ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОТРАЖЕНИЯ И ОРГАНИЗАЦИИ СОЦИАЛЬНЫХ ВЗАИМООТНОШЕНИЙ

пояснительная записка к проекту

|  |  |
| --- | --- |
|  | Заказчик:  В.О. Фатхулова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |
|  | Выполнили:  И.А. Искужин,  В.М. Костусенко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

**2021**

АННОТАЦИЯ

Пояснительная записка к проекту содержит постановку и веб-приложение решения задачи «ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОТРАЖЕНИЯ И ОРГАНИЗАЦИИ СОЦИАЛЬНЫХ ВЗАИМООТНОШЕНИЙ».

Веб-приложение «Write Me» написана на языке C# в среде программирования Visual Studio 2019 с использованием сервера баз данных MySQL Workbench 8.0 CE и веб-технологии ASP Net. Core и предназначена для работы в браузерах любой операционной системы и отлажена на данных контрольного примера.

# СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| Введение | 4 |
| 1. Постановка задачи | 5 |
| * 1. Описание предметной области | 5 |
| * 1. Описание входной информации | 7 |
| * 1. Описание выходной информации | 8 |
| * 1. Концептуальное моделирование | 9 |
| * 1. Логическое моделирование | 10 |
| * 1. Описание структуры базы данных | 11 |
| * 1. Контрольный пример | 15 |
| * 1. Общие требования к программному продукту | 19 |
| 1. Экспериментальный раздел | 21 |
| 1. Описание программы | 21 |
| 1. Руководство пользователя | 30 |
| 1. Протокол тестирования программного продукта | 37 |
| 1. Меры по обеспечению защиты информации | 60 |
| Заключение | 62 |
| Приложения | 63 |
| Список сокращений | 90 |
| Список использованных источников | 91 |

# ВВЕДЕНИЕ

В условиях стремительных технологических прорывов появилась возможность автоматизации сферы транспортных услуг по перемещению пассажиров между городами. В связи с увеличением пассажирооборота возникла необходимость в создании автоматизированной информационной системы междугородных маршрутов.

Для этого необходимо разработать программный продукт, который упростит контроль и взаимодействие с междугородными пассажирскими маршрутами.

В первом разделе курсового проекта должны быть решены следующие задачи:

* изучить предметную область;
* описать входную и выходную информацию;
* изучить концептуальную модель;
* изучить логическую модель;
* описать структуру базы данных;
* описать контрольный пример и общие требования к программному продукту.

Во втором разделе курсового проекта должны быть решены следующие задачи:

* описать программу;
* описать руководство пользователя;
* описать протокол тестирования программного продукта и меры по обеспечению защиты информации.

В заключении подведены итоги работы над курсовым проектом, обобщены выводы и рекомендации.

## 1 Описание предметной области

Проектируемая информационная система предназначена для дистанционного взаимодействия между людьми при помощи инструментов автоматизированной информационной системы "Write Me", каждый ее пользователь сможет создать свой виртуальный портрет - сформировать аккаунт, в котором они смогут указать подробные данные о себе (ФИО, дату рождения, страну проживания, город и другое). По этой информации аккаунт пользователя смогут найти другие участники информационной системы. Наличие аккаунта уже позволяет использовать механизмы поиска людей, а также общения, для знакомств и создания социальных отношений между людьми, которые имеют схожие интересы.

Интернет-продукт должен быть разграничен функционально по уровню доступа из следующего списка:

* пользователь;
* администратор.

Для пользователя, информационная система должна обеспечивать возможность выполнения перечисленных ниже функций:

* отправление жалобы на пользователя или пост;
* поиск людей;
* просмотр каталога входящих заявок;
* просмотр каталога диалогов;
* просмотр каталога друзей;
* просмотр каталога исходящих заявок
* просмотр каталога личных постов;
* просмотр личного профиля;
* просмотр новостной ленты.

Для модератора, информационная система должна обеспечивать возможность выполнения перечисленных ниже функций:

* просмотр каталога жалоб;
* рассмотрение жалоб и принятие решения по ней.

А также иметь все функциональные составляющие пользователя.

Для администратора, информационная система должна обеспечивать возможность выполнения перечисленных ниже функций:

* добавление новости проект.

А также иметь все функциональные составляющие пользователя и модератора.

## 2 Техническое задание

### 2.2 Описание входной информации

Основным источником информации для взаимодействия между пользователями является личная информация личных аккаунтов пользователей, а также новостная лента системы.

Описание входных документов представлено в таблице 2.2.1.

#### Таблица 2.2.1 – Описание входных документов

|  |  |  |
| --- | --- | --- |
| Наименование документа (шифр) | Дата поступления документа | Откуда поступает документ |
| Личная информация пользователя | При регистрации | Пользователь |
| Новости | По мере необходимости | Администратор |

### 2.3 Описание выходной информации

Выходными данными являются:

* личный профиль пользователя;
* новостная лента.

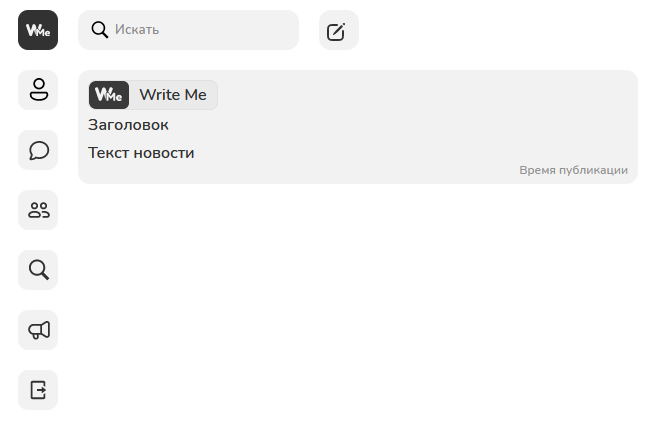
Описание входных документов представлено в таблице 2.3.1.

#### Таблица 2.3.1 – Описание выходных документов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Кол-во экз. | Куда передаются | Поля сортировки | Поля группировки | Итоги |
| Личный профиль | Независимо от времени | 1 | Пользователю | - | - | - |
| Новостная лента | Независимо от времени | 1 | Пользователю | - | - | - |

На рисунке 2.3.1 представлен шаблон выходного документа «Личный профиль».

#### Рисунок 2.3.1 – Шаблон «Личный профиль»



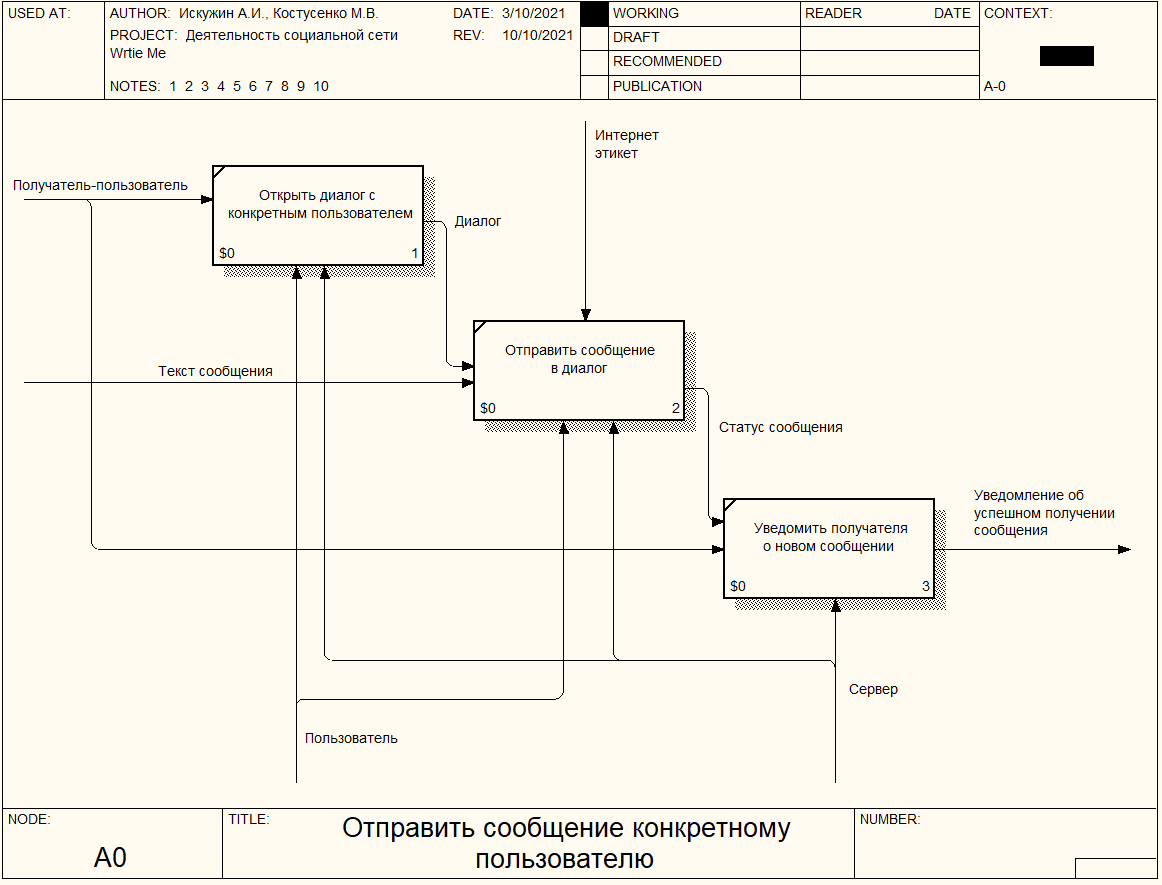
#### Рисунок 2.3.2 – Шаблон «Новостная лента»

## 3 Диаграмма Ганта

## 4 Проектирование ПО

### 4.1 Контекстная диаграмма и декомпозиция 1-го уровня

#### Рисунок 4.1.1 – контекстная диаграмма



#### Рисунок 4.1.2 – декомпозиция 1-го уровня

### 4.2 Диаграмма вариантов использования

Концептуальная модель – это модель предметной области. Компонентами модели являются объекты и взаимосвязи. Концептуальная модель включает описание объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области. Взаимосвязи между объектами являются частью концептуальной модели и должны отображаться в базе данных. Взаимосвязь может охватывать любое число объектов.

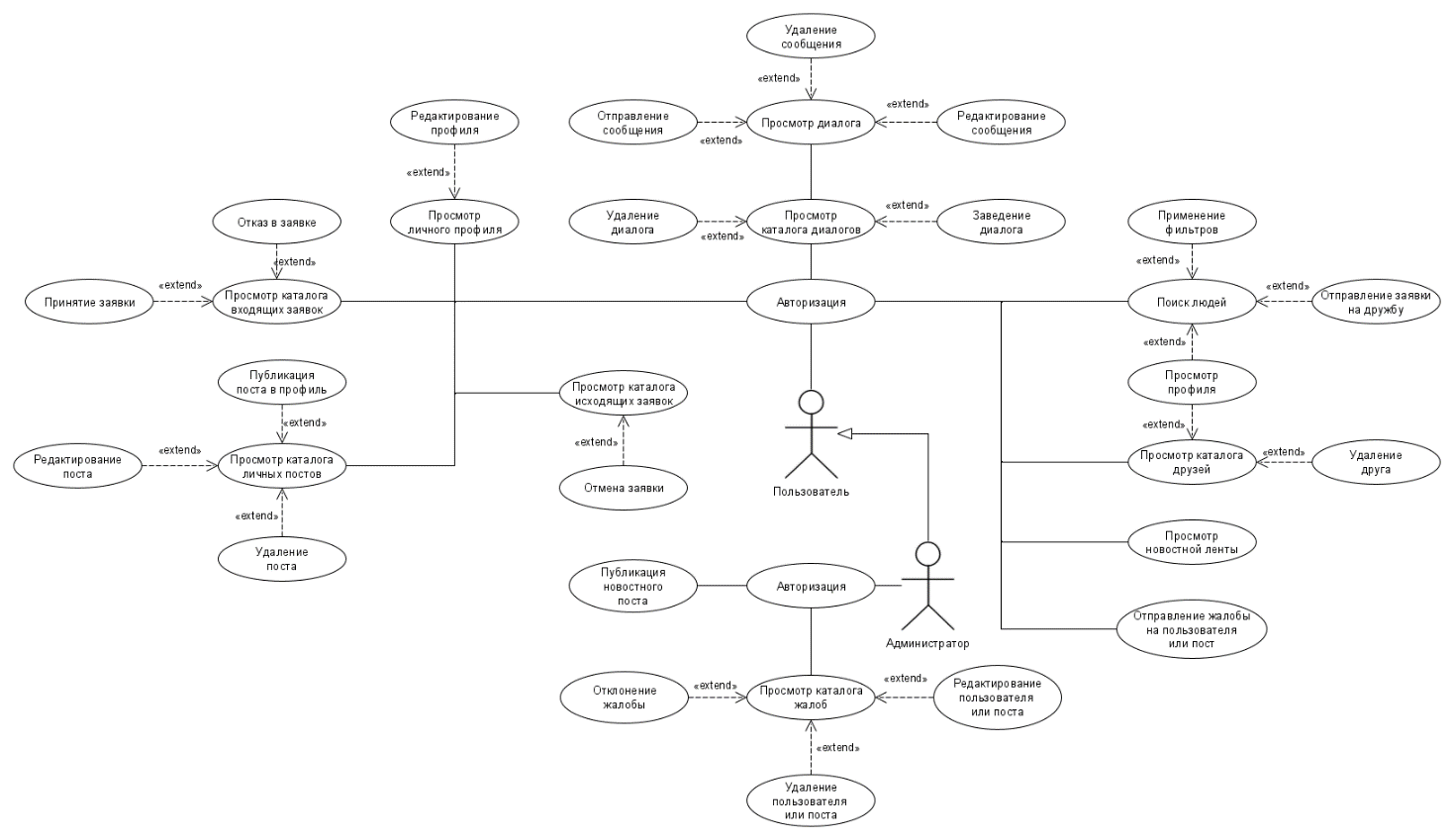
Основными элементами модели являются сущности, связи между ними и их свойства (атрибуты).

Сущность – это класс однотипных объектов, информация о которых должна быть учтена в модели.

Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Каждая сущность в модели изображается в виде прямоугольника с наименованием.

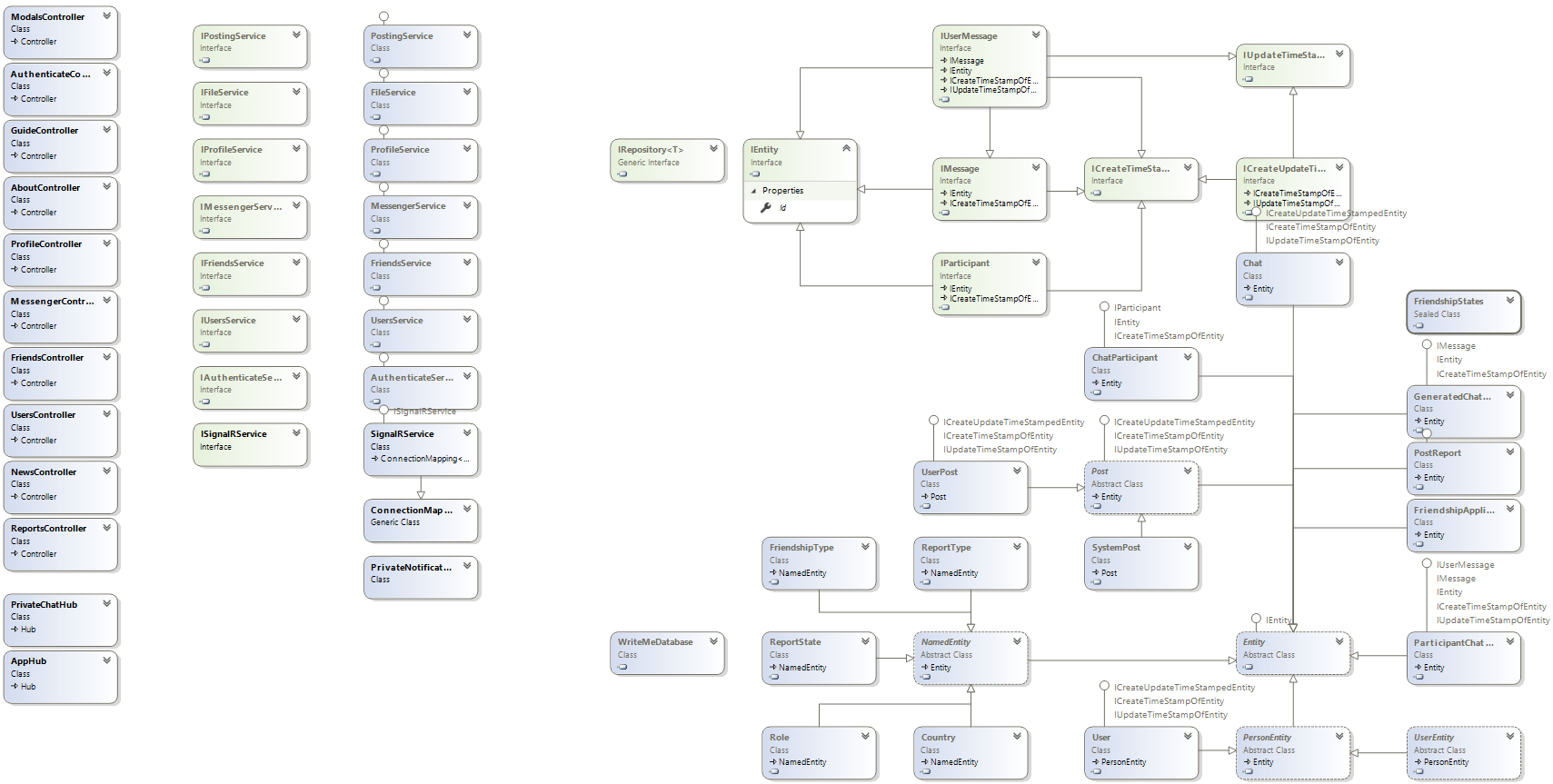
У сущностей выделяются ключевые атрибуты – ключ сущности – это один или более атрибутов, уникально определяющих данную сущность.

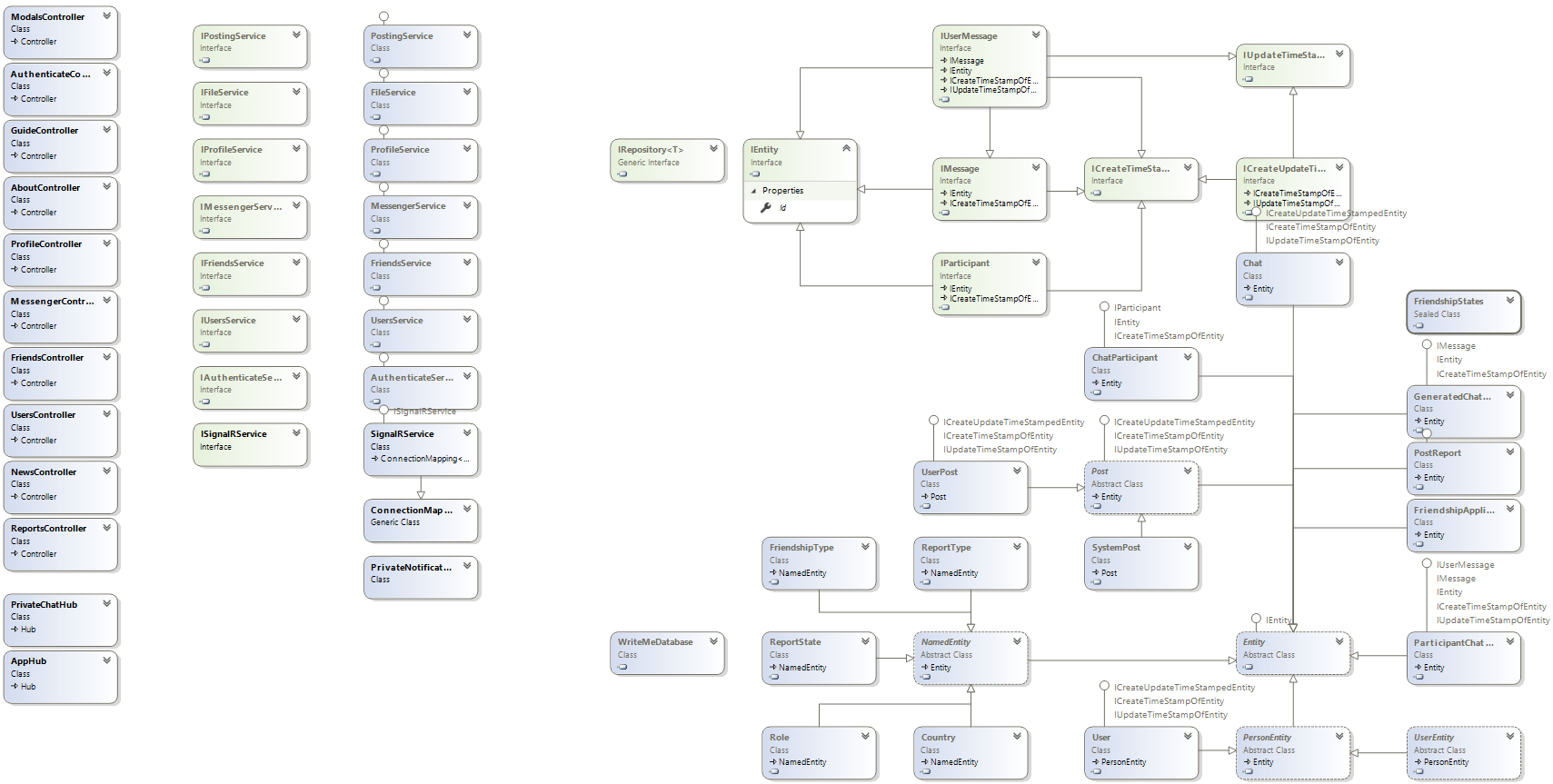
Вместе эти элементы образует диаграмму вариантов использования в соответствие с рисунком 4.2.1.



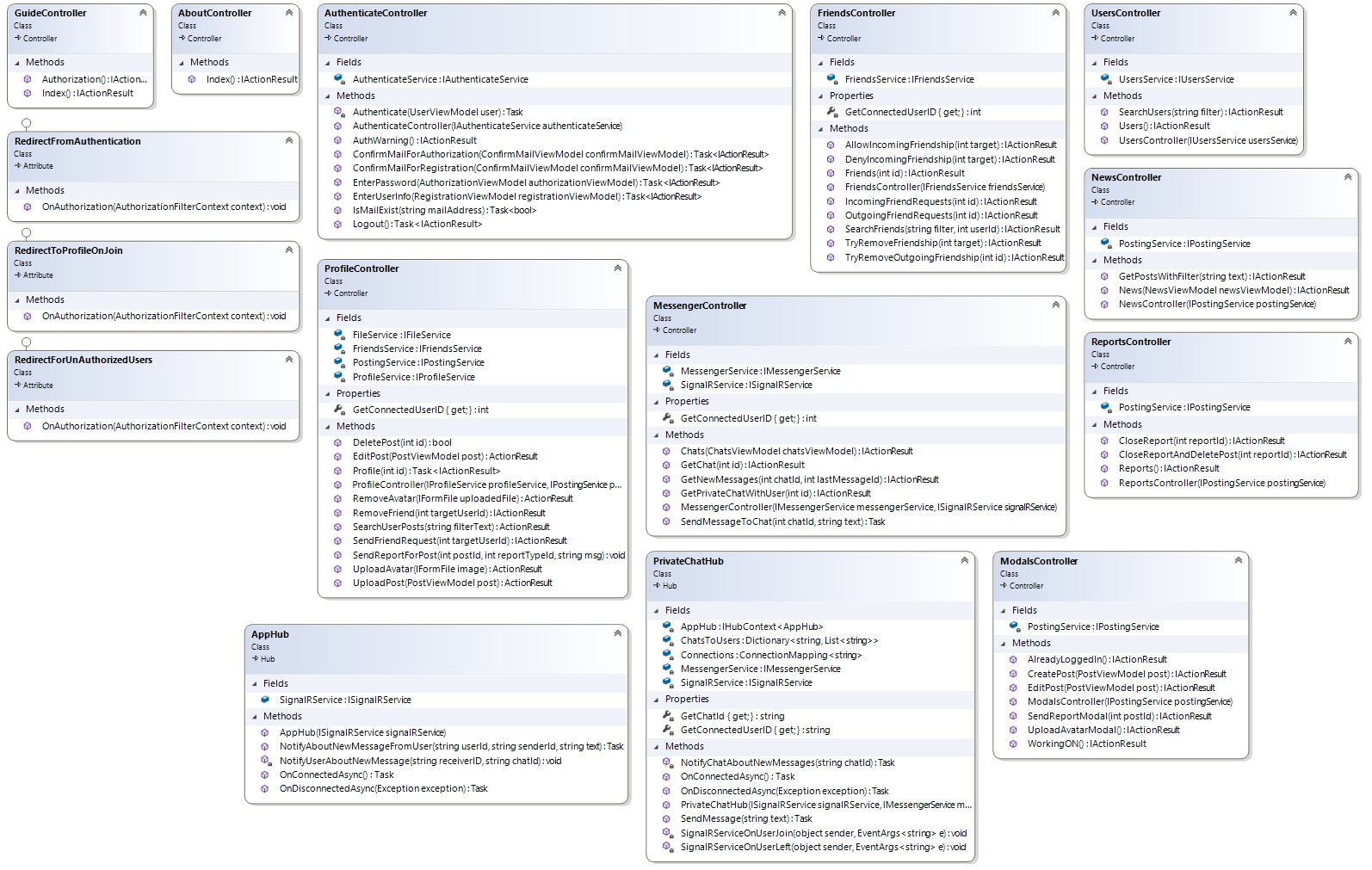
#### Рисунок 4.2.1 – UCD диаграмма

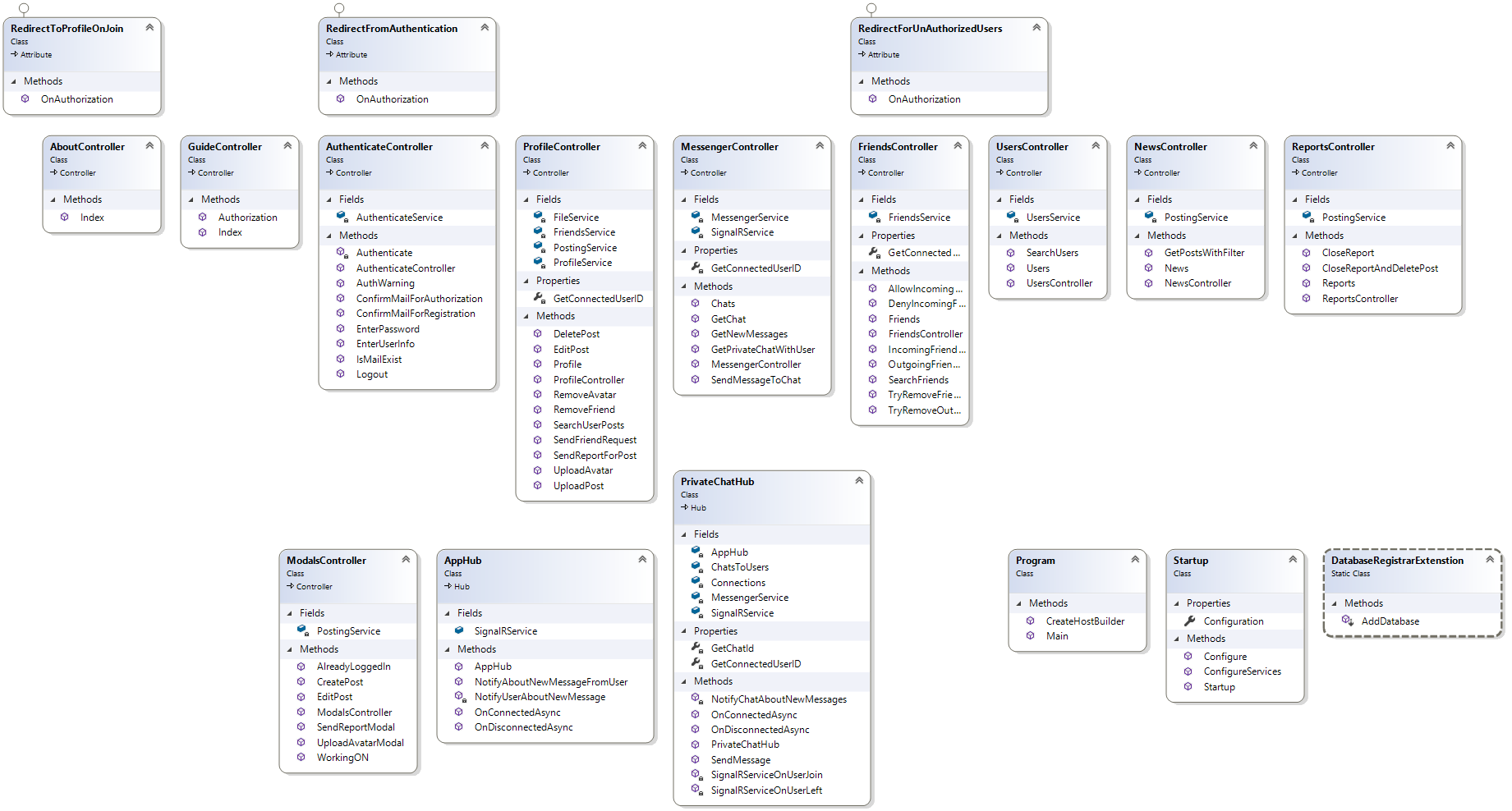
### 4.3 Диаграмма классов



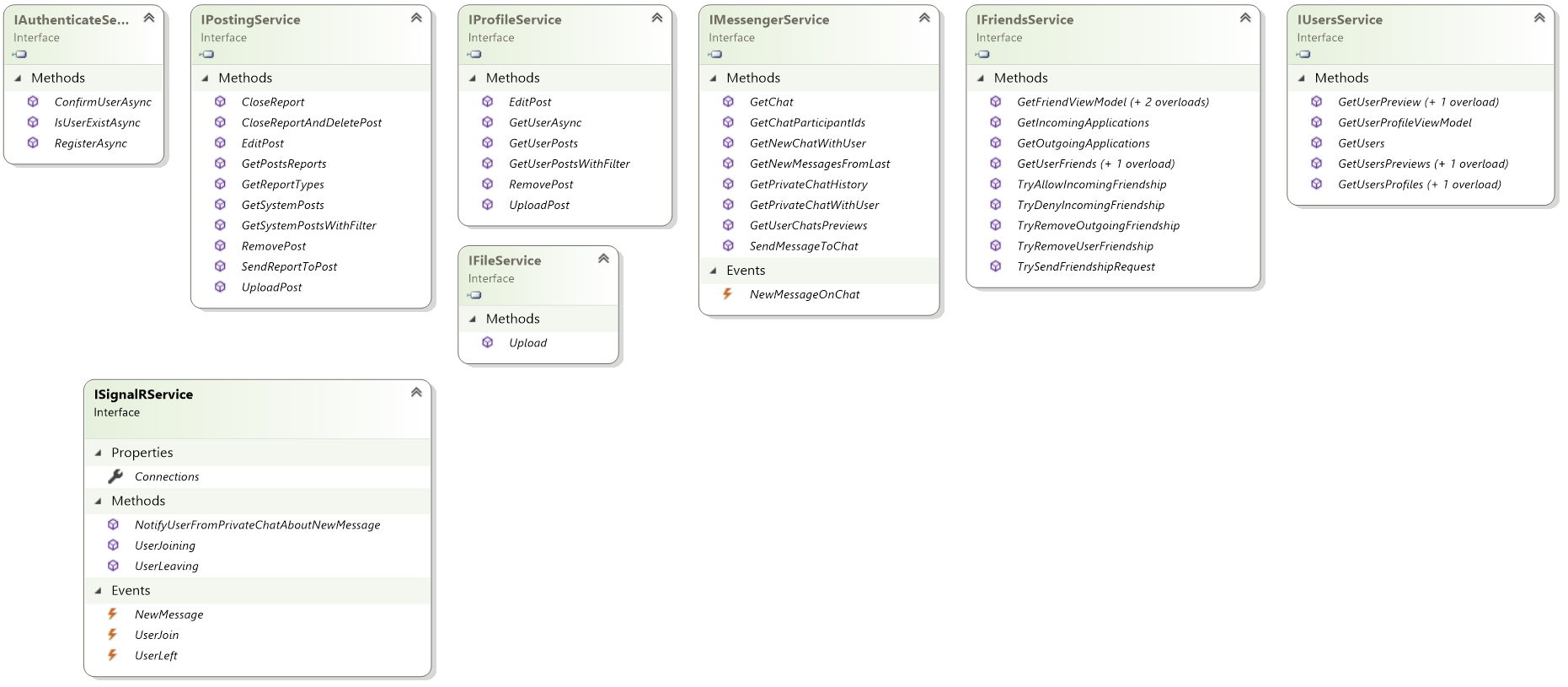


#### Рисунок 4.3.1 – общая UML диаграмма классов

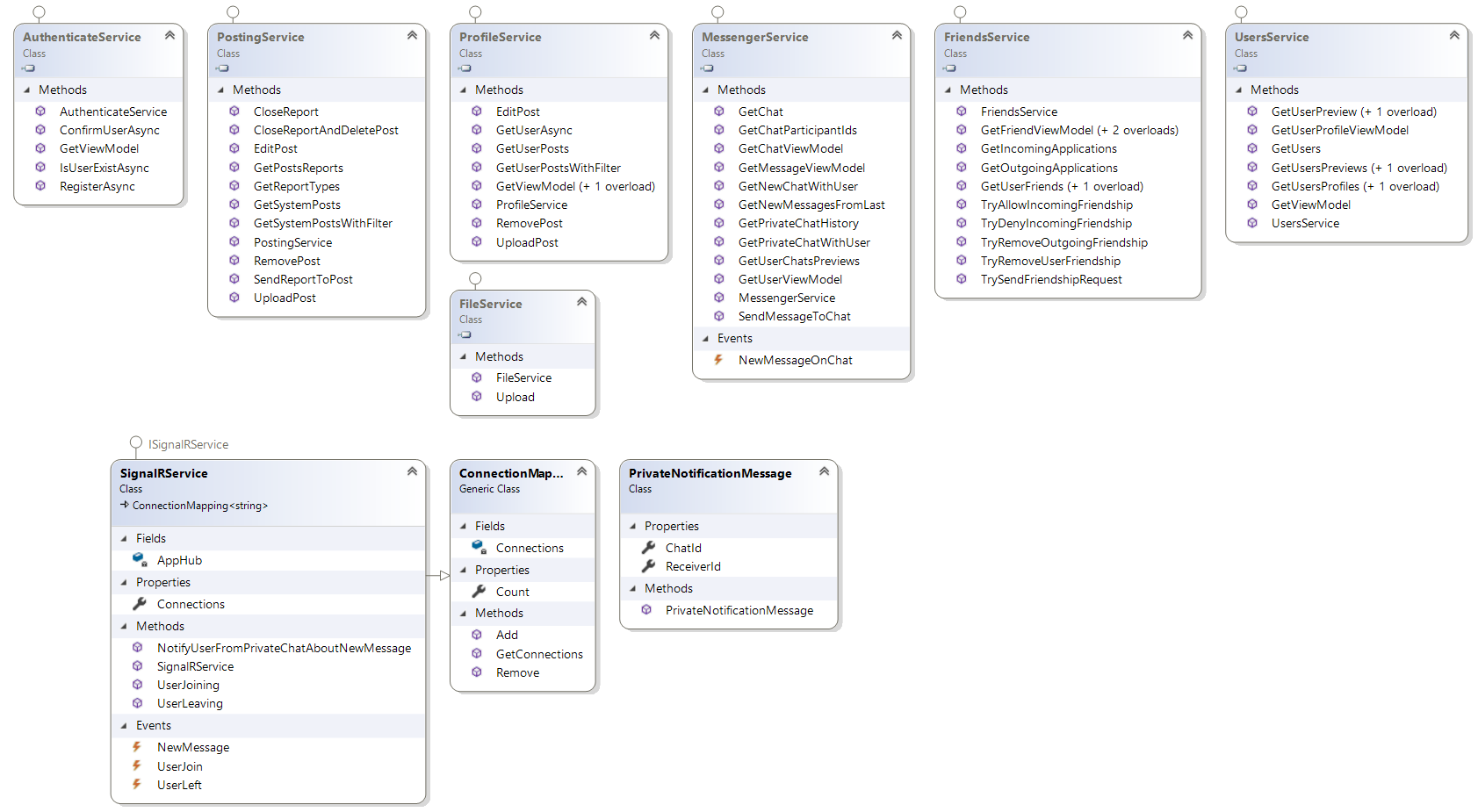




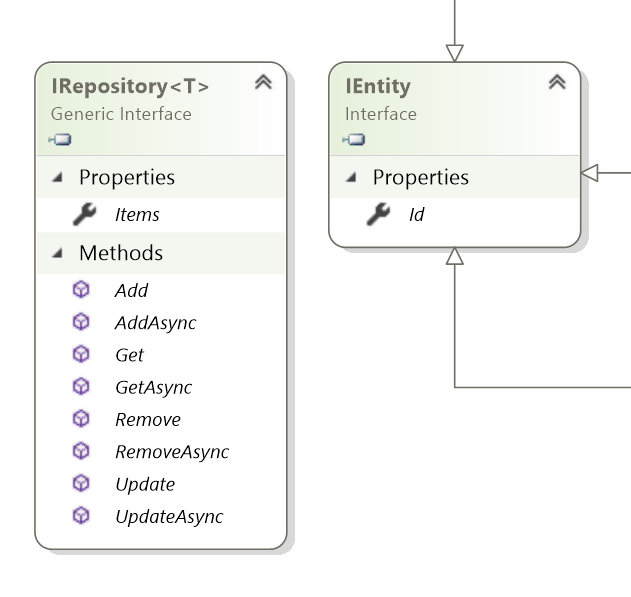
#### Рисунок 4.3.2 – UML диаграмма классов контроллеров



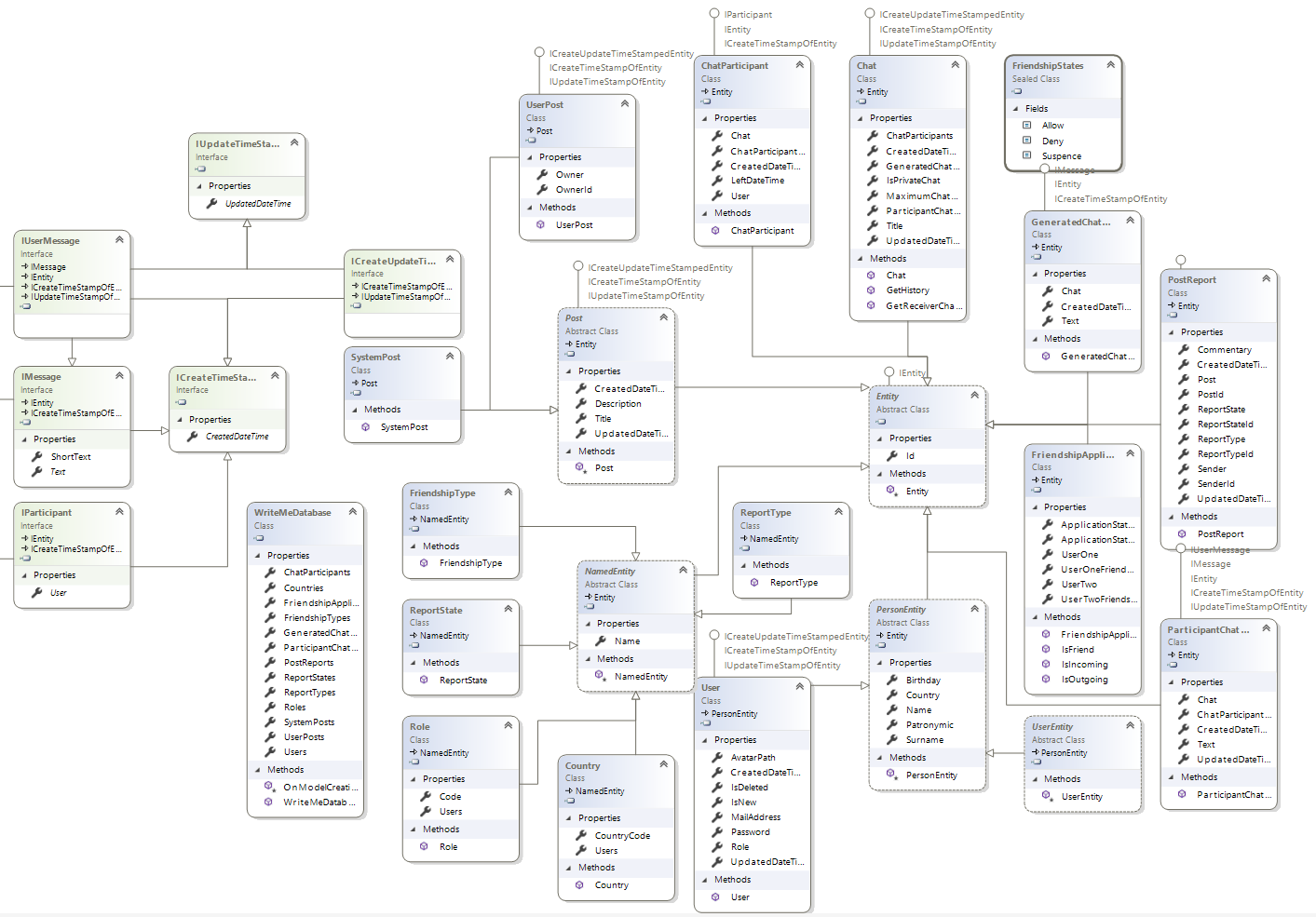
#### Рисунок 4.3.3 – UML диаграмма классов интерфейсов сервисов



#### Рисунок 4.3.4 – UML диаграмма классов сервисов



#### Рисунок 4.3.5 – UML диаграмма классов репозитория



#### Рисунок 4.3.6 – UML диаграмма классов слоя базы данных

## 5 Описание структуры базы данных

Для создания базы данных используется СУБД MySQL

Среда Workbench 8.0 CE.

### Таблица 5.1 – Описание структуры базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Физическое имя таблицы (Логическое имя таблицы) | | | | |
| countries (страны) | | | | |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа[[1]](#footnote-1) |
| Id | Код страны | INT |  | PK |
| CountryCode | Идентификационный код страны | INT |  | - |
| Name | Наименование страны | VARCHAR | 256 | - |
| roles (роли) | | | | |
| Id | Код роли | INT |  | PK |
| Name | Наименование роди | VARCHAR | 256 | - |
| Code | Короткий код | LONGTEXT |  | - |
| users (пользователи) | | | | |
| Id | Код пользователя | INT |  | PK |
| MaillAddress | Почта пользователя | LONGTEXT |  | - |
| Password | Пароль пользователя | LONGTEXT |  | - |
| RoleId | Код роли | INT |  | FK |
| IsNew | Статус нового пользователя | TINYINT | 1 | - |
| IsDeleted | Статус удаленного пользователя | TINYINT | 1 | - |
| CreatedDateTime | Штамп создания профиля | DATETIME | 6 | - |
| UpdatedDateTime | Штамп обновления профиля | DATETIME | 6 | - |
| Name | Имя пользователя | LONGTEXT |  | - |
| Surname | Фамилия пользователя | LONGTEXT |  | - |
| Patronymic | Отчество пользователя | LONGTEXT |  | - |
| Birthday | Дата рождения пользователя | DATETIME | 6 | - |
| AvatarPath | Путь к иконке профиля | LONGTEXT |  | - |
| CountryId | Код страны | INT |  | - |
| chatParticipants (Участники чата) | | | | |
| Id | Код участника | INT |  | PK |
| ChatId | Код чата | INT |  | FK |
| UserId | Код пользователя | INT |  | FK |
| CreatedDateTime | Штамп создания/вхождения в чат | DATETIME | 6 | - |
| LeftDateTime | Штамп выхода из чата | DATETIME | 6 | - |
| chats (чаты) | | | | |
| Id | Код чата | INT |  | PK |
| Title | Заголовок чата | LONGTEXT |  | - |
| MaximumChatParticipants | Максимальное количество участников в чате | INT |  | - |
| IsPrivateChat | Является сущность приватным чатом | TINYINT | 1 | - |
| CreatedDateTime | Штамп создания чата | DATETIME | 6 | - |
| UpdatedDateTime | Штамп обновления чата | DATETIME | 6 | - |
| generatedChatMessaged (сгенерированные сообщения чата) | | | | |
| Id | Код сообщения | INT |  | PK |
| ChatId | Код чата | INT |  | FK |
| Text | Текст сообщения | LONGTEXT |  | - |
| CreatedDateTime | Штамп создания сообщения | DATETIME | 6 | - |
| participantChatMessages (сообщения участников чата) | | | | |
| Id | Код сообщения | INT |  | PK |
| ChatParticipantSenderId | Код отправителя, участника чата | INT |  | FK |
| ChatId | Код чата | INT |  | FK |
| Text | Текст сообщения | LONGTEXT |  | - |
| CreatedDateTime | Штамп отправки сообщения | DATETIME | 6 | - |
| UpdatedDateTime | Штамп изменения сообщения | DATETIME | 6 | - |
| userPosts (посты пользователей | | | | |
| Id | Код поста | INT |  | PK |
| OwnerId | Код пользователя, создателя | INT |  | FK |
| Title | Заголовок поста | TEXTLONG |  | - |
| Description | Текст поста | TEXTLONG |  | - |
| CreatedDateTime | Штамп создания поста | DATETIME | 6 | - |
| UpdatedDateTime | Штамп обновления поста | DATETIME | 6 | - |
| systemPosts (системные посты) | | | | |
| Id | Код поста | INT |  | PK |
| Title | Заголовок поста | LONGTEXT |  | - |
| Description | Текст поста | LONGTEXT |  | - |
| CreatedDateTime | Штамп создания поста | DATETIME | 6 | - |
| UpdatedDateTime | Штамп обновления поста | DATETIME | 6 | - |
| postReports (Жалобы на посты) | | | | |
| Id | Код жалобы | INT |  | PK |
| PostId | Код поста | INT |  | FK |
| SenderId | Код отправителя | INT |  | FK |
| Commentary | Кооментарий жалобы | LONGTEXT |  | - |
| ReportTypeId | Код типа жалобы | INT |  | FK |
| ReportStateId | Код состояния жалобы | INT |  | FK |
| CreatedDateTime | Штамп создания жалобы | DATETIME | 6 | - |
| UpdatedDateTime | Штамп обновления жалобы | DATETIME | 6 | - |
| reportsStates (Состояния жалоб) | | | | |
| Id | Код состояния жалобы | INT |  | PK |
| Name | Наименование состояния | VARCHAR | 256 | - |
| reportTypes (Типы жалоб) | | | | |
| Id | Код типа жалобы | INT |  | PK |
| Name | Наименование типа жалобы | VARCHAR | 256 | - |
| friendshipApplications (Дружественные связи) | | | | |
| Id | Код связи | INT |  | PK |
| UserOneID | Код первого пользователя | INT |  | FK |
| UserTwoId | Код второго пользователя | INT |  | FK |
| ApplicationStateUserOne | Состояние заявки первого пользователя | TINYINT | 1 | - |
| ApplicationStateUserTwo | Состояние заявки второго пользователя | TINYINT | 1 | - |
| UserOneFrendshipTypeId | Код типа типа связи первого пользователя | INT |  | FK |
| UserTwoFriendshipTypeID | Код типа связи первого пользователя | INT |  | FK |
|  |  |  |  |  |
|  |  |  |  |  |

Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется логической моделью. Процесс построения логической модели базы данных должен опираться на определённую модель данных (реляционная, сетевая, иерархическая). В нашем случае информационная система создаётся в среде Workbench 8.0 CE и будет представлять собой реляционную базу данных.

В реляционных моделях данных объекты и взаимосвязи между ними представляются с помощью таблиц.

Таблица в реляционной модели называется отношением.

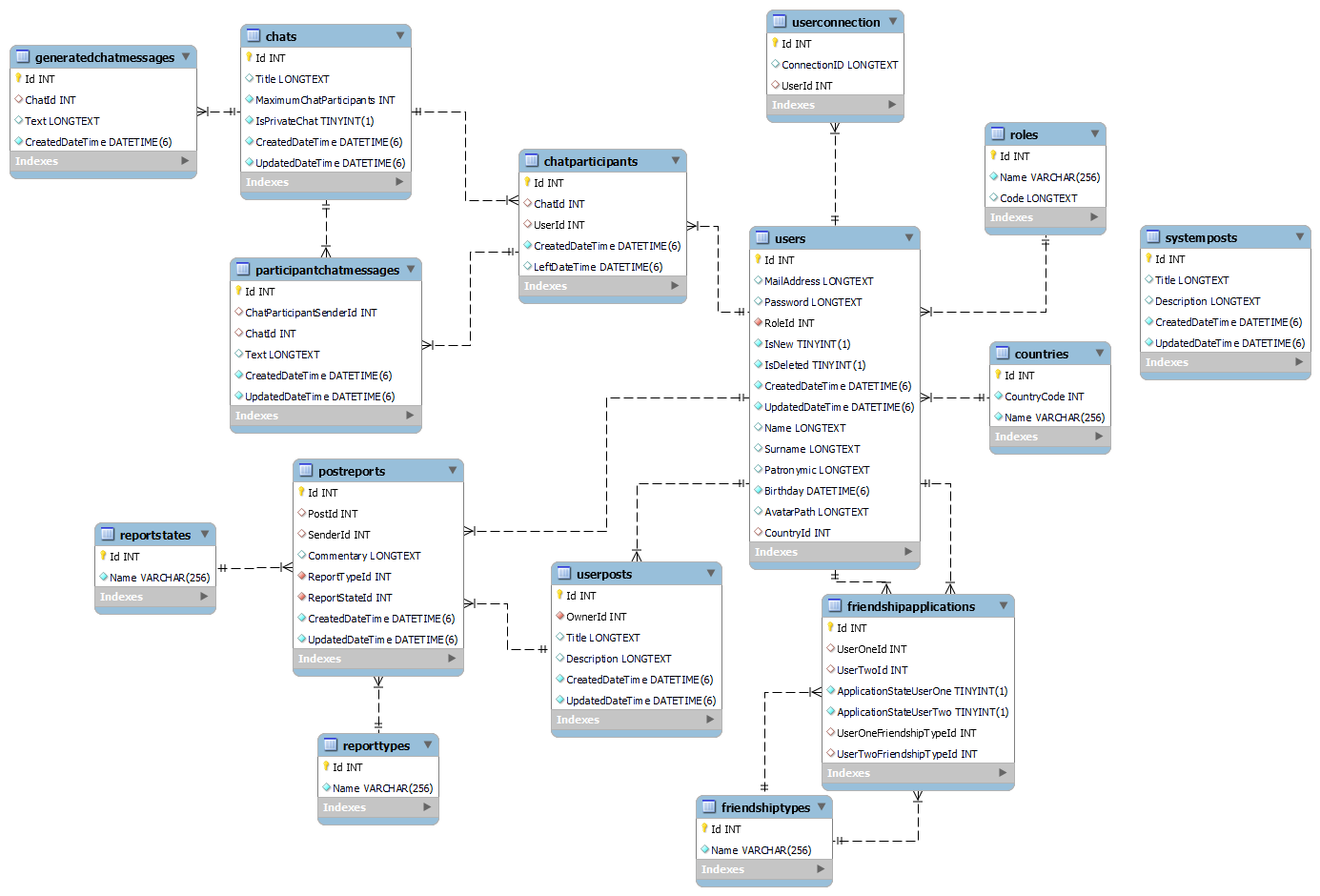
Атрибут (поле) – любой столбец в таблице.

Кортежи (записи) – строки таблицы.

Таблицы связаны между собой при помощи ключевых полей.

Ключ – это поле, позволяющее однозначно идентифицировать запись в таблице. Ключ может быть простым (состоит из одного поля) или составным (из нескольких полей).

В реляционных базах данных логическое проектирование приводит к разработке схемы данных в соответствии с рисунком 5.1.1.



### Рисунок 5.1 – EER модель

## 6 Описание программы

Описание программы содержит: описание контроллеров, сервисов, классов, методов.

|  |  |
| --- | --- |
| Процедуры | Назначение |
| Сервис AuthenticateService – сервис аутентификации | |
| ConfirmUserAsync | Авторизация пользователя |
| RegisterAsync | Регистрация пользователя |
| IsUserExist | Проверка существования пользователя |
| GetViewModel (User) | Получить сущность пользователя в виде модели представления |
| Сервис FileService – сервис работы с файлами | |
| Upload | Загрузить фото |
| Сервис FriendsService – сервис взаимотношений | |
| GetViewModel (User) | Получить сущность пользователя в виде модели представления |
| GetViewModel (FriendshipApplication) | Получить сущность связи отношения между двумя пользователями в виде модели представления |
| GetFriendViewModel | Получить сущность пользователя в виде модели представления |
| GetFriendFromApplication | Получить сущность друга из связи отношений в виде модели представления |
| GetUserFriends (int) | Получить список друзей указанного пользователя |
| GetUserFriends (int, string) | Получить список друзей указанного пользователя, соответствующие фильтру |
| GetIncomingApplications | Получить список входящих заявок на дружбу указанного пользователя |
| GetOutgoingApplications | Получить список исходящих заявок на дружбу указанного пользователя |
| TryRemoveOutgoingFriendship | Отменить исходящую заявку на дружбу |
| TryRemoveFriendship | Удалить пользователя из друзей |
| TryResponseIncomingFriendship | Изменить статус входящей заявки |
| TryAllowIncomingFriendship | Принять входящую заявку |
| TryDeniIncomingFriendship | Отказать во входящей заявке |
| TrySendFriendship | Отправить заявку на дружбу |
| Сервис MessengerService – сервис мессенджер | |
| NewMessagesOnChat | Событие, реагирует на новые сообщения в чате |
| GetUserChatsPreviews | Получить список моделей представления превью чатов |
| GetPrivateChatHistory | Получить историю указанного чата |
| GetChat | Получить модель представления указанного чата |
| GetPrivateChatWithUser | Получить приватный чат с указанным пользователем |
| GetNewChatWithUser | Создать приватный чат с указанным пользователем |
| SendMessageToChat | Отправить сообщение в чат |
| GetNewMessagesFromLast | Получить новые сообщения после указанного сообщения |
| GetChatParticipantsIds | Получить список участников чата |
| GetChatViewModel | Получить сущность чата в виде модели представления |
| Сервис PostingService – сервис записей | |
| GetSystemPostViewModel | Получить сущность записи системы в виде модели представления |
| IsAdmin | Является ли указанный пользователь админом |
| GetSystemPosts | Получить список системных записей |
| GetSystemPostsWithFilter | Получить список системных записей, соответствующих фильтру |
| UploadPost | Выложить запись |
| EditPost | Отредактировать запись |
| RemovePost | Удалить запись |
| SendReportToPost | Отправить жалобу на пост |
| GetPostsReports | Получить список жалоб на посты |
| GetReportTypes | Получить список вариантов жалоб на пост |
| CloseReport | Закрыть жалобу |
| CloseReportAndDeletePost | Закрыть жалобу и удалить пост |
| Сервис ProfileService – сервис профиля | |
| GetUsersPostsWithFilter | Получить список постов указанного пользователя, соответствующие фильтру |
| GetUsersPosts | Получить список постов указанного пользователя |
| GetUserAsync | Получить пользователя |
| UploadPost | Выложить запись |
| EditPost | Отредактировать запись |
| RemovePost | Удалить запись |
| GetViewModel (UserPost) | Получить модель представления записи пользователя |
| GetViewModel (User) | Получить модель представления пользователя |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### 6.1 Назначение программы

### 6.2 Условия выполнения программы

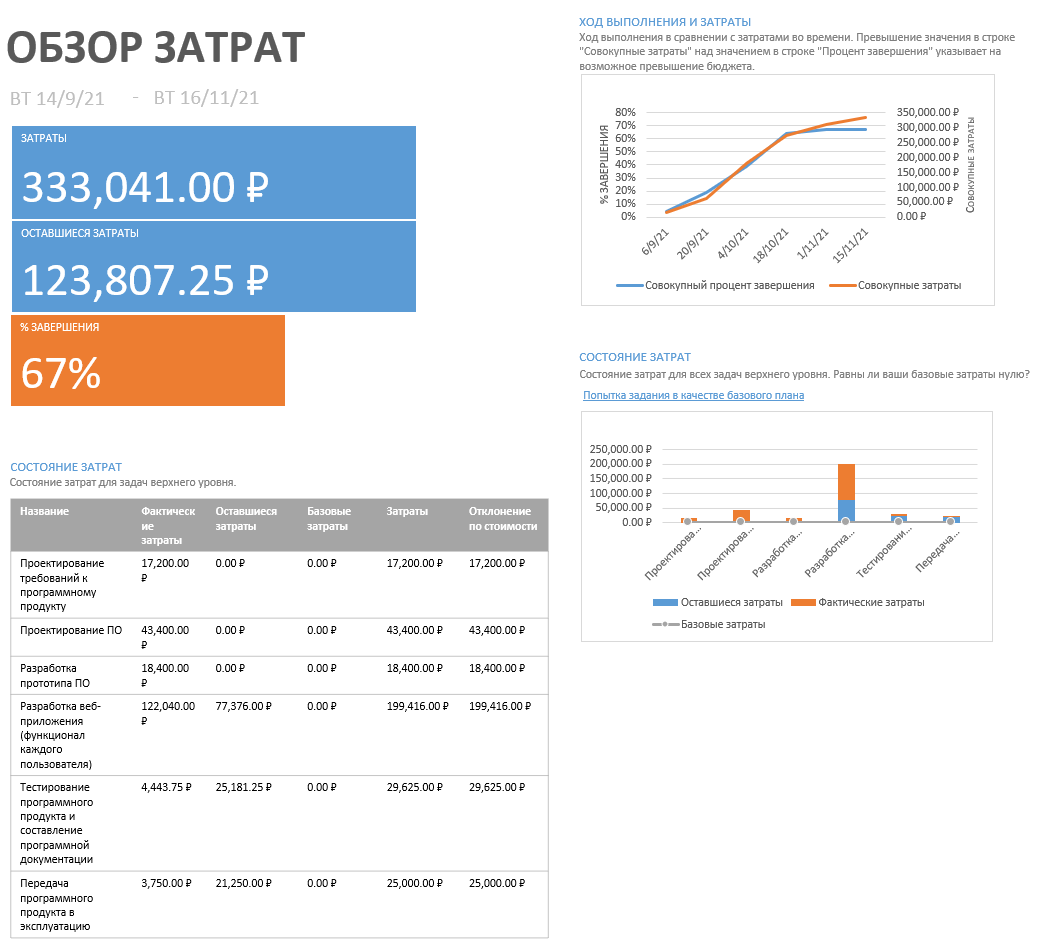
### 6.3 Выполнение программы

## 7 Руководство пользователя

## 8 Протокол тестирования ПО

## 9 Экономические затраты

Обзор затрат изображено на рисунке 9.1



### Рисунок 9.1 – экономические затраты

### 2.4 Меры по обеспечению защиты информации

Под угрозой безопасности понимается событие или действие, которое может привести к разрушению, искажению или несанкционированному использованию данных системы, включая хранимую и обрабатываемую информацию, а также программные средства.

Идентификация пользователей позволяет устанавливать конкретного пользователя, работающего с системой. Право доступа к определенным информационным ресурсам, программам и наборам данных предоставляется ограниченному контингенту лиц, и система должна распознавать пользователей, работающих с системой. Идентификация пользователей производится с помощью паролей.

При входе проходит сравнение введенного пароля пользователем с паролём, находящимся в информационной системе.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта были разработаны структура и алгоритм работы приложения «Мой автобус». При этом были изучены особенности реализации компонентов информационной системы. Результатом работы стало создание программного продукта.

Программное обеспечение написано на языке C# с использованием среды Visual Studio 2019 и СУБД MY SQL SERVER 8.0.

Была проведена эксплуатация и отладочное тестирование программного обеспечения. По результатам отладочного тестирования были устранены некоторые недостатки, в частности были обнаружены и исправлены неточности в реализации алгоритма. После этого было написано руководство пользователя.

С помощью приложения на основе данных контрольного примера были получены результаты, которые полностью совпадают с выходной информацией контрольного примера.

# Приложение А

# СПИСОК СОКРАЩЕНИЙ

MS – Microsoft;

MySQL – MySQL Server;

PK – Первичный ключ;

FK – Внешний ключ;

СУБД – Система управления базами данных;

UCD (Use Case Diagram) – Диаграмма последовательности действий;

EER (Enhanced entity-relationship model) – Расширенная модель «сущность-связь»;

Прецедент – сервисы (функции), которые наша система предоставляет пользователям;

Алгоритм Дейкстры – алгоритм нахождения короткого пути;

ТС – транспортное средство;

Госномер – Государственный автомобильный номерной знак.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Электронные ресурсы:

1 «Вебинар WPF + MVVM 1 – 4» [Видео плейлист] / Шмачилин П.А / МАИ – Московский авиационный институт; Режим доступа <https://www.youtube.com/playlist?list=PL-p05fYs48r-HOAA4_5qZfS0rTIka3fgc>, свободный.

2 «Полное руководство по языку программирования C# 9.0 и платформе .NET 5» [Электронный ресурс] / Сообщество о программировании Metanit; 2012 – 2021; Режим доступа [https://metanit.com/sharp/tutorial/](https://metanit.com/sharp/tutorial/%3c), свободный. Загл. с экрана – Яз. рус., англ.

1. PK – первичный ключ (primary key)  
   FK – внешний ключ (foreign key) [↑](#footnote-ref-1)