Assignment1

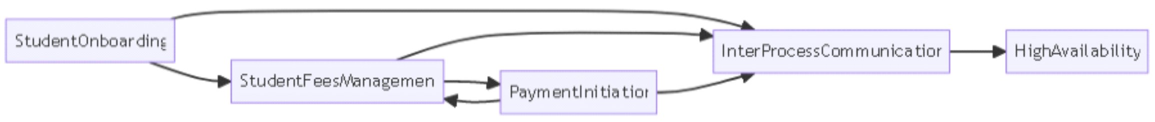
**Problem Statement**:
A popular education society "Edu-Excellence" is setting up a new school in the city. They want a "School Management Software" that should match the high standards of the school that they aspire to build. They have hired you to provide them with the software that should enable them to manage all aspects of running a school successfully, but they would want the application to grow gradually with the growth of the new school.

The software should follow the architectural design principles. It should be cohesive, loosely coupled, agile, flexible, and distributed. Provide a design for the above-mentioned.

Requirement (refer to the Problem Statement) with the following functions:

1. Manage students' onboarding
2. Manage students' fees
3. Fees payment to be initiated internally by student's management service
4. Mention the details of inter-process communication
5. Keep high availability into account in your design

**Microservice Architecture**: **School-Management**
based on the requirements and design principles mentioned in the problem statement below are the services:



### *1. Student Onboarding Microservice*
The Student Onboarding microservice will handle the registration of new students into the school system. This service will store student data and generate unique student IDs, which will be used by other microservices to identify students. This microservice will also handle validation of student data, and will communicate with other services to ensure that all necessary information is collected.

### *2. Student Fees Management Microservice*
The Student Fees Management microservice will manage the fees of students, including creating fee structures and managing fee payments. This microservice will interact with the Student Onboarding microservice to ensure that student fees are properly assigned to the correct student accounts.

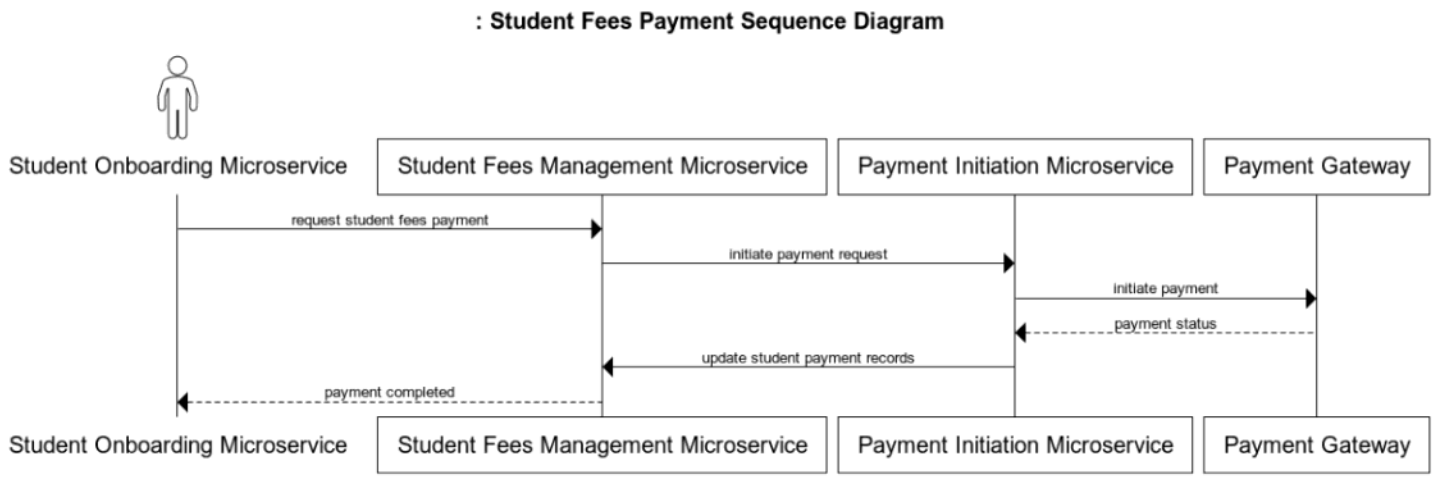
### *3. Payment Initiation Microservice*
The Payment Initiation microservice will initiate payments for student fees. This microservice will communicate with the Student Fees Management microservice to obtain the necessary payment information, and will then initiate payment processing through an external payment gateway. Once payment is complete, this microservice will communicate with the Student Fees Management microservice to update student payment records.

### *4. Inter-Process Communication*
The microservices will communicate with each other using a lightweight messaging

protocol, such as HTTP or AMQP. Each microservice will have a well-defined API, and will use a service registry to locate other microservices. The service registry will maintain a list of all available microservices, along with their current status and location.

## 5. *High Availability*

To ensure high availability, each microservice will be deployed on multiple instances, with load balancing and auto-scaling mechanisms in place. Each instance will be deployed in a separate availability zone or data center, to ensure that the system remains available in the event of a single-point-of-failure. Additionally, the microservices will be designed to be resilient to failure, with retry and circuit-breaking mechanisms in place to handle temporary network outages or service disruptions.


: Student Fees Payment Sequence Diagram
Student Onboarding Microservice
Student Fees Management Microservice
Payment Initiation Microservice
Payment Gateway
request student fees payment
initiate payment request
initiate payment
payment status
update student payment records
payment completed
Student Onboarding Microservice
Student Fees Management Microservice
Payment Initiation Microservice
Payment Gateway