Project Log

April 03rd,

Problem Statement 7: Scenario: During emergency situations, such as natural disasters, accidents, or medical emergencies, effective coordination among first responders, victims, and support services is crucial. However, the current methods of communication and coordination often fall short, leading to delays in response, overlapping efforts, and confusion. For example, in the aftermath of a natural disaster, rescue teams might struggle to locate victims due to a lack of real-time information, while victims may have difficulty accessing the help they need. Similarly, medical emergencies may see delays in care because responders and hospitals are not fully synchronized. The chaos that often accompanies emergencies exacerbates these issues, making it clear that a more streamlined, technology-driven approach is needed.

|  |  |  |  |
| --- | --- | --- | --- |
| Group | Learner1 | Learner2 | Project Number |
| 7 | Aswin B Prasad | Soundarya S | 7 |

April 03rd,

Agenda

1. Discuss about the project and find the important points.

* Responders:
* Support Service:
* Disaster Api-Global Api

Observation done

**Abstract:**

The Emergency Management System is designed to optimize emergency response efficiency and streamline resource allocation during crisis situations. This platform harnesses cutting-edge web technologies, including **Angular** for the frontend, **ASP.NET Core** for the backend, and **SQL Server** for data management, all hosted on **Azure cloud infrastructure**.

Key features include a victim reporting interface by a responder, a rescuer dashboard for real-time data access, and a resource management module for emergency managers. Additionally, integrating a **public API**, such as **reliefapi**, will enhance location tracking and improve overall situational awareness.

By offering a robust and unified solution for disaster response coordination, this project aims to **save lives and mitigate the impact of emergencies** effectively.

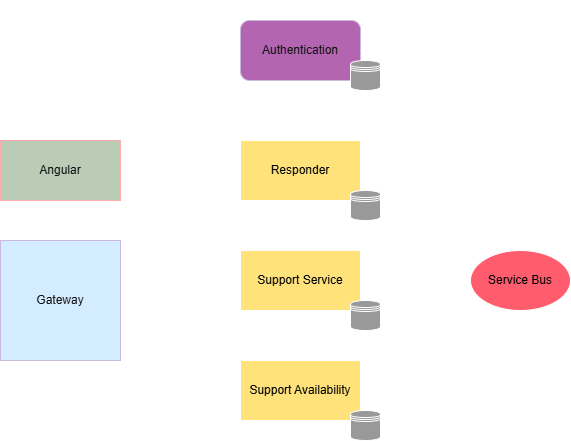
April 04th,

**Microservices:**

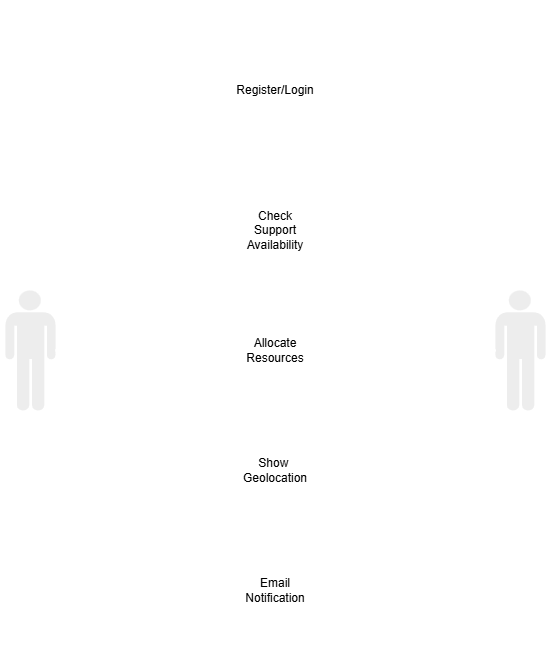
1. Responder Api -victim reporting
2. SupportService Api -allocating resource
3. SupportAvailability Api - resource availability
4. AuthApi
5. Gateway
6. Aggregator –connecting all the apis

April 05th,

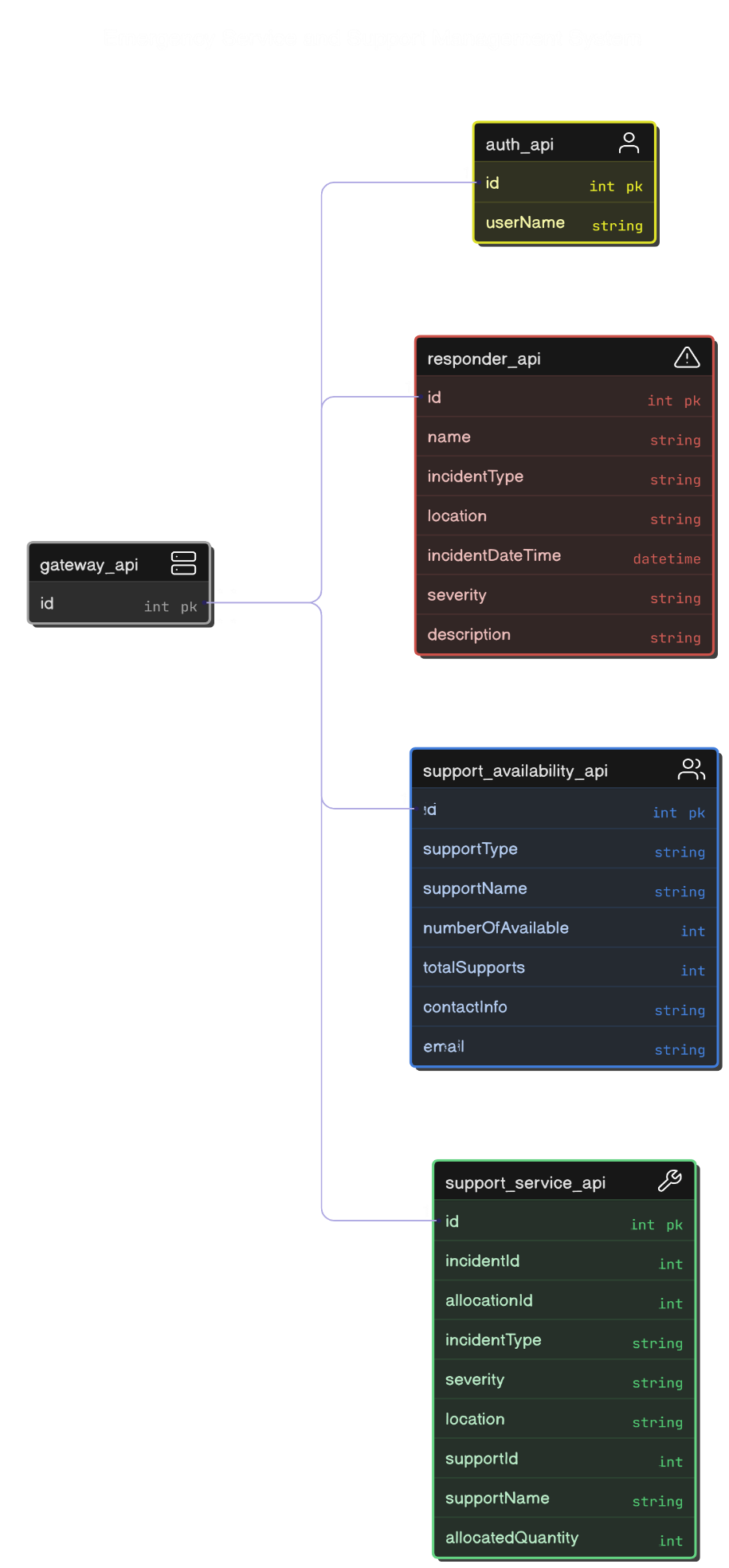
**Architecture Diagram:**

****

**Usecase Diagram:**

****

**Class Diagram:**

****