



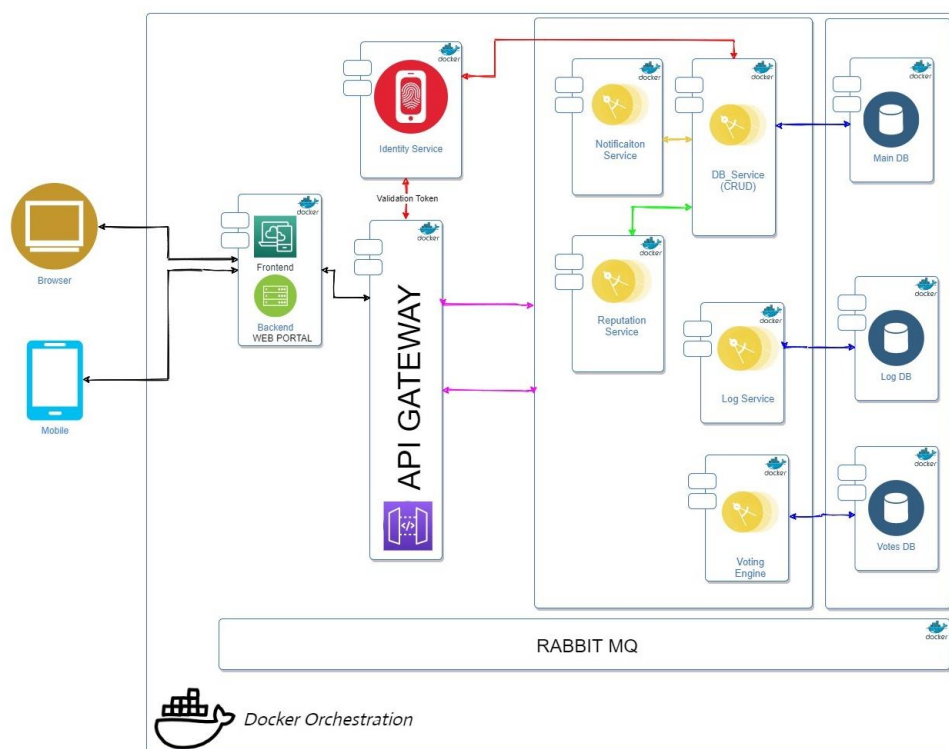
SERVICES DAO PLATFORM ARCHITECTURE

PLATFORM ARCHITECTURE SUMMARY

The Services DAO platform consists of several micro-services which supports Docker containerization. Applications will be developed with .NET Core 3.1 Framework and will be unified in a single solution file. Model View Controller (MVC) pattern will be used in the UI layer which is a single web application and WebApi pattern will be used in backend applications. MySQL databases will be used for data storage. There will be three databases that will be used for logging, voting and general data storage.

Every application will be connected to the RabbitMQ message broker for communication. Also Http requests will be accepted between applications for request-response type tasks.

Test projects which contains unit tests and integration tests will be included in the solution. In addition docker compose files will be included for easier installation and testing.





SERVICE DEFINITIONS

DAO_WebPortal

Project Type: .NET Core 3.1 MVC

DAO_WebPortal will be the presentation layer which users can access from web browsers. It will have a public landing page and login-register pages. After successful login users can see inner views in the platform according to their authorization types. Web portal will have a responsive design which can be used in desktop and mobile devices.

Dependencies:

- .NET Core PagedList
- Newtonsoft.Json

Dependency Reference:

- <https://github.com/dncuug/X.PagedList>
- <https://github.com/JamesNK/Newtonsoft.Json>

DAO_ApiGateway

Project Type: .NET Core 3.1 WebApi

DAO_ApiGateway will be a service which accepts requests from DAO_WebPortal and transmit these requests to backend services. It protects backend layer from direct interaction. Api gateway functionality will be provided by “Ocelot” which is an open-source library.

Dependencies:

- Ocelot
- JwtBearer

References:

- <https://github.com/ThreeMammals/Ocelot>



DAO_IdentityService

Project Type: .NET Core 3.1 WebApi

DAO_IdentityService will have the authentication and authorization functionality. It will work with JWT tokens and controls every request passing through DAO_ApiGateway. It will also implement login, register and password reset functionalities.

Dependencies:

- JwtBearer

DAO_DbService

Project Type: .NET Core 3.1 WebApi

DAO_DbService is the access layer to the main database of DAO platform. Main database will include every table except log tables and voting tables. DbService will contain simple CRUD operations and custom query methods. Database connection will be made by EntityFramework and model context will be included in the project.

Dependencies:

- EntityFrameworkCore
- PagedList.Core
- AutoMapper

Dependency Reference:

- <https://github.com/dncuug/X.PagedList>
- <https://github.com/AutoMapper/AutoMapper>

DAO_LogService

Project Type: .NET Core 3.1 WebApi

DAO_LogService is the logging service of the DAO platform. It will receive user logs, error logs and application logs from RabbitMQ and Http Requests. LogService will have it's own database which only contains log tables. Database connection will be made by EntityFramework and model context will be included in the project.

Dependencies:



- EntityFrameworkCore
- PagedList.Core
- AutoMapper

Dependency Reference:

- <https://github.com/dncuug/X.PagedList>
- <https://github.com/AutoMapper/AutoMapper>

DAO_VotingEngine

Project Type: .NET Core 3.1 WebApi

DAO_VotingEngine will provide the core functionality of the DAO platform. Every voting operation will be received by VotingEngine and will be recorded in a separate vote database. Every vote will contain related project identity and user identity. However project identities and user identities won't be checked in this application. Voting engine will assume request parameters are correct because it can be implemented to another project and should work independently. Database connection will be made by EntityFramework and model context will be included in the project.

Dependencies:

- EntityFrameworkCore
- PagedList.Core
- AutoMapper

Dependency Reference:

- <https://github.com/dncuug/X.PagedList>
- <https://github.com/AutoMapper/AutoMapper>

DAO_ReputationService

Project Type: .NET Core 3.1 WebApi

DAO_ReputationService will be the reputation management service of the DAO platform. Reputations will be stored in the DAO main database and will be accessed by DAO_DbService but reputation management operations will be done in this application.



DAO_NotificationService

Project Type: .NET Core 3.1 WebApi

DAO_NotificationService will handle notification requests like email, push notification and SMS. In the first version only email notification will be available.

Helpers

Project Type: .NET Core 3.1 Class Library (DLL)

Helpers is a class library project which contains utility methods and class definitions which will be used in every application in the system. It will be referenced to all applications.

Dependencies:

- Newtonsoft.Json
- RabbitMQ Client

Dependency Reference:

- <https://github.com/JamesNK/Newtonsoft.Json>
- <https://github.com/rabbitmq>