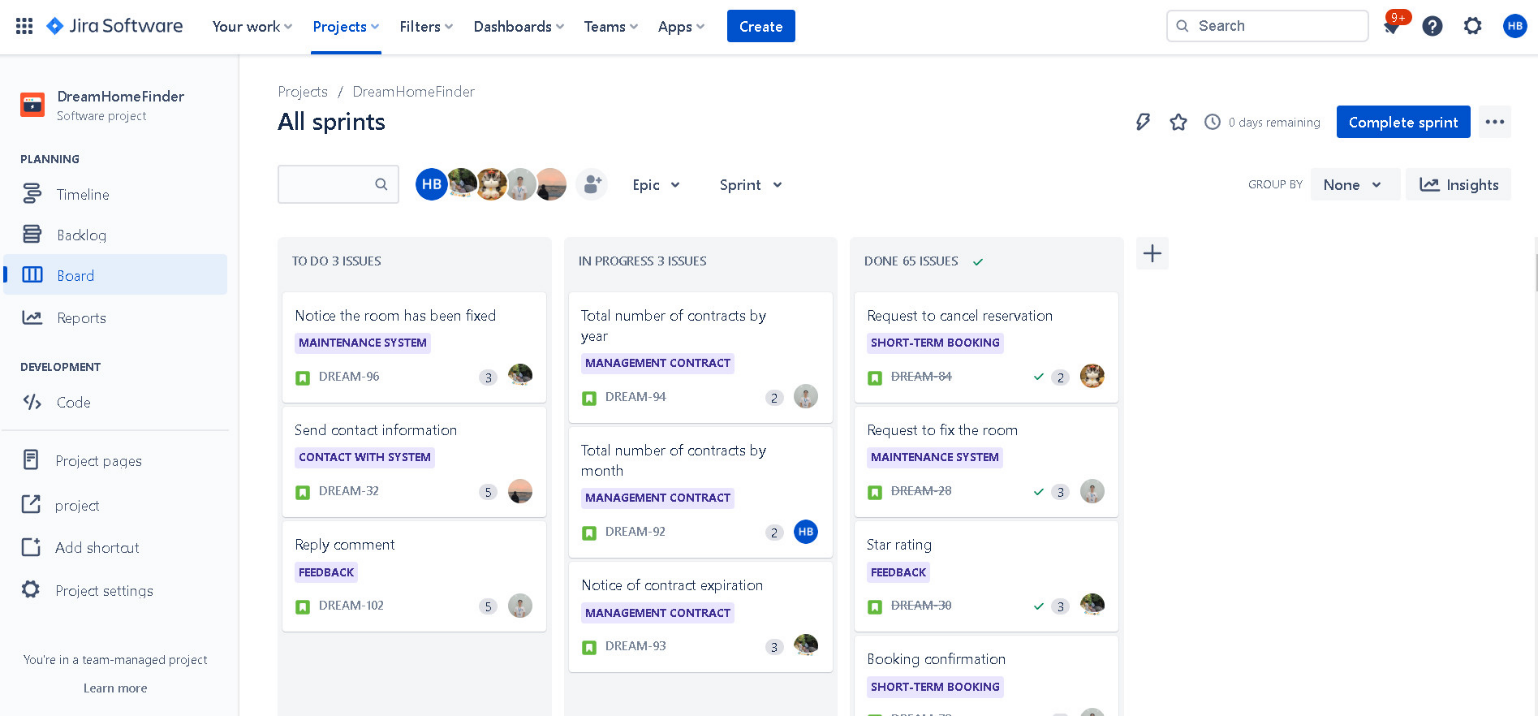
**

*Figure 3.2 – Github branches*

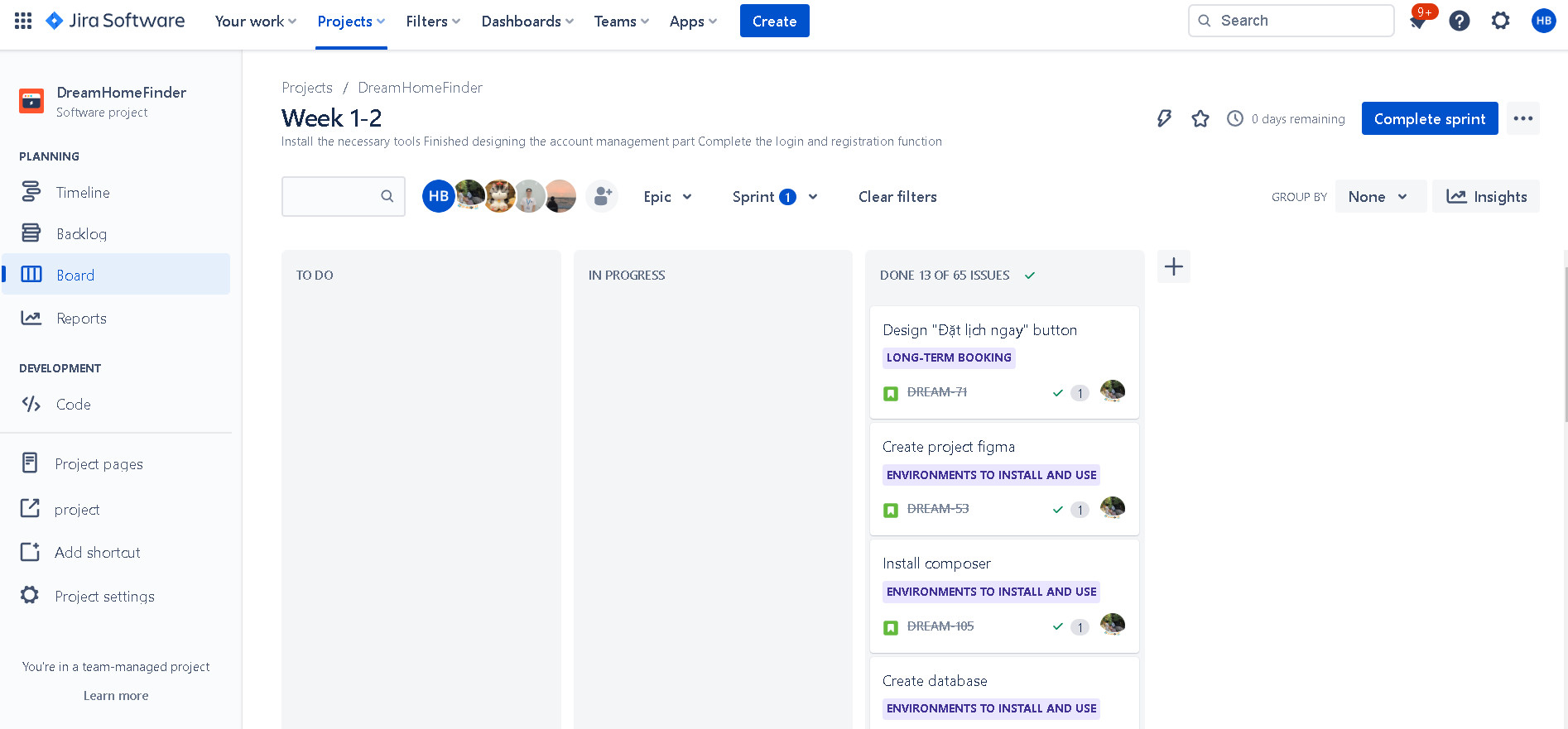
**

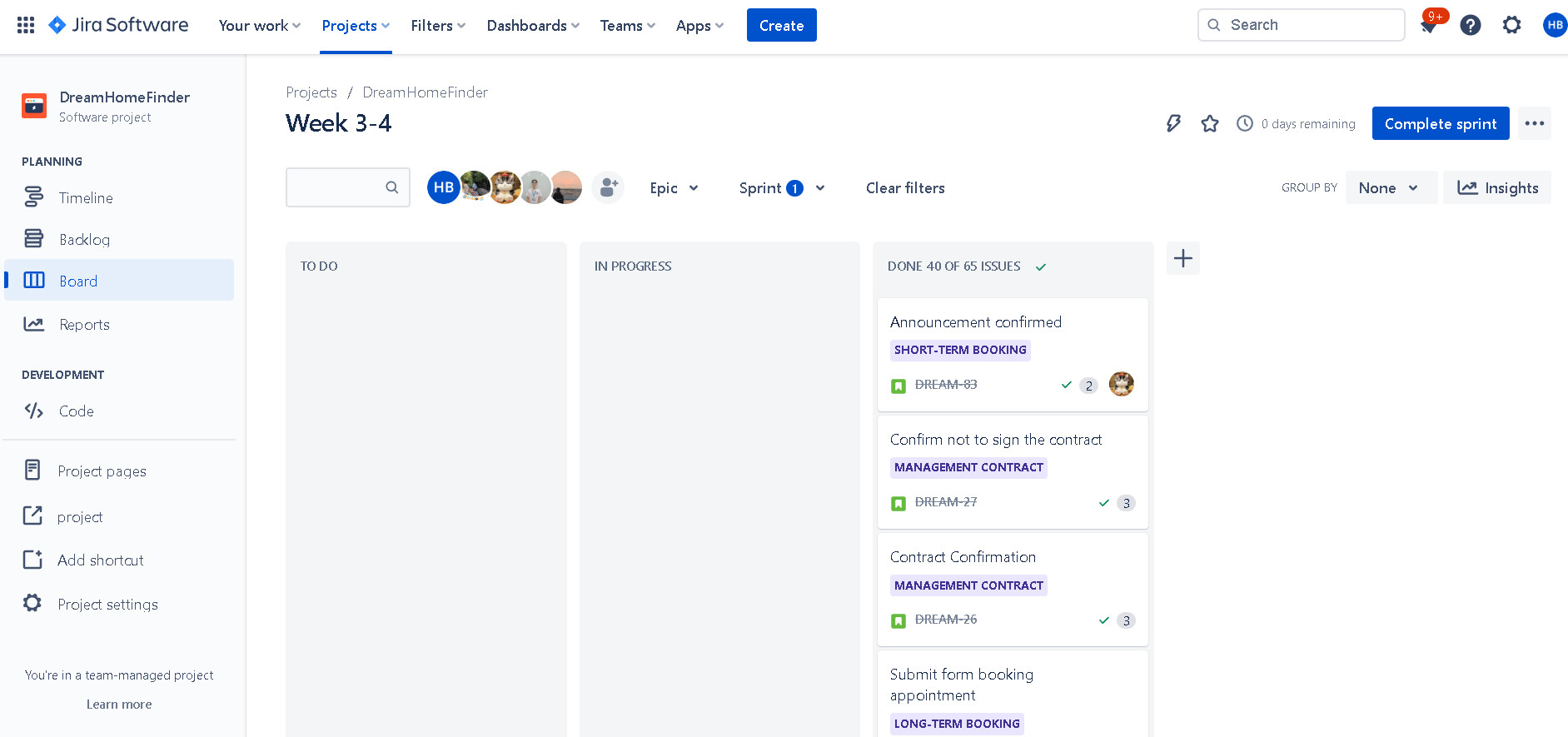
*Figure 3.3 – Github branches*

* 1. ***Jira board images***

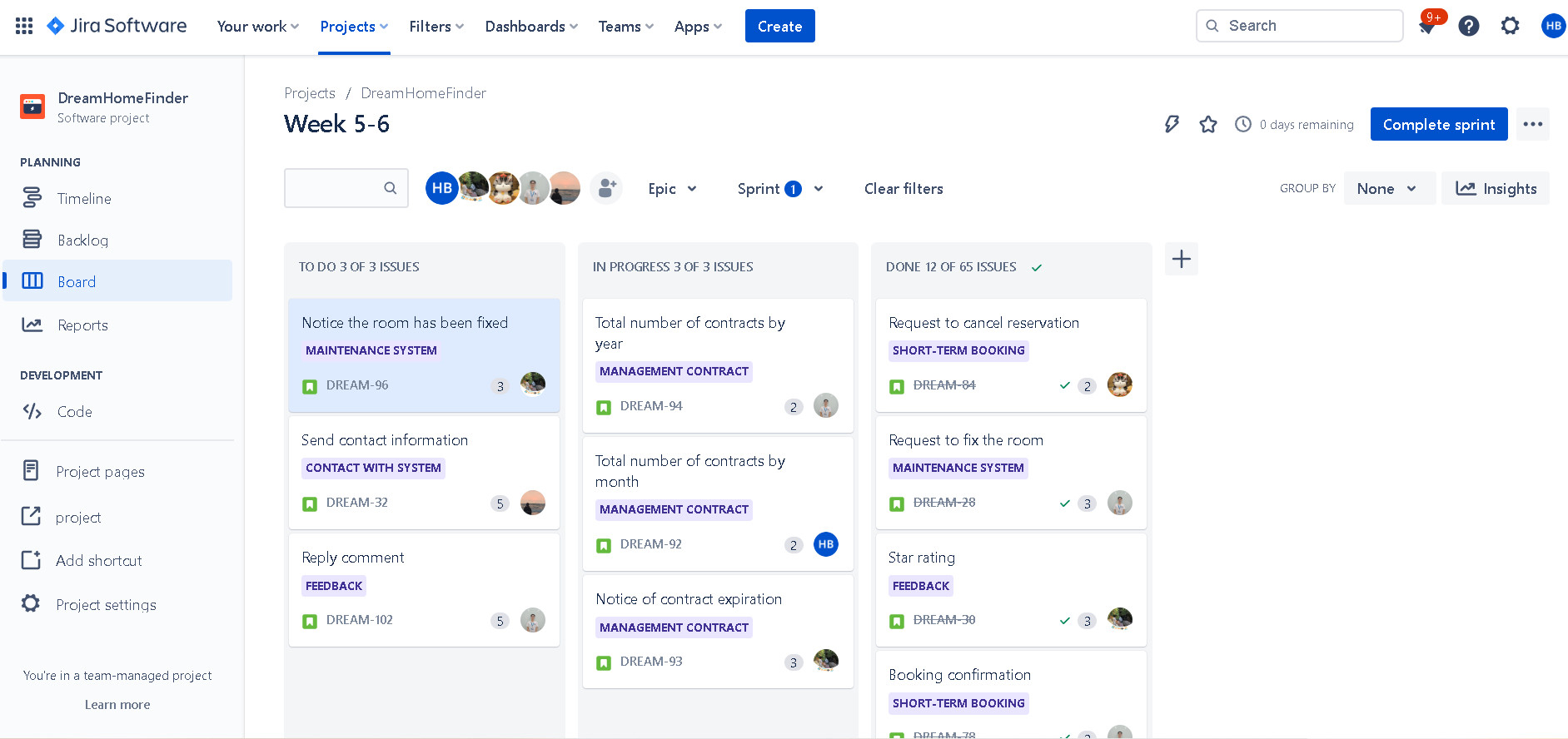
To ensure the project runs on schedule and with quality, the team has applied Agile and Scrum to project management. Here are some pictures of the Sprint Board through the stages:

*Figure 3.4 - Jira Board - All sprints*



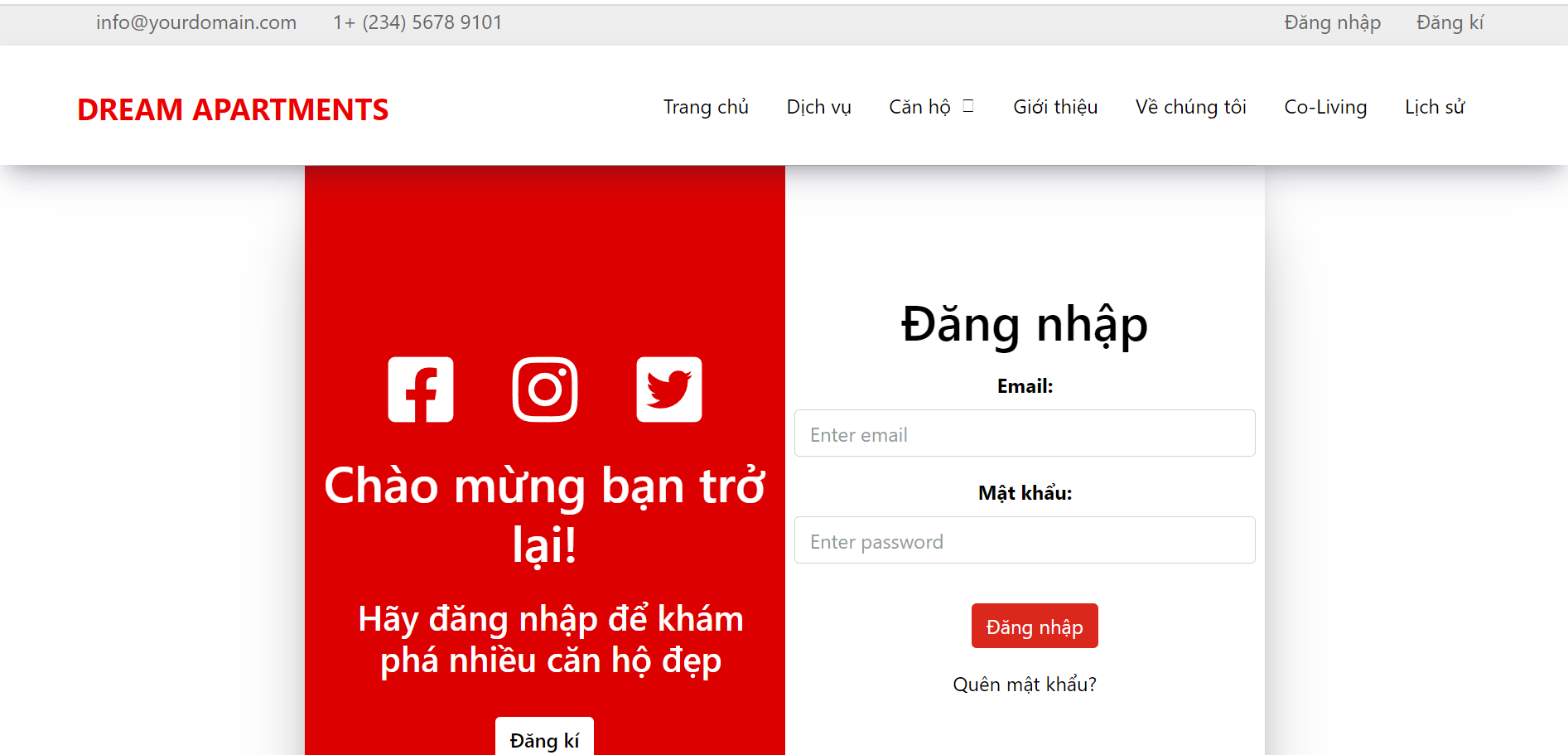
*Figure 3.5 - Jira Board - Week 1-2*

*Figure 3.6 - Jira board - Week 3-4*

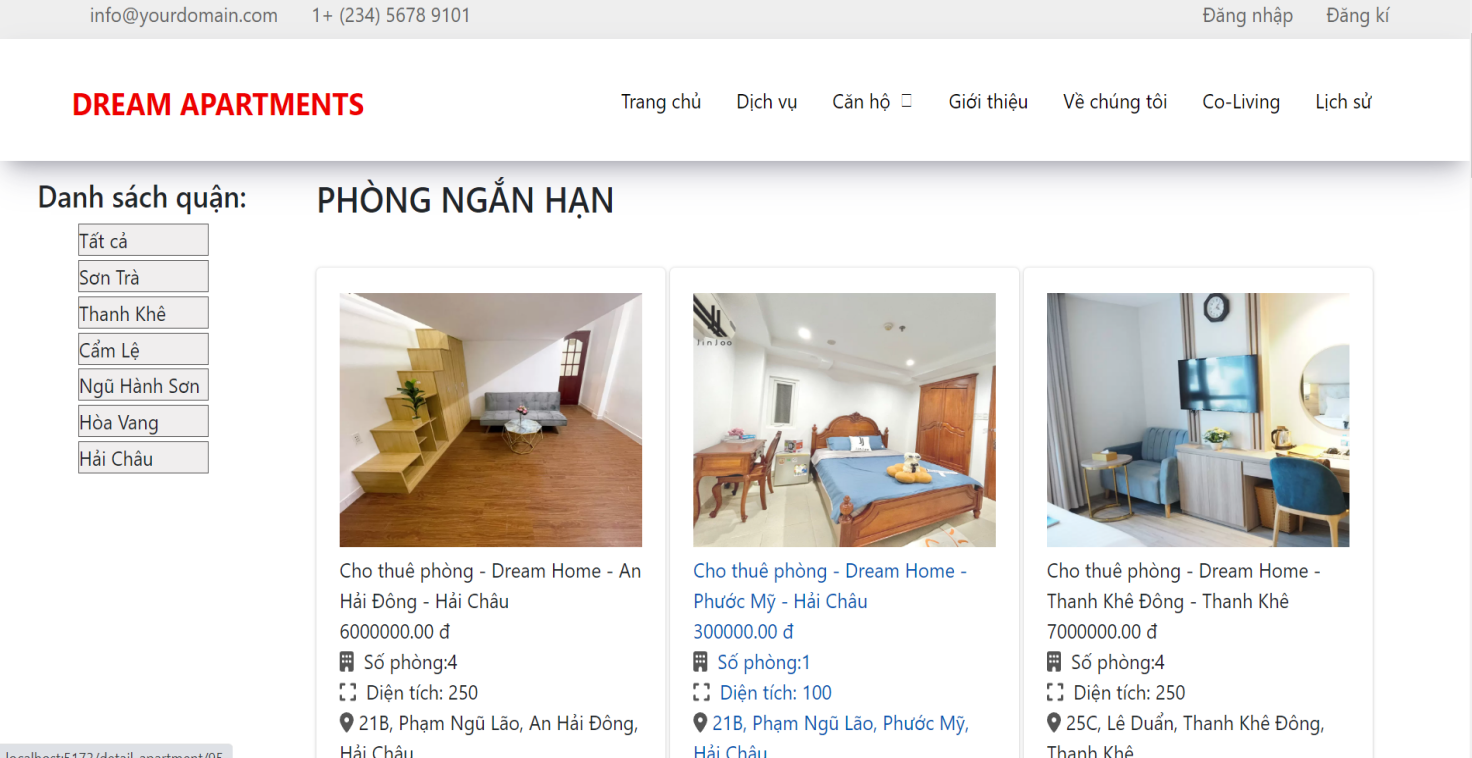


*Figure 3.7 - Jira board - Week 5-6*

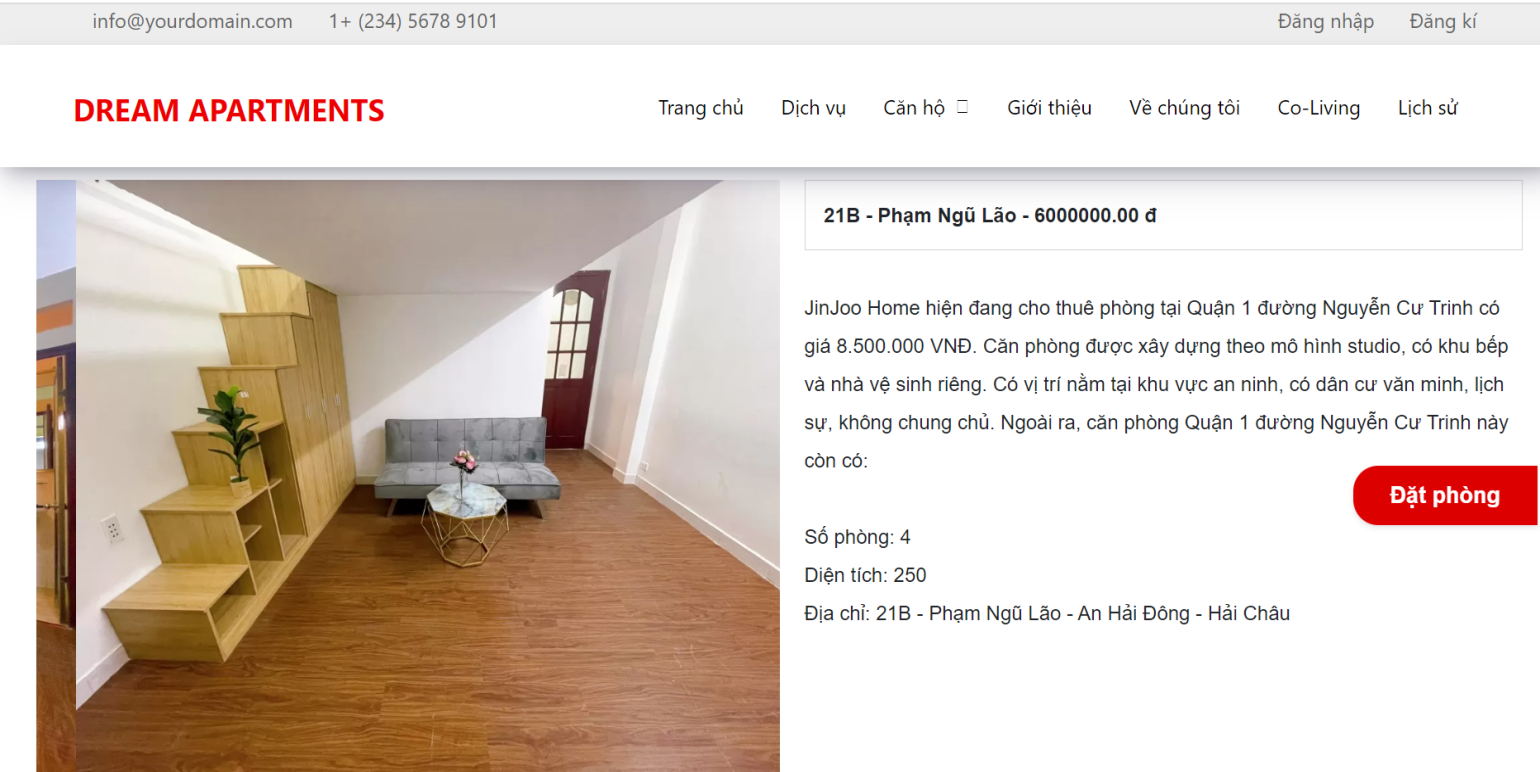
* 1. ***Images of the final application***

 Here are some pictures of the group's final application:

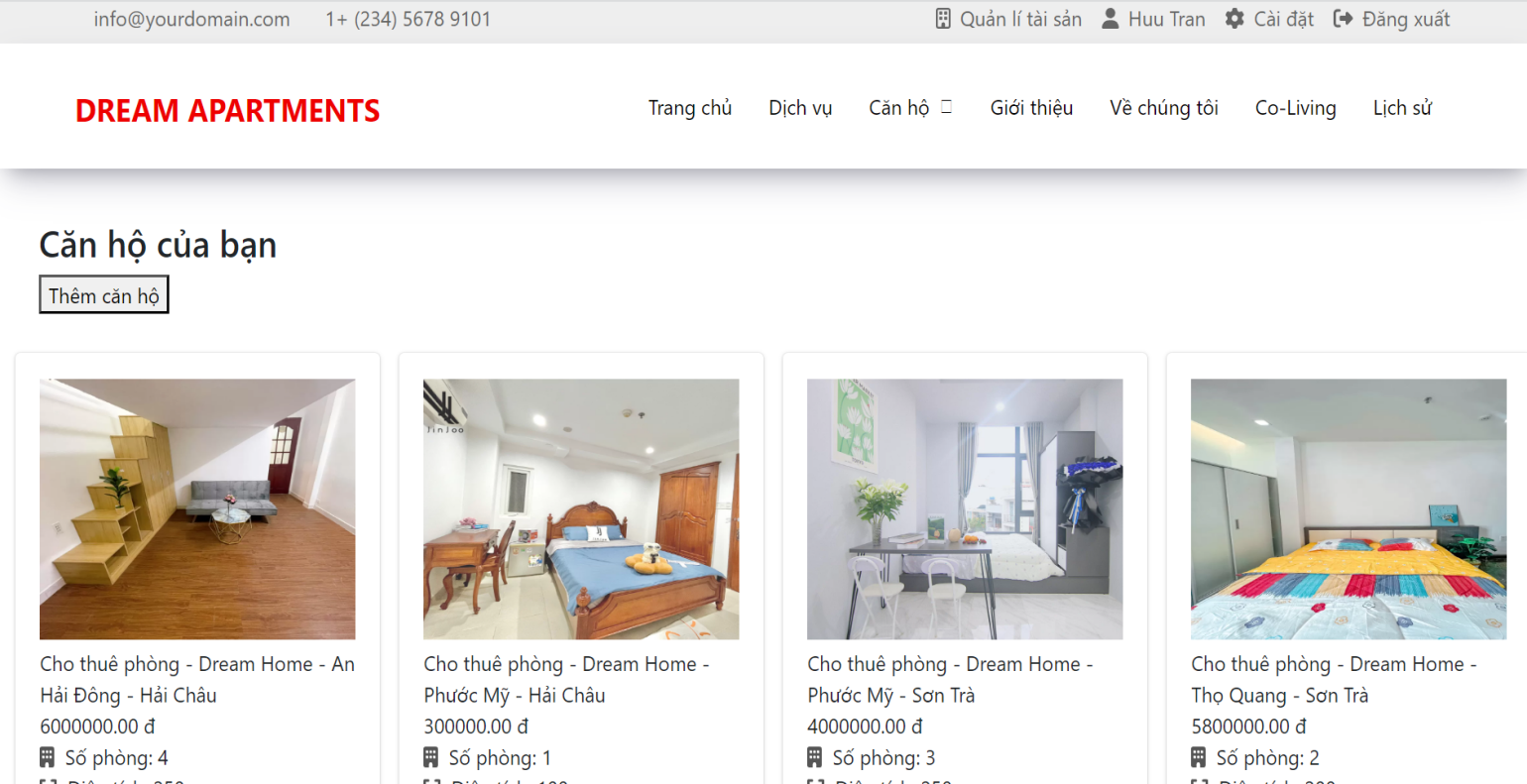
*Figure 3.8– Final application – Login page*



*Figure 3.9 – Final application – Apartments page*



*Figure 3.10 – Final application – Detail page*

**

*Figure 3.11 – Final application – Property management page*

1. **Conclusion**
   1. ***What went well***

- The team has effectively used Jira and the Scrum model to manage the project.

- The team members were active, proactive and open to share the problems encountered in the daily meetings for the whole team to solve.

- The group effectively took advantage of the support of mentor and teacher Tuan Anh in the process.

- Achieve 90% of the goals set out for the project.

- The members complete the assigned tasks within the deadline.

- During the project implementation, the team did not have any conflicts, on the contrary, the team is doing well in giving suggestions and complementing each other.

- We have built a website with a clear and user-friendly interface along with functions that meet user requirements.

* 1. ***What did not go well***

- There are advanced functions that are not possible

- Some members are still late due to the difficult function but actively reported in Sprint Review to transfer that function to the next Sprint.

- Using Github is not effective in managing source code

- Determining the difficulty of the incorrect function leading to the failure of the sprint

- Have not really solved the difficult problems in the process of making the project

* 1. ***Lessons learned and further improvements*** 
     1. Lessons learned

- Know and understand the importance of project management. When running a project, it is necessary to have a strict management process to ensure the progress and quality of the work.

- Learn to design and analyze a website from scratch.

- Access to working with Git and Github to manage source code.

- In the process of working with Jira, Scrum, and GitHub, we have learned how to effectively use these tools.

- Improve communication in the working environment through group meetings.

- Know how to manage time and prioritize work to ensure progress.

- Learn how to log bugs and fix bugs from Mr. Tuan Anh.

- Understand the combination code between ReactJs and Laravel.

4.3.2. Further improvements

- Use Github more professionally. Make sure all members understand the git workflow.

- Define the website's functions in more detail, more clearly to avoid not reaching the goal

- Need to design a better and more consistent website interface.