Testing plan

Testing Tools:

Backend (Spring Boot):

- **Unit Tests:** We will employ JUnit for robust backend unit testing, focusing on business logic, data validation, and database interactions.
- **Integration Tests:** We will conduct thorough integration testing to ensure collaboration between different backend components.

Frontend (React):

- **Unit Tests:** Utilizing Jest for React component testing, we will validate UI rendering, state changes, and user interactions.
- **Integration Tests:** Implementing Cypress for end-to-end testing, we aim to ensure the frontend communicates effectively with the backend.

CI/CD (GitHub Actions):

- **Continuous Integration:** GitHub Actions will automate unit and integration tests on each pull request and push to the main branch.
- **Continuous Deployment:** Utilizing GitHub Actions, we plan to deploy to a staging environment, enabling further testing post-build.

Testing Approach:

Unit Testing:

We will conduct unit tests for each backend and frontend module, ensuring individual components function as intended.

Integration Testing:

Verification that different modules collaborate seamlessly and communicate correctly is crucial to our testing approach.

End-to-End Testing:

Performing end-to-end tests, we aim to validate the complete application flow, from user interactions to backend processes.

User Acceptance Testing (UAT):

We plan to engage stakeholders, including condo owners and management, in acceptance testing to ensure the application meets their specific requirements.

Testing Metrics

Code Coverage:

We aim for a minimum of 80% code coverage for both backend and frontend codebases.

Defect Density:

Monitoring the number of defects identified per unit of code will allow us to assess code quality. We plan to use *Datadog* for future monitoring.

Test Case Execution and Pass Rate:

Tracking test case execution and pass rates will help gauge the overall health of the testing process.

Testing Coverage

Backend:

Ensuring comprehensive coverage of business logic, API endpoints, and database interactions is integral to our testing plan.

Frontend:

Our testing approach includes covering UI components, state management, and user interactions.

Acceptance Tests

1. **User Profile Creation:**
    - We will verify that users can create/update profileswith the necessary information.
2. **Registration Key Validation:**
    - Ensuring the system correctly validates and processes registration keys is a critical acceptance test.
3. **Condo Management Features:**
    - We plan to validate the creation of property profiles, uploading of condo files, and entry of unit details.
4. **Financial System:**
    - Testing condo fee calculation, presentation, operational budget recording, and generation of annual financial reports will be a focus.
5. **Reservation System:**

- Our approach involves validating the setup of common facilities, the reservation process, and availability display.

6. **Request Submission and Assignment:**
   - We will test the submission of various requests by condo owners and their correct assignment to employees.

7. **Additional and Bonus Features:**
   - Forum functionality, event organization, and login using Gmail or other accounts will be thoroughly tested.