|  |
| --- |
| **Project Overview** |

Develop two microservices: KBMGrpcService and KBMHttpService. The purpose of this project is to evaluate the developer's ability to implement essential microservice features, handle state management, and expose both gRPC and HTTP APIs.

1. KBMGrpcService
   1. Exposes a grpc api
      1. Create Organization
         1. Request: name(required), address
         2. Response: organization id
      2. Get Organization by id
         1. Response (name, address, createdAt, updatedAt)
      3. Query Organizations paginated
         1. Request: page, pageSize, orderBy, direction, queryString(to search in name or address)
         2. Response: page, pageSize, total, list of organizations(name, address, createdAt)
      4. Update Organization by id
      5. Request: name, address
      6. Delete Organization by id
      7. Create User
         1. Request: name, username, email
         2. Response: user id
      8. Get User by id
         1. Response: name, username, email, createdAt, updatedAt
      9. Query users
      10. Update User
      11. Delete User
      12. Associate User to organization
      13. Disassociate User from organization
      14. Query users for organization
          1. Request: organizationId, page, pageSize, orderBy, direction, queryString(to search in name, username or email)
      15. Response: page, pageSize, total, list of users (name, username, email, createdAt)
   2. Business layer(services)
      1. organization
      2. name is unique among valid organizations
      3. user
         1. username is unique among valid users
         2. email is valid email
         3. email is unique among valid users
   3. Data layer
      1. Database (Microsoft SQL Server recommended docker image: <http://mcr.microsoft.com/mssql/server:2017-latest> or newer)
         1. Entities:
            1. Organization (Id, Name)
            2. User (Id, Name, Username, Email)
            3. Any other if required
         2. All database entities will also have auditable fields: createdAt(required), updatedAt, deletedAt
         3. Entities can only be soft deleted
   4. Validations and constraints
      1. organization name is unique among valid organizations
      2. user username is unique among valid users
      3. user email is valid email
      4. user email is unique among valid users
2. KBMHttpService
   1. Expose an http api as a proxy to the grpc api.

**Evaluation Criteria**

* **Completion**: Meeting all requirements.
* **Code Quality**: Readability, structure, documentation, and use of best practices.

**Submission Requirements**

* **Source Code**: Complete source code in a Git repository. There should be separate commits for the different features, not all project in one big commit.
* **Documentation**: A README file with instructions for setting up and running the microservices.