

ELETRIC NET

Abel Teixeira - 113655 Product Owner
Bruno Lopes - 68264 Team Manager/Devops Master
Tiago Albuquerque - 112901 QA Manager



Product concept

Problem:

Fragmentation in the EV charging ecosystem with:

- Inability to make advance reservations
- Lack of real-time visibility
- Inefficient operational management

Solution:

Unified platform with:

- Smart reservations by time slot
- Consumption and cost dashboard
- Centralized management for operators
- Discounts during off-peak hours



Scenarios

- 1** Joana intends to reserve a charging station from the map and receives a confirmation notification.
- 2** Carlos reports an issue with a charging station, accesses the management area, and marks the station as "Under Maintenance."
- 3** André logs into the platform to view a chart showing the total number of charging sessions per month and their respective costs.
- 4** Marta notices that the stations are frequently overloaded between 6 PM and 8 PM, so she sets a 25% discount for reservations made between 12 AM and 6 AM.

Epics

- 1** **Station Search**
Allow users to view available charging stations on a map.
- 2** **Slot Booking**
Enable users to reserve charging slots at specific times.
- 3** **Charging Session**
Support starting a charging session and tracking consumption.
- 4** **Personal Dashboard**
Show users a history of charging sessions with monthly costs.
- 5** **Station Management**
Let operators manage station availability, maintenance, and usage stats.
- 6** **Off-Peak Discounts**
Enable dynamic discounts to reduce peak-time congestion.



User Stories

The screenshot shows a Jira user story card for 'US3: Como Joana, quero ver a disponibilidade em tempo real para evitar deslocar-me a postos ocupados.' The card includes sections for Description, Child work items, Linked work items, Test Coverage, and Details. The 'Details' section is expanded, showing assignee (Bruno Lopes), labels (Frontend), parent (EN-30 Procura de Estações), team (None), sprint (I5 +2), story point estimate (None), fix versions (None), development (Open with Atlassian for VS...), reporter (Bruno Lopes), and automation (Rule executions).

EN-30 / EN-33

US3: Como Joana, quero ver a disponibilidade em tempo real para evitar deslocar-me a postos ocupados.

+ Add Apps

Description
Add a description...

Child work items
0% Done

T...	Key	Summary	P...	A...	Status
EN-78	Time Open slots	Time Open slots	=		REVIEW/TEST

Linked work items

is tested by

EN-57 Unit and integration tests	DONE	=
EN-110 whenRegister_withInvalidRole_thenRet...	DONE	=
EN-85 Example #1.1	DONE	=

Test Coverage

Add Tests Execute

Analysis & Scope
Scope: Latest Final Status

OK

Done ✓ Done

Details

Assignee Bruno Lopes

Labels Frontend

Parent EN-30 Procura de Estações

Team None

Sprint I5 +2

Story point estimate None

Fix versions None

Development Open with Atlassian for VS...
Create branch
Create commit

Reporter Bruno Lopes

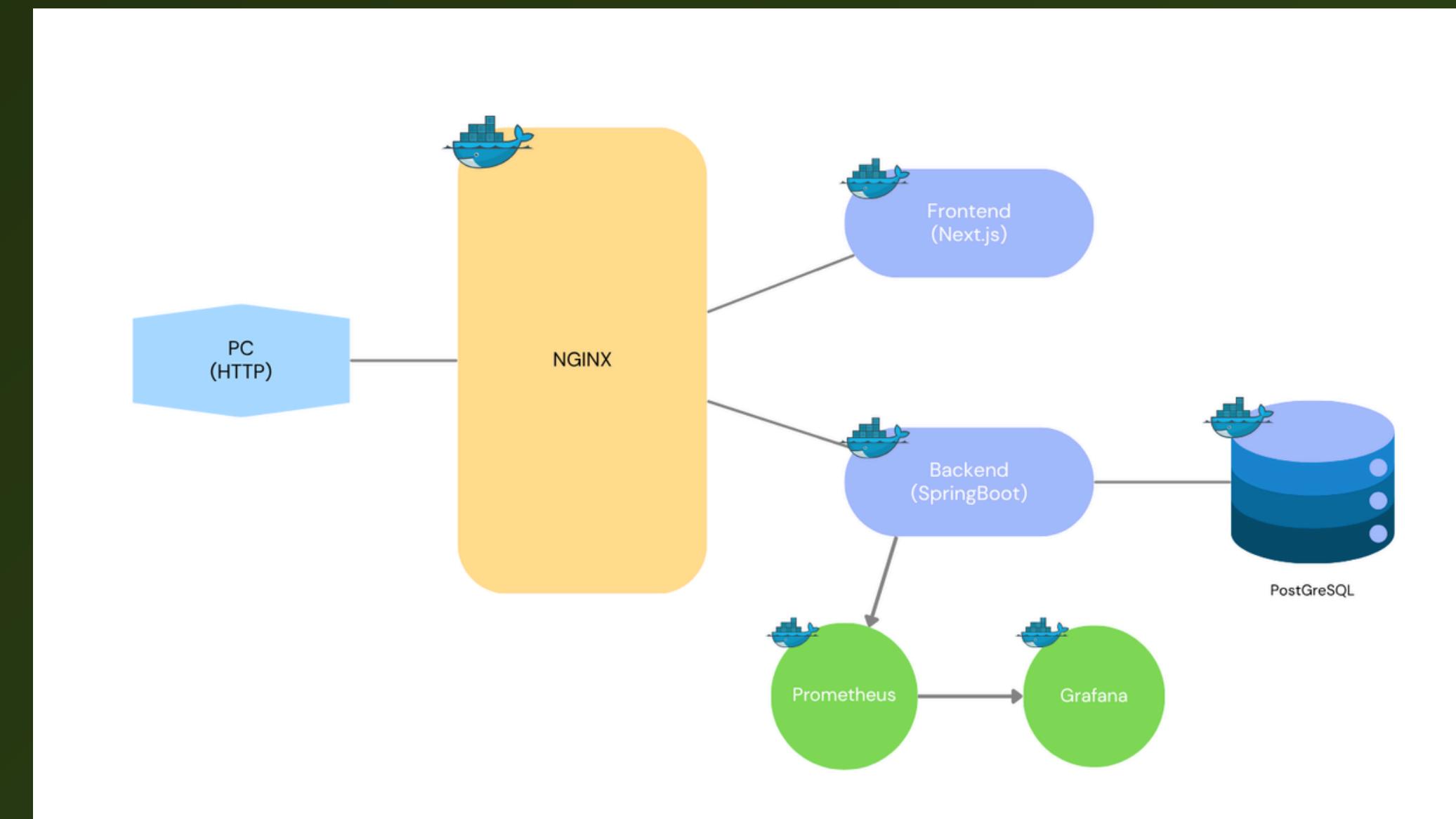
Automation Rule executions

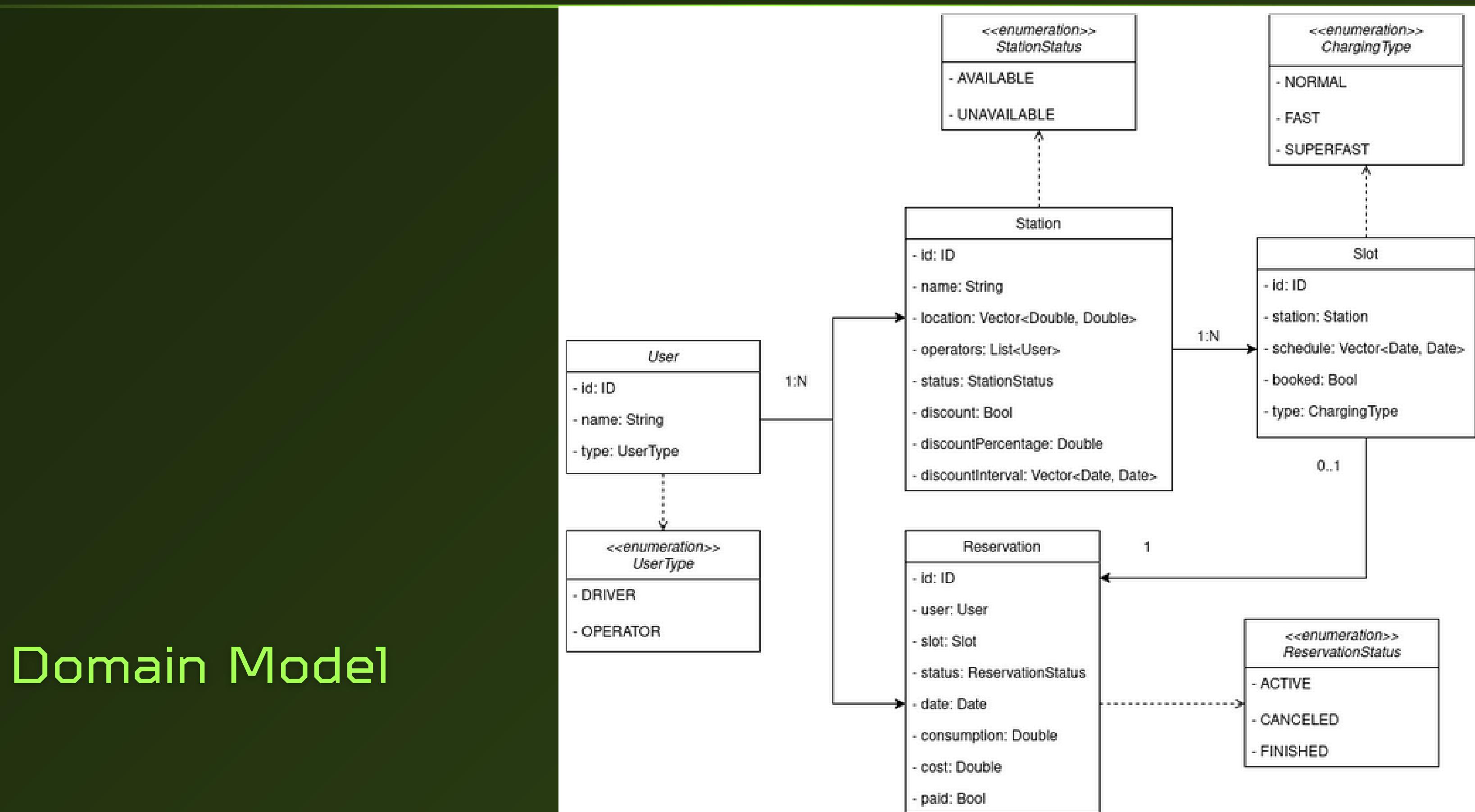
Created May 9, 2025 at 6:13 PM Updated 1 minute ago

Configure

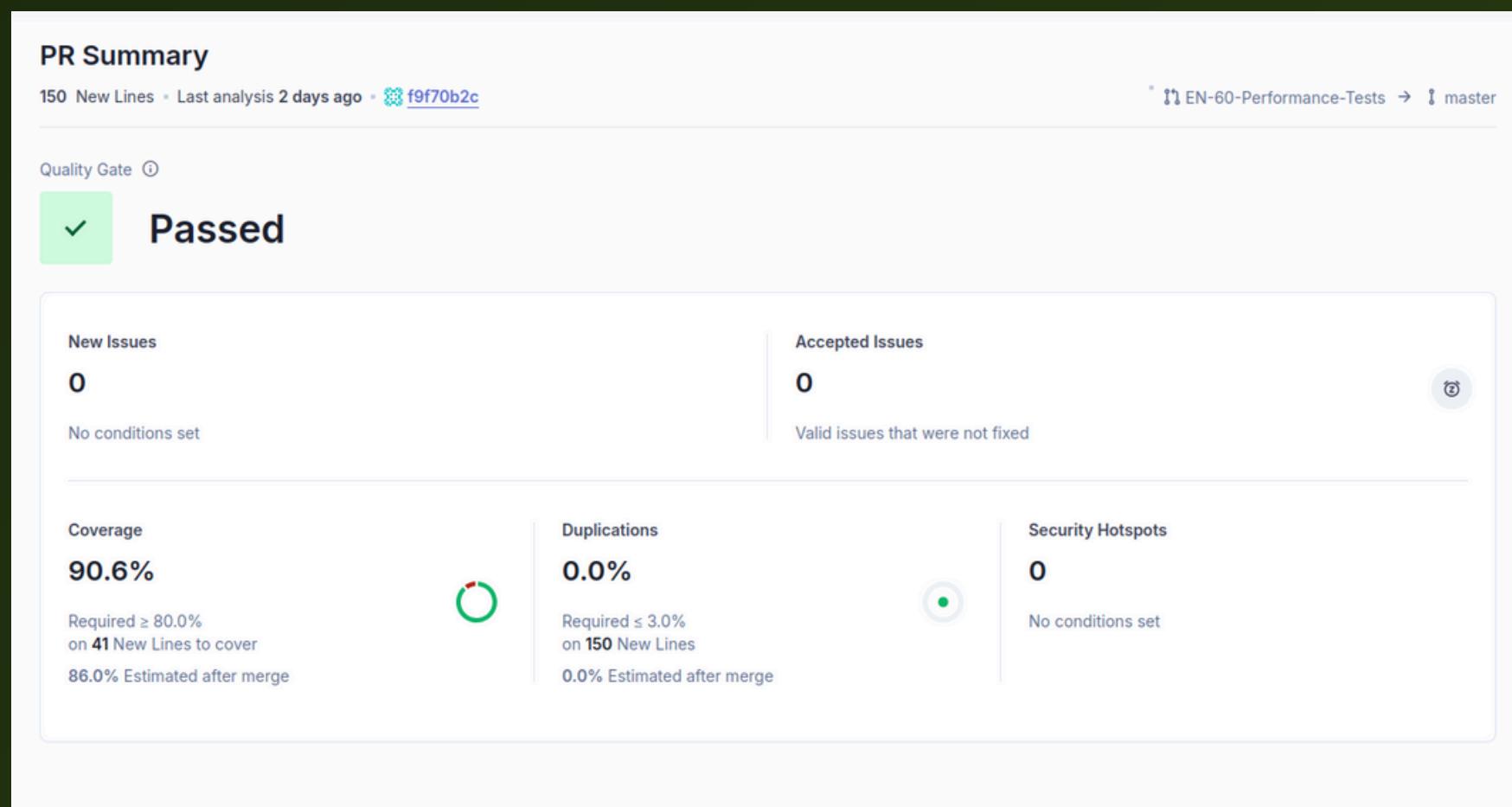
Software architecture

- FRONTEND
 - React
 - Vite
 - Tailwind CSS
- BACKEND
 - SpringBoot
- POSTGRESQL
 - Persistante data with reservations and chargers
- NGINX
 - Proxy to improve security on authentication
- DOCKER
 - All componentes containerize
- MONITOR
 - Actuator=>Prometheus=>Grafana





- Sonarqube



- Coding Style

- **ESLint**

- JavaScript
 - TypeScript

- **Checkstyle**

- Java
 - Google Style Guide

Code Quality
Management



Automated Tests:

- Unit Tests: JUnit (Backend $\geq 80\%$ coverage)
- Integration: Testcontainers (PostgreSQL)
- E2E: Cucumber + Playwright (Frontend)

QA stack

Static Analysis:

- SonarQube Quality Gate:
- Coverage $\geq 80\%$
- Duplication $\leq 3\%$
- CodeQL for vulnerabilities

Traceability:

- User Stories → Tests (Xray + Jira)
- Requirements Coverage: 100% MVP

OK

Status	Key	Summary	Test Status
DONE	EN-85	Example #1.1	PASSED
DONE	EN-110	whenRegister_withInvalidRole_thenReturn400	PASSED

CI Pipeline

Triggered via pull request 4 days ago
Brupez synchronize #22 [EN-73-Add-notifications](#)

Status	Total duration	Artifacts
Success	8m 9s	1

build.yml
on: pull_request

```
graph LR; A[SonarQube Analysis] -- "4m 25s" --> B[Deploy]; B -- "1m 15s" --> C[Lighthouse]; C -- "2m 9s" --> D[Cleanup]; D -- "2s" --> E[Check Status]; E -- "3s" --> F[ ];
```

Artifacts
Produced during runtime

Name	Size	Digest
lighthouse-results	189 KB	sha256:c1b8b886bd21b48c8508cdcc3dd80d982b4d186c14da9ad4861fe182095d73a0

github-actions bot commented 2 days ago

Lighthouse Results

- performance: 61/100
- accessibility: 82/100
- best-practices: 100/100
- seo: 83/100

coverage

sonarqubecloud bot commented 2 days ago

Quality Gate passed

Issues

- 0 New issues
- 0 Accepted issues

Measures

- 0 Security Hotspots
- 90.6% Coverage on New Code
- 0.0% Duplication on New Code

[See analysis details on SonarQube Cloud](#)



Code Review

Hotfix1- fixing login on vm #24
Brupez merged 6 commits into `master` from `hotfix1` 3 days ago

Merged

Brupez removed the request for review from [TiagoAlb12](#) 3 days ago

Brupez approved these changes 3 days ago [View reviewed changes](#)

Brupez left a comment

Owner ...

good, keep it up!

Brupez merged commit `cd8d877` into `master` 3 days ago
18 checks passed

View details Revert

Pull request successfully merged and closed [Delete branch](#)

You're all set — the `hotfix1` branch can be safely deleted.

Conversation 6 Commits 12 Checks 4 Files changed 18 +596 -181

TiagoAlb12 commented 5 days ago

Collaborator ...

This pull request introduces changes to enhance the reservation management system by adding a new endpoint for fetching reservation details by ID, improving the data model, and updating the frontend to display reservation details in a UI. It also includes several bug fixes and performance optimizations.

Reviewers

Copilot ttabelhaxd

Functional Tests

```
public class BookingStationSteps { & Brupez
    @When("I enter {string} in the email field") & Brupez
    public void iEnterInTheEmailField(String email) { page.getByLabel(text: "Email").fi
    @When("I enter {string} in the password field") & Brupez
    public void iEnterInThePasswordField(String password) { page.locator(selector: "#p
    @When("I enter {string} in the confirm password field") & Brupez
    public void iEnterInTheConfirmPasswordField(String password) { page.locator(selector: "#c
    @And("I select {string} from the role dropdown") & Brupez
    public void iSelectFromTheRoleDropdown(String role) {
        Locator roleDropdown = page.getRole(AriaRole.COMBOBOX, new Page.GetByRoleOpt
        roleDropdown.selectOption(role);
    }

    @When("I click the register button") & Brupez
    public void iClickTheRegisterButton() {
        page.getRole(AriaRole.BUTTON, new Page.GetByRoleOptions().setName("Sign Up"))
    }

    @Then("I should be redirected to the login page and homepage should be visible")
    public void iShouldBeRedirectedToLoginPage() {
        page.waitForURL("**/login");
        assertTrue(page.url().contains("/login"), message: "URL should contain '/login"
        page.getRole(AriaRole.BUTTON, new Page.GetByRoleOptions().setName("Sign In"))
    }

    // Add Charger Admin Side
}

Feature: Booking charger Management
As a user of the ElecricNET I want to be able to book a charger so tha
Scenario Outline: Authenticated User
Given I navigate to "http://localhost"
When I click on the "Sign Up" button
And I enter "<name>" in the name field
And I enter "<email>" in the email field
And I enter "<password>" in the password field
And I enter "<password>" in the confirm password field
And I select "<role>" from the role dropdown
And I click the register button
Then I should be redirected to the login page and homepage should be
Examples:
| name | email | password | role |
| John Doe | john.doe@example.com | Password123! | USER |
| Alice Smith | alice.smith@gmail.com | Secret456! | USER |
| Bob Brown | bob.brown@hotmail.com | Secure789! | ADMIN |

Scenario: Admin creates a new charger
Given I navigate to "http://localhost"
And I enter "bob.brown@hotmail.com" in the email field
And I enter "Secure789!" in the password field
And I click on the "Sign In" button
Then I should see the admin dashboard

When I click on the "Add Charger" button
And I enter "Fast Charger Porto" in the charger name field
And I enter "Porto" in the location field
And I select "ULTRA_FAST" from the charging type dropdown
And I enter "350" in the power field
```

Unit and Integration Tests

```

package controllers;

> import ...;

@ExtendWith(MockitoExtension.class) & Brupez
@ContextConfiguration(classes = EleticNET_BackendApplication.class)
class AuthControllerTest {

    @Mock 2 usages
    private AuthenticationManager authManager;

    @Mock 8 usages
    private UserDetailsServiceImpl userDetailsService;

    @Mock 2 usages
    private JwtUtil jwtUtil;

    @BeforeEach & Brupez
    void setup() {
        AuthController authController = new AuthController(authManager, userDetailsService, jwtUtil);
        MockMvc mockMvc = MockMvcBuilders
            .standaloneSetup(authController)
            .build();
        RestAssuredMockMvc.mockMvc(mockMvc);
    }

    @Test & Brupez
    void whenLogin_withValidCredentials_thenReturn200AndToken() {
        AuthRequest request = new AuthRequest();
        request.setEmail("test@example.com");
        request.setPassword("password");

        UserDetails userDetails = new User(username: "test@example.com", password: "password",
            List.of(new SimpleGrantedAuthority(role: "ROLE_USER")));

        when(userDetailsService.loadUserByUsername("test@example.com")).thenReturn(userDetails);
        when(jwtUtil.generateToken(anyString(), anyString())).thenReturn(value: "test-token");
        when(userDetailsService.getUserId(any())).thenReturn(value: 1L);
        when(userDetailsService.getUserName(any())).thenReturn(value: "Test User");
    }
}

```

```

class ReservationServiceTest {

    @Test & Brupez
    void getAllReservations_Success() {
        when(reservationRepository.findAll()).thenReturn(List.of(reservation));
        List<ReservationResponseDTO> result = reservationService.getAllReservations();

        assertThat(result).hasSize(expected: 1);
        ReservationResponseDTO dto = result.get(0);
        assertThat(dto.getId()).isEqualTo(reservation.getId());
        assertThat(dto.getUserId()).isEqualTo(user.getId());
        assertThat(dto.getSlotId()).isEqualTo(slot.getId());
        assertThat(dto.getStationName()).isEqualTo(station.getName());
        assertThat(dto.getChargingType()).isEqualTo(slot.getChargingType().name());
    }

    @Test & Brupez
    void createReservation_UserNotFound_ReturnEmpty() {
        when(userRepository.findById(1L)).thenReturn(Optional.empty());
        when(slotRepository.findById(1L)).thenReturn(Optional.of(slot));

        Optional<ReservationResponseDTO> result = reservationService.createReservation(requestDTO);

        assertThat(result).isEmpty();
        verify(slotRepository, never()).save(any());
        verify(reservationRepository, never()).save(any());
    }

    @Test & Brupez
    void createReservation_SlotNotFound_ReturnEmpty() {
        when(userRepository.findById(1L)).thenReturn(Optional.of(user));
        when(slotRepository.findById(1L)).thenReturn(Optional.empty());

        Optional<ReservationResponseDTO> result = reservationService.createReservation(requestDTO);

        assertThat(result).isEmpty();
        verify(slotRepository, never()).save(any());
        verify(reservationRepository, never()).save(any());
    }
}

```

Product tour - Demo



PMI Board

Plus	Minus	Interesting
Almost 90% code coverage	Started treating tests at the middle of development	Leading with data coming from external API's to take the most advantage of them
Robust CI/CD pipelines	Cannot book external (from maps API) charger slots	Sonarqube integrated in the pipeline
Good workflow and communication within the team	Selenium Testing (used Playwright instead)	Integration of Xray with Jira and communicate with the team “without communicating”



THANK
YOU!
QUESTIONS?



<http://deti-tqs-05.ua.pt/>



<https://github.com/Brupez/EletreicNET>