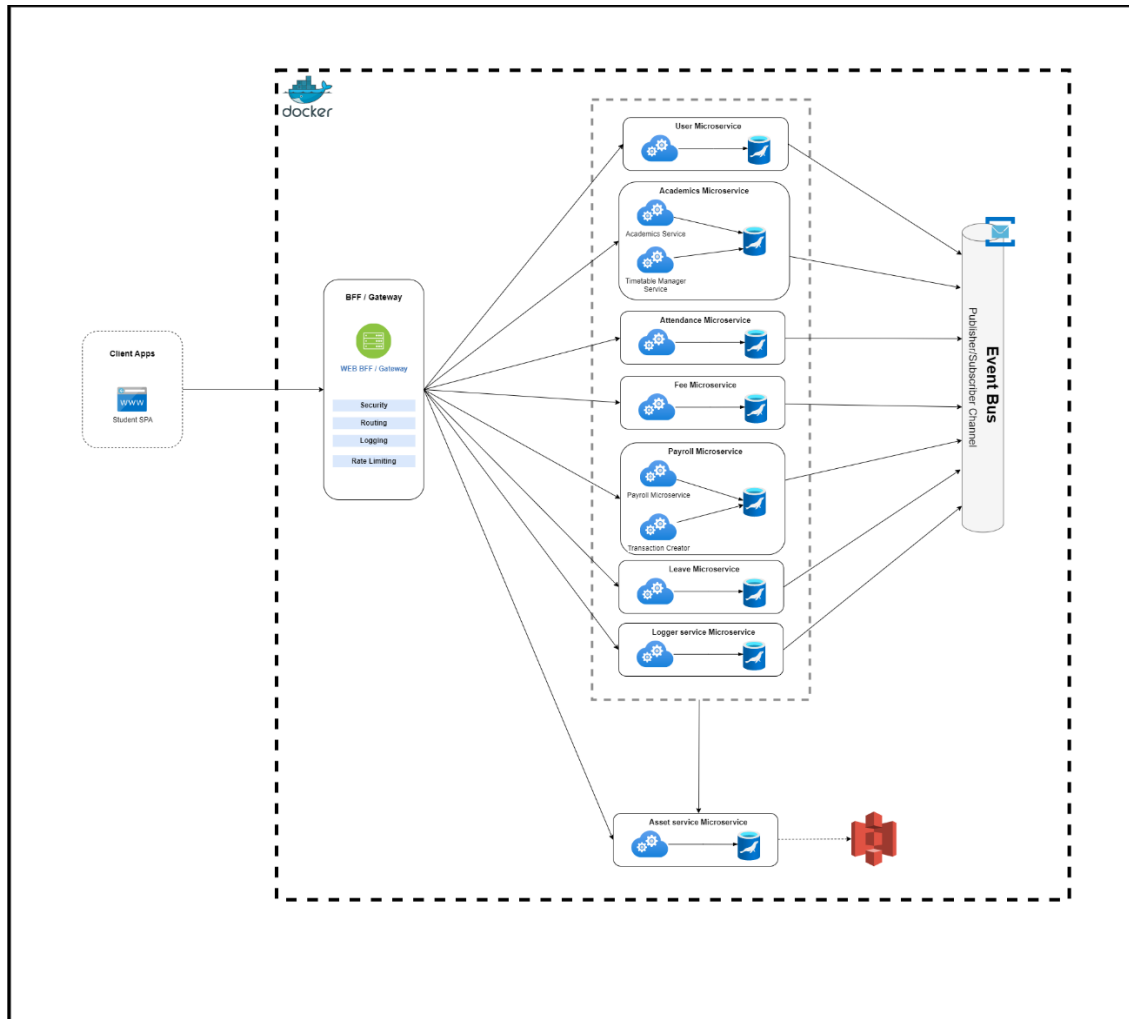


School Management System Design

A. Architecture Diagram



B. Microservices Definitions

- User/ Authentication Service

Entity	Attributes	Domain Events	Domain Services
User	id : string fName : string lName : string emailId : string, contactNum : string, password : string, userRole : string	<ul style="list-style-type: none">• When a user is created or updated the user entity sends the UserCreated / UserUpdated event which will have the newly created / updated user information.	
Student	id : string, userId : string (REF), regId : string, advisor : User[1-1] (id, fName, lName, contactNum, userRole), parent : User[1-1] (id, fName, lName, contactNum, userRole),		
Faculty	id : string, userId : string (REF), regId : string, designation : string		
Parent	id : string, userId : string (REF), occupation : string,		

Endpoints

- validate-user - POST (Login)
- refresh-token - POST (access_token, refresh_token, expiry)
- student - POST, GET, PATCH, DELETE
- faculty - POST, GET, PATCH, DELETE
- parent - POST, GET, PATCH, DELETE

- Academics service

Entity	Attributes	Domain Events	Domain Services
Syllabus	id : string subjectName : string subjectCode : string syllabus : string, totalHours : string, expectedOutcome : string, references : string	<ul style="list-style-type: none"> • When a class is created or updated the user entity sends the ClassCreated / ClassUpdated event which will have the newly created / updated class information. 	<ul style="list-style-type: none"> • A timetableManager service is introduced in the boundary context to update a days timetable based on the availability of a faculty • An email is send to the students when they are assigned to a new class
Class	id : string, className : string, classInCharge : User[1-1] (id, fName, lName, contactNum, userRole), students : User[1-1] (id, fName, lName, contactNum, userRole), classFees : number		
Timetable	id : string, classId : string (REF), syllabusId : string, hour : string, day : datetime, taughtBy : User[1-1] (id, fName, lName, contactNum, userRole),		
User	id : string, userId : string, fName : string, contactNum : string, userRole : strin		

Endpoints

- class - POST, GET, PATCH, DELETE
- syllabus - POST, GET, PATCH, DELETE
- timetable - POST, GET, PATCH, DELETE

- Attendance service

Entity	Attributes	Domain Events	Domain Services
Attendance	id : string, isPresent : Boolean, user : User[1-1] (id, fName, lName, contactNum, userRole), type : Ref, schedule : Timetable[1-1] (id, hour, day) , remark : string,		<ul style="list-style-type: none"> • If a student is marked absent in an hour an email/sms will be triggered to the parent of the student.
Attendance Type	id : string, type : string		
User	id : string, userId : string, fName : string, contactNum : string, userRole : string		
Timetable	id : string, scheduleId : string, hour : string, day : datetime		

Endpoints

- attendance - POST, GET, PATCH, DELETE, BATCH
- attendance_type - POST, GET, PATCH, DELETE

- Leave service

Entity	Attributes	Domain Events	Domain Services
Leaves Applied	id : string, appliedBy : User[1-1] (id, fName, lName, contactNum, userRole), leaveType : Ref, leaveFrom : date, leaveTo : date, isHalfDay : boolean, status : string, purpose : string, description : string, approvedBy : User[1-1] (id, fName, lName, contactNum, userRole),		<ul style="list-style-type: none"> • Notify attendance service about the students who are on leave on a particular day.
Leaves Allocated	id : string, user : User[1-1] (id, fName, lName, contactNum, userRole), leaveType : Ref, totalNumOfLeave : number, numOfLeaveAvailable : number,		
LeaveType	id : string, typeName : string		
User	id : string, userId : string, fName : string, contactNum : string, userRole : string		

Endpoints

- applied_leaves - POST, GET, PATCH, DELETE
- leaves_allocated - POST, GET, PATCH, DELETE
- apply_leave - POST, GET, PATCH, DELETE

Payroll service

Entity	id: string,	Domain Events	Domain Services
Payroll	id: string, amount: number, accountNumber: number, IFSCNumber: String, user : User[1-1] (id, fName, lName, contactNum, userRole), upId : string		<ul style="list-style-type: none">• SALARY STATUS manager• Salary processor service to transfer monthly salary to faculty
Transaction	id: string, payedAmount : number, createdDate : dateTime txnStatus : string, paymentMode : string, txnId : string, payroll : Payroll[1-1] (id, fName, lName, contactNum, userRole),		
User	id : string, userId: string, fName: string, contactNum : string, userRole : string		
Leave	id: string, leave : Leave[1-1] (id, fName, lName, contactNum, userRole),		

Endpoints

- payroll/:userId - POST, GET, PATCH, DELETE
- transaction - POST, GET, PATCH, DELETE

- Fee service

Entity	id: string,	Domain Events	Domain Services
Charge	id : string student : User[1-1] (id, fName, lName, contactNum, userRole), chargeType : Ref, Amount : number, chargeName : string, lastDate : date		• FEE STATUS / REMINDER • Create recurring charges per month for students
Class	id : string, className : string, classFees : number students : User[1-]* (id, fName, lName, contactNum, userRole),		
User	id : string, userId : string, fName : string, contactNum : string, userRole : string		
ChargeType	id : string, typeName : string		

Endpoints

- charge - POST, GET, PATCH, DELETE
- charge-type - POST, GET, PATCH, DELETE