Coding Exercise - backend

Task Delivery

The following will be provided to the candidate, both should be attached on this document:

- a pdf documentation with the requirements, guidelines and deliverable results.
- a zip file of the template project

Requirements

Listed below is the requirement of the project to be delivered, required features are expected to be complete. Candidate who will also implemented optional features will be given more chance.

Required features

Create a **Maven** project to implement a couple of simple micro-services with **Spring Boot Framework.** Micro services are listed below:

1-Authorization Service. This micro service is responsible for Authorizing the User by Username and Password.

2-User Profile Service. This micro service will save profile informations such as Address and Phone number.

User should be able to create profile information, update or delete them after logging in to the system.

- POST /login (Synchronous, just send the username and password in the post body. User will be authenticated and can perform the below operations.)
- POST /profile (Synchronous rest call between Authorization and Profile Service, user will be creating their own profile info if they are authenticated.)
- PUT /profile (Event based) (If user is authenticated they can update their own profile only.)
- DELETE /profile (Event based) (If user is authenticated they can delete their own profile only.)

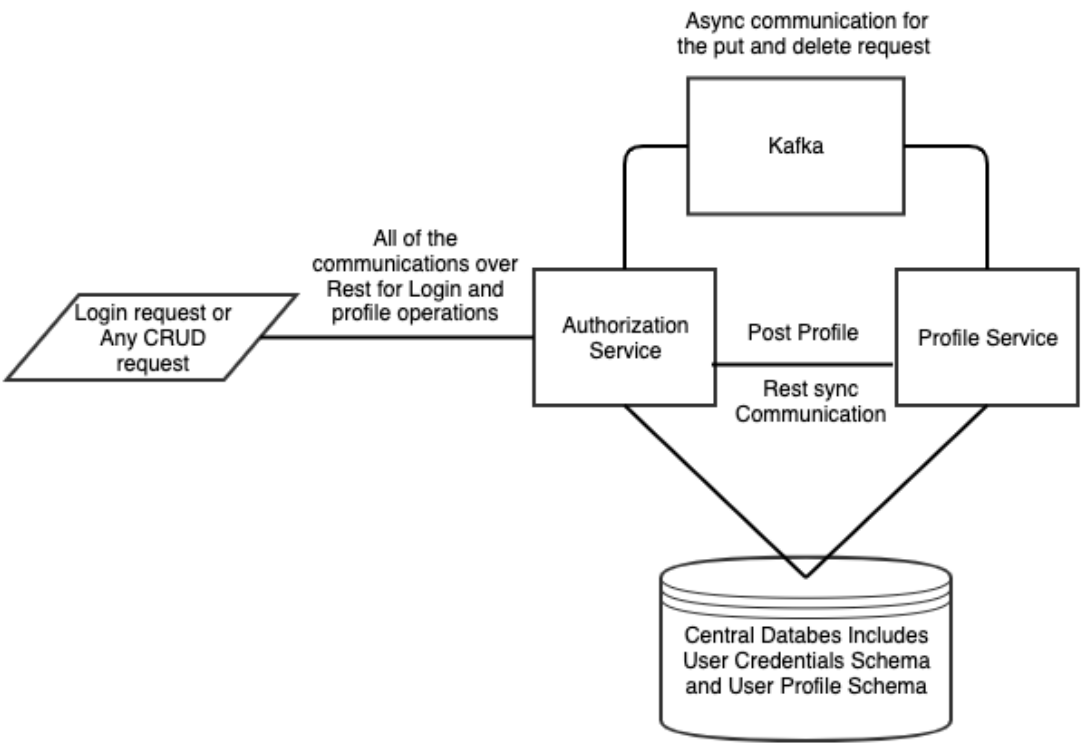
APIs will perform CRUD operations using a persistence framework on a Profile entity with the following fields:

- Address (String)
- Phone number

You do not need to implement the on boarding (creation of the username/password in the authorization database). You can add those manually in the database.

All of the requests need to go through Authorization service. Authorization service will act as API Gateway too.

The authorization service should serve the login endpoint. A simple post call with username and password in the post body. The moment User posts the correct username and password, they will be able to post and create the user profile info. Username and Password will be stored on authorization database only and user-profile will be stored on the profile service's database.



Provide **unit tests** for controllers, services and DAO classes.

The project should be hosted in local **GIT** repository. Please make sure you have committed all of your changes in the Git.

Optional features

Use Reactive (Mono/Flux)

Guidelines

- You can use any IDE of your choice (IntelliJ, Eclipse, Netbeans ...)

- Spring Annotation is preferred over XML.
- For simplicity use an in-memory DB like (HSQLDB or H2)
- Use Spring boot
- Please follow the provided project structure
- The submitted project will be evaluated for implementation completeness, code quality, documentation)
- Be prepared to discuss any part of the application during the in-person interview.

Deliverable Results

A zip file of the project containing the source code and .git folder (local GIT repo with a proper history of all commits), without build artifacts and compiled classes. The project should not be checked into any of the remote repository sites.

Document important configurations, references or design concerns in the README.md file.

The project should be easily locally executable without any extra configuration with the following commands:

```
mvn clean install
mvn spring-boot:run
```

The application should be accessible by the following url: http://localhost:9090/assignement

Template Project

A web application project based on Maven standard directory layout is provided:

POM file

Contains a barely empty **war packaging** POM file without any dependency or plugin.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.amdocs.media</groupId>
<artifactId>assignement</artifactId>
<packaging>war</packaging>
<version>1.0-SNAPSHOT</version>

<build>
<finalName>assignement</finalName>
</build>
</project>
```

Package structure

To further emphasize the project structure, in the source folders for java test created the following package:

Package	Description
`com.amdocs.media.assignement.config`	package for web application configuration classes
`com.amdocs.media.assignement.controller`	package for Spring controller classes
`com.amdocs.media.assignement.service`	package for service classes
`com.amdocs.media.assignement.dao`	package for DAO classes

Documentation

An empty README.md file is available at the project root directory.