**Предметная область:**

автоматизация юридического сопровождения судебного производства гражданско-правовых дел.

Функционал позволяется вести судебные дела, а именно актуализировать информацию о ходе судебного дела, добавлять и изменять стороны по делу, добавлять и изменять документы по делу, ставить и закрывать задачи по делу.

Дело ведет юрист. Юрист может выступать в деле под разными ролями: куратор, ответственный юрист по делу, помощник юриста по делу, просто юрист.

Дело обладает необходимым минимальным достаточным количеством полей для отображения основной информации по делу.

Ход информации по делу отображается в виде ряда завешенных и плановых задач. Результат задачи: новое событие в деле.

В деле имеется ряд входящей и исходящей документации, досье по делу.

Дела привязываются к проекту - финансовой организации и формируют портфель проекта.

По каждому проекту предполагается статистическая отчетность, отчетность установленного вида, средства управления делами в контексте распределения дел и задач между юристами.

Подразумевается подключение системы ко внешним источникам (сайты судов, ФССП и тд) для обмена документами, мониторинга информации.

**Домен бизнес-логики:**

Реализуется минимально необходимый и достаточный для работы блок ведения судебных дел.

Ключевые сущности: судебное дело, документ по делу, сторона по делу, юрист, ведущий дело, задача по делу.

Диаграмма сущностей содержится в приложении (ER-диаграмма).

Юрист может вести одно или несколько дел в качестве главного юриста или помощника.

Юрист закрывает задачи в деле, каждая задача отнесена к делу.

Доле может содержать одну или несколько сторон по делу.

Дело может содержать одно или несколько документов по делу.

Каждая сущность оперирует различным списком справочников, которые хранятся в единой таблице, объединяясь в группы.

**Краткое описание архитектуры приложения:**

Web-приложение MPA: серверное приложение - .NetCore C# с RestAPI,

Предполагаема СУБД- MSSQL Srever, клиентское приложение – Angular.

Так как пользователь может работать с несколькими делами одновременно в разных окнах, все формы в масштабируемом контексте имеют различную параметризующую модель, особенную логику в контексте роли пользователя, а также различный контекст наполнения данными, то использование SPA не целесообразно.

В данном задании реализуется серверная часть приложения. За основу взят паттерн MVC.

Реализуется слой работы с данными, где хранится классы сущностей, модели (если необходимо), DTO, репозитарии - DAL.

Также обычно выделяется слой для хранения аналитической части и хранении бизнес логики – SL, но в данном случае эта логика хранится в папке сервисов основного проекта приложения, так как не было необходимости в выделении отдельного слоя. При более тяжелой логике слой бы был выделен.

И реализован основной слой для работы с представлениями, получаемыми с клиента через REST API, где происходит предобработка данных с клиента, обмен данными с клиентом, содержатся контроллеры и классы по работы с ними.

Работа с репозиториями и entity осуществляется через unitofwork.