Project: The Implementation of an Application to Support covid-19 Management

Course: CS505- Intermediate Topics in Database Systems (Spring-2023)

Team Name: Team Alpha Bravo
Team Members: Srijata Maji (12598195) & Mir Al-Masud (12602303)
Date: 04/19/2023

INTRODUCTION

Project Covid19 has three sets of data, containing information on events related to testing, hospitals, and vaccinations. These events could arrive one at a time or in batches. Additionally, we received a CSV list of hospitals with details such as their zip code and number of beds.

DESIGN APPROACH

Technology

Language	Java
Database	OrientDB, Apache Derby
Queuing Service	RabbitMQ
Libraries	Siddhi CEP
Build Tool	Maven

Database

	Purpose
Graph Database: Orient DB	Storing Patient and Event relations
Relational/Embedded Database: Apache Derby	Storing Hospital and Vaccine data
Queue: RabbitMQ	Sending Patient, Hospital and Vaccination feeds

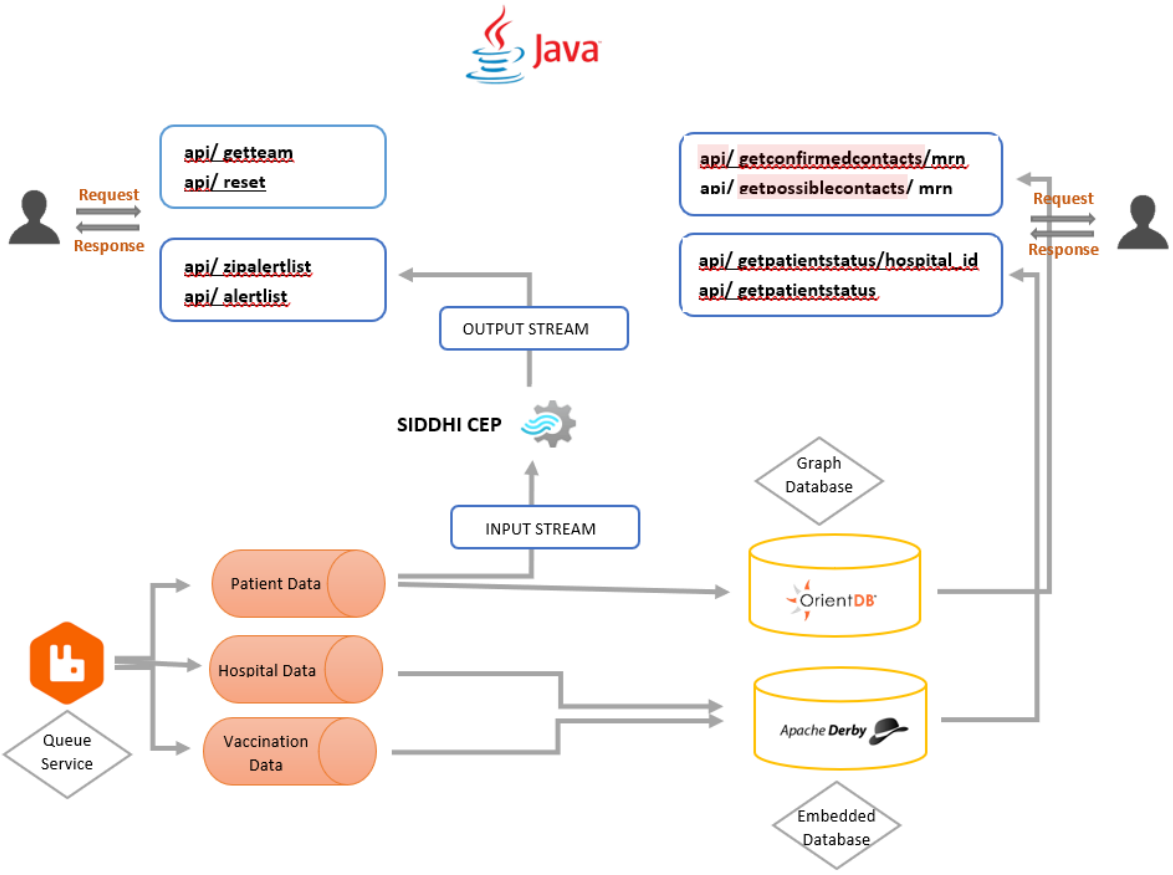
Purpose

- Siddhi CEP
  Siddhi Streaming SQL is designed to process event streams in a streaming manner, detect complex event occurrences, and notify them in real-time.

- Graph Database – Orient DB
  Orient DB is a document database overlain by a graph database. The document database provides the advantages of one-direction link relationships, key/value pairs, and object-oriented models. The graph database adds vertex and bi-directional edge relationships and speed benefits. It is highly flexible and scalable.

- Relational/Embedded Database – Apache Derby
  Derby database has a small footprint, i.e., it occupies less space and it is easy to use and deploy it. Embed with Java Application – Derby provides an embedded database engine which can be embedded in to Java applications and it will be run in the same JVM as the application.

ARCHITECTURE



DEVELOPMENT PROCESS

API Management Functions

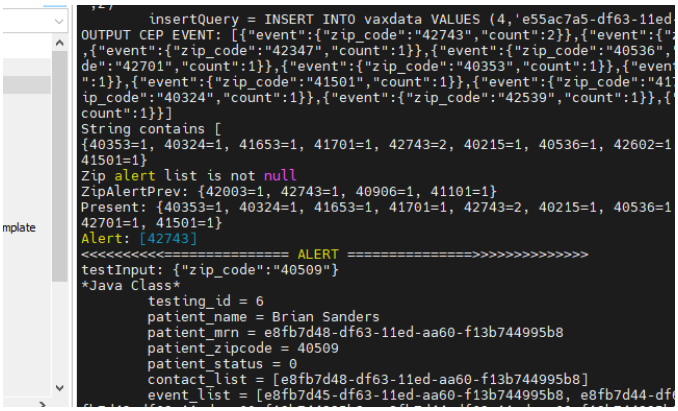
1. MF 1: API for name of team and list of student ids that are part of the team
   Team and member's names are static data. This data was hardcoded in the API.

2. MF 2: This API is used for resetting all data, one **must reset all data, but schemas can remain** resetting data means dropping the database or ending the connection to the database.

Real-time Reporting Functions - We Used CEP feeds to process real time data.

3. RTR 1: API alert on zip code that is in alert state based on growth of cases.

4. RTR 2: API alert on statewide when at least five zipcodes are in alert state (based on RT1)



Contact Tracing Functions

We used Orient DB to store streamed data and contact tracing

5. CT 1: API to determine groups of people who have reported that they came in contact with one another.

6. CT 2: API to determine return number of people who might have have been in contact if they attended the same event

Logic and Operational Functions

We used Apache Derby Embedded database to facilitate Operational functions api.

7. OF 1: API to provide summarized patient status per hospital

8. OF 2: API to provide summarized patient status per stat (all hospitals)

CHALLENGES

1. Initially we intended to work with python. However proper python libraries were not available in VM. Then we switched to Java template provided by Dr. Cody Bumgardner.
2. Orient DB sessions were terminating now and then. We struggled through it.

TESTING PLAN

1. While development we made sure of handing exceptions and null values.
2. Closing result set and database connections.

CONCLUSION AND LEARNING OUTCOME

Working on this project helped us develop a deeper understanding of the different types of database engines available and how they can be used to solve different types of problems. It helped us gain a better understanding of the design and thought process that goes into building a complex system like a database.

By working with multiple types of database engines, we had to consider a variety of factors, such as performance, scalability, reliability, and security, among others. We had to make decisions about which database engine to use

in different situations, based on factors like the size of the data set, the complexity of the queries, and the specific needs of the application.

CONTRIBUTIONS

Srijata Maji

1. Design the Graph Database
2. Pre-processing data from RabbitMQ Streams
3. Data pre-processing for Graph Database
4. Report document

Mir Al-Masud

1. Siddhi CEP
2. Design the Embedded Database
3. Data pre-processing for Apache Derby
4. Report document