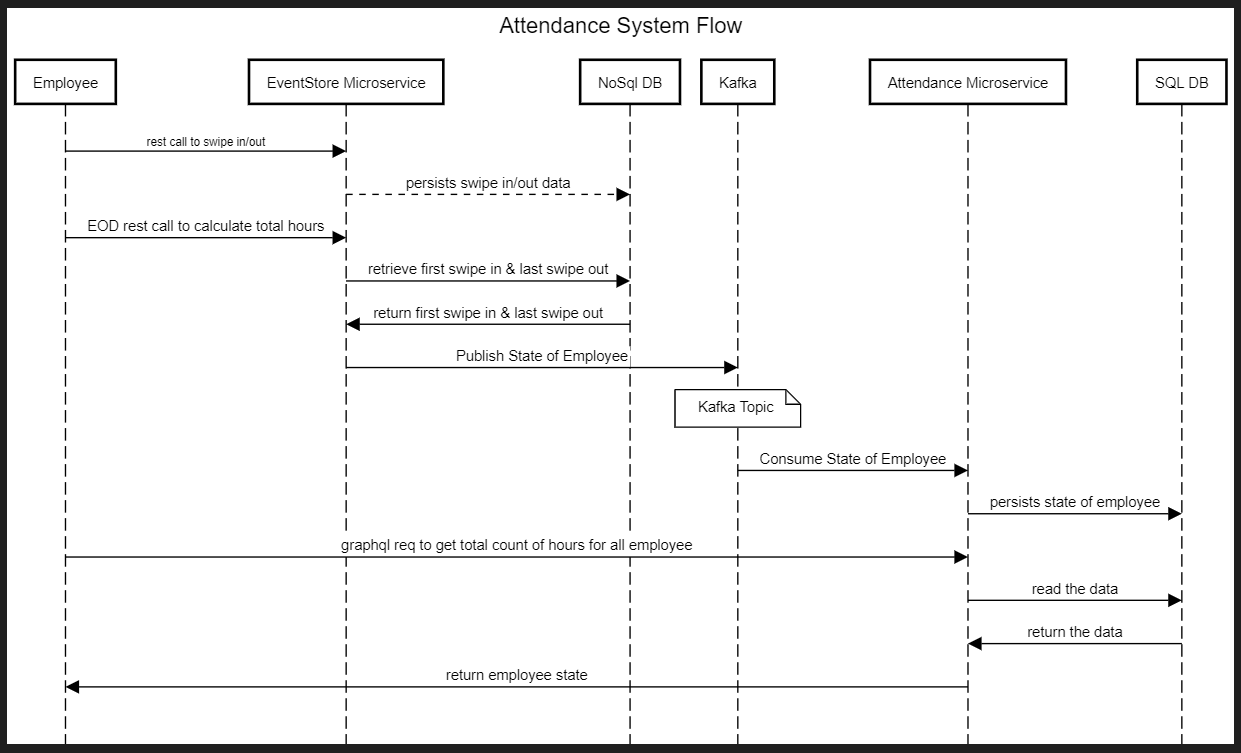
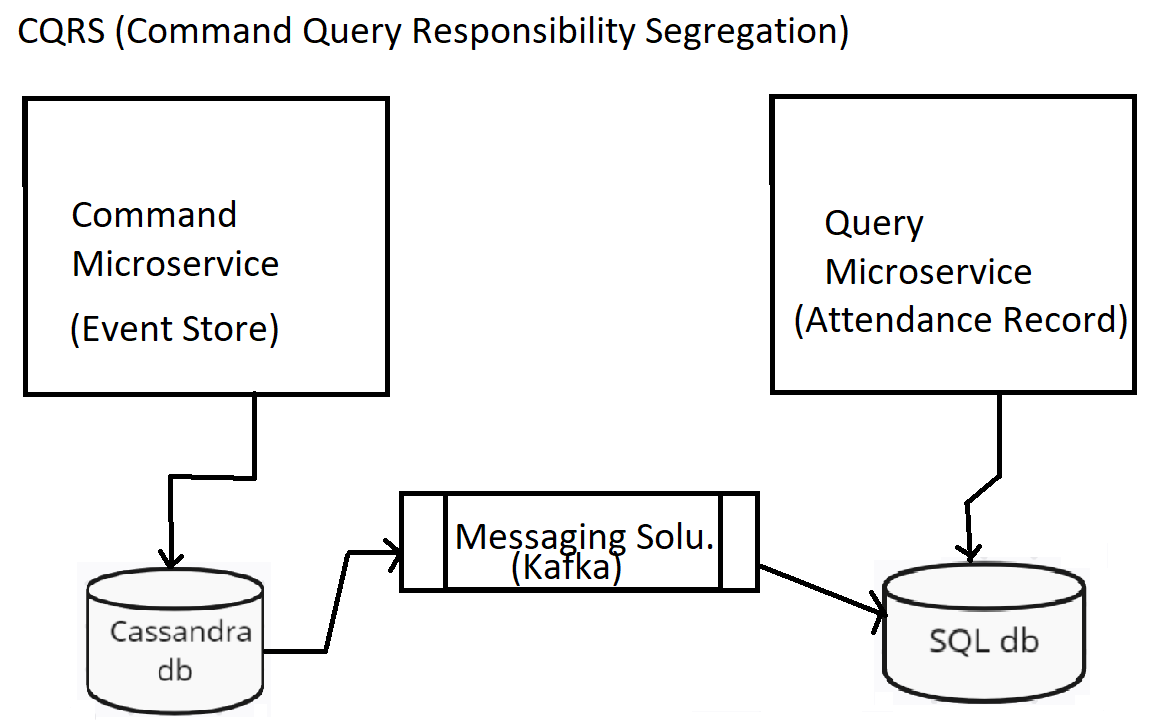
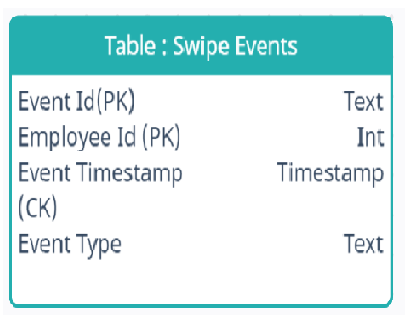
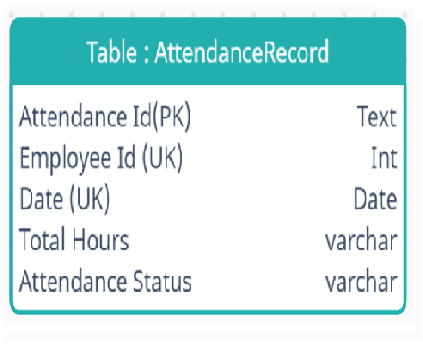
## Milestone 1: Architecture Artifacts & Design

1. **Solution Architecture:** 
   1. **Logical Architecture**:
      1. **Event Store Microservice**:
         1. Responsible for persisting all swipe in and out in the event store.
         2. Exposes REST Endpoints to receive swipe in/out requests from employees.
         3. Upon receiving a swipe event, using cassandara db (NoSQL) to store swipe events.
         4. Exposes REST Endpoint to calculate total hours of ach employee at end of day, applies attendance formula to determine the attendance status. And publish employee state to Kafka.
      2. **Attendance Report Microservice**:
         1. Listens to the kafka topic for employee state.
         2. Store the employee data to read db(MYSQL).
         3. Exposes Graphql Endpoint to retrieve attendance information of employee.
      3. **Apache Kafka**:
         1. Acts as a message broker for streaming events b/w servies.
         2. Apache Kafka will serve as the event bus for asynchronous communication b/w microservices.
         3. EventStore Microservice will publish events to Kafka topics, and other microservices will subscribe to these topics to consume events.
      4. **Database**:
         1. Separate databases for read and write operations to optimize for read-heavy operations.
         2. A relational database (RDBMS) will be used for the read model, optimized for querying and reporting.
         3. The write model will be stored in the event store, which can be a NoSQL database optimized for event storage and retrieval.
   2. **Physical Architecture** 
      1. **Event Store:** Cassandra for storing swipe events.
      2. **Microservice:** Spring Boot for building REST and DGS framework for building Graphql endpoint**.**
      3. **Kafka:** Apache kafka for event streaming.
      4. **Read Database server:** MySQL for storing attendance data.
      5. **Kubernetes:** for container orchestration.
   3. **Business Context Diagram :** boundaries and interaction b/w component**.** 
      1. **Event Management Context :** responsible for managing swipe events and their persistent into the event store, calculates total hours and publish onto kafka.
      2. **Attendance Management Context :** manages employee attendance based on the events consumed from kafka and stores it in the attendance db.
   4. **EDA strategy and implementation with MS architecture**
      1. **Swipe Events Handling:** 
         1. When employee swipes in/out , the swipe Microservices receives the REST call and store the swipe data into Cassandra db.
         2. At the EOD, Periodically or upon a receiving a trigger, it calculates the total hours for each employee using the given formula(first swipe in and last swipe out).
         3. Once calculated, it publishes the attendance data to a kafka topic.
      2. **Attendance Persistence :** 
         1. The Attendance Microservice listens to the kafka topic.
         2. And then persist the data into the read store.
         3. Expose Graphql endpoint to retrieve the attendance record.
   5. **NFRS managed:**
      1. **Scalability, Reliability, performance, Security, monitoring & logging.**
2. **Sequence Diagram** : 
3. **Design** :
   1. DDD & Bounded Contextv – given in PPT. all diagrams have been created in power point. Under project directory check doc folder.
   2. **Ready Heavy CQRS**

Separating the command (writing swipe events, calculating attendance) and query (attendance reporting) responsibilities ensures scalability and independence. This patterns is efficient for read and write.

* 1. **DB Selection (NOSQL + SQL)** ,
     1. Set up a RDBMS db (MySQL) for the read store. Define tables for persisting attendance data.
     2. Set up Cassandra db (NoSQL) for the write. Define tables for persisting swipe events.
  2. **ER Diagram**





* **Employee**: Represents the employee information. Each employee has a unique employee\_id.
* **SwipeEvent**: Represents the swipe-in and swipe-out events. Each event is associated with a specific employee (employee\_id). The event\_timestamp denotes when the event occurred, and the event\_type indicates whether it was a swipe-in or swipe-out event.
* **Attendance Status**: represents the status of employee whether employee was absent, or present (half day/full day).
* **Hours:** represent total hours employees was in the building.

Relationships:

* An Employee can have multiple SwipeEvents (one-to-many relationship).
* Each SwipeEvent is associated with one Employee (many-to-one relationship).

## Milestone 2: Set up MVP

1. CONFIGURATION
   1. Configure Monitoring Dashboards - 7
   2. Use of Build Tools - 7
   3. Deployment of services on Cloud managed AKS using CI/CD Pipelines (Desired) – Refer to the Bonus Marks section below
   4. Kafka Set up

For this milestone, check microservices.

## Milestone 3: Build Phase

1. Microservices Ecosystem with Springboot
   1. Features Implemented
   2. Springboot annotations
   3. Transactions
   4. configurations
2. API Security
   1. O-Auth2.0
   2. API Gateway
3. Logging & Tooling
   1. Observability: Grafana, cloud monitoring
4. Exception handling

All above 4 points has been done in microservices.

## Milestone 4: Testing Phase

Junit has been written for each microservices.