

**Mentor Appointment Scheduling Application**

**Software Design Document**

– Ho Chi Minh, May 2022 –

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) **5**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 5

[1.1 Back-end Code Packages](#_heading=h.o1jjkkxx1xtq) 5

[1.2 Font-end Code Packages](#_heading=h.2n3nh6lv69e4) 7

[2. Database Schema](#_heading=h.1fob9te) 8

[II. Code Designs](#_heading=h.3znysh7) **9**

[1. Major Features](#_heading=h.7nng70z330a) 9

[a. Class Diagram](#_heading=h.8h7mj61ay2ua) 9

[b. Class Specifications](#_heading=h.k72kqiejqpsz) 9

[MajorsController Class](#_heading=h.l8zdv3lx4a0) 9

[IMajorService Interface](#_heading=h.v9e6y6f21bp0) 10

[MajorService Class - Implement from IMajorService](#_heading=h.vl67a1iqkm5o) 10

[IMajorRepository Interface](#_heading=h.yhoxnot346cs) 11

[MajorRepository Class - Implement from IMajorRepository](#_heading=h.w7pt9yosy77m) 11

[Major Class](#_heading=h.7910n66wzcwf) 11

[MajorResponse Class](#_heading=h.kdzga8qik2fm) 12

[MajorCreateRequest Class](#_heading=h.dhc4bfggu12z) 12

[MajorUpdateRequest Class](#_heading=h.cz6pf7dkl4re) 12

[2. Subject Features](#_heading=h.y8iy8wus6qu7) 13

[a. Class Diagram](#_heading=h.bsuxvh5g8y7m) 13

[b. Class Specifications](#_heading=h.h565ku8axea3) 13

[SubjectsController Class](#_heading=h.5dhr8b4hc7rp) 13

[ISubjectService Interface](#_heading=h.l6mn0pe5vgbb) 14

[SubjectService Class - Implement from ISubjectService](#_heading=h.7v4ua9xylk4j) 14

[ISubjectRepository Interface](#_heading=h.4qojmjsea7hp) 14

[SubjectRepository Class - Implement from ISubjectRepository](#_heading=h.j582lssdjk0q) 15

[Subject Class](#_heading=h.nob1q38hzb9w) 15

[SubjectResponse Class](#_heading=h.w0qxos5i5epw) 15

[SubjectDetailResponse Class](#_heading=h.fso29syuozpp) 16

[SubjectCreateRequest Class](#_heading=h.4b974hv4i30v) 16

[SubjectUpdateRequest Class](#_heading=h.r61ihygb7h46) 16

[3. MasUser Features](#_heading=h.1w1jf5i53vjk) 18

[a. Class Diagram](#_heading=h.cfj91r5pnyq8) 18

[b. Class Specifications](#_heading=h.2nnro4qmx6aw) 18

[4. MentorSubject Features](#_heading=h.9c2q2j86dxag) 19

[a. Class Diagram](#_heading=h.r0xttp4ottby) 19

[b. Class Specifications](#_heading=h.uqg6fdy4zpwg) 19

[MentorSubjectsController Class](#_heading=h.omwpmbtr1psy) 19

[IMentorSubjectService Interface](#_heading=h.da39v22i654r) 20

[MentorSubjectService Class - Implement from IMentorSubjectService](#_heading=h.h11cpt21wbrc) 20

[IMentorSubjectRepository Interface](#_heading=h.ppomfcfu0tw) 21

[MentorSubjectRepository Class - Implement from IMentorSubjectRepository](#_heading=h.hx615hjyl73) 21

[MentorSubjectResponse Class](#_heading=h.urewwut0sxw3) 21

[MentorSubjectRegisterRequest Class](#_heading=h.x5yghrrtvpg4) 22

[MentorSubjectUpdateRequest Class](#_heading=h.iwbj4rtslh48) 22

[5. Slot Features](#_heading=h.x87rc1cgjs8h) 23

[a. Class Diagram](#_heading=h.vm9mmnjli2te) 23

[b. Class Specifications](#_heading=h.dguu2gff11h) 23

[SlotsController Class](#_heading=h.tmlc0ka17qq5) 23

[ISlotService Interface](#_heading=h.8k2qeq95rkwm) 24

[SlotService Class - Implement from ISlotService](#_heading=h.bqlqyhlkz0vk) 24

[ISlotRepository Interface](#_heading=h.9vg3kqwy1llg) 24

[SlotRepository Class - Implement from ISlotRepository](#_heading=h.5z92xfwsrmo9) 25

[Slot Class](#_heading=h.lsduiwpr0rjf) 25

[SlotResponse Class](#_heading=h.m87yxseyz1fk) 25

[SlotDetailResponse Class](#_heading=h.tkg76bq2d3ft) 26

[SlotCreateRequest Class](#_heading=h.u8jt2vanpu6i) 26

[6. Appointment Features](#_heading=h.or316b2m0koz) 27

[a. Class Diagram](#_heading=h.mto2z6xb845n) 27

[b. Class Specifications](#_heading=h.z6x89qv7e49k) 27

[7. Question Features](#_heading=h.3f2ivj48v50w) 28

[a. Class Diagram](#_heading=h.5fdwgdvr3ydp) 28

[b. Class Specifications](#_heading=h.bhrmliu9hda1) 28

[8. Rating](#_heading=h.606n4qu8hvbr) 29

[a. Class Diagram](#_heading=h.1y2efggcxpv0) 29

[b. Class Specifications](#_heading=h.yxbz1zg7opui) 29

[III. Database Tables](#_heading=h.17dp8vu) **29**

[1. Majors](#_heading=h.3rdcrjn) 29

[2. Subjects](#_heading=h.u8pcylkwstiu) 30

[3. MasUsers](#_heading=h.qgurofn495io) 30

[4. MentorSubjects](#_heading=h.hmdmzen222sr) 30

[5. Slots](#_heading=h.ylzzbsaraqmn) 31

[6. Appointments](#_heading=h.zh6rjeafs8ru) 31

[7. Questions](#_heading=h.wvus24thj9ho) 31

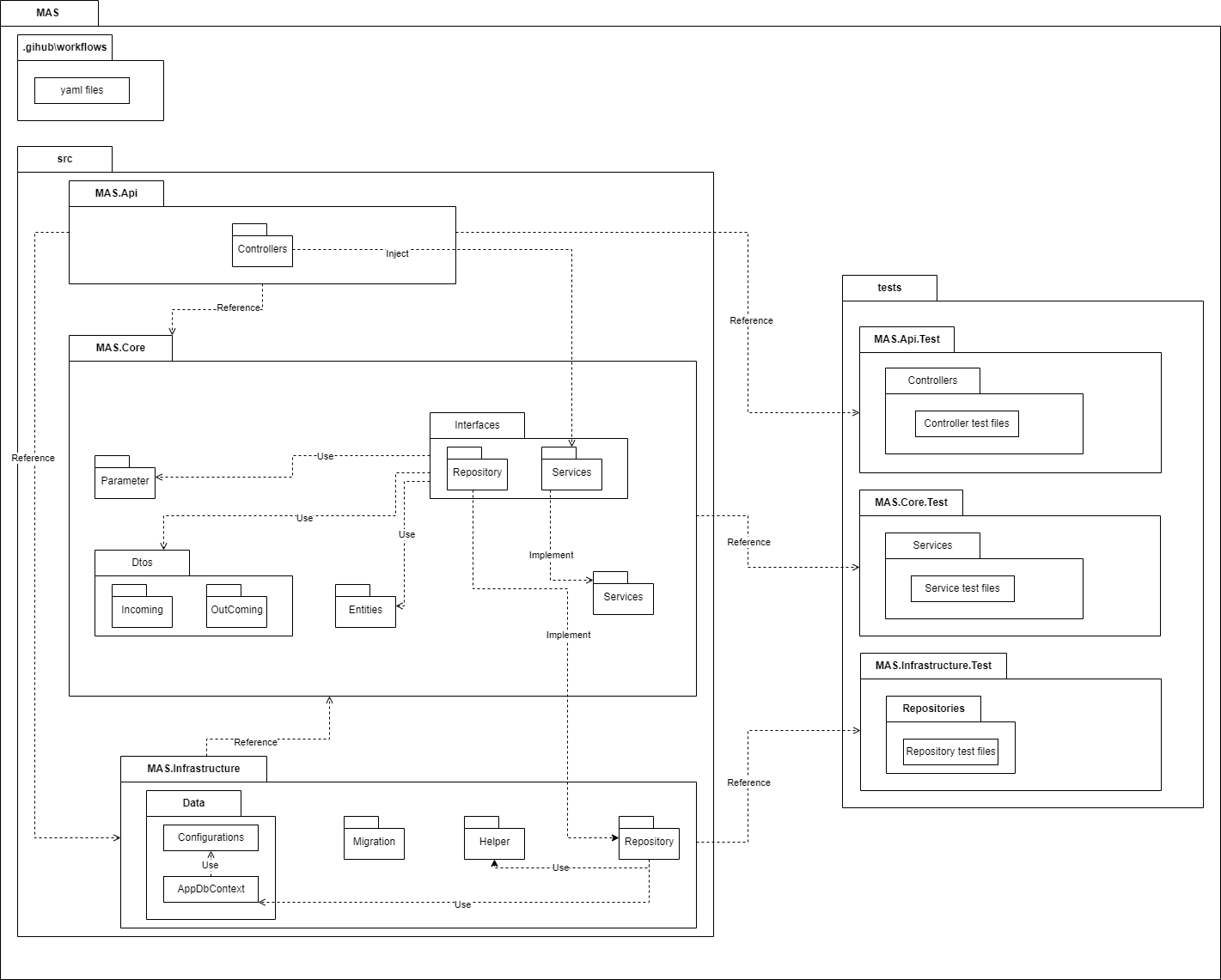
[8. Ratings](#_heading=h.34howjb4i786) 32

[9. AppointmentSubjects](#_heading=h.65t0c588ds5b) 32

# I. Overview

## 1. Code Packages/Namespaces

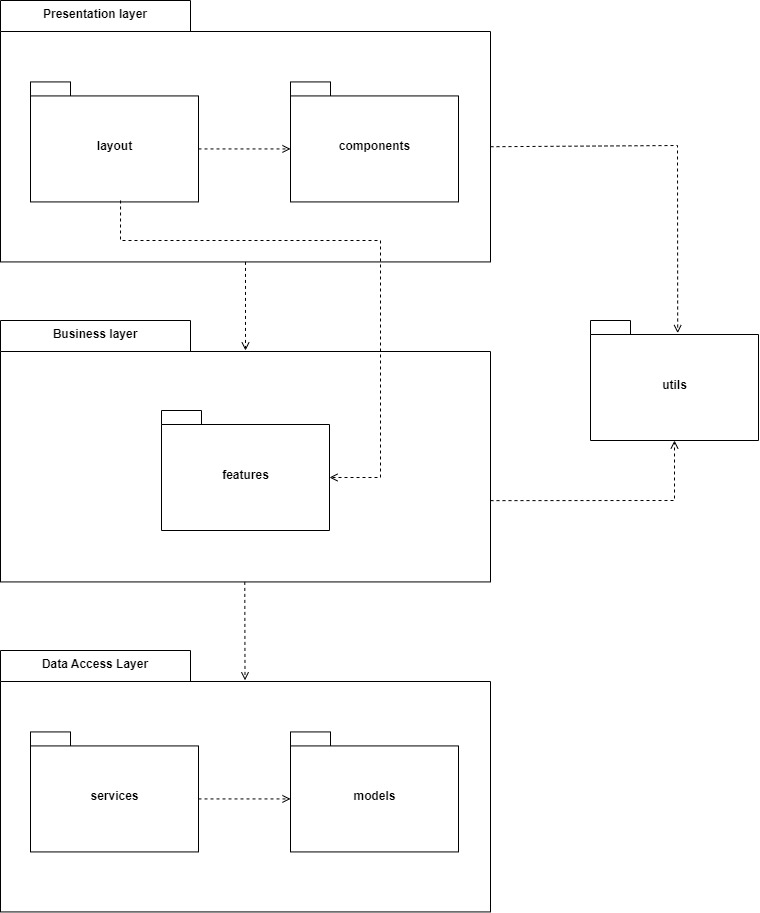
### 1.1 Back-end Code Packages



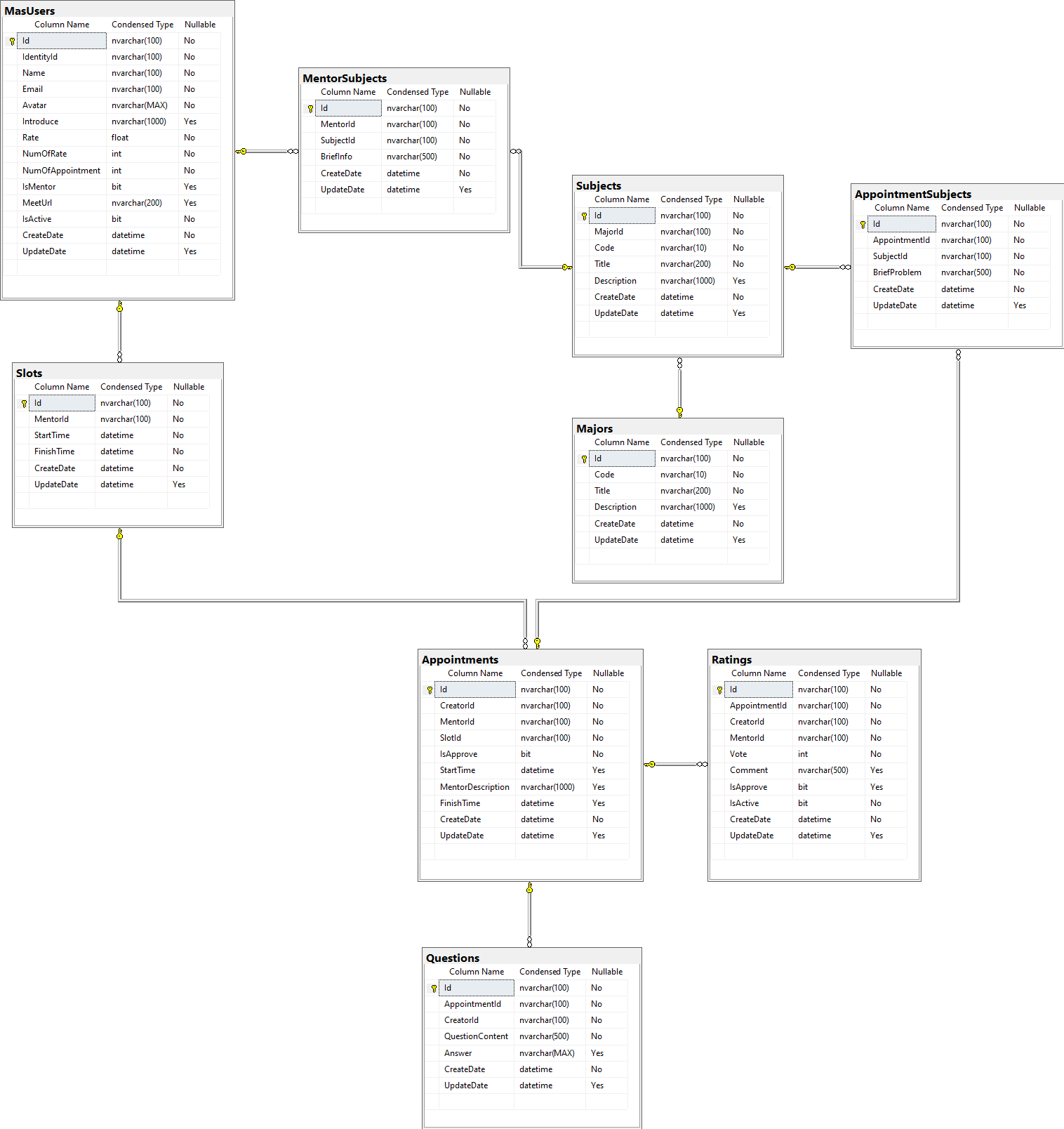
***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | .github\workflows | Contain config files (YAML files) used to work with GitHub Actions for CI/CD source code. |
| 02 | src | Contains the main source code of project API, include MAS.Api, MAS.Core and MAS.Infrastructure. |
| 03 | MAS.API | Contains the source code of the .NET 5 API, it is a Web API project. It has two main packages: Helpers and Controllers. |
| 04 | Helpers | Contains config files for the API such as service injection (declare the dependency injection into Dependency Inject Container of Asp.Net Core), config for swagger UI to easy testing API on Web browser, auto Seed Data file use to seed data to Database if it does not exist, constant file for the constant value of API. |
| 05 | Controllers | Contains controller files responsible for handling HTTP requests and returning responses with values. |
| 06 | MAS.Core | Contains source code of the Core Business of API, It is a Classlib project, including some main packages such as Parameters, Interfaces (contains Repository interfaces and Service interfaces), Data Transfer Object (Dtos), and Services (Implement level). |
| 07 | Entities | Contains all model classes for mapping with database tables. |
| 08 | Parameters | Contains all Parameters class files, it is used for query routes in API. |
| 09 | Exceptions | Contains files used for logging in Product Environment (Deploy in Azure). |
| 10 | Dtos | Contains Data transfer object classes used for data binding input object (Incoming package) and response mapping output object (Outcoming package). |
| 11 | Interfaces | Contains Interface classes of Repositories (which are working with Entity Framework Core and Database SQL Server to get data), and Services (which will be used by the API, they are same to the Repository Interfaces but they call and use the Repository Interfaces). |
| 12 | Services | Contains the Implement classes of the Interfaces/Services, It calls the Repository, catches the error from the input parameters from users, then passes these input parameters through the repository (Dependency Injection). |
| 13 | MAS.Infrastructure | Contains main files used for Entity Framework Core Object Relational Mapping with Database and the Repository (Implementation from the Repository Interfaces in Core layer). |
| 14 | Data | This package contains the AppDbContext class, where all entities or model classes use mapping with the table in the Database. |
| 15 | Entities | Contains all model classes for mapping with database tables. |
| 16 | Migrations | Contains migrations file, history design, and modified database. |
| 17 | Helpers | Contains files that help populate new model objects. |
| 18 | Repository | Contains files implementation of the Repository Interface in the Core layer. |
| 19 | tests | Contain test projects and test files. |

### 1.2 Font-end Code Packages



## 2. Database Schema



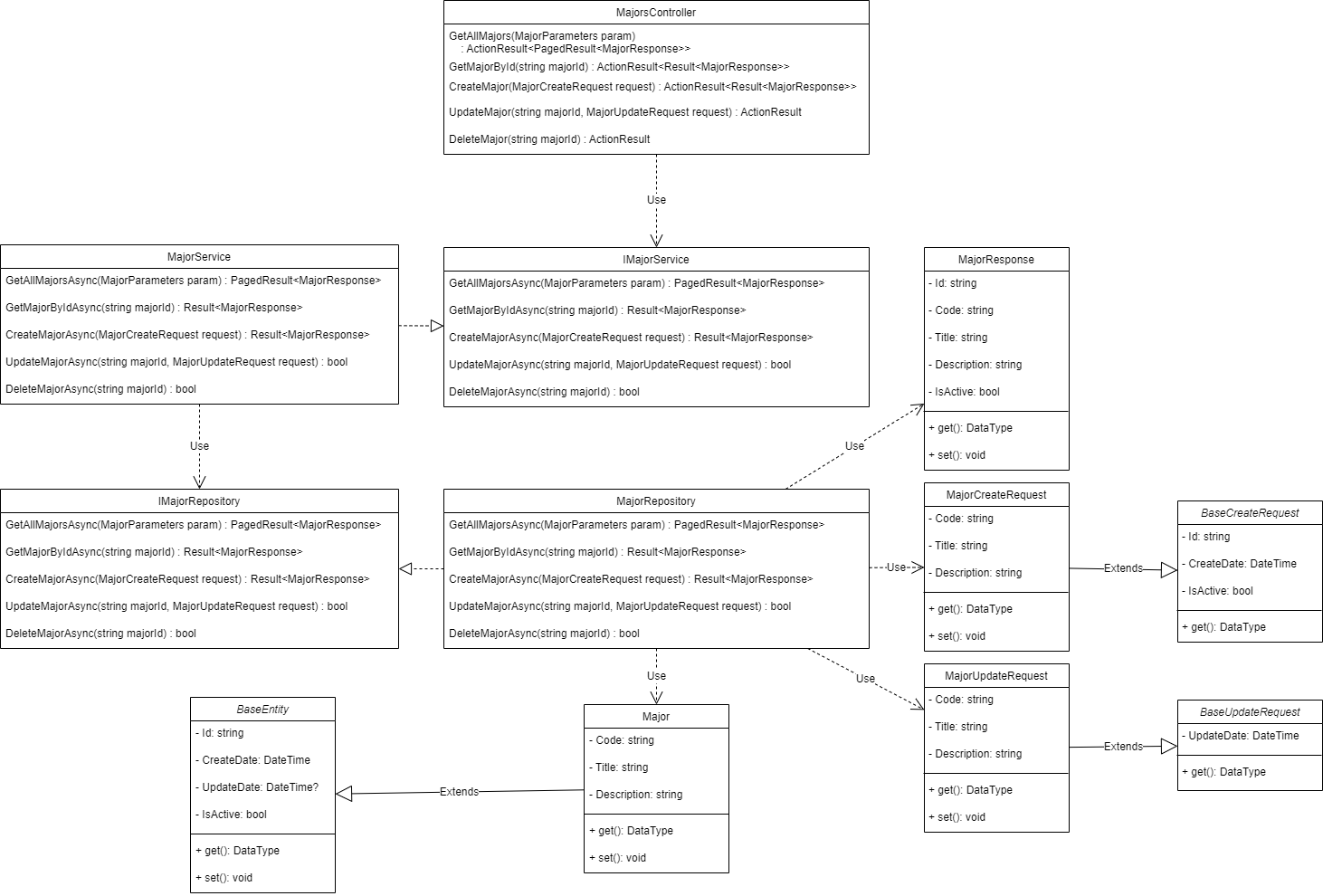
***Table descriptions & package class naming conventions are as below***

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 1 | pages | The package contains pages, responsible for grouping components together |
| 2 | components | The package contains components for specific use in a specific page. |
| 3 | utils | The package contains utility methods for common use. |
| 4 | features | The package contains classes used for handling business logic, managing state. |
| 5 | services | The package contains classes responsible for handling data between REST APIs and web application. |
| 6 | models | The package contains model classes for data binding. |

# II. Code Designs

## 1. Major Features

### a. Class Diagram



### b. Class Specifications

#### MajorsController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllMajors | Get all Majors in System, **Input** is *MajorParameters*, **Output** is *ActionResult<PagedResult<MajorResponse>>*, **Internal method processing** is *GetAllMajorsAsync* from *IMajorService* by **Inject** it into *MajorsController* |
| 02 | GetMajorById | Get a Major Info by its Id, **Input** is *majorId,* **Output** is *ActionResult<Result<MajorResponse>>*, **Internal method processing** is *GetMajorByIdAsync* from *IMajorService* by **Inject** it into *MajorsController* |
| 03 | CreateMajor | Create a Major or Add a Major into System, **Input** is *MajorCreateRequest,* **Output** is *ActionResult<Result<MajorResponse>>*, **Internal method processing** is *CreateMajorAsync* from *IMajorService* by **Inject** it into *MajorsController* |
| 04 | UpdateMajor | Update info of Major, **Input** is *majorId, MajorUpdateRequest*, **Output** is*ActionResult*, **Internal method processing** is *UpdateMajorAsync* from *IMajorService* by **Inject** it into *MajorsController* |
| 05 | DeleteMajor | Disable Major, **Input** is *majorId*, **Output** is *ActionResult*, **Internal method processing** is *DeleteMajorAsync* from *IMajorService* by **Inject** it into *MajorsController* |

#### IMajorService Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllMajorsAsync | Get all Majors in System, **Input** is *MajorParameters*, **Output** is *PagedResult<MajorResponse>* |
| 02 | GetMajorByIdAsync | Get a Major Info by its Id, **Input** is *majorId,* **Output** is *Result<MajorResponse>* |
| 03 | CreateMajorAsync | Create a Major or Add a Major into System, **Input** is *MajorCreateRequest,* **Output** is *Result<MajorResponse>* |
| 04 | UpdateMajorAsync | Update info of Major, **Input** is *majorId, MajorUpdateRequest*, **Output** is*bool* |
| 05 | DeleteMajorAsync | Disable Major, **Input** is *majorId*, **Output** is *bool* |

#### MajorService Class - Implement from IMajorService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllMajorsAsync | Get all Majors in System, **Input** is *MajorParameters*, **Output** is *PagedResult<MajorResponse>*, **Internal method processing** is *GetAllMajorsAsync* from *IMajorRepository* by **Inject** it into *MajorService* |
| 02 | GetMajorByIdAsync | Get a Major Info by its Id, **Input** is *majorId,* **Output** is *Result<MajorResponse>*, **Internal method processing** is *GetMajorByIdAsync* from *IMajorRepository* by **Inject** it into *MajorService* |
| 03 | CreateMajorAsync | Create a Major or Add a Major into System, **Input** is *MajorCreateRequest,* **Output** is *Result<MajorResponse>*, **Internal method processing** is *CreateMajorAsync* from *IMajorRepository* by **Inject** it into *MajorService* |
| 04 | UpdateMajorAsync | Update info of Major, **Input** is *majorId, MajorUpdateRequest*, **Output** is*bool*, **Internal method processing** is *UpdateMajorAsync* from *IMajorRepository* by **Inject** it into *MajorService* |
| 05 | DeleteMajorAsync | Disable Major, **Input** is *majorId*, **Output** is *bool*, **Internal method processing** is *DeleteMajorAsync* from *IMajorRepository* by **Inject** it into *MajorService* |

#### IMajorRepository Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllMajorsAsync | Get all Majors in System, **Input** is *MajorParameters*, **Output** is *PagedResult<MajorResponse>* |
| 02 | GetMajorByIdAsync | Get a Major Info by its Id, **Input** is *majorId,* **Output** is *Result<MajorResponse>* |
| 03 | CreateMajorAsync | Create a Major or Add a Major into System, **Input** is *MajorCreateRequest,* **Output** is *Result<MajorResponse>* |
| 04 | UpdateMajorAsync | Update info of Major, **Input** is *majorId, MajorUpdateRequest*, **Output** is*bool* |
| 05 | DeleteMajorAsync | Disable Major, **Input** is *majorId*, **Output** is *bool* |

#### MajorRepository Class - Implement from IMajorRepository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllMajorsAsync | Get all Majors in System, **Input** is *MajorParameters*, **Output** is *PagedResult<MajorResponse>* |
| 02 | GetMajorByIdAsync | Get a Major Info by its Id, **Input** is *majorId,* **Output** is *Result<MajorResponse>* |
| 03 | CreateMajorAsync | Create a Major or Add a Major into System, **Input** is *MajorCreateRequest,* **Output** is *Result<MajorResponse>* |
| 04 | UpdateMajorAsync | Update info of Major, **Input** is *majorId, MajorUpdateRequest*, **Output** is*bool* |
| 05 | DeleteMajorAsync | Disable Major, **Input** is *majorId*, **Output** is *bool* |

#### Major Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Major. |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| CreateDate | DateTime | private |  |
| UpdateDate | DateTime? | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### MajorResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Major. |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### MajorCreateRequest Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Major. Auto setting value. |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| CreateDate | DateTime | private | Auto setting value. |
| IsActive | bool | private | This info is active or not. Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### MajorUpdateRequest Class

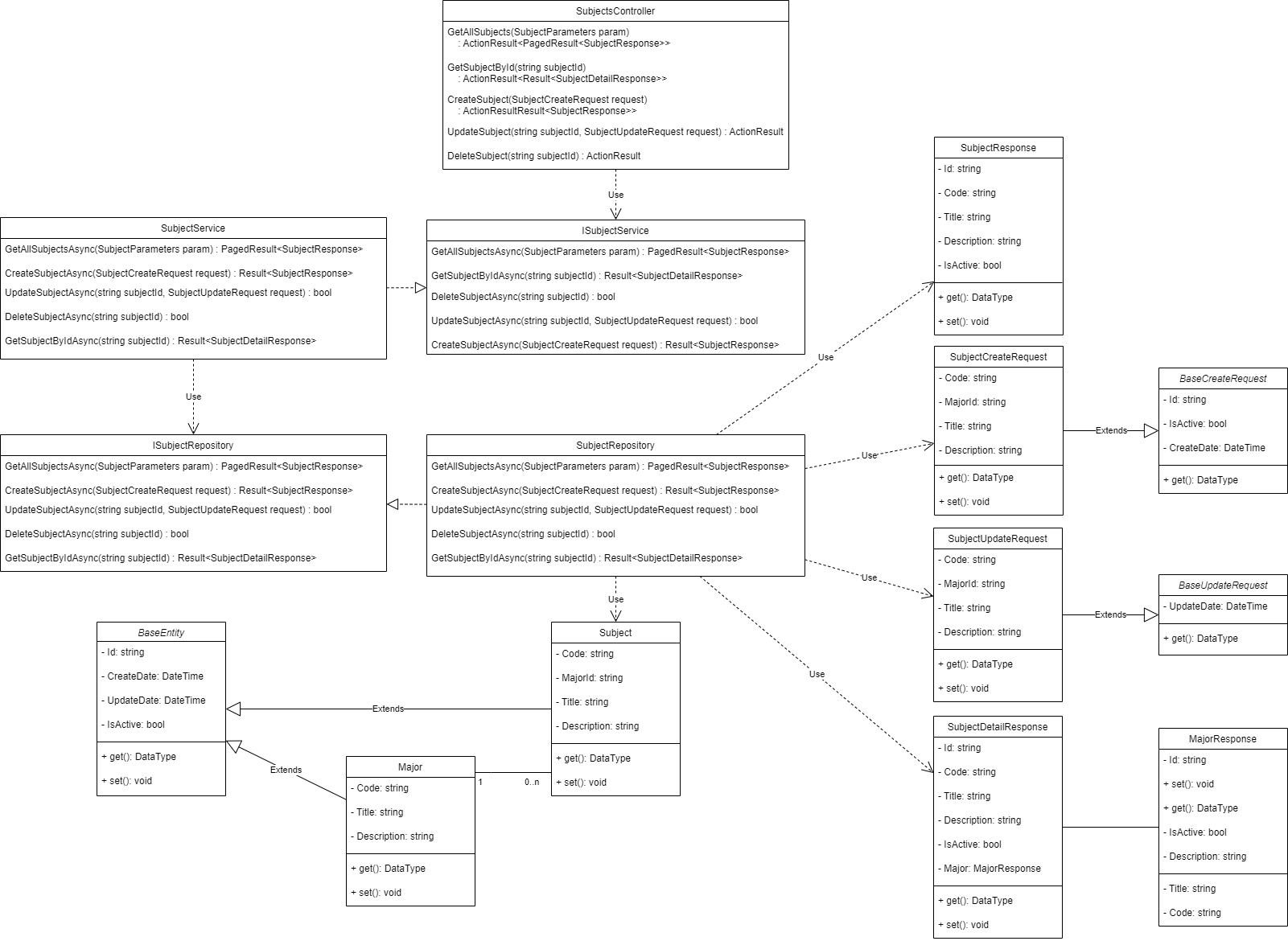
| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| UpdateDate | DateTime | private | Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

## 2. Subject Features

### a. Class Diagram



### b. Class Specifications

#### SubjectsController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjects | Get all Subjects in System, **Input** is *SubjectParameters*, **Output** is *ActionResult<PagedResult<SubjectResponse>>*, **Internal method processing** is *GetAllSubjectsAsync* from *ISubjectService* by **Inject** it into *SubjectsController* |
| 02 | GetSubjectById | Get a Subject Info by its Id, **Input** is *subjectId,* **Output** is *ActionResult<Result<SubjectDetailResponse>>*, **Internal method processing** is *GetSubjectByIdAsync* from *ISubjectService* by **Inject** it into *SubjectsController* |
| 03 | CreateSubject | Create or Add a Subject into System, **Input** is *SubjectCreateRequest,* **Output** is *ActionResult<Result<SubjectResponse>>*, **Internal method processing** is *CreateSubjectAsync* from *ISubjectService* by **Inject** it into *SubjectsController* |
| 04 | UpdateSubject | Update info of Subject, **Input** is *subjectId, SubjectUpdateRequest*, **Output** is*ActionResult*, **Internal method processing** is *UpdateSubjectAsync* from *ISubjectService* by **Inject** it into *SubjectsController* |
| 05 | DeleteMajor | Disable Subject, **Input** is *subjectId*, **Output** is *ActionResult*, **Internal method processing** is *DeleteSubjectAsync* from *ISubjectService* by **Inject** it into *SubjectsController* |

#### ISubjectService Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsAsync | Get all Subjects in System, **Input** is *SubjectParameters*, **Output** is *PagedResult<SubjectResponse>* |
| 02 | GetSubjectByIdAsync | Get a Subject Info by its Id, **Input** is *subjectId,* **Output** is *Result<SubjectDetailResponse>* |
| 03 | CreateSubjectAsync | Create or Add a Subject into System, **Input** is *SubjectCreateRequest,* **Output** is *Result<SubjectResponse>* |
| 04 | UpdateSubjectAsync | Update info of Subject, **Input** is *subjectId, SubjectUpdateRequest*, **Output** is*Result<bool>* |
| 05 | DeleteSubjectAsync | Disable Subject, **Input** is *subjectId*, **Output** is *Result<bool>* |

#### SubjectService Class - Implement from ISubjectService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsAsync | Get all Subjects in System, **Input** is *SubjectParameters*, **Output** is *PagedResult<SubjectResponse>*, **Internal method processing** is *GetAllSubjectsAsync* from *ISubjectRepository* by **Inject** it into *SubjectService* |
| 02 | GetSubjectByIdAsync | Get a Subject Info by its Id, **Input** is *subjectId,* **Output** is *Result<SubjectDetailResponse>*, **Internal method processing** is *GetSubjectByIdAsync* from *ISubjectRepository* by **Inject** it into *SubjectService* |
| 03 | CreateSubjectAsync | Create or Add a Subject into System, **Input** is *SubjectCreateRequest,* **Output** is *Result<SubjectResponse>*, **Internal method processing** is *CreateSubjectAsync* from *ISubjectRepository* by **Inject** it into *SubjectService* |
| 04 | UpdateSubjectAsync | Update info of Subject, **Input** is *subjectId, SubjectUpdateRequest*, **Output** is*Result<bool>*, **Internal method processing** is *UpdateSubjectAsync* from *ISubjectRepository* by **Inject** it into *SubjectService* |
| 05 | DeleteSubjectAsync | Disable Subject, **Input** is *subjectId*, **Output** is *Result<bool>*, **Internal method processing** is *DeleteSubjectAsync* from *ISubjectRepository* by **Inject** it into *SubjectService* |

#### ISubjectRepository Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsAsync | Get all Subjects in System, **Input** is *SubjectParameters*, **Output** is *PagedResult<SubjectResponse>* |
| 02 | GetSubjectByIdAsync | Get a Subject Info by its Id, **Input** is *subjectId,* **Output** is *Result<SubjectDetailResponse>* |
| 03 | CreateSubjectAsync | Create or Add a Subject into System, **Input** is *SubjectCreateRequest,* **Output** is *Result<SubjectResponse>* |
| 04 | UpdateSubjectAsync | Update info of Subject, **Input** is *subjectId, SubjectUpdateRequest*, **Output** is*Result<bool>* |
| 05 | DeleteSubjectAsync | Disable Subject, **Input** is *subjectId*, **Output** is *Result<bool>* |

#### SubjectRepository Class - Implement from ISubjectRepository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsAsync | Get all Subjects in System, **Input** is *SubjectParameters*, **Output** is *PagedResult<SubjectResponse>* |
| 02 | GetSubjectByIdAsync | Get a Subject Info by its Id, **Input** is *subjectId,* **Output** is *Result<SubjectDetailResponse>* |
| 03 | CreateSubjectAsync | Create or Add a Subject into System, **Input** is *SubjectCreateRequest,* **Output** is *Result<SubjectResponse>* |
| 04 | UpdateSubjectAsync | Update info of Subject, **Input** is *subjectId, SubjectUpdateRequest*, **Output** is*Result<bool>* |
| 05 | DeleteSubjectAsync | Disable Subject, **Input** is *subjectId*, **Output** is *Result<bool>* |

#### Subject Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Subject. |
| Code | string | private |  |
| MajorId | string | private | Unique identifier of Major. |
| Title | string | private |  |
| Description | string | private |  |
| CreateDate | DateTime | private |  |
| UpdateDate | DateTime? | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### 

#### SubjectResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Subject. |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### SubjectDetailResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Subject. |
| Code | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| IsActive | bool | private | This info is active or not. |
| Major | MajorResponse | private | This info has been describe in *<Major Features>* |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### SubjectCreateRequest Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Subject. Auto setting value. |
| Code | string | private |  |
| MajorId | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| CreateDate | DateTime | private | Auto setting value. |
| IsActive | bool | private | This info is active or not. Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### SubjectUpdateRequest Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Code | string | private |  |
| MajorId | string | private |  |
| Title | string | private |  |
| Description | string | private |  |
| UpdateDate | DateTime | private | Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

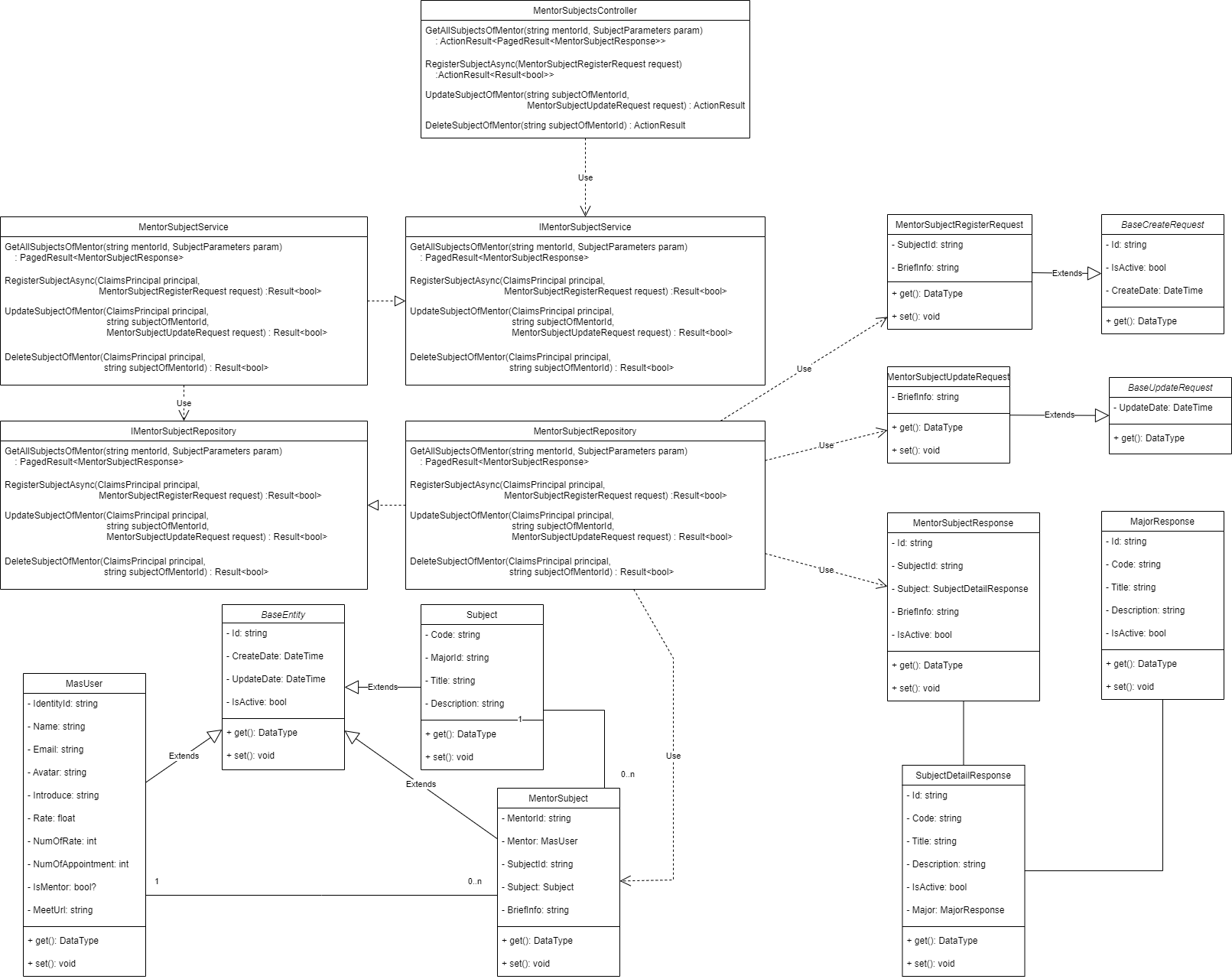
## 3. MasUser Features

### a. Class Diagram

### b. Class Specifications

## 4. MentorSubject Features

### a. Class Diagram



### b. Class Specifications

#### MentorSubjectsController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsOfMentor | Get all Subjects of a Mentor in System, **Input** is *mentorId, MentorSubjectParameters*, **Output** is *ActionResult<PagedResult<MentorSubjectResponse>>*, **Internal method processing** is *GetAllSubjectsOfMentorAsync* from *IMentorSubjectService* by **Inject** it into *MentorSubjectsController* |
| 02 | RegisterSubject | Register Subject into Mentor Profile, **Input** is *MentorSubjectRegisterRequest,* **Output** is *ActionResult<Result<bool>>*, **Internal method processing** is *RegisterSubjectAsync* from *IMentorSubjectService* by **Inject** it into *MentorSubjectsController* |
| 03 | UpdateSubjectOfMentor | Update Subject info of Mentor Profile, **Input** is *subjectOfMentorId*, *MentorSubjectRegisterRequest*, **Output** is *ActionResult*, **Internal method processing** is *UpdateSubjectOfMentorAsync* from *IMentorSubjectService*  by Inject it into *MentorSubjectsController* |
| 04 | DeleteSubjectOfMentor | Disable Subject from Mentor Profile, **Input** is *subjectOfMentorId*, **Output** is *ActionResult*, **Internal method processing** is *DeleteSubjectOfMentorAsync* from *IMentorSubjectService*  by Inject it into *MentorSubjectsController* |

#### IMentorSubjectService Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsOfMentorAsync | Get all Subjects of a Mentor in System, **Input** is *mentorId, MentorSubjectParameters*, **Output** is *PagedResult<MentorSubjectResponse>* |
| 02 | RegisterSubjectAsync | Register Subject into Mentor Profile, **Input** is *MentorSubjectRegisterRequest,* **Output** is *Result<bool>* |
| 03 | UpdateSubjectOfMentorAsync | Update Subject info of Mentor Profile, **Input** is *subjectOfMentorId*, *MentorSubjectRegisterRequest*, **Output** is *Result<bool>* |
| 04 | DeleteSubjectOfMentorAsync | Disable Subject from Mentor Profile, **Input** is *subjectOfMentorId*, **Output** is *Result<bool>* |

#### MentorSubjectService Class - Implement from IMentorSubjectService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsOfMentorAsync | Get all Subjects of a Mentor in System, **Input** is *mentorId, MentorSubjectParameters*, **Output** is *PagedResult<MentorSubjectResponse>*, **Internal method processing** is *GetAllSubjectsOfMentorAsync* from *IMentorSubjectRepository* by **Inject** it into *MentorSubjectService* |
| 02 | RegisterSubjectAsync | Register Subject into Mentor Profile, **Input** is *MentorSubjectRegisterRequest,* **Output** is *Result<bool>*, **Internal method processing** is *RegisterSubjectAsync* from *IMentorSubjectRepository* by **Inject** it into *MentorSubjectService* |
| 03 | UpdateSubjectOfMentorAsync | Update Subject info of Mentor Profile, **Input** is *subjectOfMentorId*, *MentorSubjectRegisterRequest*, **Output** is *Result<bool>*, **Internal method processing** is *UpdateSubjectOfMentorAsync* from *IMentorSubjectRepository* by **Inject** it into *MentorSubjectService* |
| 04 | DeleteSubjectOfMentorAsync | Disable Subject from Mentor Profile, **Input** is *subjectOfMentorId*, **Output** is *Result<bool>*, **Internal method processing** is *DeleteSubjectOfMentorAsync* from *IMentorSubjectRepository* by **Inject** it into *MentorSubjectService* |

#### IMentorSubjectRepository Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsOfMentorAsync | Get all Subjects of a Mentor in System, **Input** is *mentorId, MentorSubjectParameters*, **Output** is *PagedResult<MentorSubjectResponse>* |
| 02 | RegisterSubjectAsync | Register Subject into Mentor Profile, **Input** is *MentorSubjectRegisterRequest,* **Output** is *Result<bool>* |
| 03 | UpdateSubjectOfMentorAsync | Update Subject info of Mentor Profile, **Input** is *subjectOfMentorId*, *MentorSubjectRegisterRequest*, **Output** is *Result<bool>* |
| 04 | DeleteSubjectOfMentorAsync | Disable Subject from Mentor Profile, **Input** is *subjectOfMentorId*, **Output** is *Result<bool>* |

#### MentorSubjectRepository Class - Implement from IMentorSubjectRepository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllSubjectsOfMentorAsync | Get all Subjects of a Mentor in System, **Input** is *mentorId, MentorSubjectParameters*, **Output** is *PagedResult<MentorSubjectResponse>* |
| 02 | RegisterSubjectAsync | Register Subject into Mentor Profile, **Input** is *MentorSubjectRegisterRequest,* **Output** is *Result<bool>* |
| 03 | UpdateSubjectOfMentorAsync | Update Subject info of Mentor Profile, **Input** is *subjectOfMentorId*, *MentorSubjectRegisterRequest*, **Output** is *Result<bool>* |
| 04 | DeleteSubjectOfMentorAsync | Disable Subject from Mentor Profile, **Input** is *subjectOfMentorId*, **Output** is *Result<bool>* |

#### MentorSubjectResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a MentorSubject. |
| SubjectId | string | private | Unique identifier of Subject. |
| Subject | SubjectDetailResponse | private | Describe in *<Subject Features>* |
| BriefInfo | string | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### MentorSubjectRegisterRequest Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a MentorSubject. Auto setting value. |
| SubjectId | string | private | Unique identifier of a Subject. |
| BriefInfo | string | private |  |
| CreateDate | DateTime | private | Auto setting value. |
| IsActive | bool | private | This info is active or not. Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### MentorSubjectUpdateRequest Class

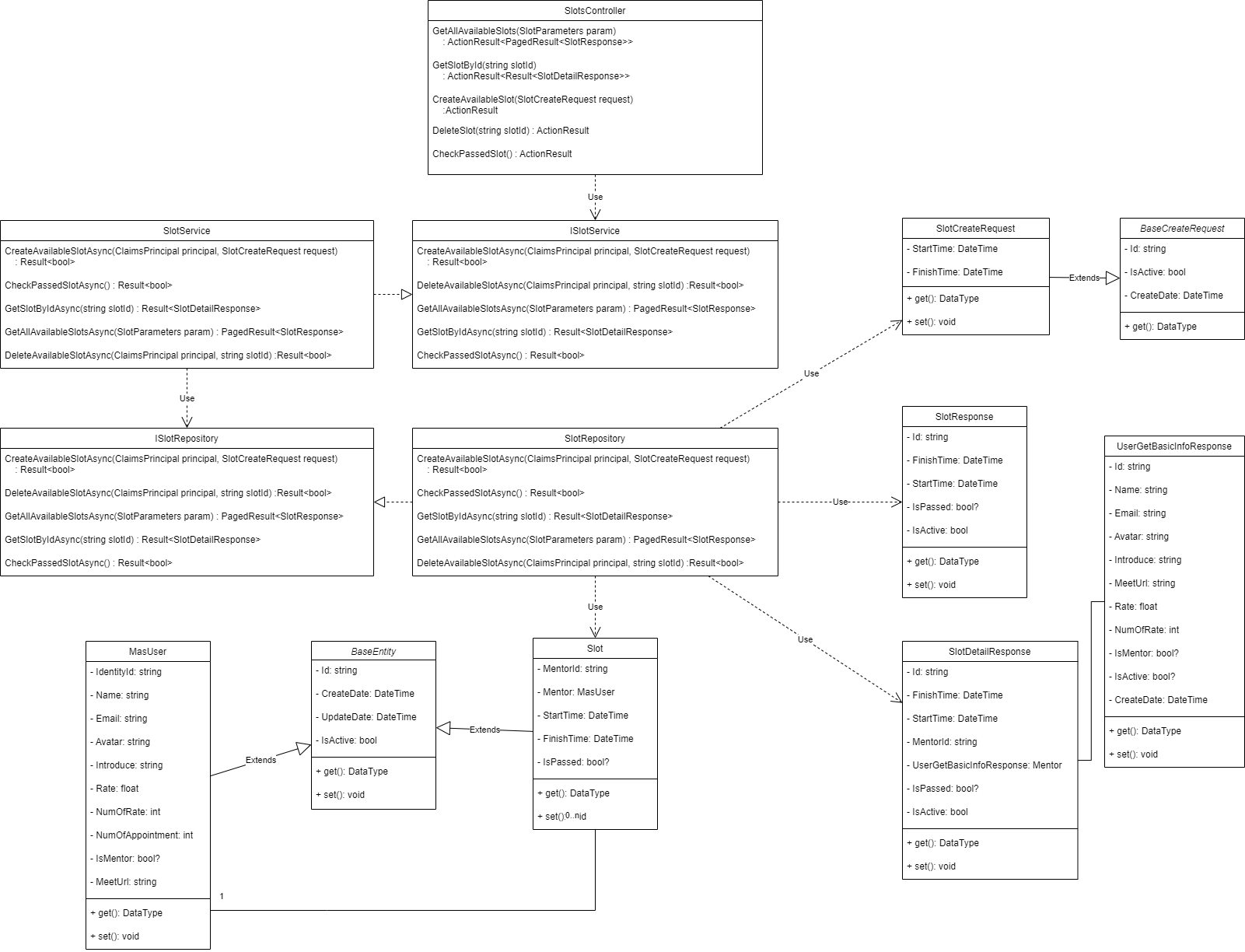
| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| BriefInfo | string | private |  |
| UpdateDate | DateTime | private | Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

## 5. Slot Features

### a. Class Diagram



### b. Class Specifications

#### SlotsController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllAvailableSlots | Get all available Slots, **Input** is *SlotParameters*, **Output** is *ActionResult<PagedResult<SlotResponse>>*, **Internal method processing** is *GetAllAvailableSlotsAsync* from *ISlotService* by **Inject** it into *SlotsController* |
| 02 | GetSlotById | Get a Slot Info by its Id, **Input** is *slotId,* **Output** is *ActionResult<Result<SlotDetailResponse>>*, **Internal method processing** is *GetSlotByIdAsync* from *ISlotService* by **Inject** it into *SlotsController* |
| 03 | CreateAvailableSlot | Create or Add a Slot, **Input** is *SlotCreateRequest,* **Output** is *ActionResult*, **Internal method processing** is *CreateAvailableSlotAsync* from *ISlotService* by **Inject** it into *SlotsController* |
| 04 | DeleteSlot | Delete or disable a Slot, **Input** is *slotId*, **Output** is*ActionResult*, **Internal method processing** is *DeleteAvailableSlotAsync* from *ISlotService* by **Inject** it into *SlotsController* |
| 05 | CheckPassedSlot | Check which Slot is passed time, none **Input** , **Output** is *ActionResult*, **Internal method processing** is *CheckPassedSlotAsync* from *ISlotService* by **Inject** it into *SlotsController* |

#### ISlotService Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllAvailableSlotsAsync | Get all available Slots, **Input** is *SlotParameters*, **Output** is *PagedResult<SlotResponse>* |
| 02 | GetSlotByIdAsync | Get a Slot Info by its Id, **Input** is *slotId,* **Output** is *Result<SlotDetailResponse>* |
| 03 | CreateAvailableSlotAsync | Create or Add a Slot, **Input** is *SlotCreateRequest,* **Output** is *Result<bool>* |
| 04 | DeleteSlotAsync | Delete or disable a Slot, **Input** is *slotId*, **Output** is*Result<bool>* |
| 05 | CheckPassedSlotAsync | Check which Slot is passed time, none **Input** , **Output** is *Result<bool>* |

#### SlotService Class - Implement from ISlotService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllAvailableSlotsAsync | Get all available Slots, **Input** is *SlotParameters*, **Output** is *PagedResult<SlotResponse>*, **Internal method processing** is *GetAllAvailableSlotsAsync* from *ISlotRepository* by **Inject** it into *SlotService* |
| 02 | GetSlotByIdAsync | Get a Slot Info by its Id, **Input** is *slotId,* **Output** is *Result<SlotDetailResponse>*, **Internal method processing** is *GetSlotByIdAsync* from *ISlotRepository* by **Inject** it into *SlotService* |
| 03 | CreateAvailableSlotAsync | Create or Add a Slot, **Input** is *SlotCreateRequest,* **Output** is *Result<bool>*, **Internal method processing** is *CreateAvailableSlotAsync* from *ISlotRepository* by **Inject** it into *SlotService* |
| 04 | DeleteSlotAsync | Delete or disable a Slot, **Input** is *slotId*, **Output** is*Result<bool>*, **Internal method processing** is *DeleteSlotAsync* from *ISlotRepository* by **Inject** it into *SlotService* |
| 05 | CheckPassedSlotAsync | Check which Slot is passed time, none **Input** , **Output** is *Result<bool>*, **Internal method processing** is *CheckPassedSlotAsync* from *ISlotRepository* by **Inject** it into *SlotService* |

#### ISlotRepository Interface

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllAvailableSlotsAsync | Get all available Slots, **Input** is *SlotParameters*, **Output** is *PagedResult<SlotResponse>* |
| 02 | GetSlotByIdAsync | Get a Slot Info by its Id, **Input** is *slotId,* **Output** is *Result<SlotDetailResponse>* |
| 03 | CreateAvailableSlotAsync | Create or Add a Slot, **Input** is *SlotCreateRequest,* **Output** is *Result<bool>* |
| 04 | DeleteSlotAsync | Delete or disable a Slot, **Input** is *slotId*, **Output** is*Result<bool>* |
| 05 | CheckPassedSlotAsync | Check which Slot is passed time, none **Input** , **Output** is *Result<bool>* |

#### SlotRepository Class - Implement from ISlotRepository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | GetAllAvailableSlotsAsync | Get all available Slots, **Input** is *SlotParameters*, **Output** is *PagedResult<SlotResponse>* |
| 02 | GetSlotByIdAsync | Get a Slot Info by its Id, **Input** is *slotId,* **Output** is *Result<SlotDetailResponse>* |
| 03 | CreateAvailableSlotAsync | Create or Add a Slot, **Input** is *SlotCreateRequest,* **Output** is *Result<bool>* |
| 04 | DeleteSlotAsync | Delete or disable a Slot, **Input** is *slotId*, **Output** is*Result<bool>* |
| 05 | CheckPassedSlotAsync | Check which Slot is passed time, none **Input** , **Output** is *Result<bool>* |

#### Slot Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Slot. |
| MentorId | string | private | Unique identifier of Mentor. |
| StartTime | DateTime | private |  |
| FinishTime | DateTime | private |  |
| IsPassed | bool? | private |  |
| CreateDate | DateTime | private |  |
| UpdateDate | DateTime? | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### 

#### SlotResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Slot. |
| StartTime | DateTime | private |  |
| FinishTime | DateTime | private |  |
| IsPassed | bool? | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### SlotDetailResponse Class

| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Slot. |
| StartTime | DateTime | private |  |
| FinishTime | DateTime | private |  |
| MentorId | string | private | Unique identifier of Mentor (MasUser). |
| Mentor | UserGetBasicInfoResponse | private | Describe in *<MasUser Features>* |
| IsPassed | bool? | private |  |
| IsActive | bool | private | This info is active or not. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

#### SlotCreateRequest Class

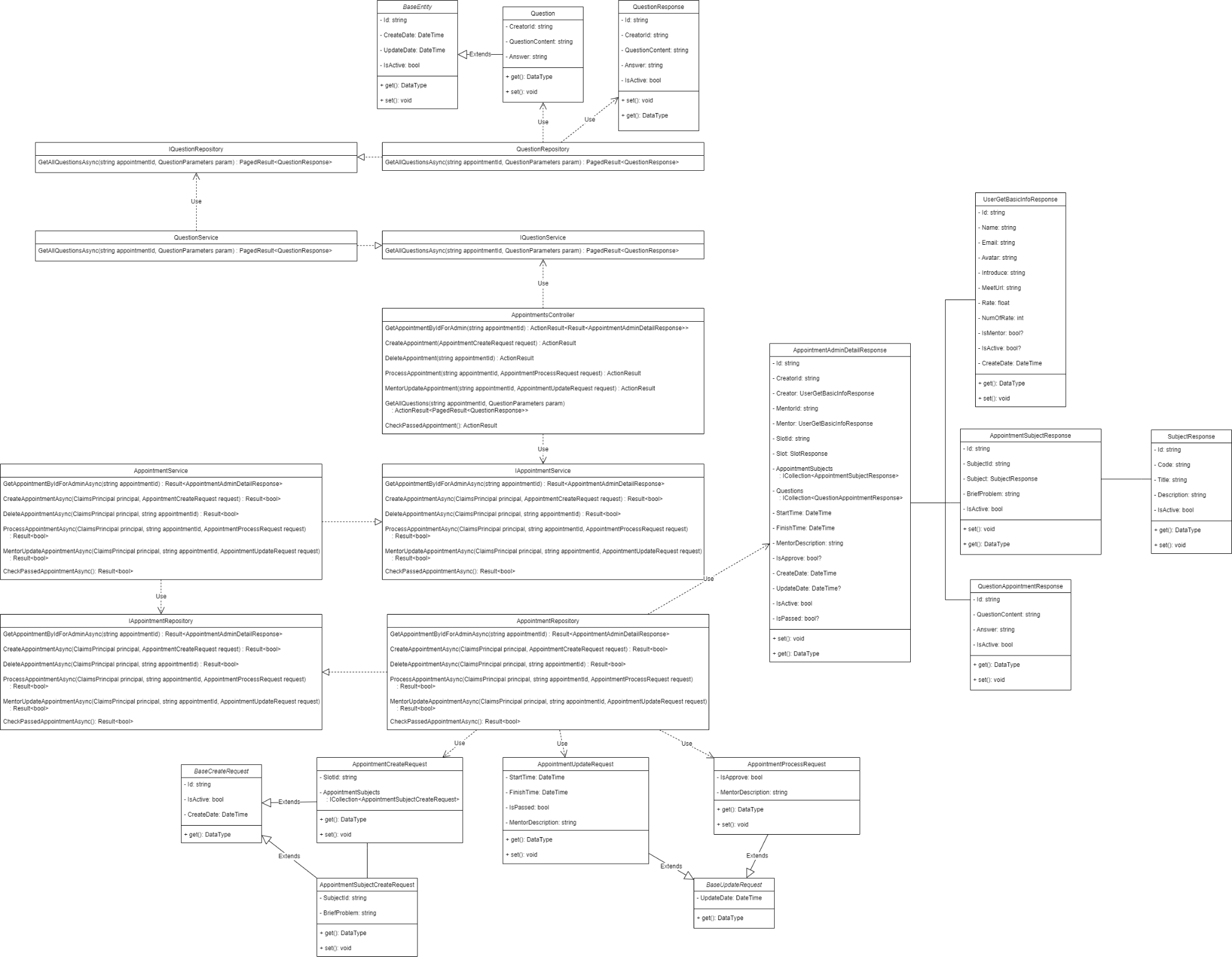
| **Attribute** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Id | string | private | Unique identifier of a Slot. Auto setting value. |
| StartTime | DateTime | private |  |
| FinishTime | DateTime | private |  |
| CreateDate | DateTime | private | Auto setting value. |
| IsActive | bool | private | This info is active or not. Auto setting value. |

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getter | Get attribute value |
| 02 | setter | Set attribute value |

## 6. Appointment Features

### a. Class Diagram



### b. Class Specifications

## 7. Question Features

### a. Class Diagram

### b. Class Specifications

## 8. Rating

### a. Class Diagram

### b. Class Specifications

# III. Database Tables

## 1. Majors

Contains Major data from FPT University.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| Code | nvarchar | 10 | No | No |  |  |
| Title | nvarchar | 200 | No | No |  |  |
| Description | nvarchar | 1000 | No | Yes |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 2. Subjects

Contain Subjects data in Major.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| MajorId | nvarchar | 100 | Yes | No | FK |  |
| Code | nvarchar | 10 | No | No |  |  |
| Title | nvarchar | 200 | No | No |  |  |
| Description | nvarchar | 1000 | No | Yes |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 3. MasUsers

Contain MAS Users data, including Student data and Mentor data, distinguish by IsMentor attribute

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| IdentityId | nvarchar | 100 | Yes | No |  | tbl.AspNetUsers primary key. |
| Name | nvarchar | 100 | No | No |  |  |
| Email | nvarchar | 100 | Yes | No |  |  |
| Avatar | nvarchar | MAX | Yes | No |  |  |
| Introduce | nvarchar | 1000 | No | Yes |  |  |
| Rate | float |  | No | No |  |  |
| NumOfRate | int |  | No | No |  |  |
| NumOfAppointment | int |  | No | No |  |  |
| IsMentor | bit |  | No | Yes |  |  |
| MeetUrl | nvarchar | 200 | No | Yes |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 4. MentorSubjects

Contain Subjects data of Mentor.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| MentorId | nvarchar | 100 | Yes | No | FK |  |
| SubjectId | nvarchar | 10 | No | No | FK |  |
| BriefInfo | nvarchar | 500 | No | No |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 

## 5. Slots

Contain Subjects data in Major.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| MentorId | nvarchar | 100 | Yes | No | FK |  |
| StartTime | datetime |  | No | No |  |  |
| FinishTime | datetime |  | No | No |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 

## 6. Appointments

Contain Appointment Data registered by Student to Mentor based on available slot of Mentor.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| CreatorId | nvarchar | 100 | Yes | No |  |  |
| MentorId | nvarchar | 100 | Yes | No |  |  |
| SlotId | nvarchar | 100 | Yes | No | FK |  |
| IsApproved | bit |  | No | Yes |  |  |
| StartTime | datetime |  | No | Yes |  |  |
| MentorDescription | nvarchar | 1000 | No | Yes |  |  |
| FinishTime | datetime |  | No | No |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 

## 7. Questions

Contain Question data in Appointment.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| AppointmentId | nvarchar | 100 | Yes | No | FK |  |
| CreatorId | nvarchar | 100 | Yes | No |  |  |
| QuestionContent | nvarchar | 500 | No | No |  |  |
| Answer | nvarchar | MAX | No | Yes |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 

## 8. Ratings

Contain Rating data on Appointment.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| AppointmentId | nvarchar | 100 | Yes | No | FK |  |
| CreatorId | nvarchar | 100 | Yes | No |  |  |
| MentorId | nvarchar | 100 | Yes | No |  |  |
| Vote | int |  | No | No |  |  |
| Comment | nvarchar | 500 | No | Yes |  |  |
| IsApprove | bit |  | No | Yes |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 

## 9. AppointmentSubjects

Contain Appointment detail, including the Subjects of this Appointment, a brief problem summary.

| **Field name** | **Type** | **Size** | **Unique** | **Allow Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nvarchar | 100 | Yes | No | PK |  |
| AppointmentId | nvarchar | 100 | Yes | No | FK |  |
| SubjectId | nvarchar | 100 | Yes | No | FK |  |
| BriefProblem | nvarchar | 500 | Yes | No |  |  |
| CreatedDate | datetime |  | No | No |  |  |
| UpdateDate | datetime |  | No | Yes |  |  |
| IsActive | bit |  | No | No |  | This data is activated or not. |

## 