Business Requirement Document

1.Functional Requirements

The functional requirements define what the system should do. They include:

- **User Management:** The system must allow users to log in and logout.
- **Dashboard:** Managers should be able to search employees by their designation or skill or location and also mention number of employees required.
- **My Team:** The system must display all the team members added into the team and has pagination if there are many members.
- **Time Zone Overlap:** The system must display the overlap time for the selected employees.
- **Schedule Meeting:** The system must display a form and, on successful submission email has to be sent to all participants containing meeting links.
- **My Timesheet:** The system must display a form on successful submission the details has to be stored in database.

2.Non-Functional Requirements

Non-functional requirements define how the system performs its functions:

- **Performance:** The system should handle at least 1000 concurrent users with response times under 200ms.
- **Scalability:** The architecture should allow horizontal scaling to support increased load.
- **Security:** Data must be encrypted in transit and at rest. Use JWT for authentication and authorization.
- **Availability:** The system should have an uptime of 99.9%, with failover mechanisms in place.
- **Maintainability:** Code should follow SOLID principles, with high modularity and low coupling.

3.Architecture Overview

The architecture is based on a microservices pattern, where each service is responsible for a specific business capability. Each microservice runs independently, communicating with others via HTTP REST APIs. The key components include:

- **API Gateway:** Acts as a single-entry point for all client requests, routing them to the appropriate microservice.
- **Authentication Service:** Manages employee and time-zone services authentication using JWT.
- **Employee Service**: Handles employee-related operations such as registration, login, search employees, display team members.
- **Time Zone Service**: Handles time-related operations such as employee timesheet, team time-zone overlap, schedule meeting.

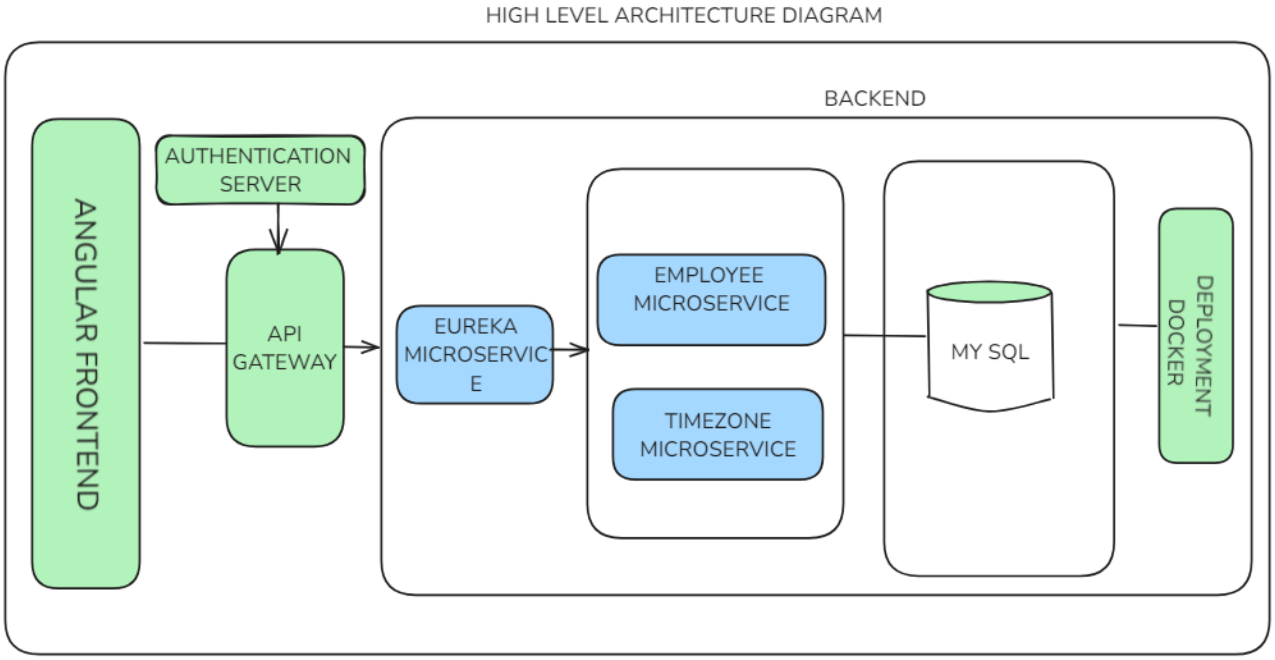


Fig-1: Architecture Diagram

4.Service EndPoints.

Each service exposes a set of RESTful APIs that can be consumed by other services or the frontend. The endpoints are as follows:

Employee Service API:

1. ‘POST /api/employees‘: Registers a new employee.
2. ‘POST /api/employees/id/skills‘: To add skills to employee.
3. ‘POST /api/employees/id/toogle-team‘: To toogle team membership.
4. ‘PUT /api/employees/id‘: To update the employee details.
5. ‘DELETE /api/employees/id ‘: To delete a employee.
6. ‘GET /api/employees/search ‘: To search employees.
7. ‘GET /api/employees/skills/{skill}‘: To find employee by skill.
8. ‘GET/api/employees/team‘: To get all the team members.

Time Zone Service API:

1. ‘POST /api/timezone‘: To save employee timezone.
2. ‘GET /api/timezone/employeeId‘:To get employee timezone.
3. ‘GET /api/ timezone /all‘: To get all employees time zone.
4. ‘GET /api/timezone/overlap‘:To get the overlapping working hours of employees.
5. ‘GET/api/timezone/suggest-meeting‘:To suggest the best time to schedule meeting.
6. ‘PUT /api/timezone/employeeId‘:Updates employee timezone.
7. ‘DELETE /api/timezone/employeeId‘: Deletes employee timezone.
8. ‘GET /api/timezone/validate-meeting-time‘: Validates meeting time.

5.Technologies Used

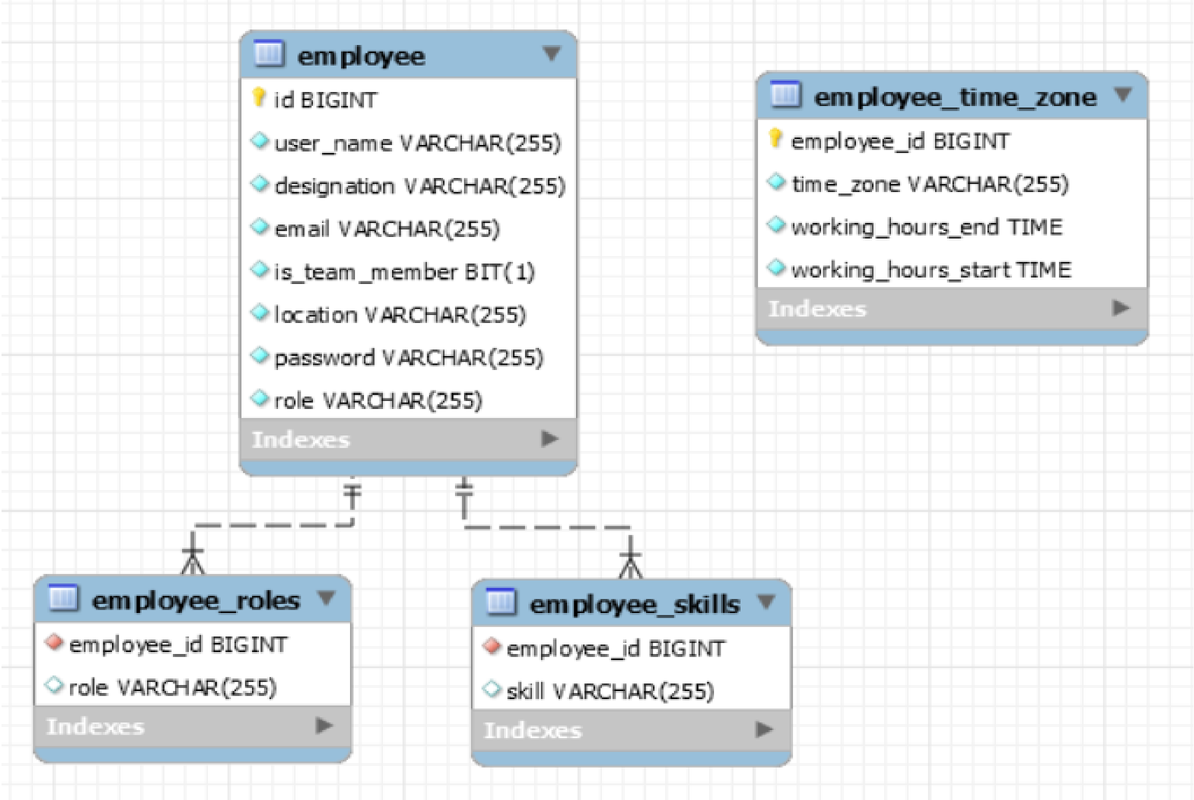
The project employs the following technologies:

- Programming Language: Java 21 for backend services.
- Frameworks:

    1. Spring Boot for building microservices.
    2. Spring Security for authentication and authorization.
    3. Spring Cloud Netflix Eureka for service discovery.
    4. Spring Cloud Gateway for API routing.

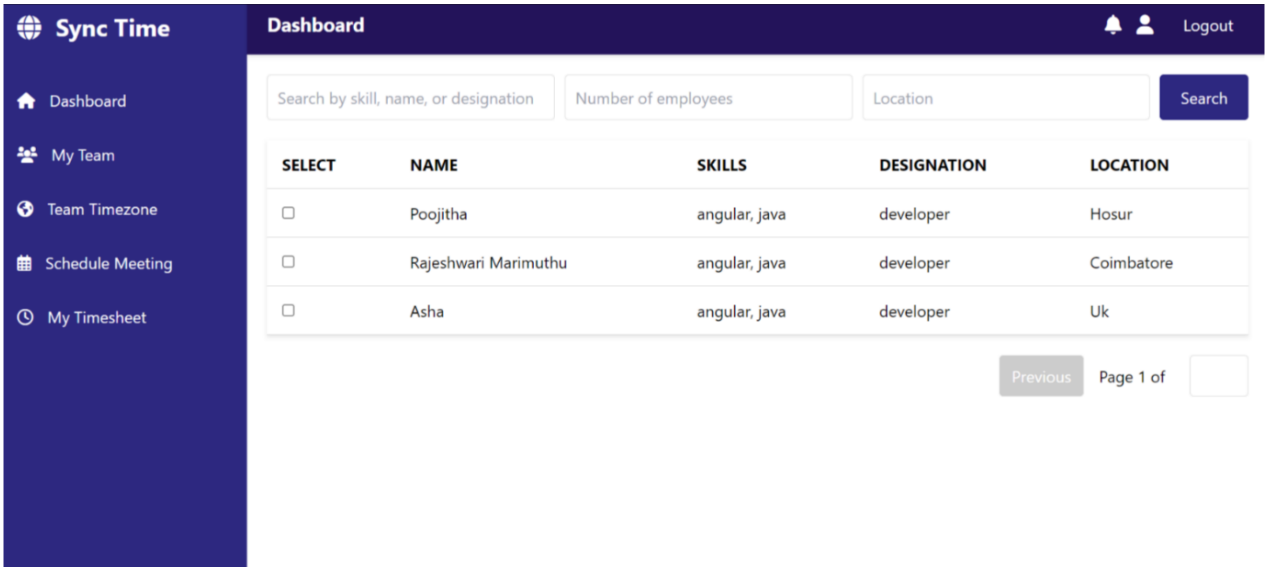
- Database: Mysql for storing persistent data.
- API Communication: RESTful APIs using Spring Web.
- Frontend: Angular for building the user interface.

6.Database Schema



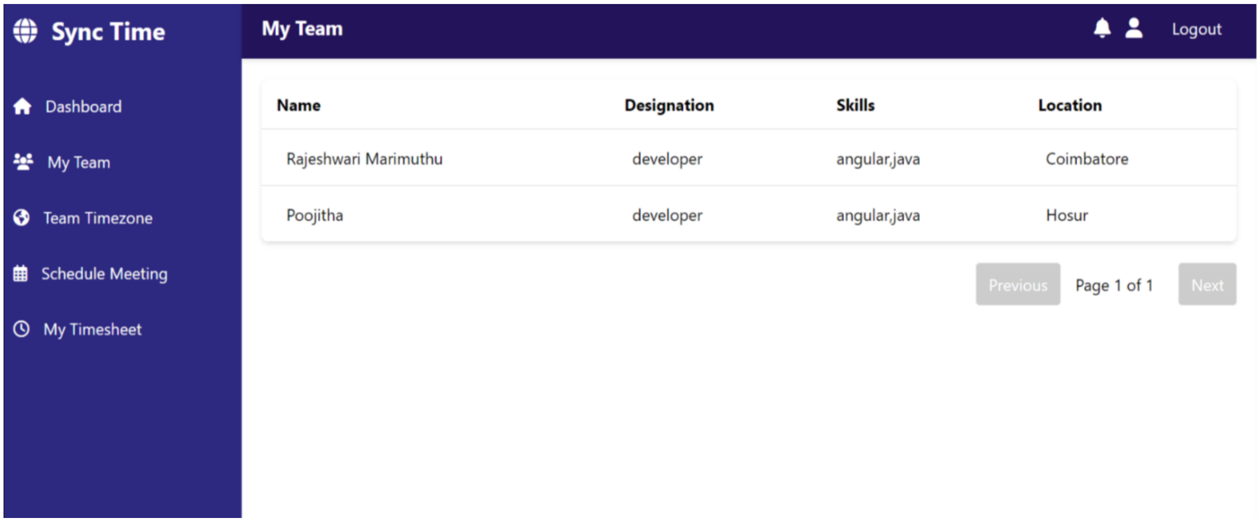
7.Wire Frame Diagrams:

7.1 Dashboard



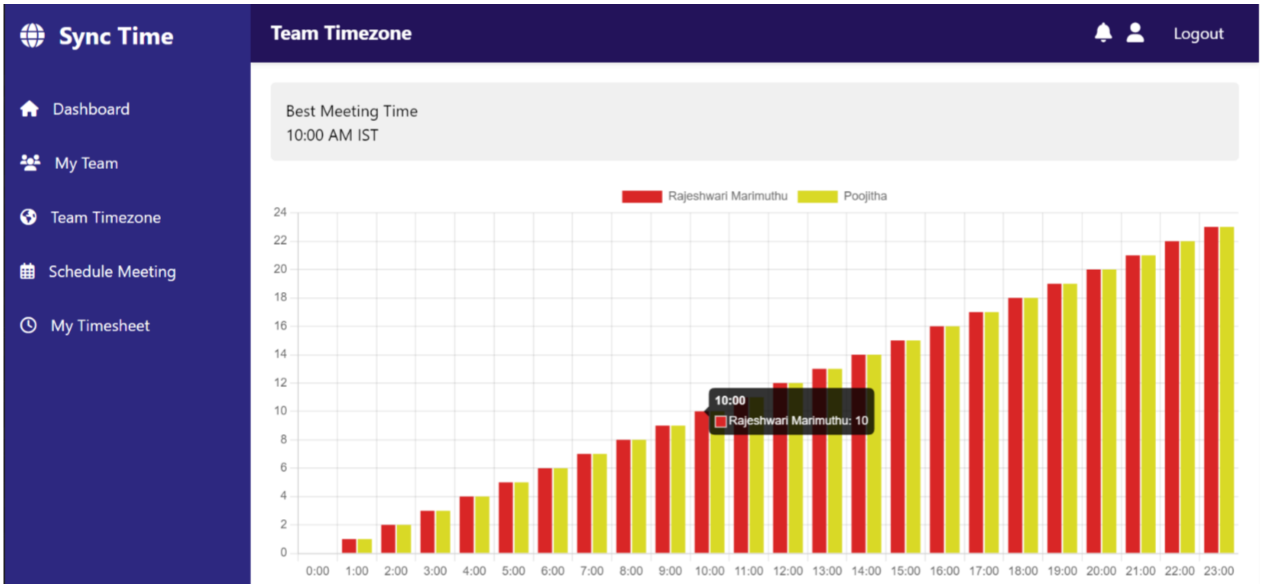
SELECT	NAME	SKILLS	DESIGNATION	LOCATION
□	Poojitha	angular, java	developer	Hosur
□	Rajeshwari Marimuthu	angular, java	developer	Coimbatore
□	Asha	angular, java	developer	Uk

7.2 My Team



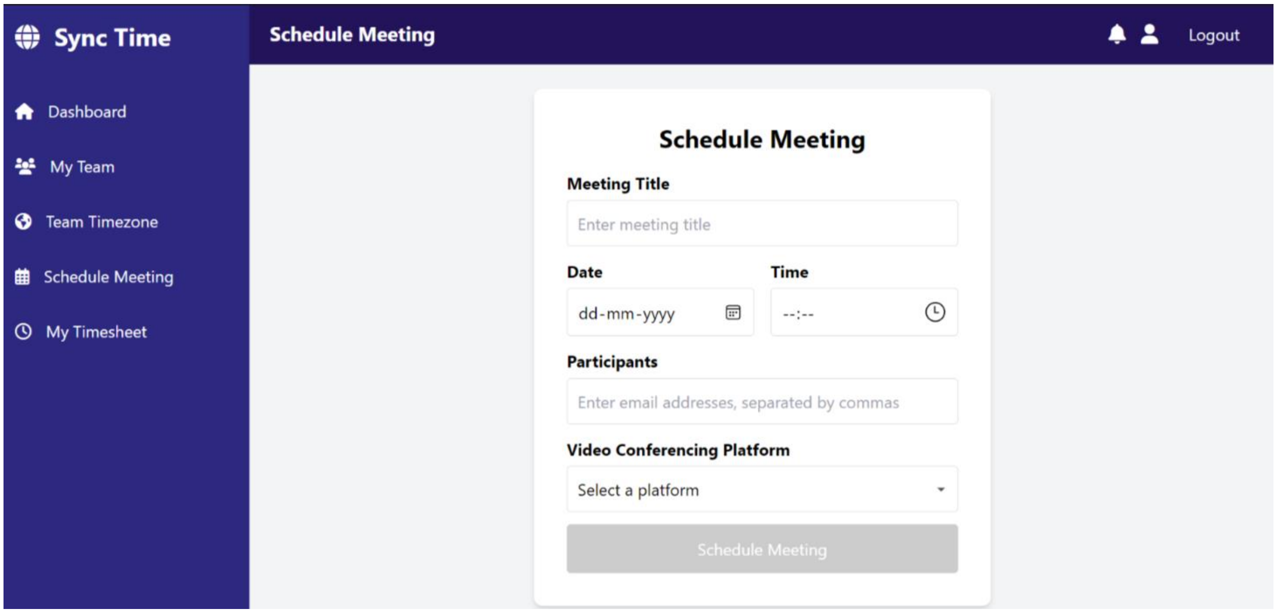
Name	Designation	Skills	Location
Rajeshwari Marimuthu	developer	angular,java	Coimbatore
Poojitha	developer	angular,java	Hosur

7.3 Team Timezone

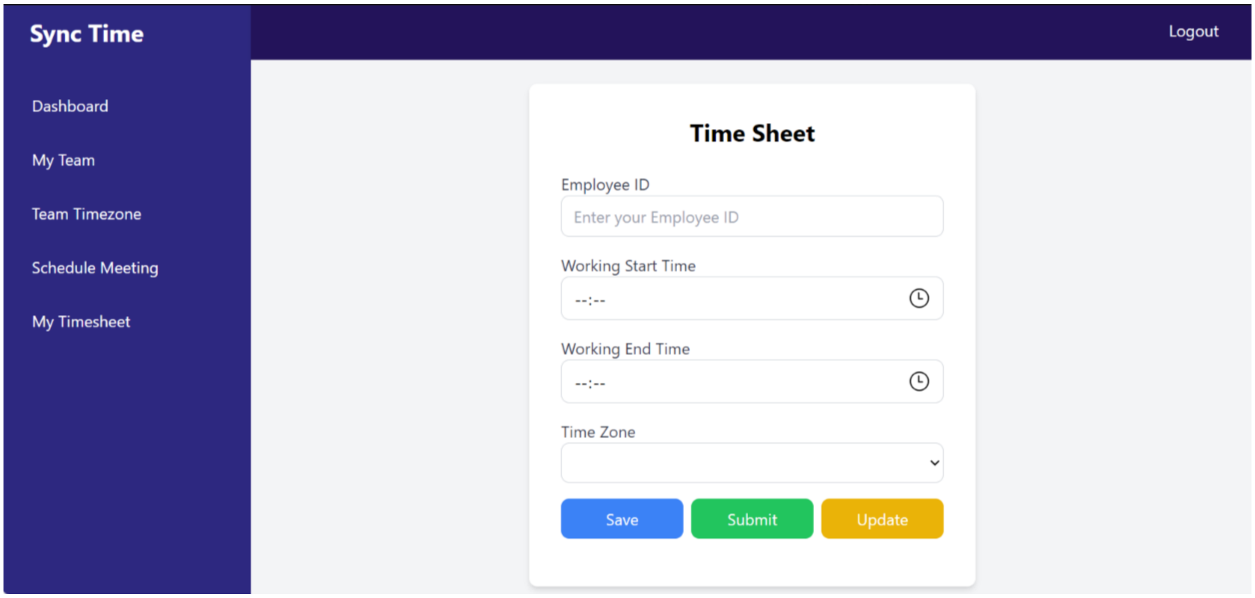


Best Meeting Time
10:00 AM IST

7.4 Schedule Meeting



7.5 My Timesheet


Sync Time
Logout
Dashboard
My Team
Team Timezone
Schedule Meeting
My Timesheet
Time Sheet
Employee ID
Enter your Employee ID
Working Start Time
--:--
Working End Time
--:--
Time Zone
Save
Submit
Update