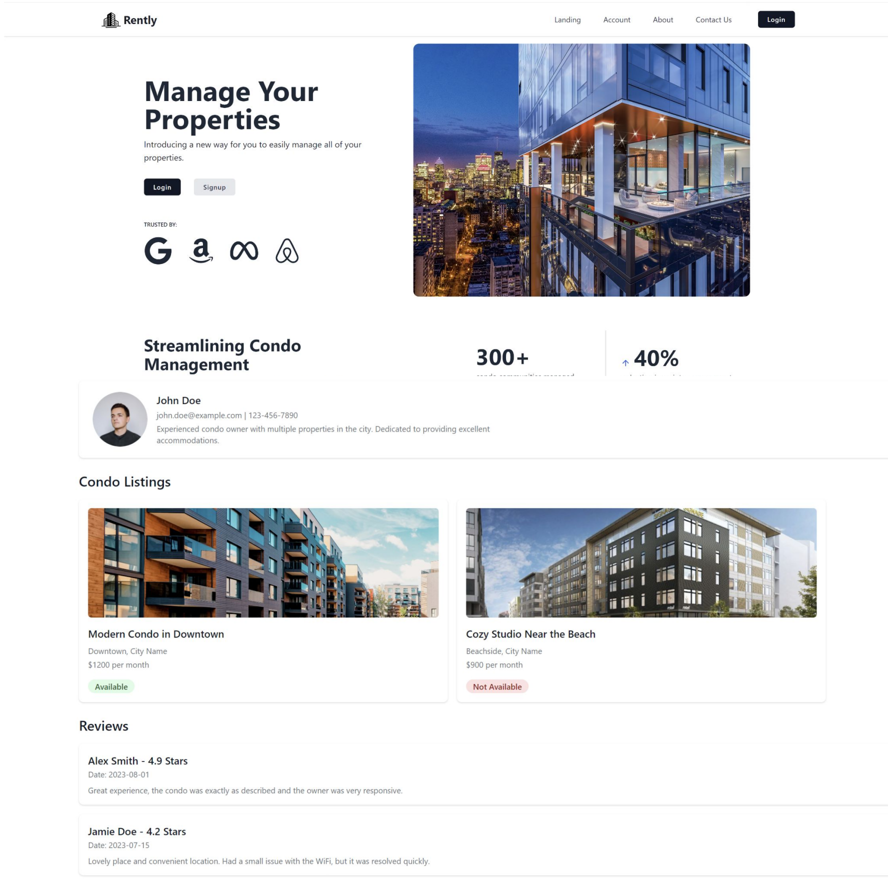
Rently

SOEN 390

Team 15

Abdelkader Habel - 40209153
Adam Boucher - 40165035
Adel Bouchatta - 40175598
Anes Khadiri - 40159080
Chems-Eddine Saidi - 40192094
Francesco Ferrato - 26642152
Omar Fares - 40162541
Oussama Cherifi - 40212275
Zakaria El Manar El Bouanani - 40190432

Rently Overview


Rently

Landing · Account · About · Contact Us · Login

Manage Your Properties

Introducing a new way for you to easily manage all of your properties.

Login · Signup

TRUSTED BY:

Streamlining Condo Management

300+

40%

John Doe
john.doe@example.com | 123-456-7890
Experienced condo owner with multiple properties in the city. Dedicated to providing excellent accommodations.

Condo Listings

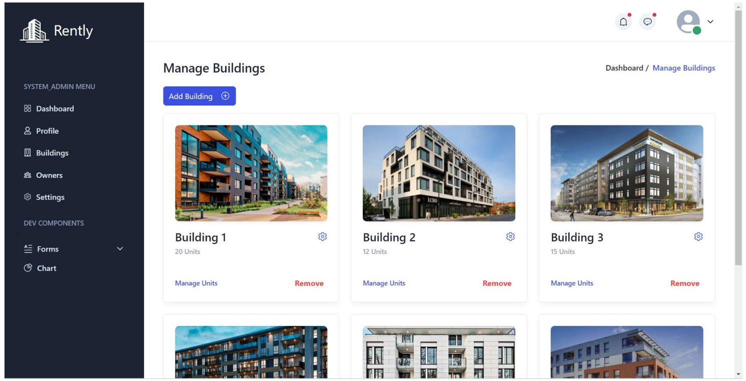
Modern Condo in Downtown
Downtown, City Name
$1200 per month
Available

Cozy Studio Near the Beach
Beachside, City Name
$900 per month
Not Available

Reviews

Alex Smith - 4.9 Stars
Date: 2023-08-01
Great experience, the condo was exactly as described and the owner was very responsive.

Jamie Doe - 4.2 Stars
Date: 2023-07-15
Lovely place and convenient location. Had a small issue with the WiFi, but it was resolved quickly.


Rently

SYSTEM_ADMIN MENU
- Dashboard
- Profile
- Buildings
- Owners
- Settings

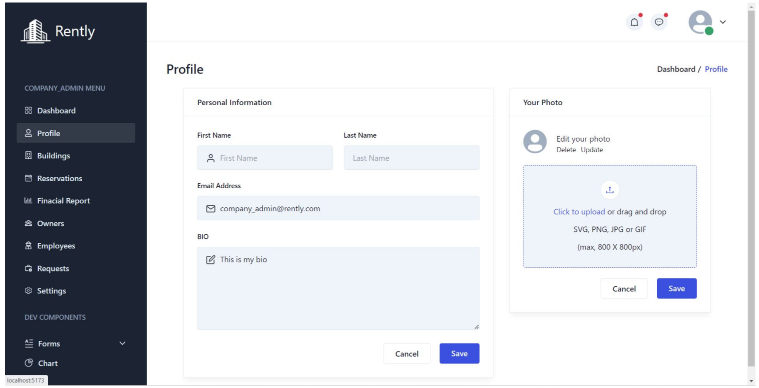
DEV COMPONENTS
- Forms
- Chart

Manage Buildings

Dashboard / Manage Buildings

Add Building

Building 1 20 Units Manage Units · Remove	Building 2 12 Units Manage Units · Remove	Building 3 15 Units Manage Units · Remove


Rently

COMPANY_ADMIN MENU
- Dashboard
- Profile
- Buildings
- Reservations
- Fiacial Report
- Owners
- Employees
- Requests
- Settings

DEV COMPONENTS
- Forms
- Chart

localhost:5173

Profile

Dashboard / Profile

Personal Information

First Name: First Name
Last Name: Last Name
Email Address: company_admin@rently.com
BIO: This is my bio

Cancel · Save

Your Photo

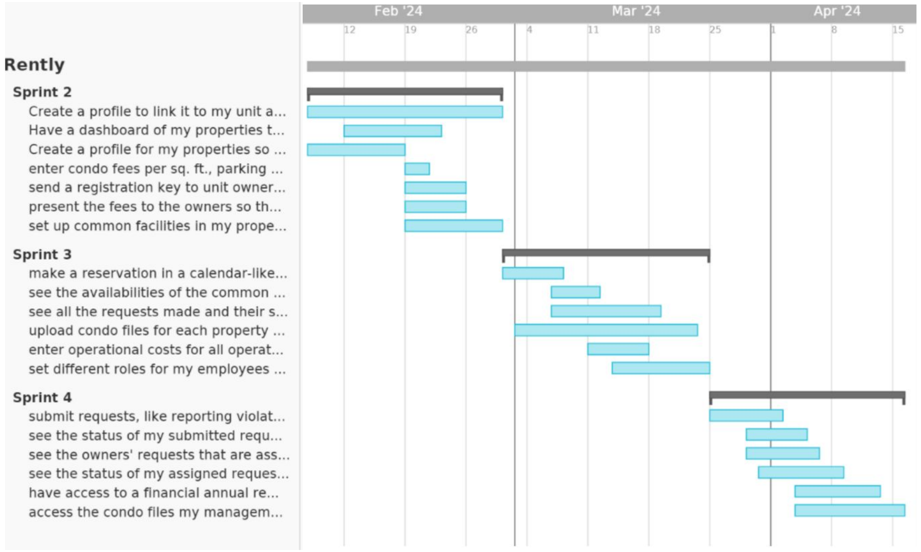
Edit your photo
Delete Update

Click to upload or drag and drop
SVG, PNG, JPG or GIF
(max, 800 X 800px)

Cancel · Save

Initial Release Plan

Planned Release



Domain Model

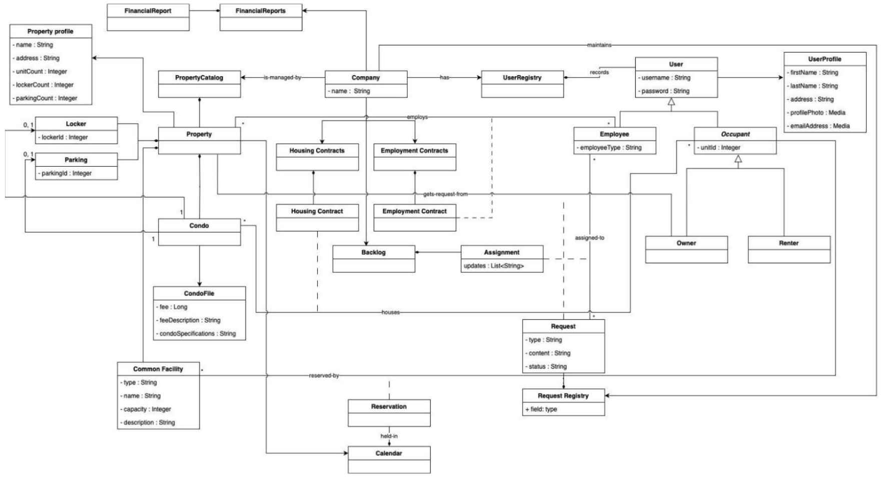
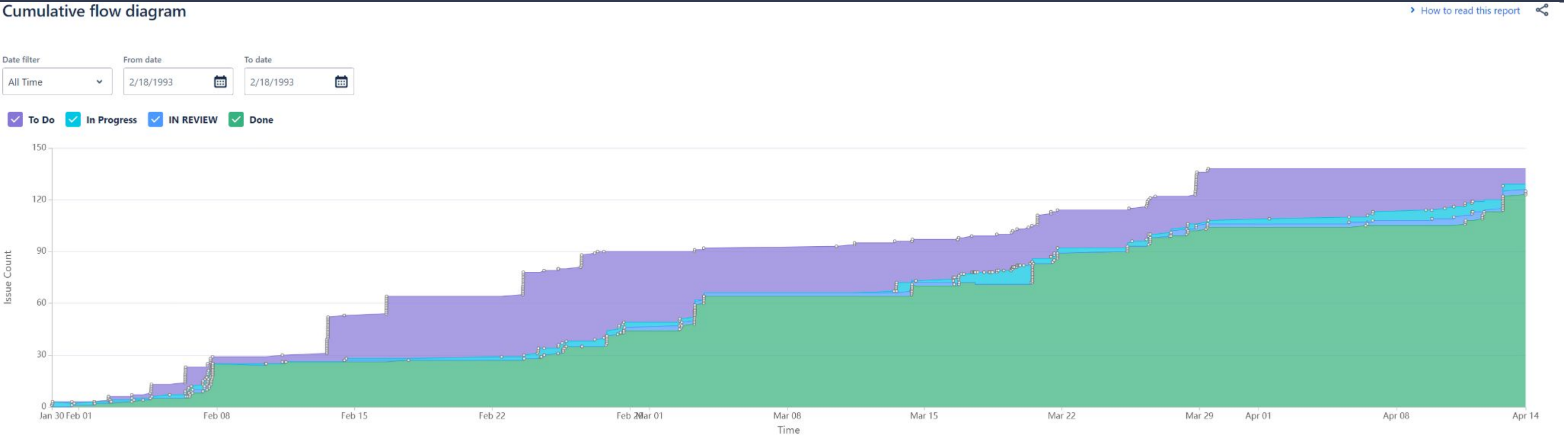
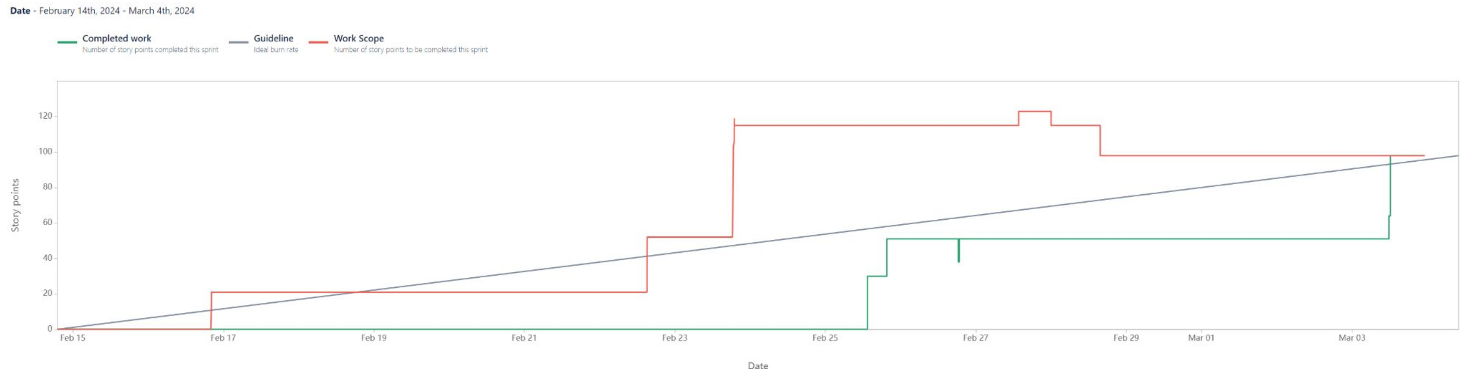


Figure 1: Domain model

Development Process

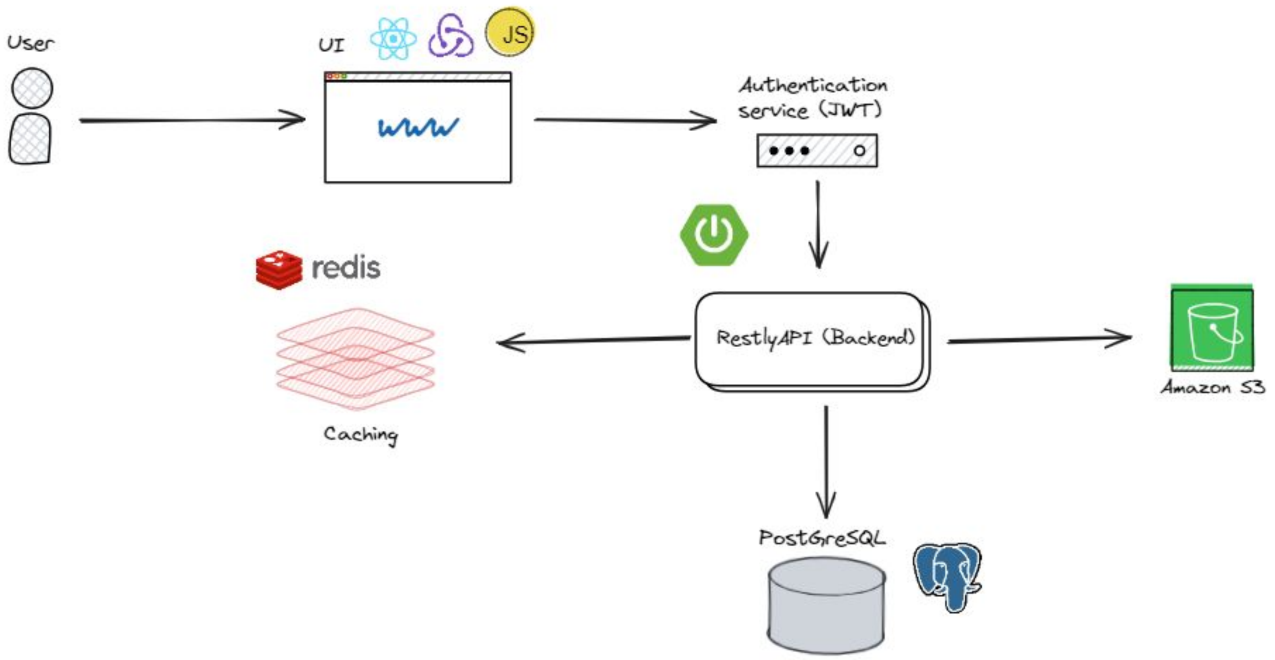




Todo, Progress, Review, Done, Repeat

Reduce Work Scope

Overall structure & technologies used



Component and architecture diagram

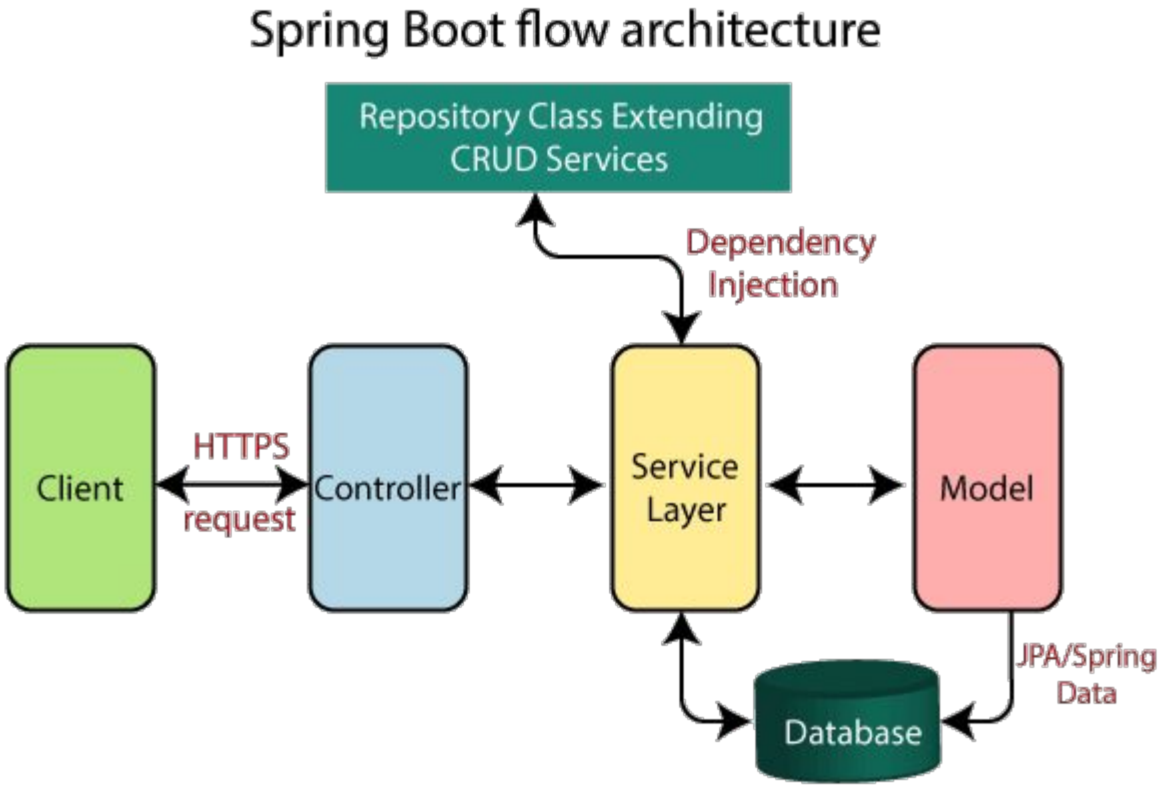
Why Spring Boot/React/PostgreSQL?

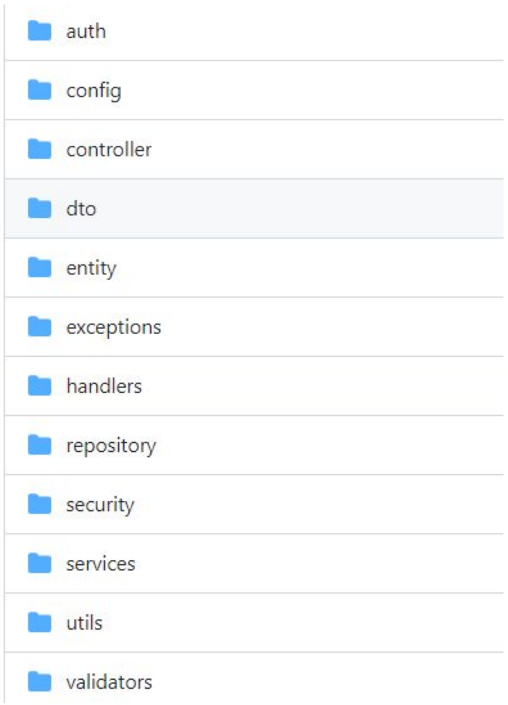
- One of the most robust backend framework
- Built on top of Java
- Easy to work with when the code becomes complexe
- Emphasize on the usage of design patterns

- Open-source & free SQL database
- Very popular in the industry
- Works very well with Spring boot and Spring JPA
- Enforce relationships and helps implement the domain model easily

- The most popular front-end framework
- Big community and can get help easily
- Heavily used in industry, good opportunity to learn it

Backend architecture - MVC

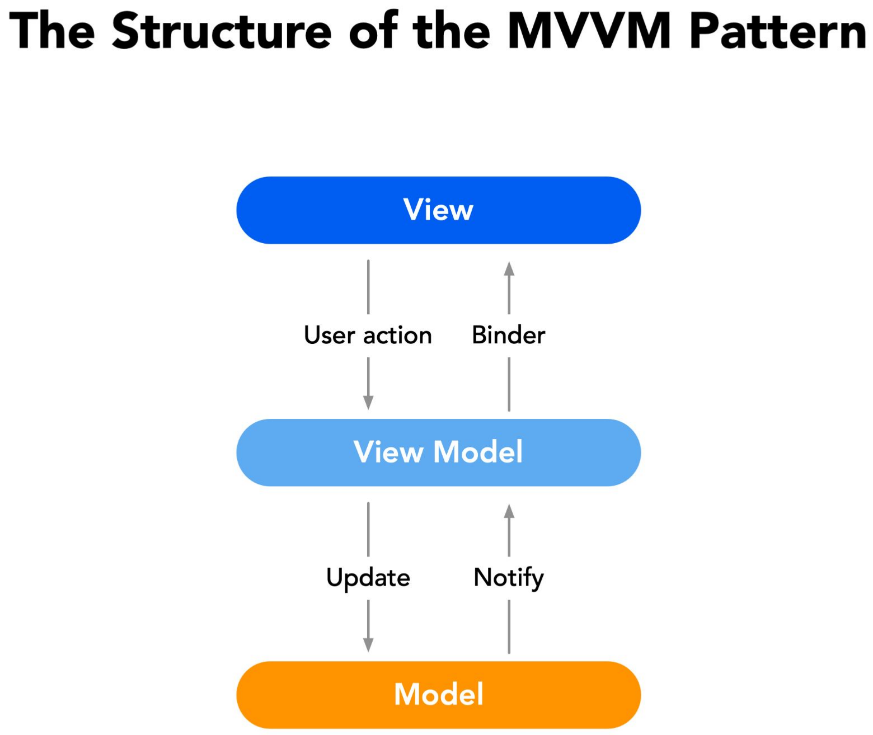


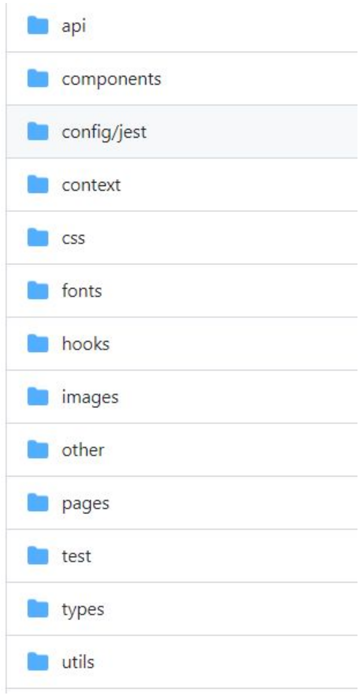


- auth
- config
- controller
- dto
- entity
- exceptions
- handlers
- repository
- security
- services
- utils
- validators

Folder structure of the backend layer

Frontend architecture - MVVM




api
components
config/jest
context
css
fonts
hooks
images
other
pages
test
types
utils

Folder structure of the frontend layer

Deployment and DB hosting

NEON

- Deploy PostgreSQL database online
- Easy to use and helps each team member to have his own DB instance

netlify

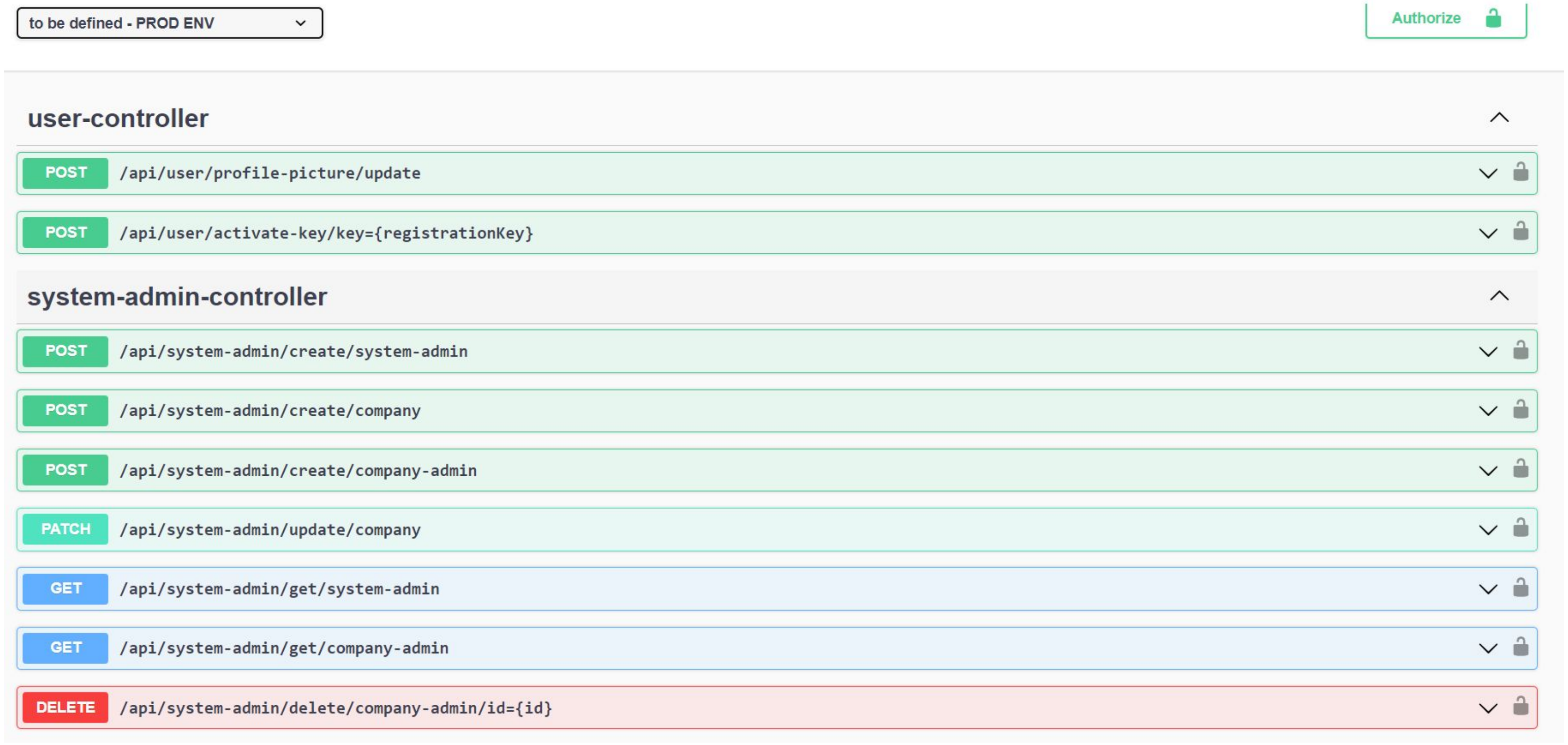
- Deploying front-end layer
- Live CI/CD integration with Github actions

Railway

- Similar to Netlify, but for the backend deployment
- Live CI/CD with github Actions

GitHub Actions

Other important tools - Swagger



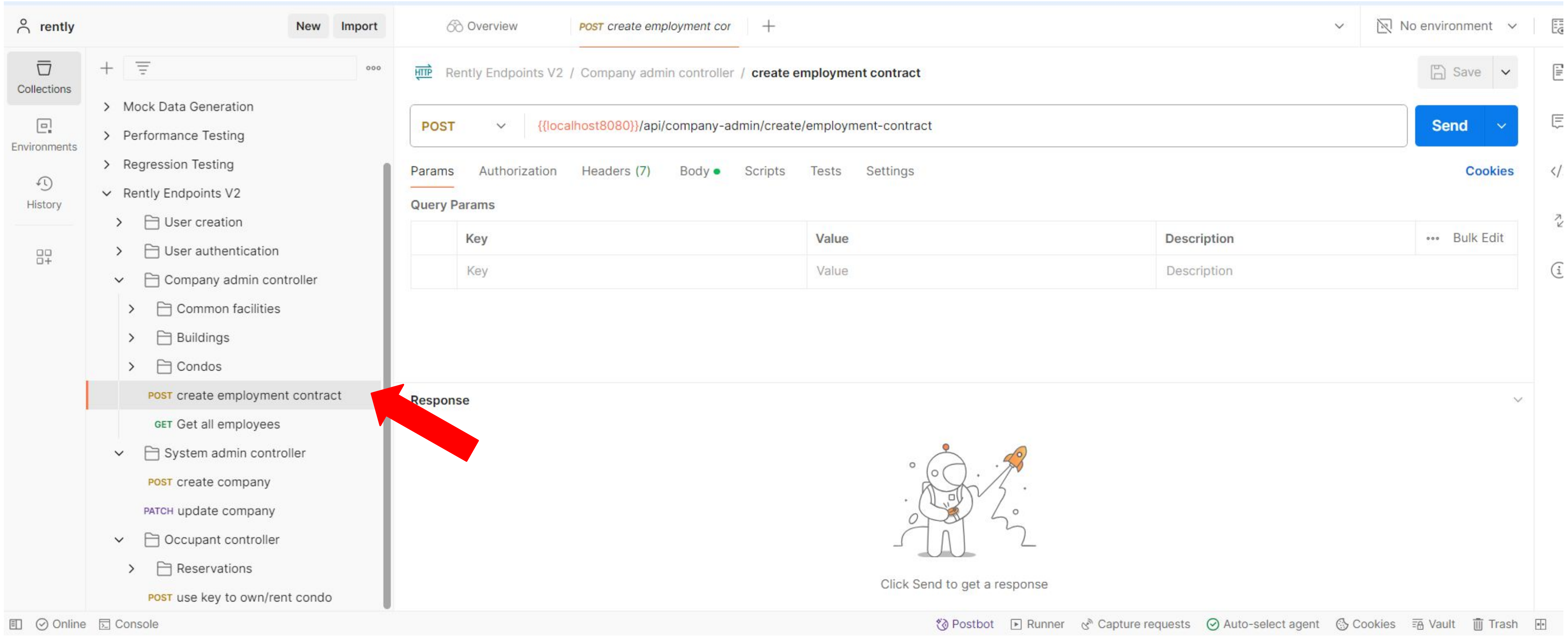
user-controller

```
POST   /api/user/profile-picture/update
POST   /api/user/activate-key/key={registrationKey}
```

system-admin-controller

```
POST   /api/system-admin/create/system-admin
POST   /api/system-admin/create/company
POST   /api/system-admin/create/company-admin
PATCH  /api/system-admin/update/company
GET    /api/system-admin/get/system-admin
GET    /api/system-admin/get/company-admin
DELETE /api/system-admin/delete/company-admin/id={id}
```

Other important tools - Postman collaboration



rently New Import

Overview POST create employment cor + No environment

Collections Environments History

- Mock Data Generation
- Performance Testing
- Regression Testing
- Rently Endpoints V2
  - User creation
  - User authentication
  - Company admin controller
    - Common facilities
    - Buildings
    - Condos
    - POST create employment contract
    - GET Get all employees
  - System admin controller
    - POST create company
    - PATCH update company
  - Occupant controller
    - Reservations
    - POST use key to own/rent condo

Rently Endpoints V2 / Company admin controller / create employment contract Save

POST {{localhost8080}}/api/company-admin/create/employment-contract Send

Params Authorization Headers (7) Body ● Scripts Tests Settings Cookies

Query Params

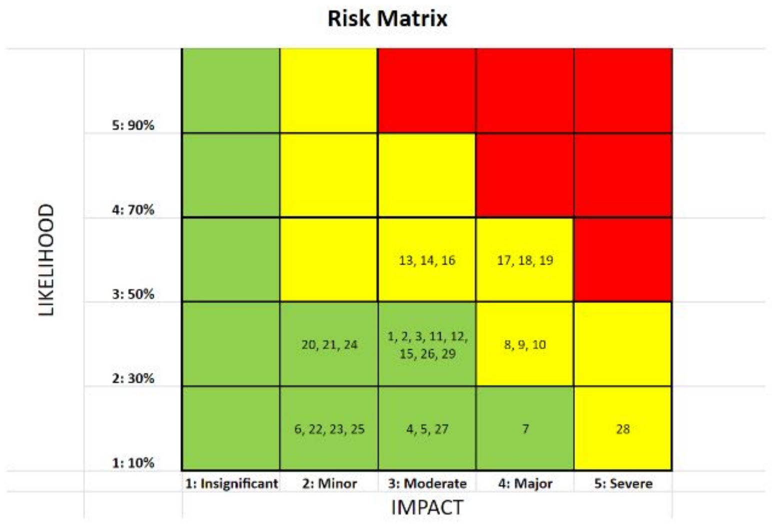
Key	Value	Description
Key	Value	Description

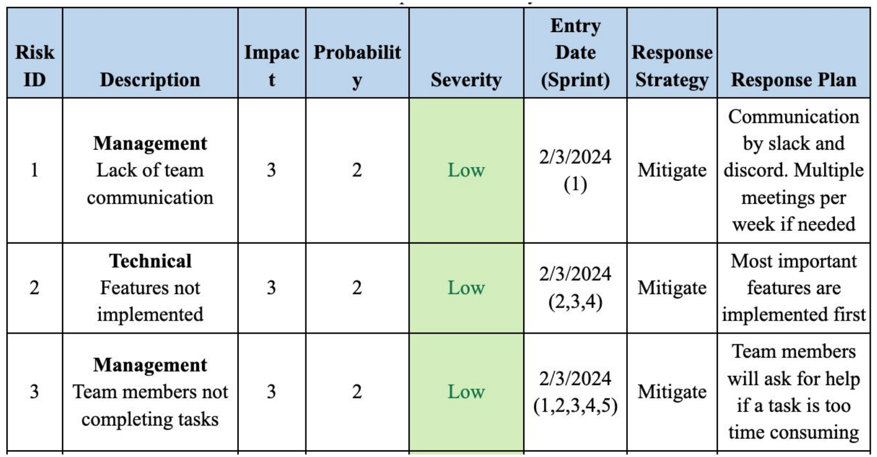
Response

Click Send to get a response

Online Console Postbot Runner Capture requests Auto-select agent Cookies Vault Trash

Risk management plan



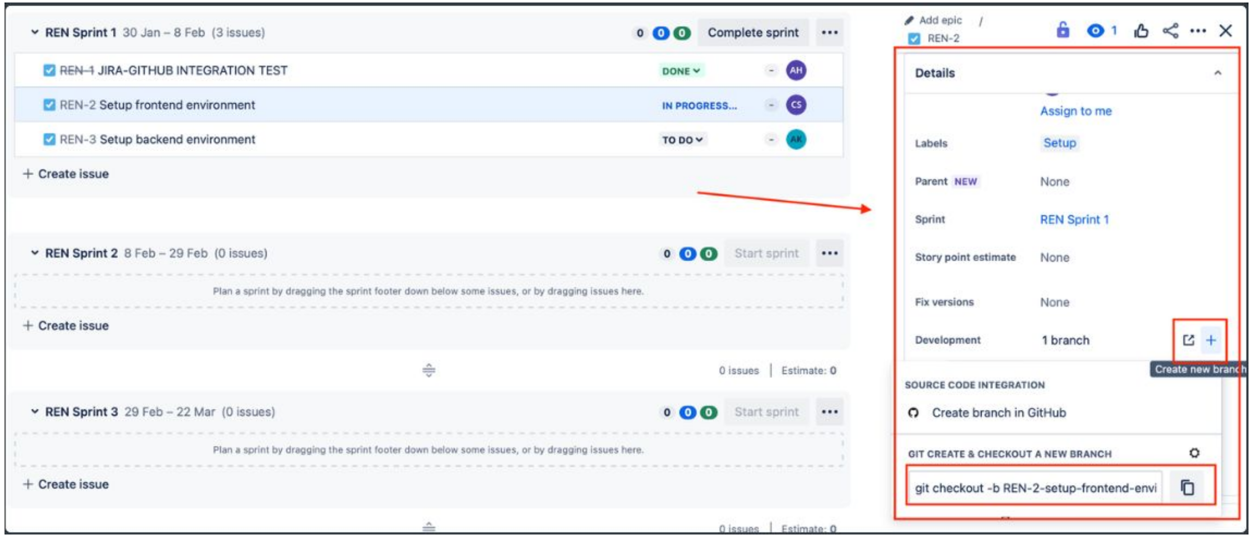


Risk ID	Description	Impact	Probability	Severity	Entry Date (Sprint)	Response Strategy	Response Plan
1	**Management** Lack of team communication	3	2	Low	2/3/2024 (1)	Mitigate	Communication by slack and discord. Multiple meetings per week if needed
2	**Technical** Features not implemented	3	2	Low	2/3/2024 (2,3,4)	Mitigate	Most important features are implemented first
3	**Management** Team members not completing tasks	3	2	Low	2/3/2024 (1,2,3,4,5)	Mitigate	Team members will ask for help if a task is too time consuming

Typically, there are 4 risk mitigation strategies used involved in project management:

- Accept
- Avoid
- Mitigate
- Transfer

Jira project management



REN Sprint 1 30 Jan – 8 Feb (3 issues) 0 0 0 Complete sprint

~~REN-1~~ JIRA-GITHUB INTEGRATION TEST	DONE	AH
REN-2 Setup frontend environment	IN PROGRESS...	CS
REN-3 Setup backend environment	TO DO	AK

+ Create issue

REN Sprint 2 8 Feb – 29 Feb (0 issues) 0 0 0 Start sprint

Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.

+ Create issue

0 issues | Estimate: 0

REN Sprint 3 29 Feb – 22 Mar (0 issues) 0 0 0 Start sprint

Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.

+ Create issue

0 issues | Estimate: 0

Add epic / REN-2

Details

Assign to me

Labels	Setup
Parent NEW	None
Sprint	REN Sprint 1
Story point estimate	None
Fix versions	None
Development	1 branch

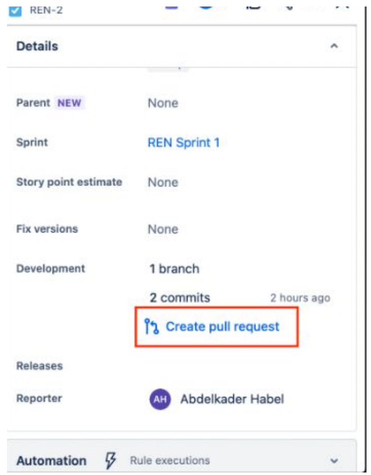
Create new branch

SOURCE CODE INTEGRATION

Create branch in GitHub

GIT CREATE & CHECKOUT A NEW BRANCH

```
git checkout -b REN-2-setup-frontend-envi
```

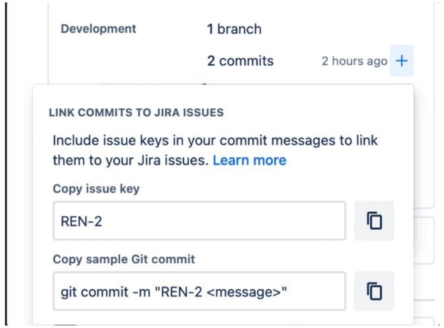


REN-2

Details

Parent NEW	None
Sprint	REN Sprint 1
Story point estimate	None
Fix versions	None
Development	1 branch
	2 commits 2 hours ago
	Create pull request
Releases	
Reporter	AH Abdelkader Habel

Automation Rule executions



Development	1 branch
	2 commits 2 hours ago

LINK COMMITS TO JIRA ISSUES

Include issue keys in your commit messages to link them to your Jira issues. Learn more

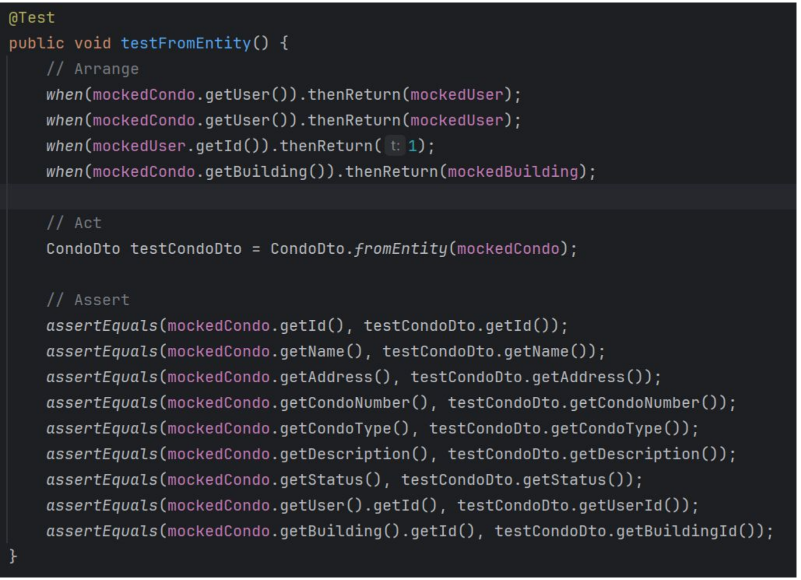
Copy issue key

```
REN-2
```

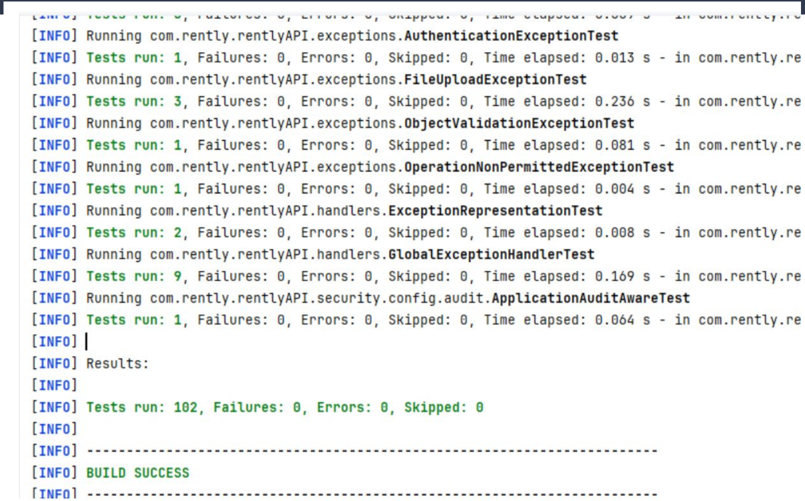
Copy sample Git commit

```
git commit -m "REN-2 <message>"
```

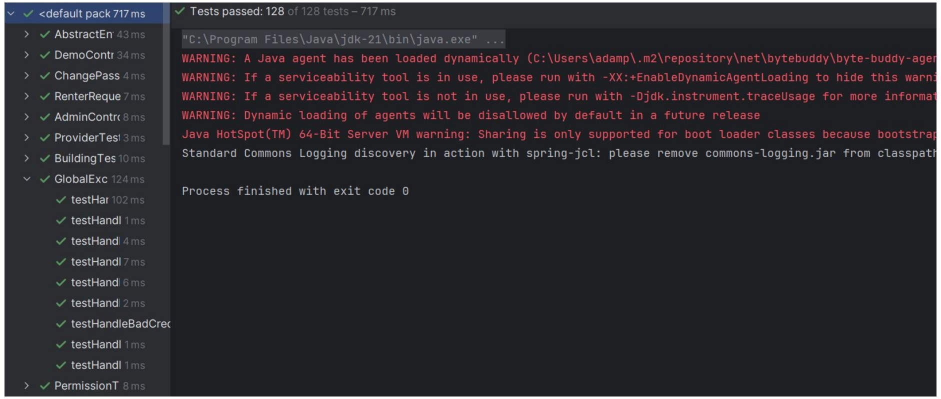
Testing - Backend



```java
@Test
public void testFromEntity() {
    // Arrange
    when(mockedCondo.getUser()).thenReturn(mockedUser);
    when(mockedCondo.getUser()).thenReturn(mockedUser);
    when(mockedUser.getId()).thenReturn( t: 1);
    when(mockedCondo.getBuilding()).thenReturn(mockedBuilding);

    // Act
    CondoDto testCondoDto = CondoDto.fromEntity(mockedCondo);

    // Assert
    assertEquals(mockedCondo.getId(), testCondoDto.getId());
    assertEquals(mockedCondo.getName(), testCondoDto.getName());
    assertEquals(mockedCondo.getAddress(), testCondoDto.getAddress());
    assertEquals(mockedCondo.getCondoNumber(), testCondoDto.getCondoNumber());
    assertEquals(mockedCondo.getCondoType(), testCondoDto.getCondoType());
    assertEquals(mockedCondo.getDescription(), testCondoDto.getDescription());
    assertEquals(mockedCondo.getStatus(), testCondoDto.getStatus());
    assertEquals(mockedCondo.getUser().getId(), testCondoDto.getUserId());
    assertEquals(mockedCondo.getBuilding().getId(), testCondoDto.getBuildingId());
}
```



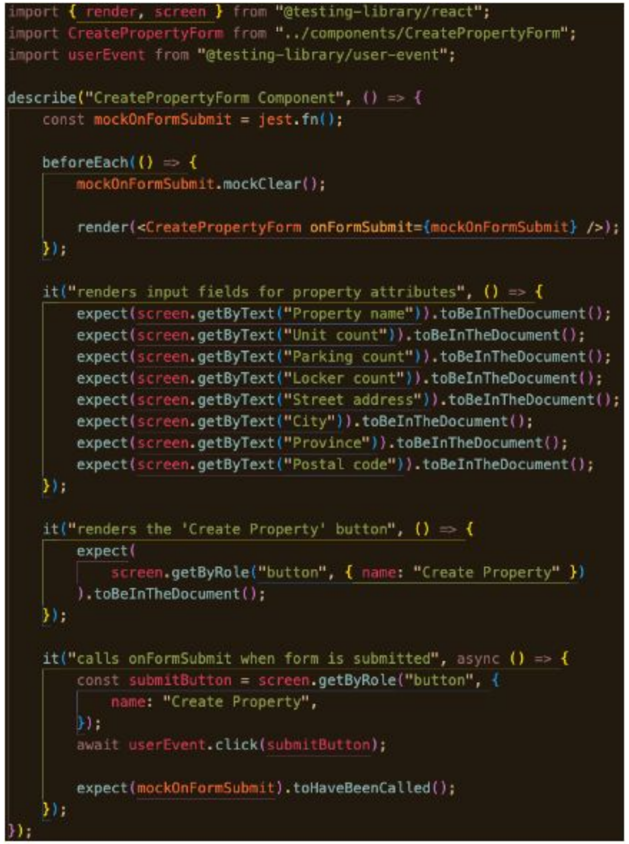
```
[INFO] Running com.rently.rentlyAPI.exceptions.AuthenticationExceptionTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.013 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.exceptions.FileUploadExceptionTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.236 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.exceptions.ObjectValidationExceptionTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.081 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.exceptions.OperationNonPermittedExceptionTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.handlers.ExceptionRepresentationTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.008 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.handlers.GlobalExceptionHandlerTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.169 s - in com.rently.re
[INFO] Running com.rently.rentlyAPI.security.config.audit.ApplicationAuditAwareTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.064 s - in com.rently.re
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 102, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```



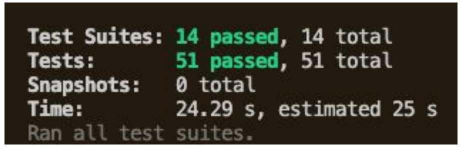
```
✓ <default pack 717 ms
  ✓ AbstractEn 43 ms
  ✓ DemoContr 34 ms
  ✓ ChangePass 4 ms
  ✓ RenterReque 7 ms
  ✓ AdminContr 8 ms
  ✓ ProviderTes 3 ms
  ✓ BuildingTes 10 ms
  ✓ GlobalExc 124 ms
    ✓ testHar 102 ms
    ✓ testHandl 1 ms
    ✓ testHandl 4 ms
    ✓ testHandl 7 ms
    ✓ testHandl 6 ms
    ✓ testHandl 2 ms
    ✓ testHandleBadCre
    ✓ testHandl 1 ms
    ✓ testHandl 1 ms
  ✓ PermissionT 8 ms
```

```
✓ Tests passed: 128 of 128 tests – 717 ms

"C:\Program Files\Java\jdk-21\bin\java.exe" ...
WARNING: A Java agent has been loaded dynamically (C:\Users\adamp\.m2\repository\net\bytebuddy\byte-buddy-agen
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warni
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more informat
WARNING: Dynamic loading of agents will be disallowed by default in a future release
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath

Process finished with exit code 0
```

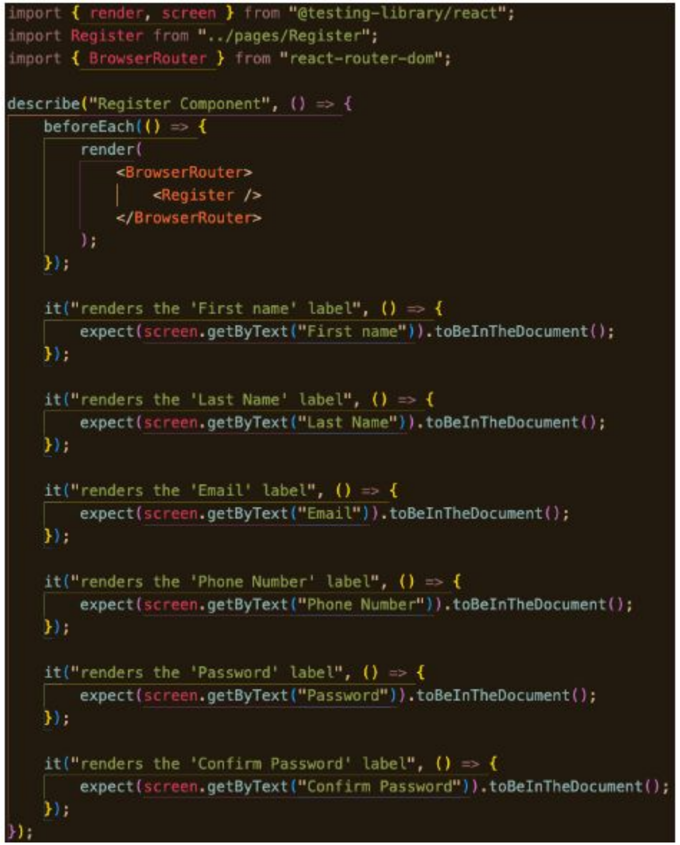
Testing with Jest - Frontend



```javascript
import { render, screen } from "@testing-library/react";
import CreatePropertyForm from "../components/CreatePropertyForm";
import userEvent from "@testing-library/user-event";

describe("CreatePropertyForm Component", () => {
    const mockOnFormSubmit = jest.fn();

    beforeEach(() => {
        mockOnFormSubmit.mockClear();

        render(<CreatePropertyForm onFormSubmit={mockOnFormSubmit} />);
    });

    it("renders input fields for property attributes", () => {
        expect(screen.getByText("Property name")).toBeInTheDocument();
        expect(screen.getByText("Unit count")).toBeInTheDocument();
        expect(screen.getByText("Parking count")).toBeInTheDocument();
        expect(screen.getByText("Locker count")).toBeInTheDocument();
        expect(screen.getByText("Street address")).toBeInTheDocument();
        expect(screen.getByText("City")).toBeInTheDocument();
        expect(screen.getByText("Province")).toBeInTheDocument();
        expect(screen.getByText("Postal code")).toBeInTheDocument();
    });

    it("renders the 'Create Property' button", () => {
        expect(
            screen.getByRole("button", { name: "Create Property" })
        ).toBeInTheDocument();
    });

    it("calls onFormSubmit when form is submitted", async () => {
        const submitButton = screen.getByRole("button", {
            name: "Create Property",
        });
        await userEvent.click(submitButton);

        expect(mockOnFormSubmit).toHaveBeenCalled();
    });
});
```



```
Test Suites: 14 passed, 14 total
Tests:       51 passed, 51 total
Snapshots:   0 total
Time:        24.29 s, estimated 25 s
Ran all test suites.
```



```javascript
import { render, screen } from "@testing-library/react";
import Register from "../pages/Register";
import { BrowserRouter } from "react-router-dom";

describe("Register Component", () => {
    beforeEach(() => {
        render(
            <BrowserRouter>
                <Register />
            </BrowserRouter>
        );
    });

    it("renders the 'First name' label", () => {
        expect(screen.getByText("First name")).toBeInTheDocument();
    });

    it("renders the 'Last Name' label", () => {
        expect(screen.getByText("Last Name")).toBeInTheDocument();
    });

    it("renders the 'Email' label", () => {
        expect(screen.getByText("Email")).toBeInTheDocument();
    });

    it("renders the 'Phone Number' label", () => {
        expect(screen.getByText("Phone Number")).toBeInTheDocument();
    });

    it("renders the 'Password' label", () => {
        expect(screen.getByText("Password")).toBeInTheDocument();
    });

    it("renders the 'Confirm Password' label", () => {
        expect(screen.getByText("Confirm Password")).toBeInTheDocument();
    });
});
```

Challenges

- Difficult to have a team of 9 people, we got through it with good planning and assigning tasks from the beginning.

- Massive refactoring challenge during sprint 3 that slowed us down in the backend.

- Handling the deployment of the frontend and backend together. We faced difficulties for the frontend on Netlify because of configuration files.

What we learned

- It is very important to assign tasks from the very beginning and choose team members that will be “team leads”
- The design is a crucial part of the development process, it helps avoid many problems down the road. This was learned when the design was not followed initially and we had to redo most of the backend.
- How to use new technologies, like Spring Boot, SwaggerUI, Postman, Jira

Most importantly, we learned how to collaborate better in a team setting where everyone is responsible for something important