XIDERAL

SELF-DRIVING VEHICLE MANAGEMENT SYSTEM

FINAL PROJECT

Java 17 Backend Academy MTY

August 2024

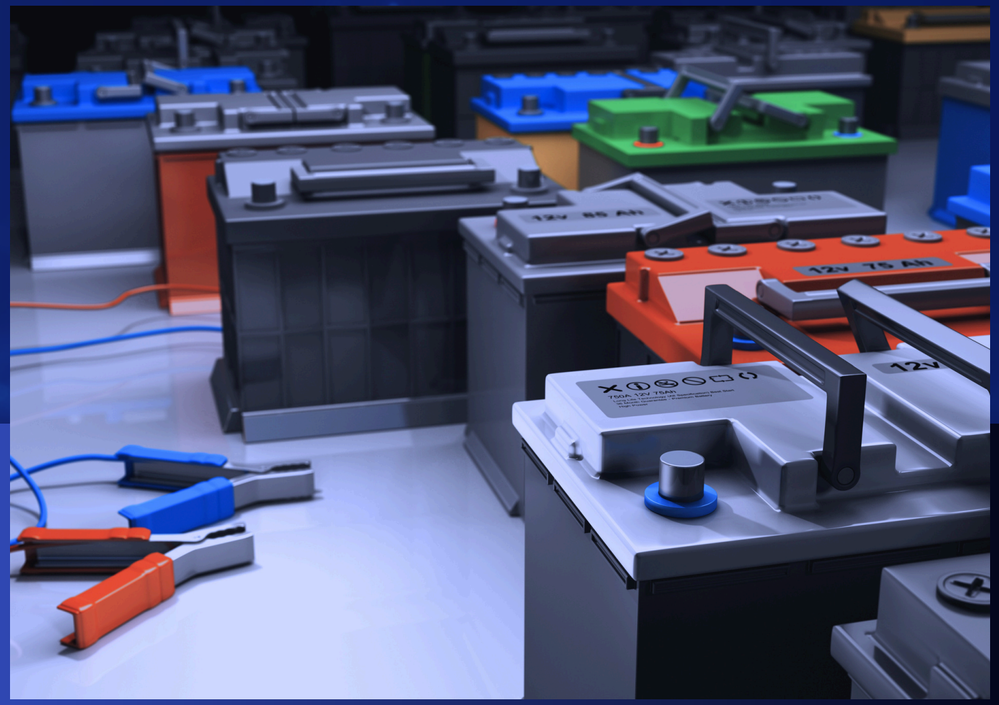
Frida Cano Falcón

https://github.com/FCANOF

XIDERAL

EXECUTIVE SUMMARY

XIDERAL



PROBLEM STATEMENT

- Growing need for efficient battery and sensor management in autonomous vehicles.
- Importance of minimizing downtime due to battery failures.

The management and tracking of battery charges in self-driving vehicles is crucial to ensure their efficiency, detect failures early, and optimize vehicle operations.

XIDERAL

SOLUTION OVERVIEW

Our system provides real-time monitoring, logging, and historical data tracking for battery charges and sensor data, ensuring operational reliability for autonomous vehicles.

OBJECTIVES

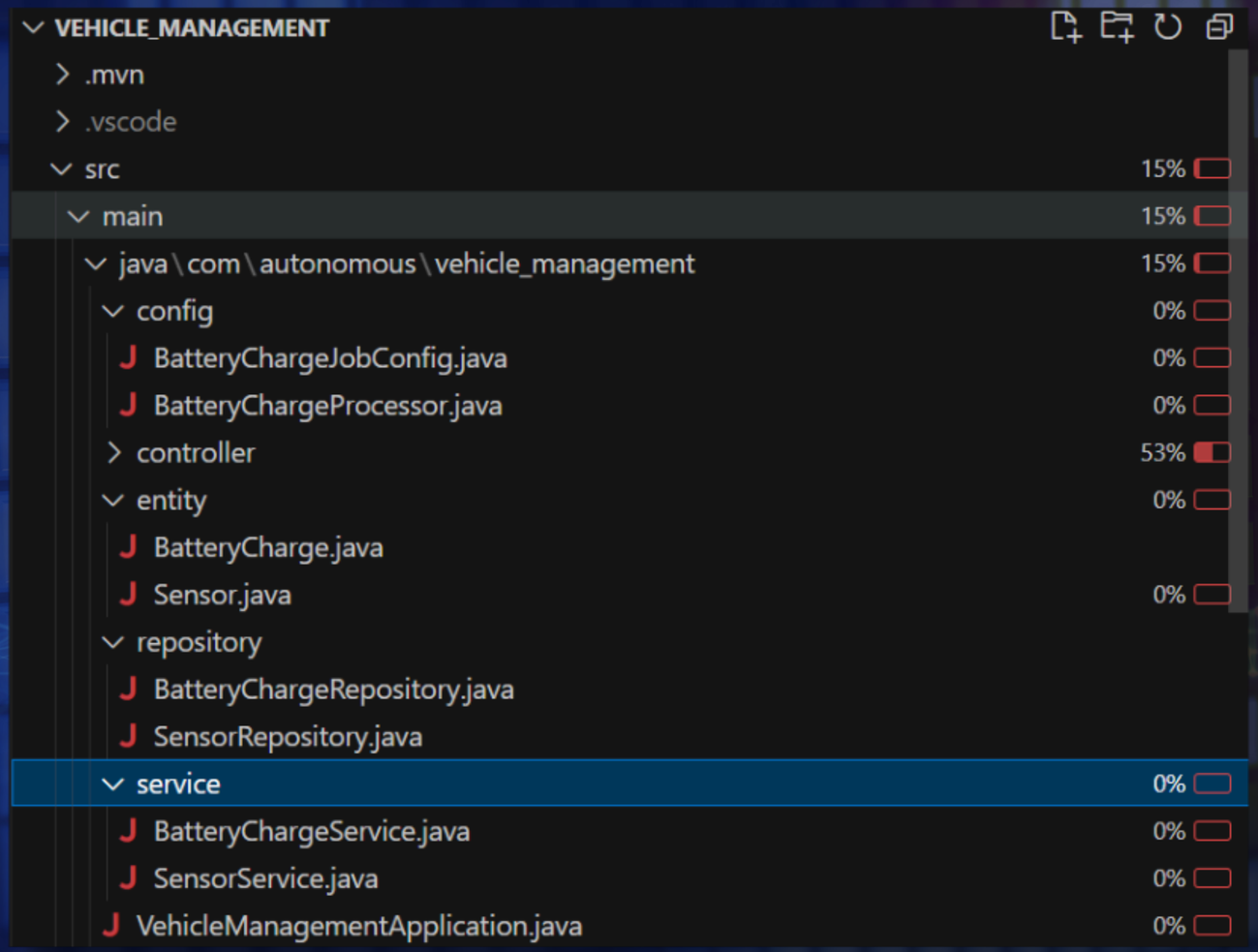
1. Record battery charge start and end times.
2. Track which chargers and subsystems were used.
3. Detect and log any issues during charging.
4. Ensure integration with existing sensor monitoring systems.

XIDERAL

SYSTEM OVERVIEW

XIDERAL

SYSTEM ARCHITECTURE



Battery Charge Monitoring Module

Describe how the system logs the start, end, and status of battery charges.

Sensor Monitoring Module

Explain how sensor data is tracked and stored.

XIDERAL

BATTERY CHARGE & SENSORS MODULES


```
"id": 5,
"batteryId": "A",
"chargeStartTime": "2024-09-09 10:00:00",
"chargeEndTime": "2024-09-09 11:00:00",
"chargingArea": "Software",
"chargerId": "A",
"chargeStatus": "successful",
"issueDetected": false,
"issueDescription": "NULL"
```

```
"sensorName": "multisense",
"sensorType": "camera",
"status": "calibrated"
```


ENTITIES

XIDERAL

BATTERY CHARGE & SENSORS MODULES

GET

- index
- getAll
- getById

POST

- create
- importCSV

PUT

- update

DELETE

- delete

REQUESTS

XIDERAL

TECHNOLOGIES USED

Java 17
for backend development.

Spring Data JPA
for database interactions.

Spring Batch
for processing large volumes of data, especially with CSV file imports.

Objects Oriented Programming

MySQL
for the database.

JUnit & Mockito
for unit testing and test coverage.

Lombok
for simplifying the development process by reducing boilerplate code.

XIDERAL

CHALLENGE & SOLUTIONS

XIDERAL

DATABASE

```java
@PostMapping("/importBatteriesChargeHistory")
public void importCsvToDBJob() {
    JobParameters jobParameters = new JobParametersBuilder()
            .addLong(key:"startAt", System.currentTimeMillis()).toJobParameters();
    try {
        jobLauncher.run(job, jobParameters);
    } catch (JobExecutionAlreadyRunningException | JobRestartException | JobInstanceAlreadyCompleteException | JobParametersInvalidException e) {
        e.printStackTrace();
    }
}
```

CHALLENGE

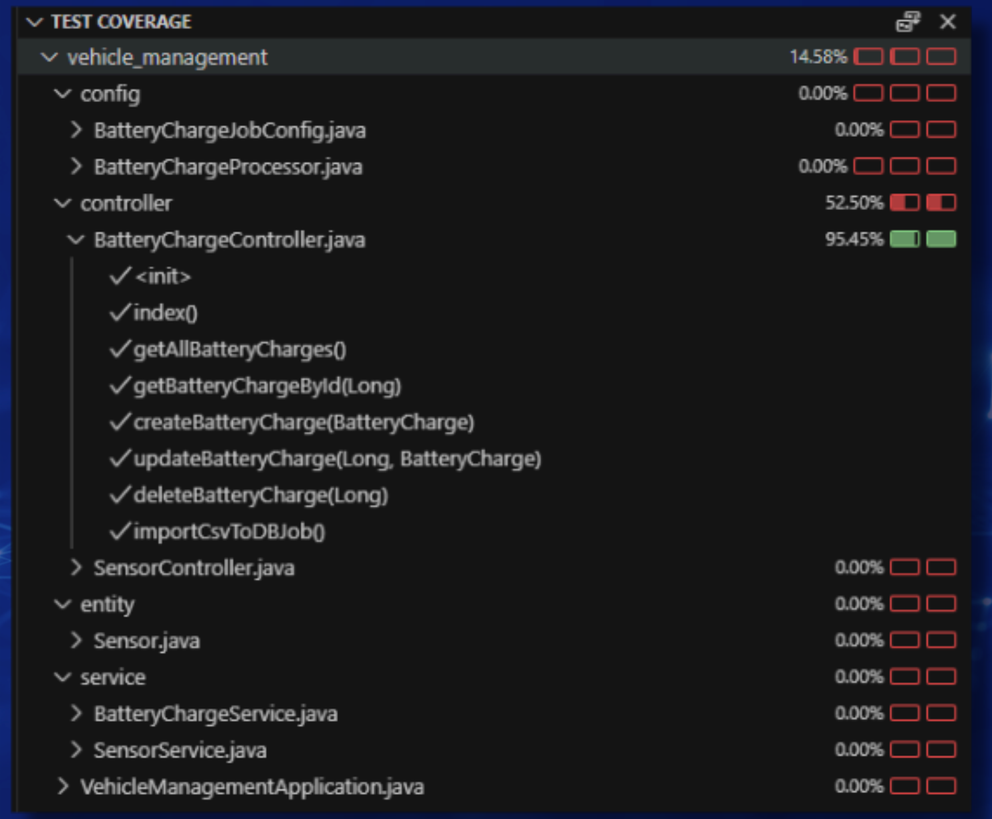
Managing large volumes of data, particularly sensor data and battery charge logs.

SPRING BATCH

we were able to efficiently import and export large datasets from and to CSV files. This ensures that the system can scale and handle large volumes of information.

XIDERAL

TESTING



CHALLENGE

Ensuring data accuracy and reliability, especially during CSV file imports.

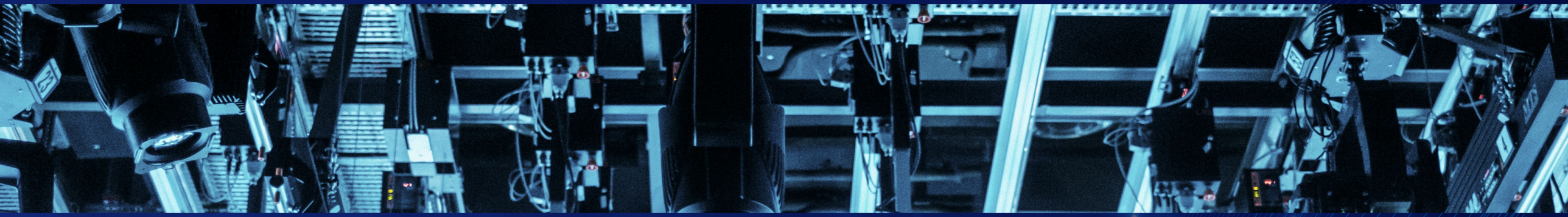
UNIT TESTING

We implemented extensive unit tests using JUnit and Mockito to ensure data integrity at every step.
Our test coverage reached **95.45%**, ensuring confidence in the system's robustness.

XIDERAL

CONCLUSION

XIDERAL



THANK YOU

https://github.com/FCANOF