***Spring Boot Project (Inbox App)***

It’s an application which is messaging app.

We are going to use technologies like (tech stack)

* Spring boot for back-end
* Spring data Cassandra for data layer
* Apache Cassandra for database which is a NoSQL database.
* Spring Security for security (OAuth login with GitHub)
* Thyme leaf for webpage rendering

Project requirements

* Java installation
* IDE
* Hosted Cassandra free account

**Que. What is Apache Cassandra?**

Apache Cassandra is an open source non-relational, or NoSQL, a database that enables continuous availability, tremendous scale, and data distribution across multiple data centres and cloud availability zones.

**Que. What is hosted database? What is Astra DB?**

In database hosting, a third party offers the hardware and infrastructure to run a database of the client’s choosing, often in the cloud. They also configure the environment for secure access, ensure resources are available to scale the database as needed, and offer managed services based on requirements.

DataStax Astra DB is a cloud-native, scalable Database-as-a-Service built on Apache Cassandra.

**Functional requirements**

To approach a design, we should first get started with listing down the requirements-

1. Compose message
2. Send message
3. View my messages
4. Folder/ label organization user folder (sent/ draft)
5. Reply, reply all
6. View single message

**Non-Functional requirements**

1. High availability
2. High scalability
3. Authentication and authorization
4. UI

* Login page (GitHub OAuth login)

Login

* Home page

* Message view

Database

LB

**Apache Cassandra**

Spring Boot

Thyme leaf

Spring Security

GitHub

**Conceptual Data modelling**

In case of relational database, referred as Entity relationship model. It allows us to identify the main actors, and the relationship b/w them.

* User and message have many-to-many relationship, as one user can of m number of messages and a message can be sent to multiple users.
* A label or folder is owned to by a single user. User can hold many labels. So, it is one-to-many relationship.
* A message can be associated with multiple label, but for now the implementation is one to many.

**ER DIAGRAM**

sees

**User**

**n** **m**

**Messages**

**Label/folder**

owns

**1** n

Associated with

m

**n**

**DATA SCHEMA**

1. **Table for labels/folders**

**folders\_by\_user**

user\_id **K (partition key) text**

label **C (clustering column) text**

unread\_message\_count **text**

**folders\_by\_user**

user\_id **K (partition key) text**

label **C (clustering column) text**

color **text**

**PRIMARY KEY**

1. **Messages for user and folder**

**messages\_by\_user\_folder**

user\_id **K (partition key) text**

label **K (partition key) text**

message\_id **text**

id **C (clustering column) timeuuid**

to **text**

subject **text**

is-read (flag) **Boolean**

Apache Cassandra doesn’t have sorting while fetching data so we are introducing timestamp as clustering column which will be sorting data in descending order.

1. **Message details by id**

**messsge\_by\_id**

id **K (Partition key) timeuuid**

from **text**

to **List<text>**

Subject **text**

Body **text**

**What is time UUID?**

Setting up hosted Cassandra instance?

It is available totally free with resources limit. So you don’t need to use any card or anything.

DataStax - <https://astra.datastax.com/3be5bbb6-5013-4fc0-8391-6736766b83aa>

Click on create database-

Provide a name- inbox-app

Provide keyspace - main

Choose cloud server and region click create database. Wait for some time, status will be updated to active from pending.

Now to use this we need to generate a token.

**Enum SchemaAction**

<https://docs.spring.io/spring-data/cassandra/docs/current/api/org/springframework/data/cassandra/config/SchemaAction.html>

**CREATE-** create each table as necessary

**CREATE\_IF\_NOT\_EXISTS-** create each table as necessary

**NONE-** take no schema actions (used in production mostly)

**RECREATE-** create each table as necessary dropping the first if it exists

**RECREATE\_DROP\_UNUSED-** drop each table in keyspace, create as necessary (only for development)