

Task 1

- Only QA , PM, and ADMIN can submit tickets.
- Only QA can submit bug and test case tickets and PM can only submit feature request ticket. But, ADMIN can submit all types of tickets.
- Only RD and ADMIN can resolved tickets.
- Only ticket owner can delete their own tickets, but ADMIN can edit and delete all types of tickets.
- Description and summary are not only required when submit tickets for QA, but also the same applies to PM.
- Only QA, PM, ADMIN are authorized to perform add ticket function.
- Users must be authenticated and authorized to use ticket system. Otherwise, users will be redirected to account registration page.
- If users are not authorized, they will be redirected to access denied page.

Task 2

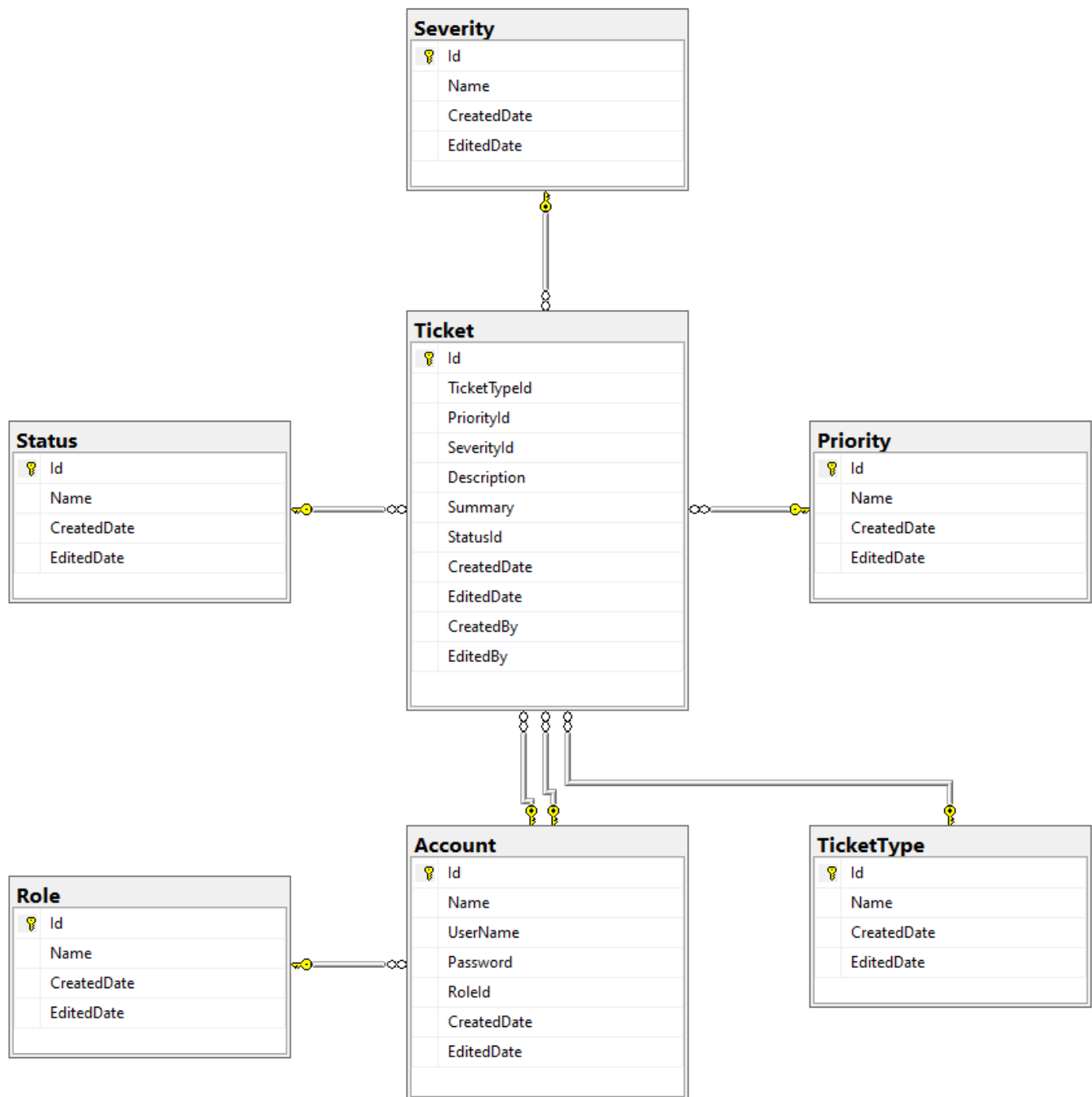
- Ticket System project on github only fills minimum requirements and some features of the assignment, because it would take so much time to full implement all requirements and fix all possible errors.
- Technologies and skills used in the project are listed below:
 - Cookies
 - Session
 - Command Query Responsibility Segregation (CQRS) pattern
 - Repository pattern
 - Mediatr pattern
 - XUnit
 - FluentAssertion
 - AutoMapper
 - Dapper
 - Dependency Injection (IOC)
 - Clean Architecture (DDD)
 - Antiforgery Token
 - Cancellation Token
 - Task / Async Await
 - .Net Core MVC
 - Authentication
 - Authorization
 - Pagination
 - BEM CSS naming convention
 - SQL Server / T-SQL

Task 3

- Basic UI of Ticket System and how they system works can be shown during interview.
- Ticket System architecture is base on clean architecture and some DDD principles.
- Class diagram would be much better to explain during interview since it would have take so much time to draw.

- Some basic UI to fulfill requirements are shown below.

Table Diagrams



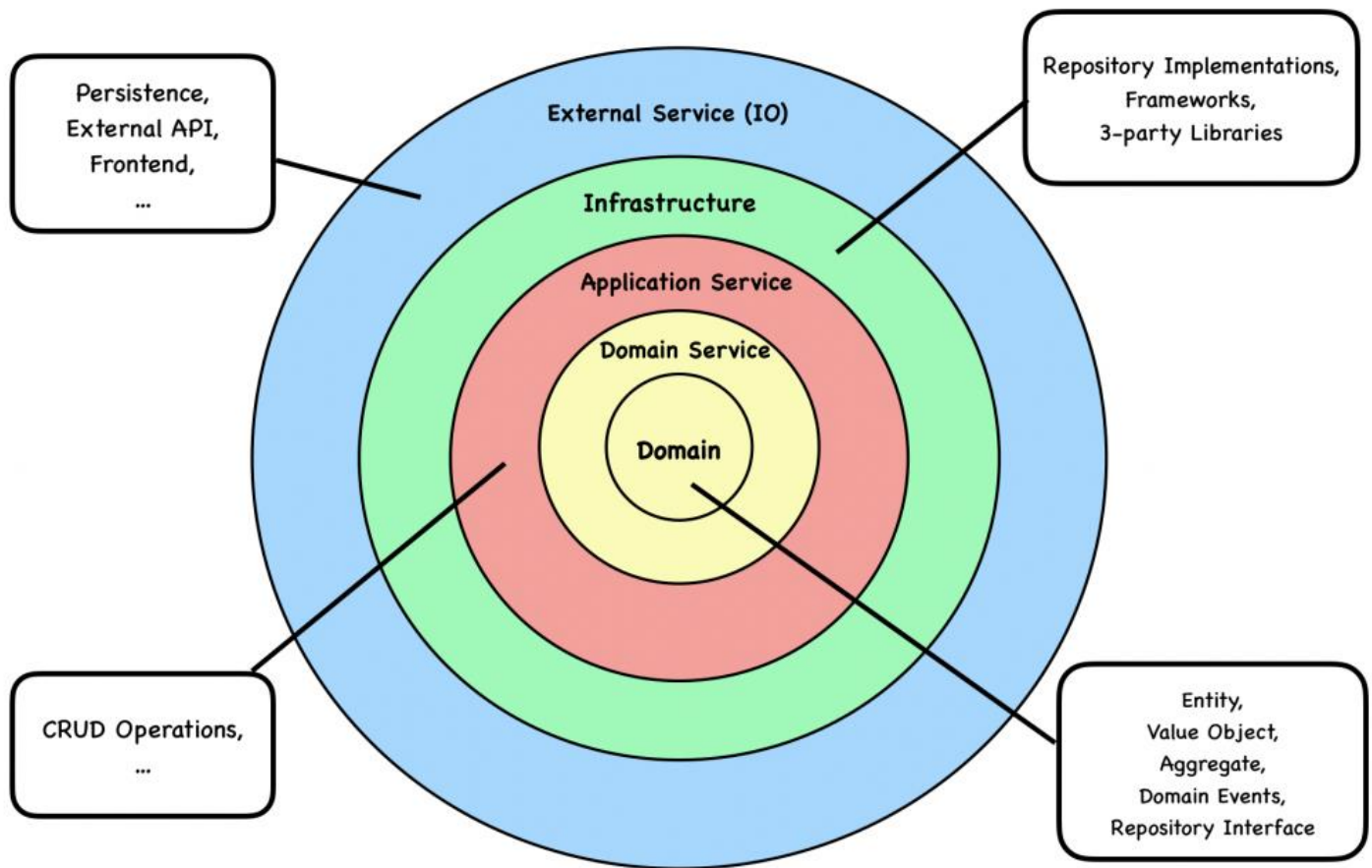
Architect of System

Base on clean architecture and DDD, all layers should point inward to domain layers.

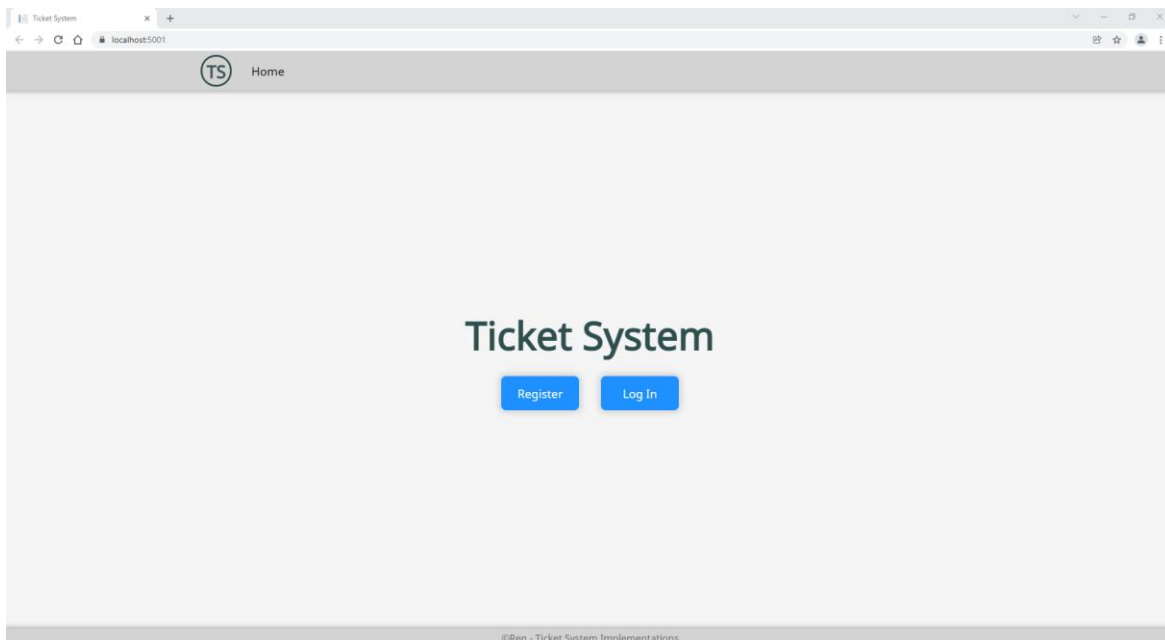
- Presentation
 - Web (reference application and infrastructure)
- Infrastructure (reference application)
- Core
 - Application (reference Domain)

- Domain

Architect Diagrams



Home Page



Register Page

Register

localhost:5001/Account/Register

Home

Register

Name

Username

Password

Role

Register

©Ren - Ticket System Implementations

Login Page

Log In

localhost:5001/Account/Login

Home

Log In

Username

Password

Log In

©Ren - Ticket System Implementations

Ticket Page

Ticket

localhost:5001/Ticket

TS

Home

Ticket

Log Out

Ticket

Add

	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
1	Bug	Urgent	Critical	Description1.	Summary1.	MichaelQA	QA	Work in progress	
2	Bug	Urgent	Error	Description2.	Summary2.	MichaelQA	QA	Resolved	
3	Bug	High	Error	Description3.	Summary3.	MichaelQA	QA	Cancelled	
4	Bug	High	Normal	Description4.	Summary4.	MichaelQA	QA	Work in progress	
5	Bug	Medium	Critical	Description5.	Summary5.	MichaelQA	QA	Resolved	

First

Previous

Next

Last

©Ren - Ticket System Implementations

Add Ticket Page

Add Ticket

localhost:5001/Ticket/PostTicketForm?

TS

Home

Ticket

Log Out

Add Ticket

TicketType

Select Ticket Type

Priority

Select Priority

Severity

Select Severity

Description

Please enter description

Summary

Please enter summary

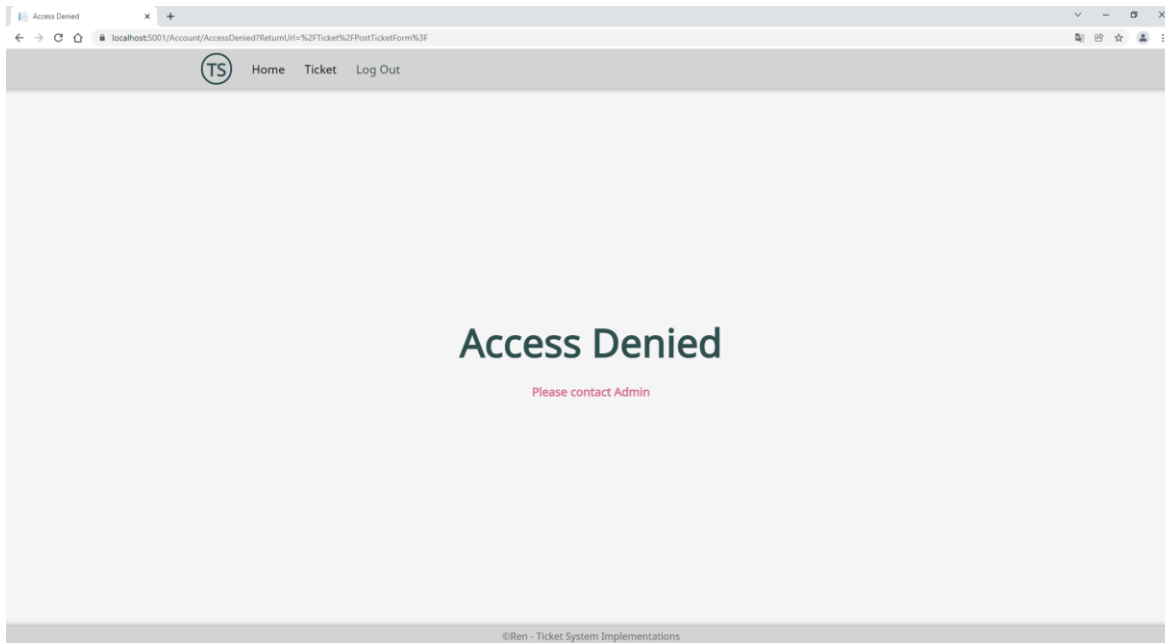
Status

Select Status

Add

©Ren - Ticket System Implementations

Access Denied Page



Task 4

- Due to time concern, information below only illustrate general ideas and designs for web api documents and json formats. Of course, more detailed information can be added later.

General API Documents

GET

- <https://.../Ticket>
- Show all tickets.

GET

- <https://.../Ticket?page={number}>
- Show all tickets by certain page number

GET

- <https://.../Ticket/{id}>
- Show a ticket by querying id.

POST

- <https://.../Ticket>
- Add a ticket into database with information provided in request body.

PUT

- <https://.../Ticket/{id}>
- Edit a ticket with information provided in request body by querying id.

DELETE

- <https://.../Ticket/{id}>
- Delete a ticket by querying id.

General Json Format

```
{
  "Tickets" : [
    { "id" : 1, "summary" : "summary1...", "description" : "description1..." },
    { "id" : 2, "summary" : "summary2...", "description" : "description2..." },
    { "id" : 3, "summary" : "summary3...", "description" : "description3..." }
  ],
  "StatusCode" : "200",
  "ErrorMessage" : "No error message",
  "Url" : "https://.../Ticket",
  "JwtToken" : "eyJhbGciOiIs.eyJzdWIiOiIxMjM5MDIyfQ.SflKxwRJSM",
  "ExpiredDatetime" : "2021/12/25 17:00:30"
}
```