**3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

В данном разделе описывается функционирование и структура разрабатываемого программного средства.

Взаимоотношения между классами разрабатываемого программного обеспечения приведены на диаграмме классов ГУИР.400201.111 РР.1.

**3.1 Описание модели данных**

**3.1.1 Таблица Messages**

Данная таблица служит для хранения данных о сообщениях в диалогах.

Поля таблицы Messages:

* id – первичный ключ;
* SenderId – идентификатор отправителя;
* SenderUsername – имя отправителя;
* Recipientld – идентификатор получателя;
* RecipientUsername – имя получателя;
* Context – текст;
* DateRead – дата прочтения;
* MessageSent – сообщение отправлено;
* SenderDeleted – отправитель удален;
* RecipientDeleted – пользователь удален.

Колонки id, SenderId, Recipientld имеют тип «integer». Колонки SenderUsername, RecipientUsername, Context имеют тип «text». Колонки DateRead, MessageSent имеют тип timestamp и заполняется автоматически. Колонки SenderDeleted, RecipientDeleted имеют тип «boolean».

**3.1.2 Таблица Groups**

Данная таблица служит для хранения информации о группах пользователей.

Поля таблицы Groups:

* Name – название группы.

Колонка Name имеет тип «text».

**3.1.3 Таблица Connections**

Данная таблица хранит информацию о подключившихся пользователях.

Поля таблицы Connections:

* ConnectionId – идентификатор подключения;
* Username – имя пользователя;
* GroupName – название группы.

Колонки ConnectionId, Username, GroupName имеют тип «text».

**3.1.4 Таблица AspNetUsers**

Данная таблица служит для хранения данных о сообщениях в диалогах.

Поля таблицы AspNetUsers:

* Id – первичный ключ;
* UserName – имя;
* NormalizedUserName – нормализованное имя пользователся;
* Email – почта;
* NormalizedEmail – нормализованная почта;
* EmailConfirmed – подтвержденная почта;
* PasswordHash – хэш пароля;
* SecurityStamp – печать безопасности;
* ConcurrencyStamp – печать параллельности;
* PhoneNumber – Номер телефона;
* PhoneNumberConfirmed –Подтвержденный номер телефона;
* TwoFactorEnabled – двухфакторный режим;
* LockoutEnd – блокировка отключена;
* LockoutEnabled – блокировка включена;
* AccessFailedCount – Счетчик неудачных попыток доступа;
* KnownAs – известность;
* Created – создатель;
* LastActive – последний раз активен;
* Gender – пол;
* Introduction – описание;
* Interests – интересы;
* City – город;
* Country – страна.

Колонки Id, AccessFailedCountимеют тип «integer». Колонки PasswordHash, SecurityStamp, ConcurrencyStamp, PhoneNumber, KnownAs, Gender, Introduction, LookingFor, Interests, City, Country имеют тип «text». Колонки LockoutEnd, DateOfBirth, Created, LastActive имеют тип timestamp и заполняется автоматически. Колонки EmailConfirmed, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnabled имеют тип «boolean». Колонки UserName, NormalizedEmail, Email, NormalizedEmail имеют тип «varchar».

**3.1.5 Таблица AspNetUserTokens**

Данная таблица хранит информацию о токенах пользователя.

Поля таблицы AspNetUserTokens:

* UserId – идентификатор пользователя;
* LoginProvider – логин пользователя;
* Name – имя;
* Value – значение.

Колонка UserId имеет тип «integer». Колонки LoginProvider, Name, Value имеют тип «text».

**3.1.6 Таблица AspNetUserRoles**

Данная таблица хранит информацию о ролях пользователя.

Поля таблицы AspNetUserRoles:

* UserId – идентификатор пользователя;
* RoleId – логин пользователя.

Колонки UserId, RoleId имеют тип «integer».

**3.1.7 Таблица AspNetUserLogins**

Данная таблица хранит информацию о ролях пользователя.

Поля таблицы AspNetUserLogins:

* UserId – идентификатор пользователя;
* ProviderKey – идентификатор отправителя;
* ProviderDisplayName – отображаемое имя отправителя;
* LoginProvider – логин отправителя.

Колонки ProviderKey, ProviderDisplayName, LoginProvider имеют тип «text». Колонка UserId имеет тип «integer».

**3.1.8 Таблица AspNetUserClaims**

Данная таблица хранит информацию о претензиях пользователя.

Поля таблицы AspNetUserClaims:

* Id – первичный ключ;
* UserId – идентификатор пользователя;
* ClaimType – тип претензии;
* ClaimValue – обоснование претензии.

Колонки Id, UserId имеет тип «integer». Колонки ClaimType, ClaimValue имеют тип «text».

**3.1.9 Таблица AspNetRoles**

Данная таблица хранит информацию о ролях.

Поля таблицы AspNetRoles:

* Id – первичный ключ;
* Name – имя пользователя;
* NormalizedName – настоящее имя;
* ConcurrencyStamp – печать параллельности.

Колонка Id имеет тип «integer». Колонки Name, NormalizedName имеют тип «varchar». Колонка ConcurrencyStamp имеет тип «text».

**3.1.10 Таблица AspNetRolesClaims**

Данная таблица является связующей для таблиц AspNetRoles и AspNetUserClaims.

Поля таблицы AspNetRolesClaims:

* Id – первичный ключ;
* RoleId – идентификатор роли;
* ClaimType – тип претензии;
* ClaimValue – обоснование претензии.

Колонки Id, RoleId имеет тип «integer». Колонки ClaimType, ClaimValue имеют тип «text».

**3.1.11 Таблица \_EFMigrationsHistory**

Данная таблица является связующей для таблиц AspNetRoles и AspNetUserClaims.

Поля таблицы \_EFMigrationsHistory:

* MigrationId – идентификатор миграции;
* ProductVersion – версия.

Колонки MigrationId, ProductVersion имеет тип «varchar».

**3.1.12 Таблица Photos**

Данная таблица служит для хранения фотографий.

Поля таблицы Photos:

* Id – первичный ключ;
* Url – адрес фотографии;
* IsMain – отвечает за отображение;
* PublicId – публичный ключ;
* AppUserId – ключ пользователя в системе.

Поле Id будет автоматически заполнено. У различных колонок могут быть разные типы данных. У колонок Url, PublicId тип данных «text». Данный тип используется для текстовых данных переменной длины без поддержки Unicode, такой тип не может служить для хранения двоичных данных. Колонки Id, AppUserId имеют тип «integer». Колонка IsMain имеет тип «boolean».

**3.1.13 Таблица Launches**

Данная таблица служит для хранения данных о запусках.

Поля таблицы Launches:

* Id – исходный идентификатор запуска;
* PublicId – публичный идентификатор запуска.
* Name – название запуска;
* Description – описание запуска.

Колонка Id имеет тип «integer». Колонки PublicId, Name, Description имеют тип «text».

**3.2 Описание структуры и взаимодействия между классами**

Рассмотрим классы разрабатываемого приложения.

**3.2.1 Класс BaseApiController**

Данный класс-контроллер является главным контроллером приложения, от которого наследуются все остальные контроллеры приложения. BaseApiController в свою очередь наследуется от ConrollerBase, это базовый модуль, который предоставляет большое количество методов для работы с контроллерами.

**3.2.2 Класс AccountController**

Данный класс-контроллер предназначен для регистрации пользователя.

Методы класса:

* Register – публичный метод экземпляра класса, используется для полной регистрации пользователя в приложении;
* Login – публичный метод экземпляра класса, используется для входа пользователя в систему;
* UserExists – публичный метод экземпляра класса, используется для проверки существующих пользователей. При попытке создать пользователя с имеющимися уже в системе логином, не даст создать новую запись и выведет предупреждение.

**3.2.3 Класс AdminController**

Данный класс-контроллер предназначен для описания возможностей главного пользователя.

Методы класса:

* GetUserWithRoles – публичный метод экземпляра класса, используется для получения списка пользователей с присвоенными им ролями;
* EditRoles – публичный метод экземпляра класса, используется для изменения роли пользователя.

**3.2.4 Класс BuggyController**

Данный класс-контроллер предназначен для обработки ошибок.

Методы класса:

* GetNotFound – публичный метод экземпляра класса, используется для вывода ошибки поиска. Если информация не найдена по заданному запросу, выводится ошибка;
* GetServerError – публичный метод экземпляра класса, используется для вывода ошибки сервера;
* GetBadRequest – публичный метод экземпляра класса, используется для вывода ошибки запроса. При некорректном запросе выводит ошибку.

**3.2.5 Класс FallbackController**

Данный класс-контроллер используется для связи с Angular.

Методы класса:

* Index – публичный метод экземпляра класса, используется для получения всех позиций, находящихся в приложении и отображения их пользователю.

**3.2.6 Класс LaunchController**

Данный класс-контроллер используется для обработки запросов с запусками.

Методы класса:

* CreateLaunch – публичный метод экземпляра класса, предназначен для создания запуска;
* GetLaunches – публичный метод экземпляра класса, предназначенный для получения списка запусков;
* GetLaunch – публичный метод экземпляра класса, предназначенный для получения запуска;
* UpdateLaunch – публичный метод экземпляра класса, предназначенный для обновления запуска;
* DeleteLaunch – публичный метод экземпляра класса, предназначенный для удаления запуска.

**3.2.7 Класс MessageController**

Данный класс-контроллер используется для обработки запросов, связанных с сообщениями.

Методы класса:

* GetMessagesForUser – публичный метод экземпляра класса, используется для получения всех сообщений, отправленных пользователем;
* DeleteMessage – публичный метод экземпляра класса, используется для удаления сообщения в одностороннем порядке.

**3.2.8 Класс UserController**

Данный класс-контроллер используется для получения информации о пользователях.

Методы класса:

* GetUsers – публичный метод экземпляра класса, используется для получения списка пользователей в системе;
* GetUser – публичный метод экземпляра класса, используется для получения пользователя по UserName;
* UpdateUser – публичный метод экземпляра класса, используется для изменения и обновления информации в профиле пользователя;
* AddPhoto – публичный метод экземпляра класса, используется для добавления фотографии в профиль пользователя;
* SetMainPhoto – публичный метод экземпляра класса, используется для установки выбранной фотографии главной в профиле;
* DeletePhoto – публичный метод экземпляра класса, используется для удаления выбранной фотографии из профиля.

**3.2.9 Класс DataContext**

Данный класс используется для построения моделей на основе базы данных.

Методы класса:

* OnModelCreating – публичный метод экземпляра класса, используется для создания отношений между таблицами базы данных.

**3.2.10 Класс LaunchRepository**

Данный класс используется для получения информации о лайках пользователя.

Методы класса:

* Create – публичный метод экземпляра класса, предназначен для создания запуска;
* GetLaunches – публичный метод экземпляра класса, предназначенный для получения списка запусков;
* GetLaunch – публичный метод экземпляра класса, предназначенный для получения запуска;
* Update – публичный метод экземпляра класса, предназначенный для обновления запуска;
* Delete – публичный метод экземпляра класса, предназначенный для удаления запуска.

**3.2.11 Класс MesssageRepository**

Данный класс используется для получения ифнормации о сообщениях пользователя.

Методы класса:

* AddGroup – публичный метод экземпляра класса, используется для создания новой группы пользователей. На данном этапе реализации под группой понимается, диалог из двух пользователе. Но данную функцию можно расширять;
* AddMessage – публичный метод экземпляра класса, используется для добавления сообщения;
* DeleteMessage – публичный метод экземпляра класса, используется для удаления сообщения;
* GetConnections – публичный метод экземпляра класса, используется для получения соединения;
* GetGroupForConnections – публичный метод экземпляра класса, используется для получения соединения группы;
* GetMessage – публичный метод экземпляра класса, используется для получения сообщения;
* GetMessageGroup – публичный метод экземпляра класса, используется для получения сообщения группы;
* GetMessageForUser – публичный метод экземпляра класса, используется для полученя сообщения определенного пользователя;
* GetMessageThread – публичный метод экземпляра класса, используется для получения списка сообщений;
* RemoveConnection – публичный метод экземпляра класса, используется для удаления соединения;

**3.2.12 Класс Seed**

Данный класс используется для загрузки данных в базу данных при первом запуске.

Методы класса:

* SeedUsers – публичный метод экземпляра класса, используется для; добавления начальных записей о пользователе в базу данных.

**3.2.13 Класс UnitOfWork**

Данный класс используется для описания надстроек над репозиториями.

Методы класса:

* Complete – публичный метод экземпляра класса, используется для сохранения изменения внесенных в базу данных;
* HasChanges – публичный метод экземпляра класса, используется для проверки базы данных на изменения.

**3.2.14 Класс UserRepository**

Данный класс используется для получения данных о пользователе.

Методы класса:

* GetMemberAsync – публичный метод экземпляра класса, используется для получения определенного пользователя;
* GetMembersAsync – публичный метод экземпляра класса, используется для получения списка пользователей;
* GetUsersByUsernameAsync – публичный метод экземпляра класса, используется для получения пользователя по имени;
* GetUserByIdAsync – публичный метод экземпляра класса, используется для получения пользователя по идентификатору;
* GetUserGender – публичный метод экземпляра класса, используется для получения пользователя по полу;
* GetUsersAsync – публичный метод экземпляра класса, используется для получения фото пользователей;
* Update – публичный метод экземпляра класса, используется для обновления данных о пользователе.

**3.2.15 Класс CreateMessageDto**

Данный класс используется для отправки сообщений между пользователями.

**3.2.16 Класс LaunchDto**

Данный класс используется для отображения запусков.

**3.2.17 Класс LoginDto**

Данный класс используется для авторизации пользователей.

**3.2.18 Класс MemberDto**

Данный класс используется для получения информации о пользователе.

**3.2.19 Класс MemberUpdateDto**

Данный класс используется для обновления информации о пользователе.

**3.2.20 Класс MessageDto**

Данный класс используется для хранения информации о диалогах пользователей.

**3.2.21 Класс PhotoDto**

Данный класс используется для хранения информации о фотографиях пользователей.

**3.2.22 Класс RegisterDto**

Данный класс используется для авторизации пользователей.

**3.2.23 Класс UserDto**

Данный класс используется для получения информации о пользователях.

**3.2.24 Класс AppRole**

Данный класс, реализующий связь с базой данных, используется для хранения информации о роли пользователя.

**3.2.25 Класс AppUser**

Данный класс, реализующий связь с базой данных, используется для хранения информации о пользователях.

**3.2.26 Класс Connection**

Данный класс, реализующий связь с базой данных, используется для создания соединения в реальном времени между пользователями.

**3.2.27 Класс Group**

Данный класс, реализующий связь с базой данных, используется для объединения пользователей в группы.

**3.2.28 Класс Message**

Данный класс, реализующий связь с базой данных, используется для хранения информации о диалогах пользователей.

**3.2.29 Класс Photo**

Данный класс, реализующий связь с базой данных, используется для хранения информации о фотографиях пользователей.

**3.2.30 Класс UserLike**

Данный класс, реализующий связь с базой данных, используется для хранения информации о понравившихся пользователях.

**3.2.31 Класс ApiException**

Данный класс используется для обработки ошибок.

**3.2.32 Класс ApllicationServiceExstensions**

Данный класс используется для конфигурирования сущностей приложения.

Метод класса:

* AddApplicationServices – публичный метод экземпляра класса, используется для подключения сервисов.

**3.2.33 Класс ClaimsPrincipleExtensions**

Данный класс используется для получения информации о текущем пользователе.

Методы класса:

* GetUsername – публичный метод экземпляра класса, используется для получения логина текущего пользователя;
* GetUserId – публичный метод экземпляра класса, используется для получения идентификатора текущего пользователя.

**3.2.34 Класс DateTimeExtensions**

Данный класс используется для расширения функциональности типа данных DateTime.

Метод класса:

* CalculateAge – публичный метод экземпляра класса, используется для расчета возраста пользователя.

**3.2.35 Класс HttpExtensions**

Данный класс используется для расширения функциональности протокола HTTP.

Метод класса:

* AddPaginationHeader – публичный метод экземпляра класса, используется для добавления заголовков в запрос.

**3.2.36 Класс IdentityServiceExtensions**

Данный класс реализован для использования системного администрирования пользователей.

Метод класса:

* AddIdentityServices – публичный метод экземпляра класса, используется для добавления правил верификации.

**3.2.37 Класс QueryableExtensions**

Данный класс используется для расширения функциональности системного интерфейса.

Метод класса:

* MarkUnreadAsRead – публичный метод экземпляра класса, используется для изменения статуса сообщения.

**3.2.38 Класс AutoMapperProfiles**

Данный класс используется для автоматического приведения типов.

**3.2.39 Класс CloudinarySettings**

Данный класс используется для хранения настроек сервиса Cloudinary.

**3.2.40 Класс LogUserActivity**

Данный класс используется для отображения активного статуса пользователя.

Метод класса:

* OnActionExecutionAsync – публичный метод экземпляра класса, используется для изменения статуса пользователя.

**3.2.41 Класс MessageParams**

Данный класс используется для пагинации сообщений.

**3.2.42 Класс PagedList**

Данный класс используется для хранения информации в постраничном формате.

Метод класса:

* CreateAsync – публичный метод экземпляра класса, используется для создания постраничной коллекции данных.

**3.2.43 Класс PaginationHeader**

Данный класс используется для хранения информации о конфигурации постраничного формата.

**3.2.44 Класс PaginationParams**

Данный класс используется для запроса на получение информации в постраничном формате.

**3.2.45 Класс UserParams**

Данный класс используется для запроса на получение пользователей в постраничном формате.

**3.2.46 Интерфейс ILikesRepository**

Данный интерфейс используется для описания функциональности работы с лайками.

**3.2.47 Интерфейс IMessageRepository**

Данный интерфейс используется для описания функциональности работы с сообщениями.

**3.2.48 Интерфейс IPhotoService**

Данный интерфейс используется для описания функциональности работы с фотографиями.

**3.2.49 Интерфейс ITokenService**

Данный интерфейс используется для описания функциональности работы с токенами.

**3.2.50 Интерфейс IUnitOfWork**

Данный интерфейс используется для описания функциональности работы с репозиториями.

**3.2.51 Интерфейс IUserRepository**

Данный интерфейс используется для описания функциональности работы с пользователями.

**3.2.52 Класс ExceptionMiddleware**

Данный класс используется для предоставления информации об ошибках.

Метод класса:

* InvokeAsync – публичный метод экземпляра класса, используется для вывода информации об ошибке.

**3.2.53 Класс PhotoService**

Данный класс используется для работы с фотографиями.

Методы класса:

* AddPhotoAsync – публичный метод экземпляра класса, используется для добавления фотографий;
* DeletePhotoAsync – публичный метод экземпляра класса, используется для удаления фотографий.

**3.2.54 Класс TokenService**

Данный класс используется для работы с токенами.

Метод класса:

* CreateToken – публичный метод экземпляра класса, используется для создания токена.

**3.2.55 Класс MessageHub**

Данный класс используется для обработки сообщений.

Методы класса:

* OnConnectedAsync – публичный метод экземпляра класса, используется для создания соединения;
* OnDisconnectedAsync – публичный метод экземпляра класса, используется для разрыва соединения;
* SendMessage – публичный метод экземпляра класса, используется для отправки сообщения;
* AddToGroup – публичный метод экземпляра класса, используется для добавления пользователя в группу;
* RemoveFromMessageGroup – публичный метод экземпляра класса, используется для удаления пользователя из группы;
* GetGroupName – публичный метод экземпляра класса, используется для получения уникального имени группы.

**3.2.56 Класс PresenceHub**

Данный класс используется для отслеживания состояния пользователя.

Методы класса:

* OnConnectedAsync – публичный метод экземпляра класса, используется для создания соединения;
* OnDisconnectedAsync – публичный метод экземпляра класса, используется для разрыва соединения.

**3.2.57 Класс PresenceTracker**

Данный класс используется для уведомления пользователя об активном состоянии других пользователей.

Методы класса:

* UserConnected – публичный метод экземпляра класса, используется для создания пользовательского соединения;
* UserDisconnected – публичный метод экземпляра класса, используется для разрыва пользовательского соединения;
* GetOnlineUsers – публичный метод экземпляра класса, используется для получения списка онлайн пользователей;
* GetConnectionsForUser – публичный метод экземпляра класса, используется для получения списка использующихся соединений.