**Overall Architecture:**

**Diagram

Description automatically generated**

**API Gateway:**

The API Gateway is the doorway to access all of over services, no service can be directly accessed.

The API gateway is also response for authentication and authorization, I have implemented Oauth in it but didn’t got time to integrate with other services.

The API gateway is also response for inter services communication, if we are going for synchronous communication.

The best approach according to me would be asynchronous communication through some message queue so that each transaction that move from one service to other, It’s state could be maintained.

**Customer Service:**

Chart, funnel chart

Description automatically generated

The Customer Service consist of layer architecture we have 5 layer each layer has a single responsibility

along with that we have a common layer and Test layer.

**Common layer** will have the resources that will be shared between all the layers.

**Test layer** is to write unit test so we can implement the devOps , I couldn’t get time to write sample unit test.

**Web API** is responsible for request, response cycle and authorization/authentication.

**Service Layer** is responsible for transforming data and integrating with other services or entities that could be internal or third party.

**Business Layer** have all the business logic of the application it is the backbone of the application.

**Repository layer** is responsible for interacting with database, I have also implemented the the unitOfwork for re-usability of code.

**Data Versioning**:

I have added the data versioning at the database level , history of the data is maintained along with the audit logs.

**Logging:**

I have added the file logger right now but it could be changed according to server where the service will be deployed application insight could be integrated incase of deployment on azure

**Memory Cache:**

.net built in memory cache is being added if the user access the customer for the first time the value is added In the cache , after that the value are served from cache.

Authorization:

JWT service is being used for authentication and role based authorization

**Dockerfile:**

Docker file is added in to the project to contained the solution we first need to create docker image

Through following command

docker build --tag Name:1.0 .

We can push our container image to docker hub container registry to appending registry path before name

To create the container from image use following command

docker run –d –p 25565:25565 – name “containerName” “image”

-d means detached  so that this container will run in background and we will not see any logs in command prompt

-p attaching container port to host port