Cognizant

Time To Complete: 10 hrs to 12 hrs

E-Stock Market

Contents

[1 Problem Statement 2](#_Toc117103135)

[2 Application Architecture 3](#_Toc117103136)

[2.1 Microservice Architecture (Governance and Tooling/ Database and Storage): 3](#_Toc117103137)

[2.2 Microservice Architecture (Governance and Tooling/ Compute and Integration): 3](#_Toc117103138)

[2.3 Microservice Architecture (Compute and Integration/Presentation/Networking and Content Delivery): 3](#_Toc117103139)

[3 Tool Chain 4](#_Toc117103140)

[4 Business Requirements: 4](#_Toc117103141)

[5 Proposed Rest Endpoints to be exposed for CompanyManagementAPP and UserManagementApp 5](#_Toc117103142)

[5.1 Rest APIs: 5](#_Toc117103143)

[6 Rubrics/Expected Deliverables 6](#_Toc117103144)

[6.1 Engineering Concepts (Compute & Integration): 6](#_Toc117103145)

[6.2 Engineering Concepts (Security & Identity): 6](#_Toc117103146)

[6.3 Products & Framework (Database & Storage): 6](#_Toc117103147)

[6.4 Products & Framework (Governance & tooling): 6](#_Toc117103148)

[6.5 Debugging & Troubleshooting 6](#_Toc117103149)

[6.6 Code Quality/Optimizations/coverage 7](#_Toc117103150)

[7 Methodology 7](#_Toc117103151)

[7.1 Agile 7](#_Toc117103152)

# Problem Statement

**E-StockMarket** Application is a Restful Microservice application, where it allows users to manage the stocks like create, view stock price details and update and delete company details.

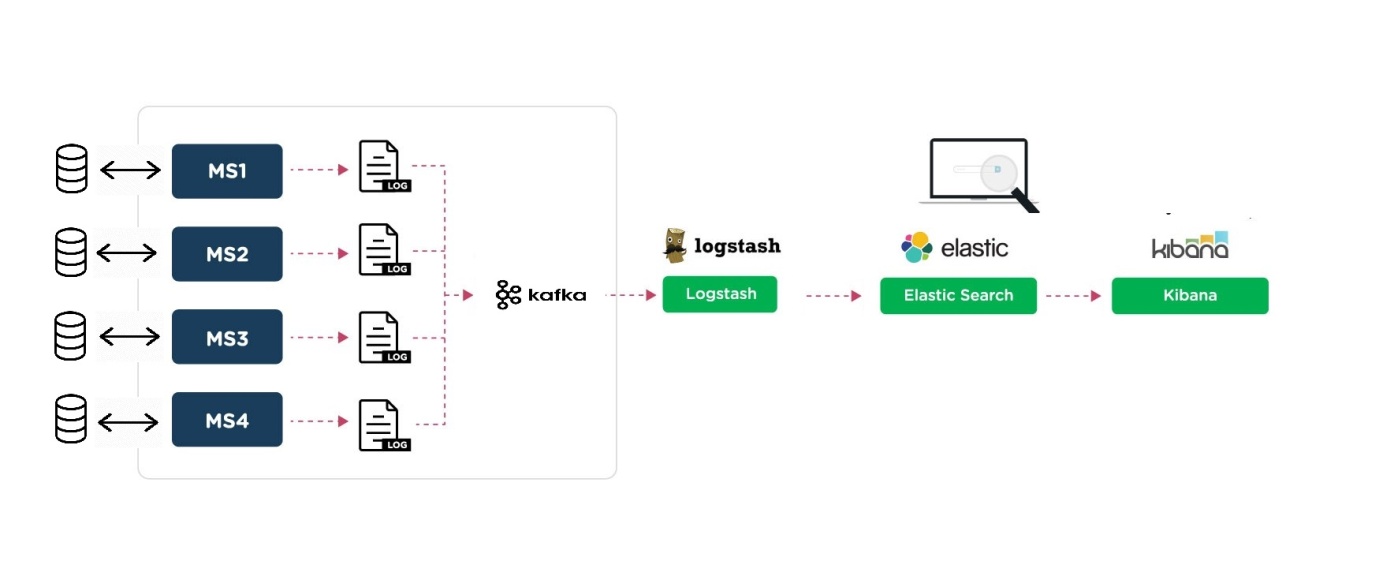
The core modules of E-Stock Market app are:

* Allows to add a new company details.
* Allows to add a new stock price detail against a company and updates the latest stock price against that same company.
* Allows to delete an existing company with all stock details.
* Allows to search the company or stock based on company ID/Code.
* Allows to display all company information and all stock detail (One-To-Many)
* Allows to update details of existing company.
* Allows displaying max, min and average stock prices between the stipulated time periods. For this purpose, there should be a frontend SPA created using Angular.

The scope includes developing the application using tool chain mentioned below.

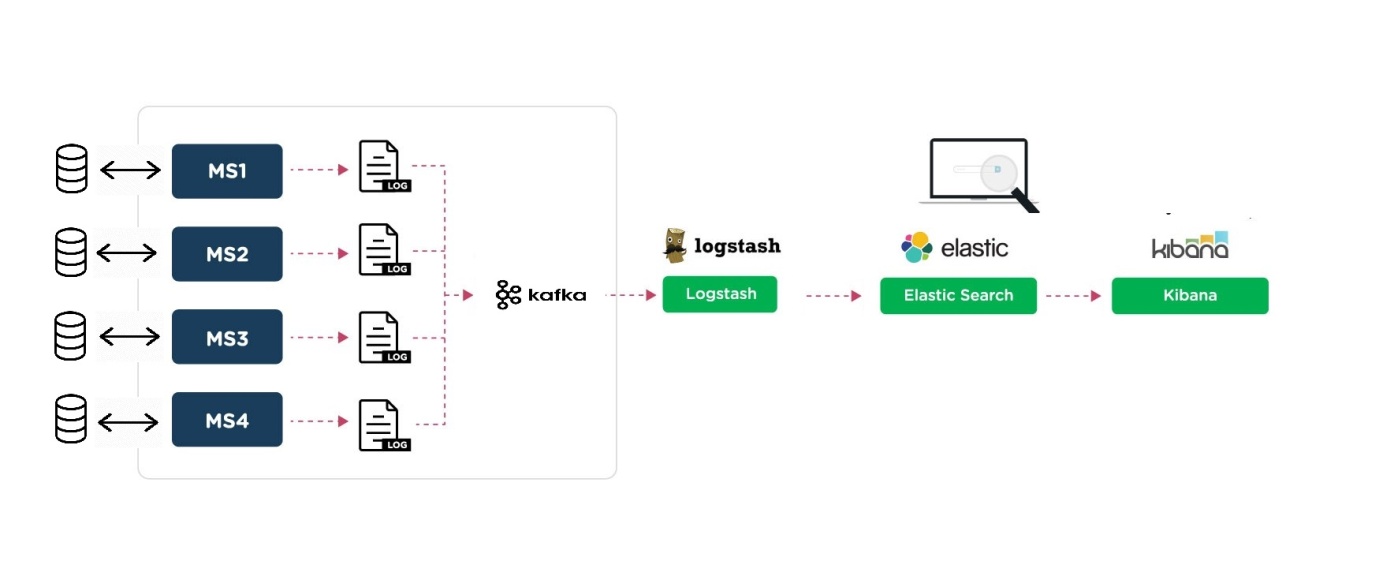
# Application Architecture

## Microservice Architecture (Governance and Tooling/ Database and Storage):



**Client App/ UI**

**Backend App**

backend integrated with UI



A green and black logo

Description automatically generated A blue and black logo

Description automatically generated A logo of a company

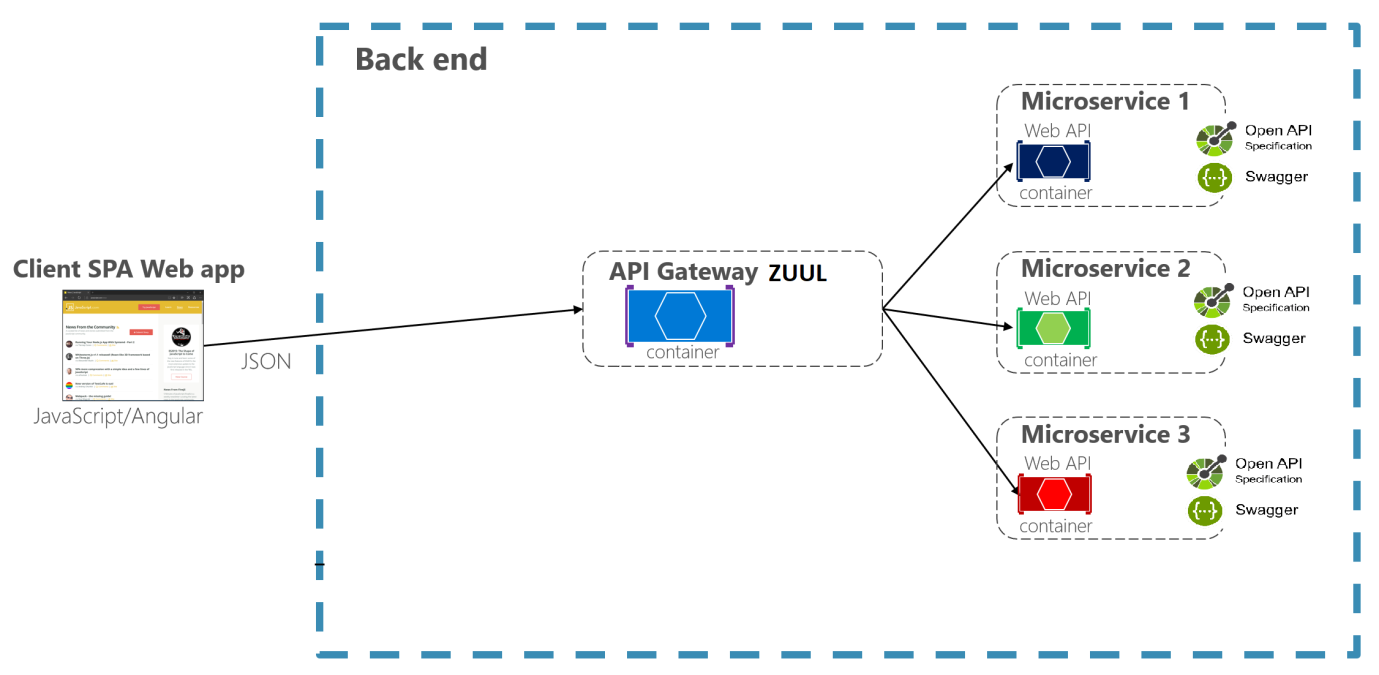
Description automatically generated A logo of an orange square with white squares and a black text

Description automatically generated A logo with text on it

Description automatically generated



## Microservice Architecture (Compute and Integration/Presentation/Networking and Content Delivery):



**POSTMAN**

**Integration Testing**

# Tool Chain

|  |  |  |
| --- | --- | --- |
| Competency | Skill | Skill Detail |
| Engineering Mindset | Networking and Content Delivery |  |
|  | Ways of Working |  |
|  | Consulting Mindset |  |
|  | DevOps |  |
|  | Code Quality/coverage | SonarQube/EclEMMA/Jacoco |
| Programming Languages | Application Language | Java, Spring Boot |
|  | Networking and Content Delivery | AWS API gateway |
| Message Broking | Kafka | Microservice communication using Pub-sub model |
|  |  |  |
|  | Database & Storage | MySQL |
|  | Governance & Tooling | Git |
|  |  | Junit |
|  |  | Mockito |

# Business Requirements:

As an application developer, develop microservices with below guidelines:

|  |  |  |  |
| --- | --- | --- | --- |
| User  Story # | User Story Name | User Story | Development |
| US\_01 | Company Registration | As a user I am be able to register a new company in E-Stock Market application  Acceptance criteria:   1. As a user I am be able to furnish following details at the time of company registration    1. Company Code    2. Company Name    3. Company CEO    4. Company Turnover    5. Company Website    6. Stock Exchange it is enlisted in (BSE, NSE etc) 2. All details fields are be mandatory 3. Company Code must be unique 4. Company Turnover must be greater than 10Cr. | API to be developed |
| US\_02 | Add Stock Price | As a user I am be able to update stock price for any company  Acceptance criteria:   * Stock price must be a fractional value. * Date & Time of the stock price is automatically fetched from server system | API to be developed |
| US\_03 | View and delete/update Company Details | As a user I am able to view and delete details of company.  Acceptance criteria:   * If a single company, details are demanded it shows complete details of company along with latest stock price. * If all company details are demanded it shows list of all companies with their latest stock price * If company is deleted, it also deletes all stock price data associated with the company. * If company stock price needs to be updated, it provides update workflow. | API to be developed |
| US\_04 | Register and Login User in UserMangementApp | User able to perform CRUD of company app after Login from UserManagementApp.  Acceptance criteria:   * A user must register and login from UserManagementApp which should generate Bearer Token. * User must use Bearer Token to access all APIs of CompanyManagementApp. * When user Registers and Logs in, notification should be published to Kafka topic. Company Microservice should subscribe to that topic and consume the notification message. | API development. Microservice communication. |

# Proposed Rest Endpoints to be exposed for CompanyManagementAPP and UserManagementApp

## Rest APIs:

|  |  |  |
| --- | --- | --- |
| **POST** | **/api/v1.0/market/company/register** | **Register a new company without stock price** |
| **GET** | **/api/v1.0/market/company/info/<companycode>** | **Fetches the Company Details** |
| **GET** | **/api/v1.0/market/company/getall** | **Fetches all the Company Details** |
| **DELETE** | **/api/v1.0/market/company/delete/<companycode>** | **Deletes a company** |
| **POST** | **/api/v1.0/market/stock/add/<companycode>** | **Add and Update new stock price** |
| **PUT** | **/api/v/1.0/market/company/put/<companycode>** | **Updates details of a particular company** |
| **POST** | **/auth/v1.0/user/add/user** | **Registers new User in UserManagementApp** |
| **GET** | **/api/v1.0/user/info/<userid>** | **Allows to search user by empID(Should not be allowed without Login)** |
| **POST** | **/auth/v1.0/user/login** | **Allows user to Login (Jwt Authentication and Authorization)** |

# Rubrics/Expected Deliverables

## Engineering Concepts (Compute & Integration):

1. As an application developer:
   1. Develop the application as a microservice architecture.
   2. Implementation as follows:
      1. Use Domain Driven Design to implement distributed architecture.
      2. Follow the Single Data Store per microservice practice.
      3. Document REST endpoints with OpenAPI/ Swagger3.
      4. Add CQRS pattern for Event Sourcing for saving and retrieving stock details(Optional)
      5. Expose all rest Endpoints using a common API Gateway AWS

## Engineering Concepts (Security & Identity):

* + - 1. As an Application Developer:
         1. Restrict the access over all write operation (secured operations) by adding authentication

## Products & Framework (Database & Storage):

1. As an application developer:
   1. Implement ORM with Spring Data JpaRepository/Jakarta and MySQL.
   2. Use MySQL for maintaining data for at least one of the microservice.

## Products & Framework (Governance & tooling):

1. As an application developer
   1. Integrate AWS API gateway to orchestrate the microservice API endpoints.
   2. Consume API- from one Microservice into another Microservice using RestTemplate/Feign Client.

## Debugging & Troubleshooting

1. Generate bug report & error logs - Report must be linked with final deliverables which should also suggest the resolution for the encountered bugs and errors.

## Code Quality/Optimizations/Coverage

1. Associates should have written clean code that is readable
2. Associate should have used the ESLint feature /PMD of Angular to ensure code quality and standard code style.
3. Unit testing in Backend using Junit with Mockito.
4. Unit testing in frontend using Karma and Jasmine.
5. EclEMMA with TestNG for code coverage or JaCoco.

# Methodology

## Agile

1. As an application developer, use project management tool along to update progress as you start implementing solution.
2. As an application developer, the scope of discussion with mentor is limited to:
   1. Q/A
   2. New Ideas, New feature implementations and estimation.
   3. Any development related challenges
   4. Skill Gaps