ASE Documentation

I. React Frontend

a. *UI Development*: Applying React component with styling frameworks (CSS, SASS, LESS) or external UI Library (e.g., Material UI) to construct the UI of the web application. -> **Lucas, Martin**

b. *State management*: Persisting, extracting, and modifying the variable (data) flow across your React components. This expectation includes Scanning and Processing QR codes. Additionally, explain how to send requests to and process responses from the backend. -> **Lucas**

c. *Cross-Origin Resource Sharing (CORS) and Cross-Site Request Forgery (CSRF) Enforcement OR Authentication & Authorization*:

- CORS & CSRF: understand how CORS and CSRF protection work on both the frontend and backend. -> **Tiago**
- Authentication & Authorization: understand how to 1) send encrypted user credentials to the backend and 2) enforce Authorization in the frontend from the token generated by the backend. -> **Lucas, Diogo, Tiago**

II. Hardware Frontend

a. *Hardware Features*: Understand how to wire and set up the embedded system, read the sensors (light sensor and RFID readers), manipulate the actuator (LED), what data are stored inside the RFID token, and why**.** Finally, how to leverage the above elements to deliver the Box Unlock and Box Management Requirements. -> **Tiago**

b. *Backend Communication*: How to send CSRF-protected requests to and process responses from the backend. -> **Lucas, Tiago**

c. *Backend Processing*: How the backend processes the inputs from the Raspberry Pi to execute a particular function. -> **Diogo, Tiago**

III. Backend

a. *Service Architecture and Implementation:* Understand the architectural layers (Filter, Controller, Service, Repository, Model) and how to develop services by leveraging the layers. Furthermore, explain how to exchange data between microservices. -> **Diogo**

b. *Security Configuration*: How to enforce CORS and CSRF protection. -> **Diogo, Tiago**

c. *Authentication and Authorization*: How to authenticate users using the Spring Boot framework or third-party API (e.g., for OAuth2), encode authorization data into a token, and extract them to authorize users in the backend. -> **Diogo, Tiago**

IV. DevOps

a. *Microservice Communication Flow*: How the API Gateway and Service Discovery deliver (and oversee, if applicable) the communication between the frontend and backend services. -> **Diogo**

b. *Build Docker Image:* For each microservice (e.g., Frontend React, API Gateway, Auth Service), explain how to build its Docker image from the source code. -> **Diogo**

c. *CI/CD Pipeline*: Configure and operate Gitlab CI/CD stages to compile, build, and deploy the system in a VM. The system is publicly accessible via the Internet. -> **Diogo**