

This system was design based on Clean Architecture, so it has the following layers:

- Application: which has the serviced and business logic
- Domain: which has domain models
- Infrastructure: where all external resources are configured, like the Database and RabbitMq
- WebApi: which is the presentation, where all the interactions happens

The Database choosed is the MySQL Server, since the data expected is structured, and to manage the database interactions Entity Framework Core was the choice, it helps to work abstractly with the database and map all the entities to update the database using migrations.

To ensure loose coupling between application objects, the mediator pattern was used with the MediatR library. To segregate the write and read operations the CQRS project pattern was used.

To guarantee that background processes run without impact the requests' performance RabbitMQ consumers were implemented using Masstransit library, processing specially the door entry history after an user requests to open a door.

In order to keep the application safe and manage the access policy, JWT-bearer authentication was used.

RabbitMQ and MySQL Server run independently using Docker.