МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии и стандарты проектирования»

Тема «Тренировочный Центр»

Исполнитель

студент 2 курса группы 5 Лагодич Юрий Сергеевич

Руководитель работы ассистент Демещик A.С.

Курсовой проект защищён с оценкой

Председатель Пацей Н.В.

Минск 2020

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc9543690)

[1. Аналитический обзор литературы 4](#_Toc9543691)

[2. Анализ требований к программному средству и разработка функциональных требований 8](#_Toc9543692)

[3. Проектирование программного средства 11](#_Toc9543693)

[4. Создание (реализация) программного средства 17](#_Toc9543694)

[5. Тестирование, проверка работоспособности и анализ полученных результатов 22](#_Toc9543695)

[6. Руководство по использованию 28](#_Toc9543696)

[ЗАКЛЮЧЕНИЕ 33](#_Toc9543697)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 34](#_Toc9543698)

[Приложение А 36](#_Toc9543699)

[Приложение Б 37](#_Toc9543700)

[Приложение В 37](#_Toc9543701)

[Приложение Г 38](#_Toc9543702)

[Приложение Д 39](#_Toc9543703)

[Приложение Е 39](#_Toc9543704)

[Приложение Ж 40](#_Toc9543705)

[Приложение З 41](#_Toc9543706)

[Приложение И 45](#_Toc9543705)

[Приложение К 45](#_Toc9543706)

# ВВЕДЕНИЕ

Тренировочные центры становятся весьма популярны в обществе, а технологии не стоят на месте, поэтому открываются новые возможности для обеспечения более комфортного использования уже привычных для нас вещей.

Возможность получения информации о нынешнем состоянии абонементов в любой момент времени и получение качественной программы тренировок без необходимости видеться с вашим тренером могут облегчить тренировочный процесс.

Программное средство «Тренировочный центр» – это программное средство, где каждый желающий может оставить информацию о своих предпочтениях в тренировках, а любой свободный тренер может предложить свою программу тренировок.

Также это предложение выдает абонементы на количество занятий, помогает следить за оставшимся количеством посещений, оставлять личную отметку по поводу тренировки чтобы было удобнее следить за прогрессом.

Приложение имеет 3 типа пользователей: администратор, пользователь, модератор. Они имеют разные полномочия. Модераторы являются тренерами и имеют возможность выдавать людям личные тренировки, выдавать абонементы на посещение спортзала, а также просматривать профиля пользователей для составления программы тренировки. Пользователи в свою очередь могут выбирать тренировку, осуществлять поиск по категориям и по описанию, предоставленную модератором, получать информацию об актуальном абонементе, добавлять личную отметку о тренировке, оставлять в профиле информацию о предпочтениях и изменять свой профиль. Администраторы имеют полные права. Они могут все что модераторы, а также пользователей и выдавать привилегии.

Также будет реализована авторизация всех пользователей. Это значит, что перед тем, начать пользоваться всем функционалом, нужно будет зарегистрироваться. Хранение всех пользователей будет осуществлено в базе данных, и при авторизации или регистрации будет непосредственное обращение к ней. Также вся информация о тренировках будет хранится в базе данных.

В программе реализовано множество проверок на корректность введенных данных, для защиты информации от потери.

# Аналитический обзор литературы

Для выполнения этого курсового проекта не обошлось без использования дополнительной литературы. Таковой является информация на сайте <https://metanit.com/>. На этом сайте имеется много информации, нужной для разработки этого курсового проекта. Также использовалась информация с сайта <http://professorweb.ru/>.

Довольно хорошо информация была представлена в лекциях на сайте https:// <https://diskstation.belstu.by:5001/>.

Самый большое количество информации было получено с официального сайта <https://docs.microsoft.com/en-us/dotnet/csharp/>. Статьи с <https://habr.com/ru/> были также очень полезны.

В качестве аналога я выбрал стартап WIP FIT он максимально приближен к моему проекту.

Суть задумки заключалась в удобном взаимодействии с приложением как клиентам так и клубам.

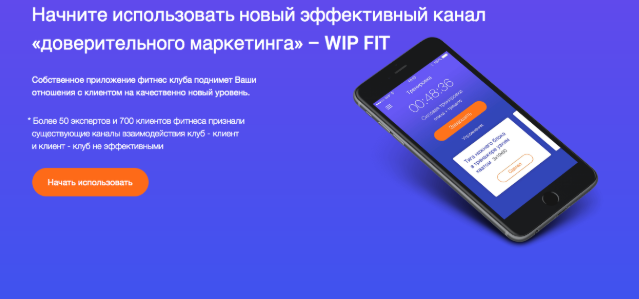


Рисунок 1.1 – Реклама приложения

Существует огромное количество программ предоставляющим клубам статистику и автоматизацию задач необходимых для работы клуба таких как

https://www.fitness365.ru/

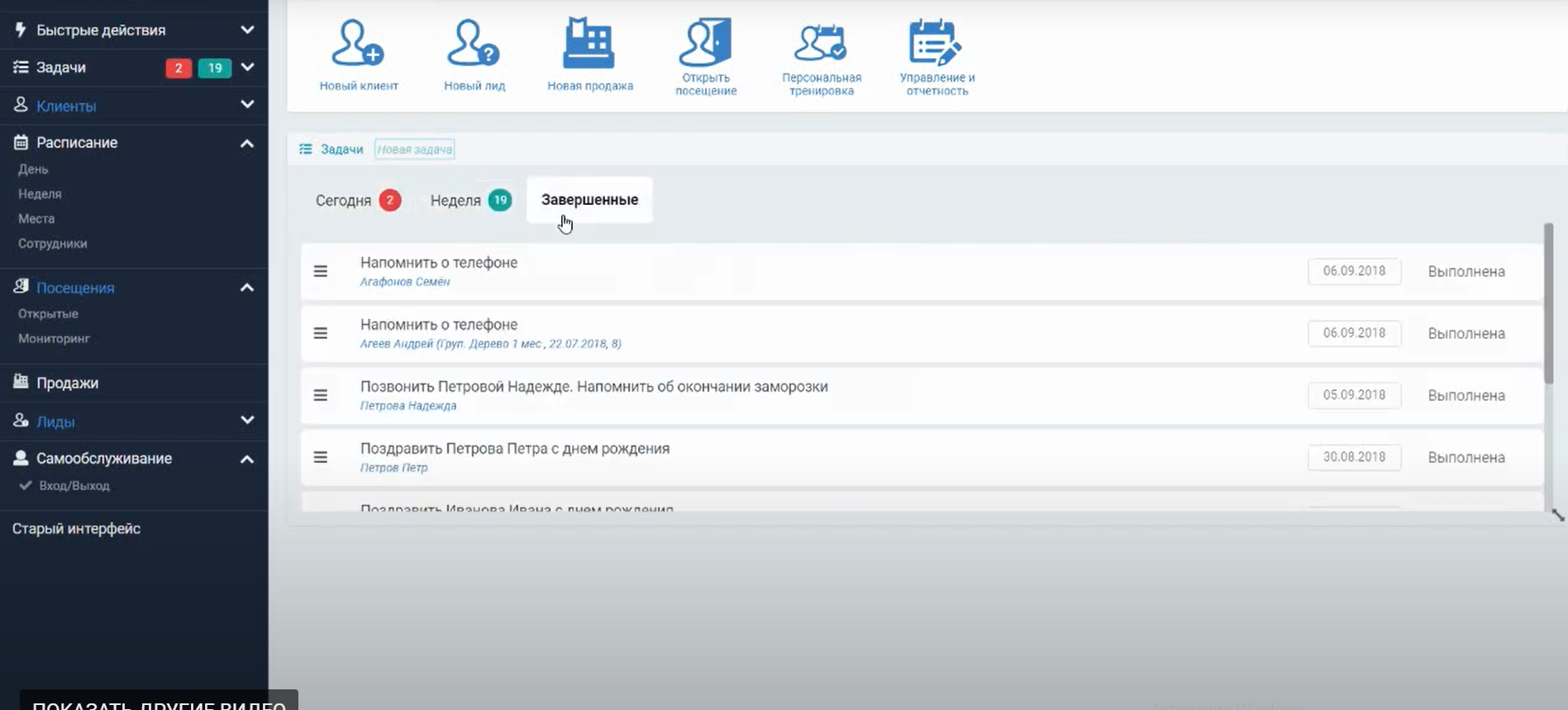


Рисунок 1.2 – Пример интерфейса fitness365

И программ, выдающих огромное количество тренировок в зависимости от ваших параметров и предпочтений на подобие Nike Training

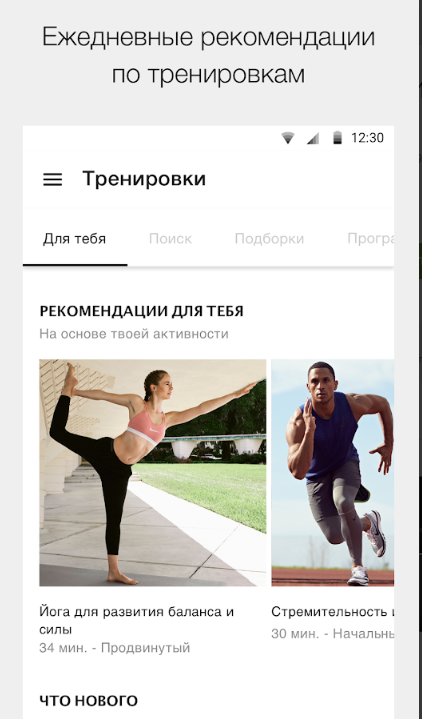


Рисунок 1.3 – Пример интерфейса Nike Training

Проанализировав нынешний рынок оказалось, что приложения, действующие как маркетинговые инструменты для клуба, совсем не работают как инструмент для клиента. То есть нет никакого смысла держать такое приложение у себя на ПК.

С другой стороны, полезных сервисов для занятий фитнессом тоже масса, но они не приносят выгоды фитнесс-клубам. Я пришел к выводу что нужно написать что-то объединяющее эти два аспекта. И нашел идею данного стартапа как наиболее похожую реализацию моей задумки.

Я взял похожий функционал для отслеживания своих тренировок как был представлен в программе Nike Training

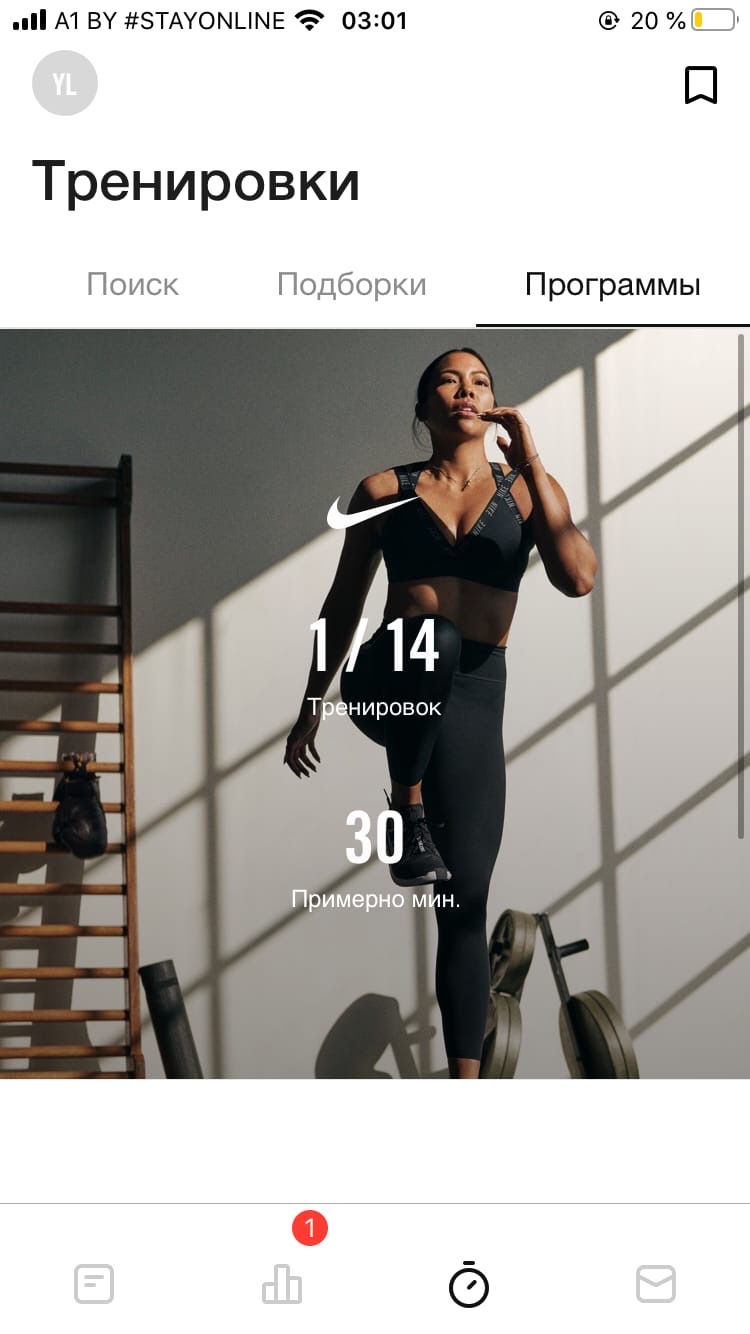


Рисунок 1.4 – Отслеживание прогресса

В программе Nike Training есть возможность отслеживать количество тренировок, получать их от профессионалов, также есть возможность отмечать свой прогресс и осуществлять поиск по критериям, что позволяет нам быстро найти полезную для нас тренировку.

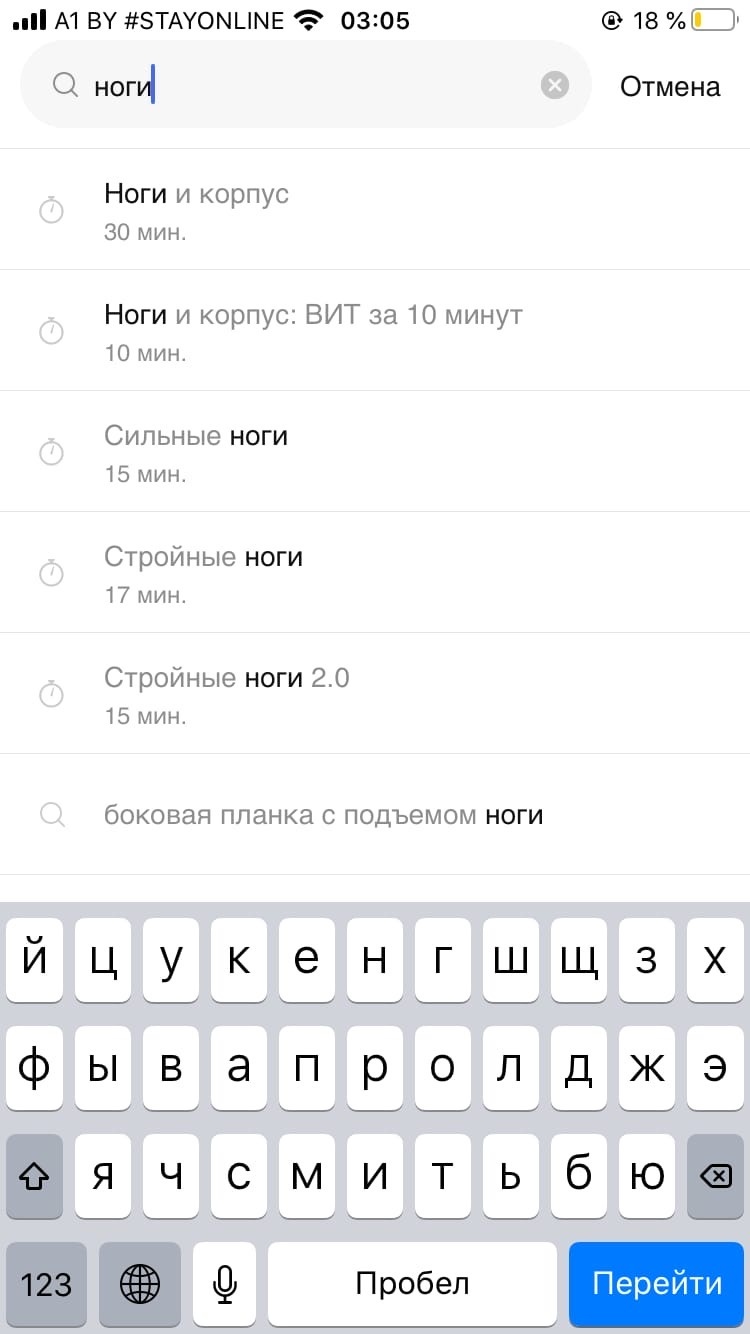


Рисунок 1.5 – Поиск по описанию

Я решил сделать проект на эту тему, так как не видел ещё desktop-версии подобного приложения для Windows.

# Анализ требований к программному средству и разработка функциональных требований

Задачей курсового проекта является разработка приложения, предназначенного для осуществления работы спортзала, а также ряда дополнительных операций: выдачи программы тренировок, поиска, получение информации о тренировках и оставления личных отметок. Абонементы и программы тренировок выдаются непосредственно с самого приложения администратором/модератором.

Программный продукт должен быть реализовать на объектно-ориентированном языке программирования C# с использованием технологии WPF. Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Window Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

Также существует урезанная версия CLR, называющаяся WPF/E, она же известна как Silverlight.

База данных была разработана в Microsoft SQL Server.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире.

SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро.
* Надежность и безопасность. SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.

Для организации баз данных MS SQL Server использует реляционную модель, которая предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Основной используемый язык запросов —Transact-SQL — реализован на структурированном языке запросов (SQL) с расширениями.

Для осуществления связи между базой данных и приложением на C# необходим посредник, которым будет являться технология Entity Framework. Она основана на платформе .NET Framework и предназначена для работы с данными.

Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Эта технология предоставляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

* Database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
* Model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
* Code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

В курсовом проектировании для разработки был выбран первый подход.

Приложение необходимо реализовать с использованием шаблона проектирования MVVM и с организацией взаимодействия с базой данных. В качестве клиентской части необходимо реализовать оконное приложение с использованием стандартных и иных библиотек. Приложение должно иметь возможность запускаться без использования интегрированных средств разработки.

Для разработанного продукта необходимо предусмотреть следующие задачи:

* Главное поле для отображения данных;
* Интуитивно понятный интерфейс;
* Механизм поиска программ тренировок;
* Возможность просмотра информации об актуальном абонементе.

По итогу работы над продуктом, его необходимо протестировать и отладить, проанализировать его возможности и, по необходимости, внести дополнения или улучшения проекта, с возможностью дальнейшего расширения проекта.

Программное средство предоставляет пользователю следующие функциональные возможности:

* Идентификация пользователей
* Получение программы тренировок
* Выборка программы тренировок по различным критериям
* Получение информации об актуальном абонементе
* Добавление личных отметок о тренировках
* Настройка информации в личном профиле

Модератор:

* Выдача программы тренировок
* Выдача абонементов
* Получение информации о пользователе

Администратор может всё то же, что и модератор, и плюс ко всему:

* Удаление пользователей любого типа
* Изменение привилегий у любого пользователя

Наглядно это видно на UML-схеме, изображённой на рисунке 2.1 (Приложение А).

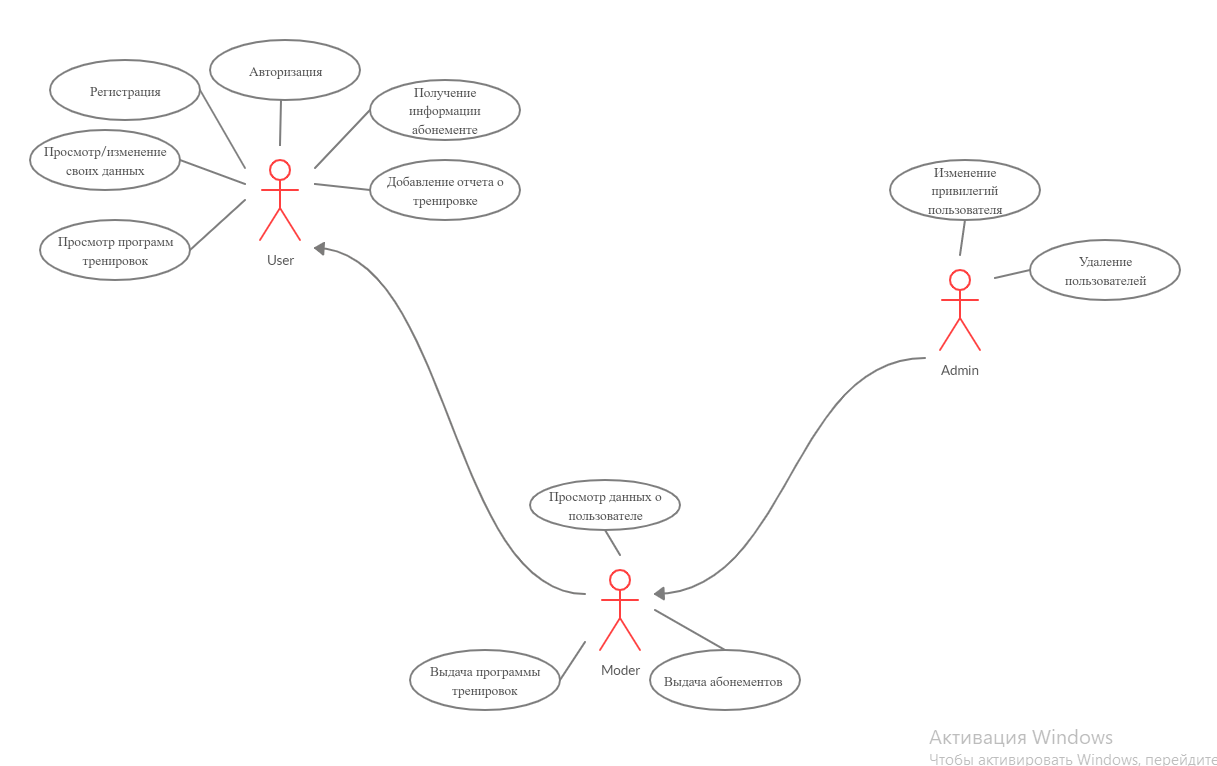


Рисунок 2.1 – «UML-диаграмма возможностей»

# Проектирование программного средства

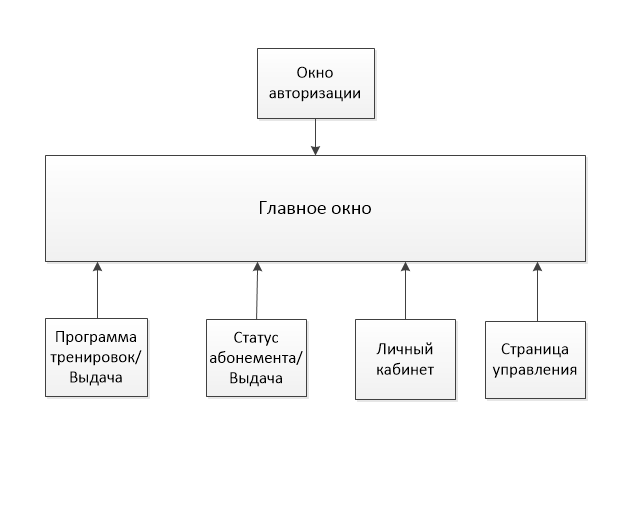


Рисунок 3.1 – «Авторизация и главное окно»

На рисунке 3.1 представлена общая схема работы приложения.

При запуске приложения у нас появляется окно авторизации, которое требует ввода логина и пароля для начала работы.

В случае, если вас нету своей учётной записи, её можно создать, нажав на кнопку «Регистрация» в окне авторизации. Вы будете направлены в окно регистрации, где вам нужно будет заполнить все поля. После успешной регистрации, вам нужно будет вернутся в окно авторизации и ввести ваши данные, указанные при регистрации.

Нажав на соответствующие кнопки в окне авторизации, у вас появится окно, где нужно будет ввести данные для входа в учётную запись, через которую производится вход.

После ввода логина и пароля в соответствующих полях, проверяется правильность ввода данных.

В случае неверного ввода данных, выведется соответствующее сообщение. В ином случае окно авторизации закроется и откроется главное окно приложения.

В главном окне работа идёт в страничном и оконных режимах.

Рассмотрим каждую страницу в отдельности.

Страница «Все тренировки» (рисунок 3.2).

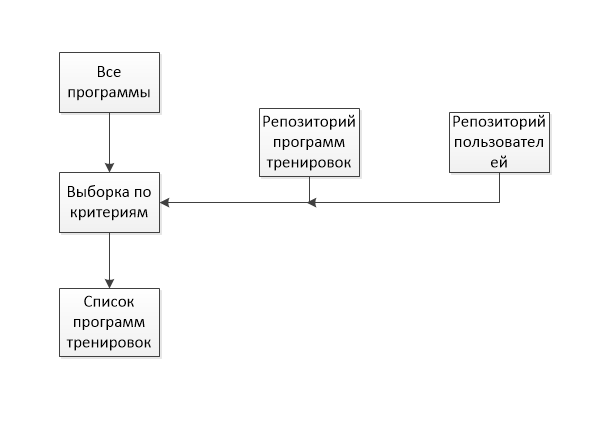


Рисунок 3.2 – «Все Тренировки»

Эта страница содержит список всех тренировок, выданных пользователю.

Также она содержит механизм поиска тренировки по определённым критериям.

Осуществить поиск можно по таким критериям:

* Слово в названии или описании
* Статус
* Комбинации вышеперечисленных

На основе критериев поиска, формируется результирующий набор. При нажатии мы видим более подробную информацию о тренировке и получаем возможность принять ее, чтобы она была отмечена как выполненная.

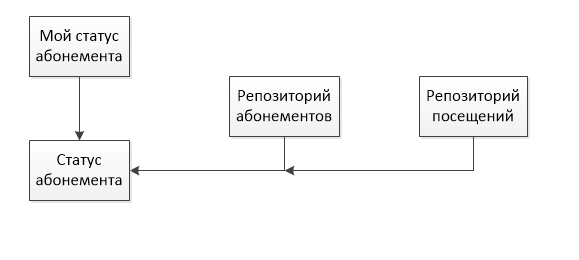


Рисунок 3.3 – «Мой статус абонемента»

На рисунке 3.3 показана схема работы страницы «Мой статус абонемента».

На этой странице пользователь получает информацию об актуальном абонементе и имеет возможность отмечать свои посещения спортзала.

Отметка тренировки происходит в отдельном окне, которое появляется при нажатии на соответствующую кнопку. В этом окне есть ограничения и коррекция недопустимых значений.

Если происходит изменение объявления из раздела «Актуальные», то после изменения, оно автоматически попадает в раздел «Непроверенные» и становится невидимым для других пользователей.

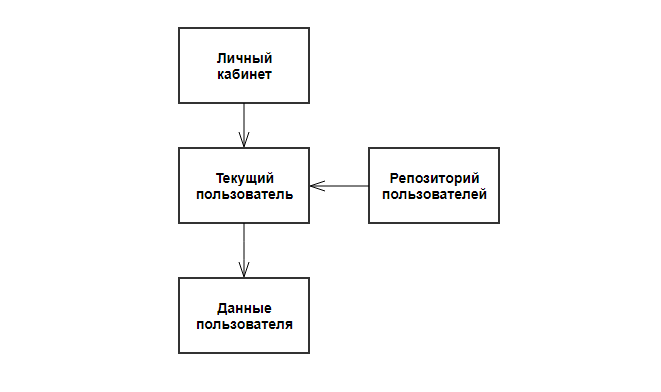


Рисунок 3.4 – «Личный кабинет»

На рисунке 3.4 показана схема страницы «Личный кабинет».

На этой странице можно редактировать свои данные, такие как:

* Имя
* Фамилия
* Mail
* Телефон
* Описание
* Картинка профиля

Под картинкой профиля находится кнопка «Загрузить», нажатие на которую приводит к открытию окна с выбором файла картинки профиля, которую вы хотите установить в качестве основой.

На этой странице есть 2 кнопки: «Сохранить изменения», «Удалить свой профиль».

При нажатии на 1-ю кнопку происходит обновление информации о текущем пользователе в базе данных.

При на нажатии на 2-ю кнопку – удаление всех абонементов, посещений и программ тренировок текущего пользователя, последующее удаление самого профиля и перенаправление на окно авторизации.

Также на эту страницу можно попасть, нажав на элемент, находящийся в верху самого окна чуть правее середины, и содержащий имя, и фамилию текущего пользователя, а также инициалы.

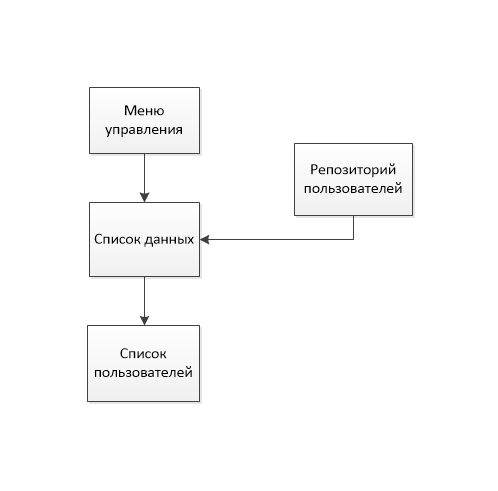


Рисунок 3.5 – «Меню управления»

В этом окне у нас производится администрирование нашего приложения.

Всего есть 3 типа привилегий:

* Пользователь
* Модератор
* Администратор

Пользователь не имеет доступ к этой странице. При попытке зайти в это меню, у него будет выведено сообщение о недостатке прав для доступа к этой странице.

Модератор имеет доступ к этой странице и имеет возможности:

* Просмотр информации о всех пользователях

Администратор имеет все те же права что и модератор, а также возможность назначать и изменять привилегии пользователей и удалять пользователей из приложения.



Рисунок 3.6 – «Страница выдачи тренировок»

При заходе с правами администратор или модератора первые две наши страницы имеют иной функционал. На странице «Страница выдачи тренировок» мы получаем возможность выдать определенному пользователю программу тренировок, для этого мы должны выбрать пользователя из списка, указать время начала выполнения, время конца и дать информацию по выполнению.

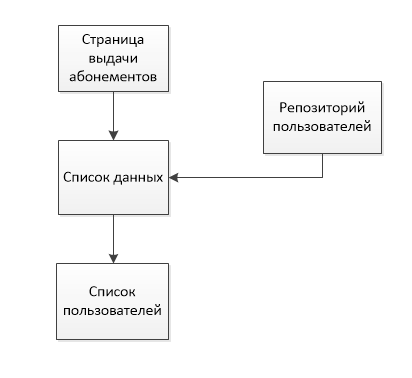


Рисунок 3.7 – «Страница выдачи абонементов»

На странице «Страница выдачи абонементов» мы получаем возможность выдать определенному пользователю абонемент, для этого мы должны выбрать пользователя из списка, количество тренировок и дату окончания абонемента.

Структура проекта представлена в приложении В.

В таблице 3.8 описана структура проекта по пакетам классов.

В этой таблице приведены основные логические составляющие пакеты, которые используются в работе приложения.

Таблица 3.8 – Описание структурных пакетов проекта

|  |  |
| --- | --- |
| Имя пакета | Описание |
| Model | Здесь описаны модели, с которыми происходит вся работа в приложении:   * Пользователь * Программа тренировок * Посещение * Абонемент |
| Repositories | Содержит классы и интерфейсы для работы с данными БД. Они являются связующим элементом между данными в БД и теми данными, с которыми работает приложение |
| Resources | Файлы стилей |
| View | Содержит все представления, которые позволяют пользователю работать с приложением. Они описывают графическую составляющую приложения. |
| ViewModel | Содержит логику, которая позволяет получить данные при помощи View, обработать их, использую при этом Model, после чего передать в базу данных. |
| App.config | Файл конфигурации приложения |
| App.xaml | Основная задача данного файла состоит в определении ресурсов, общих для приложения |

Структуры данных, а также их зависимости друг от друга представлены в приложении Б.

Как показано в приложении Б, главным классом является User, который хранит в себе поля, связанные с пользователями, от него по полю id идет связь к двум таблицам, а именно TrainProgram и Abonement. Таблица TrainProgram отвечает за программы, тренировок, которые выдаются определенному пользователю. Таблица Abonement отвечает за абонементы, выдаваемые пользователям, а также от нее зависит таблица Visiting которая хранит в себе информацию о посещениях тренировок.

Диаграмма классов показана в приложении Г.

# Создание (реализация) программного средства

При создании приложения использовался паттерн проектирование MVVM. Он заключается разделении представления от бизнес логики. Это достигается за счёт ввода новой логической конструкции ViewModel. ViewModel связывает представление и бизнес логику приложения.

Пакетная структура проекта показана на рисунке 4.1

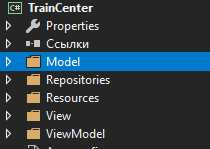


Рисунок 4.1 – «Структура пакетов проекта»

В Model находится структура модели базы данных приложения. При проектировании базы данных данного курсового проекта для взаимодействия с базой данных используется Entity Framework.

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из бд, но и получать объекты, связанные различными ассоциативными связями.

Нередко все сущности (модель) наследует интерфейс INotifyPropertyChanged или INotifyCollectionChanged, которые позволяют отлавливать изменения и правильно на их реагировать и сохранять.

Для того, чтобы реализовать интерфейс INotifyPropertyChanged, нужно объявить свойство PropertyChanged и метод OnPropertyChanged().

Для того, чтобы начать отслеживание свойства, необходимо вызывать метод OnPropertyChanged(“Имя свойства”) как показано в приложении Д.

Для удобной работы с данными, используется паттерн Repository. Репозиторий позволяет абстрагироваться от конкретных подключений к источникам данных, с которыми работает программа, и является промежуточным звеном между классами, непосредственно взаимодействующими с данными, и остальной программой. На рисунке 4.2 показана структура.

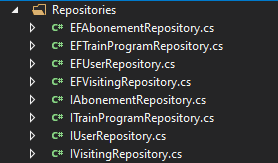


Рисунок 4.2 – «Паттерн Repository»

Интерфейсы IUserRepository, ITrainProgramRepository, IVisitingRepository, IAbonementRepository имплицируются в соответствующих классах и указывают: какие методы (функционал) должен иметь этот репозиторий. Репозитории показаны в приложении Е.

Интерфейсы репозиториев также показаны в приложении Е.

На рисунке 4.3 показана структура пакета Resources, в котором находятся файлы стилей кнопок (ButtonDictionary.xaml), а также стили для других элементов управления (Styles.xaml). Эти файлы написаны на языке разметки xaml.



Рисунок 4.3 – «Структура пакета Resources»

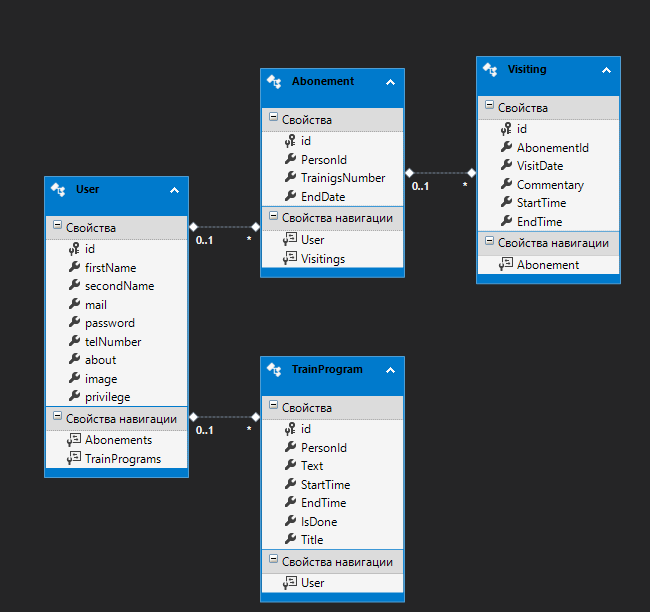
В проекте используется паттерн MVVM, а это значит, что должны присутствовать ViewModels, которые связывают View и Model. Во ViewModel информация обрабатываться и выводится/записываться в нужном виде. Структура пакета ViewModel представлена в приложении К.

В приложении И показана структура пакета View, в котором хранятся файлы представлений на языке разметки xaml, а также соответствующие им файлы на языке C#. В этом пакете находятся все «окна» и «страницы», которые мы видим, используя приложение. Назначение каждого файла описано в таблице ниже (таблица 4.4)

Таблица 4.4 – «Описание файлов View»

|  |  |
| --- | --- |
| **Название** | **Краткое описание** |
| AddMyTrainWindow.xaml (окно) | Окно добавления отметки о тренировке |
| AdminPage.xaml (страница) | Окно для управления работы приложения и ведения проверки данных |
| AlertWindow.xaml (окно) | Окно уведомления |
| AllTrainings.xaml (страница) | Страница, отображающая все программы тренировок |
| AuthWindow.xaml (окно) | Окно авторизации. Требует ввода логина и пароля |
| DialogWindow.xaml (страница) | Окно, в котором нужно выбрать вариант ответа «Да» или «Нет» на заданный там вопрос |
| GiveAbonementsPage.xaml (страница) | Окно выдачи абонементов |
| GiveTrainProgramPage.xaml (страница) | Окно выдачи программы тренировок |
| MainWindow.xaml (окно) | Главное окно приложения |
| MyStatusPage.xaml (страница) | Страница, на которой показана информация о состоянии абонемента |
| PersonAreaPage.xaml (страница) | Страница профиля, в которой можно посмотреть или изменить свои личные данные |
| RegistrationWindow.xaml (окно) | Окно регистрации нового пользователя |
| UserViewModel.xaml (окно) | Окно просмотра информации о пользователе (доступно для модератора/админа) |
| ViewWindow.xaml (окно) | Основное окно для просмотра информации о тренировке |

На рисунке 4.5 изображена диаграмма таблиц со связями в базе данных.

Рисунок 4.5 – «Диаграмма таблиц»

Описание таблицы TrainProgram(таблица 4.6)

Таблица 4.6 – «описание TrainProgram»

|  |  |
| --- | --- |
| id | Идентификатор |
| PersonId | Идентификатор пользователя |
| Text | Описание тренировки |
| StartTime | Время начала |
| EndTime | Время конца |
| IsDone | Статус выполнения |
| Title | Оглавление |

Описание таблицы Abonement (таблица 4.7)

Таблица 4.7 – «описание Abonement»

|  |  |
| --- | --- |
| id | Идентификатор |
| PersonId | Идентификатор пользователя |
| TrainingsNumber | Количество тренировок |
| EndDate | Дата конца абонемента |
|  |  |

Описание таблицы (таблица 4.8)

Таблица 4.8 – «описание Users»

|  |  |
| --- | --- |
| Id | Идентификатор |
| firstName | Имя |
| secondName | Фамилия |
| mail | Email адрес |
| password | Хеш пароля |
| telNumber | Номер телефона |
| about | О себе |
| image | Поток байтов (изображение) |
| privilege | Привилегии |

Описание таблицы Visiting (таблица 4.9)

Таблица 4.9 – «описание Visiting»

|  |  |
| --- | --- |
| id | Идентификатор |
| AbonementId | Название объявления/товара |
| VisitDate | Дата посещения |
| Commentary | Комментарий |
| StartTime | Время начала |
| EndTime | Время конца |

# Тестирование, проверка работоспособности и анализ полученных результатов

При тестировании данного приложения, были применены сценарии, которые могли бы привести к ошибке. В этой главе мы рассмотрим некоторые такие сценарии и посмотрим на их обработку.

В момент регистрации, возможна такая ситуация, в которой пользователь ничего не ввёл или ввёл неверные данные. Обработка данного сценария приведена на рисунке 5.1

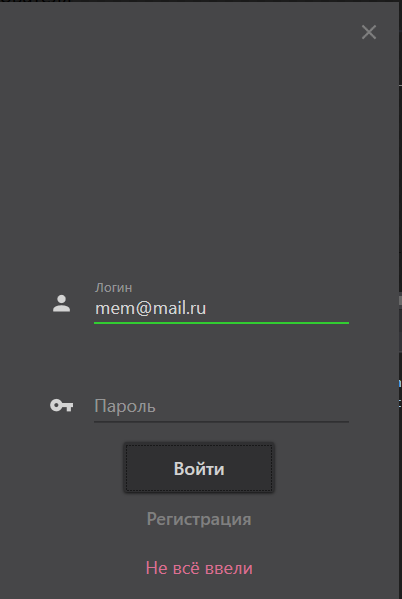


Рисунок 5.1 – «Ошибка корректности ввода»

При неверных данных – рисунок 5.2

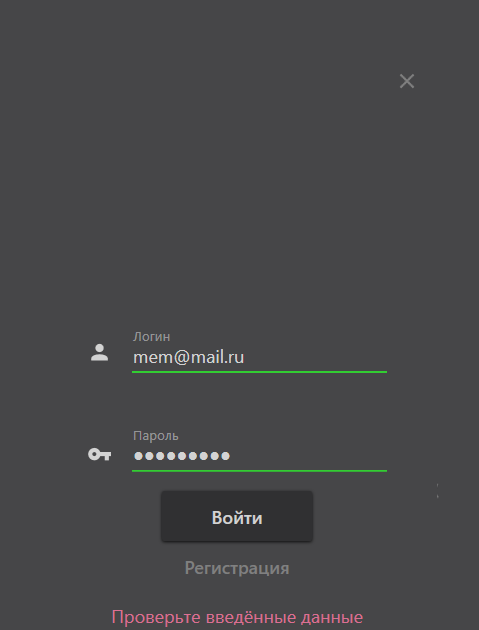


Рисунок 5.2 – «Сценарий, с неверными данными»

Также возможен такие случаи:

* Пользователь хочет зарегистрироваться, при этом не указав никаких данных (рисунок 5.3)
* Пользователь ввел некорректный email или некорректный пароль (рисунок 5.4)

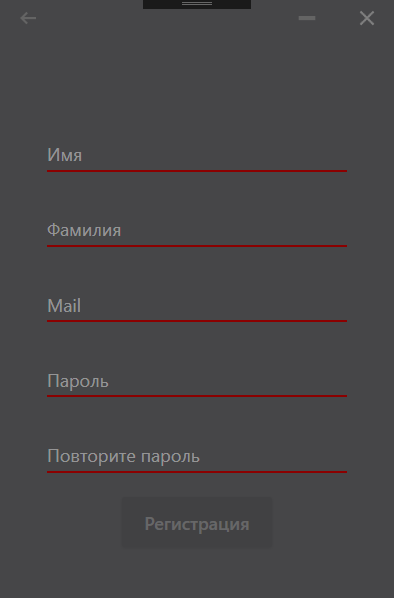


Рисунок 5.3 – «Отсутствие данных»

Как видно на рисунке 5.3, кнопка регистрации просто не будет активна, если не введены данные, необходимые для регистрации.

При несовпадении паролей будет точно такая ситуация – неактивная кнопка «Регистрация»

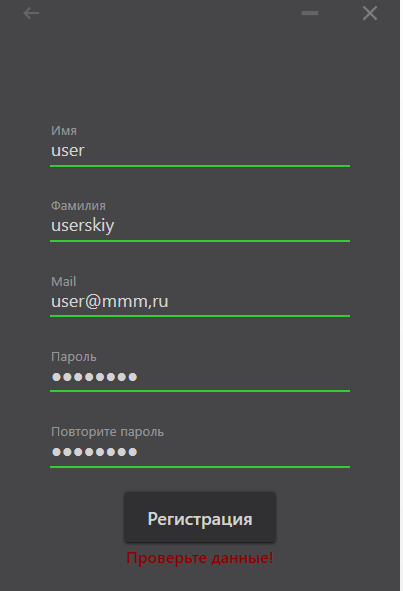


Рисунок 5.4 – «сценарий, при некорректном вводе почты»

Также возможен такой исход, при котором пользователь хочет зарегистрироваться при помощи такого mail/логина, который уже есть. Обработка такого рода сценариев показана на рисунке 5.5.

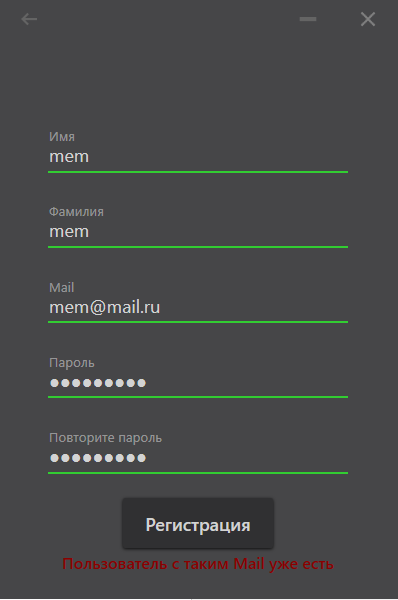


Рисунок 5.5 – «Регистрация пользователя с занятым логином»

Также может возникать такая ситуация, когда пользователь, который не имеет достаточные привилегии, хочет зайти в меню управления. Ему не даст туда зайти ограничение по привилегиям. Пример показан на рисунке 5.6

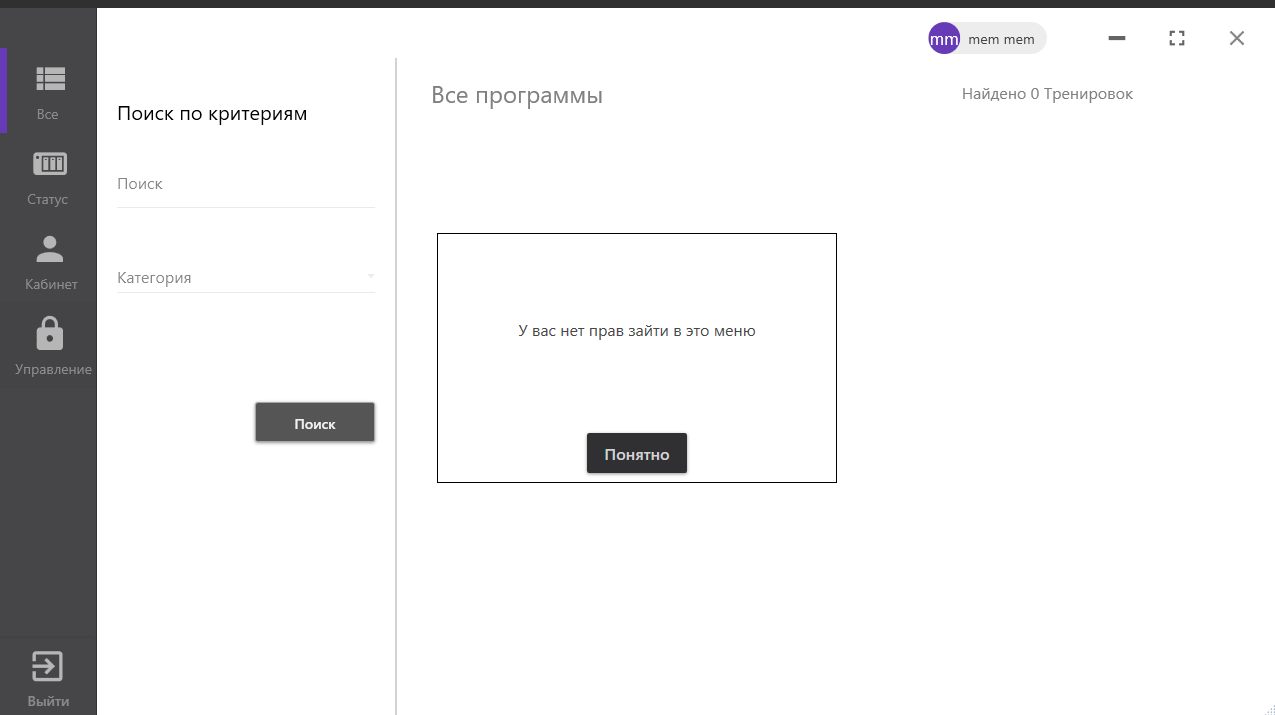


Рисунок 5.6 – «ограничение привилегий»

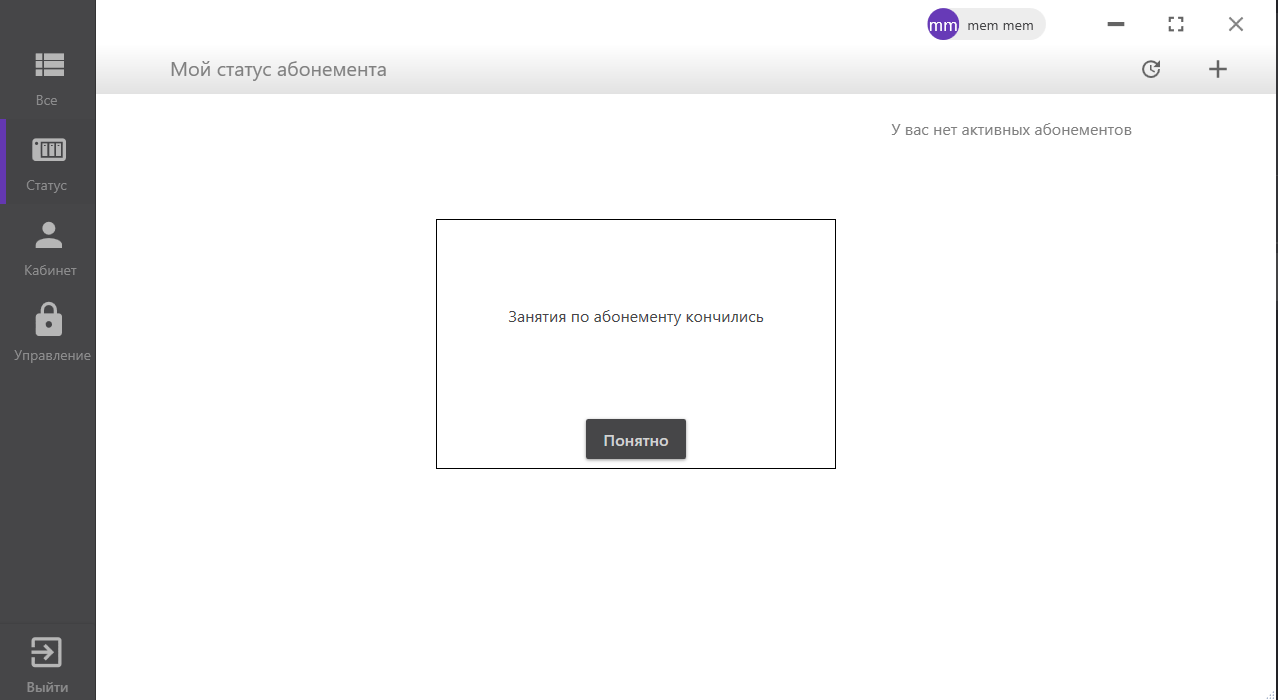
В случае, когда у вас нет активного абонемента или закончились тренировки, вы не можете добавлять личную отметку.

Рисунок 5.7 – «ограничение добавлений»

# Руководство по использованию

При запуске приложения у нас появляется окно авторизации, которое требует ввода логина и пароля для начала работы.

В случае, если вас нету своей учётной записи, её можно создать, нажав на кнопку «Регистрация» в окне авторизации. Вы будете направлены в окно регистрации, где вам нужно будет заполнить все поля. После успешной регистрации, вам нужно будет вернутся в окно авторизации и ввести ваши данные, указанные при регистрации.

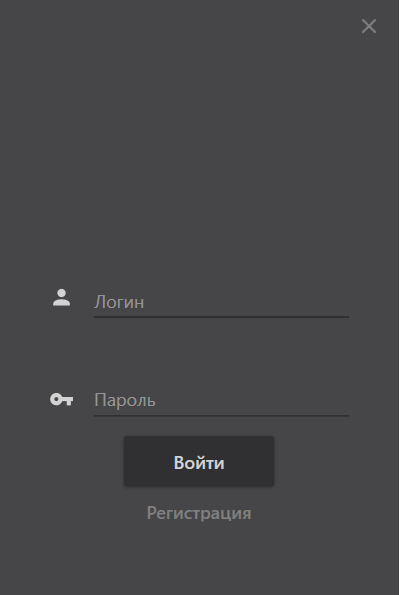


Рисунок 6.1 – «Окно авторизации»

После успешного входа, мы попадаем в главное окно, которое изображено на рисунках 6.2.1, 6.2.2

У роли пользователь и модератор/админ разные первые два окна.

В случае с пользователем мы видим рисунок 6.2.1. Здесь у пользователя есть возможность просмотреть тренировку полученную от модератора, отметить ее как выполненную и осуществить поиск тренировок по критериям.

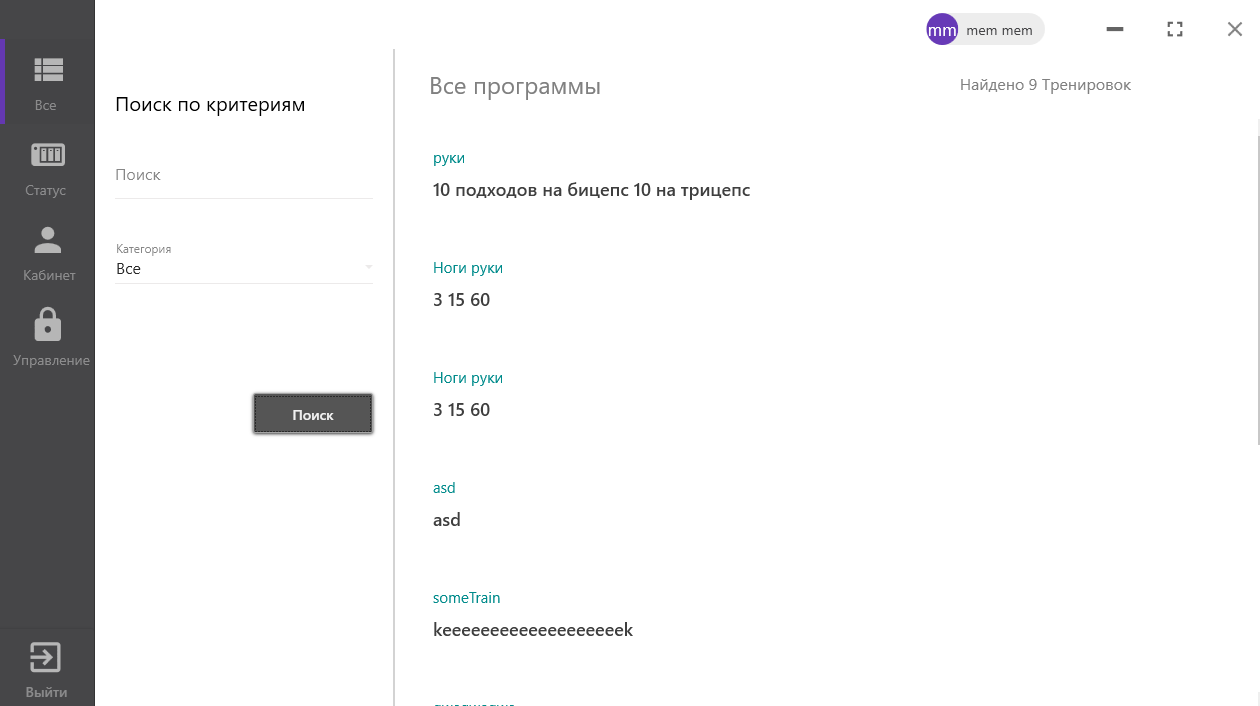


Рисунок 6.2.1 – «Главное окно пользователя»

В случае захода под привилегией модератор/админ мы увидим главное окно Рисунок 6.2.2. Здесь есть возможность выдать тренировку пользователю выбрав его из выпадающего меню, сделав описание тренировки и выбрав время начала и время конца тренировки.

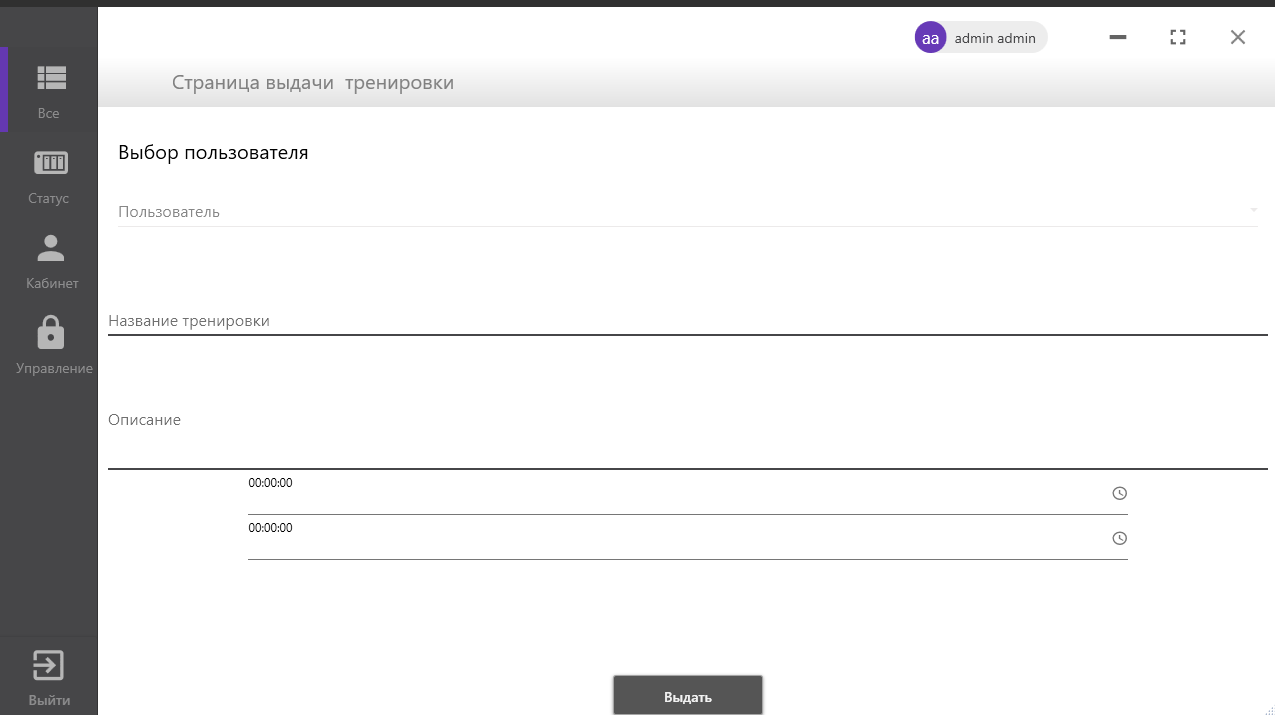


Рисунок 6.2.2 – «Главное окно пользователя»

Для получения информации о статусе актуального абонемента и добавлении личной отметки о тренировке нужно перейти на вкладку статус. Здесь у пользователя есть возможность добавления своей тренировки нажав на кнопку «+».

Ниже показан рисунок 6.3, на котором показана страница «Статус».

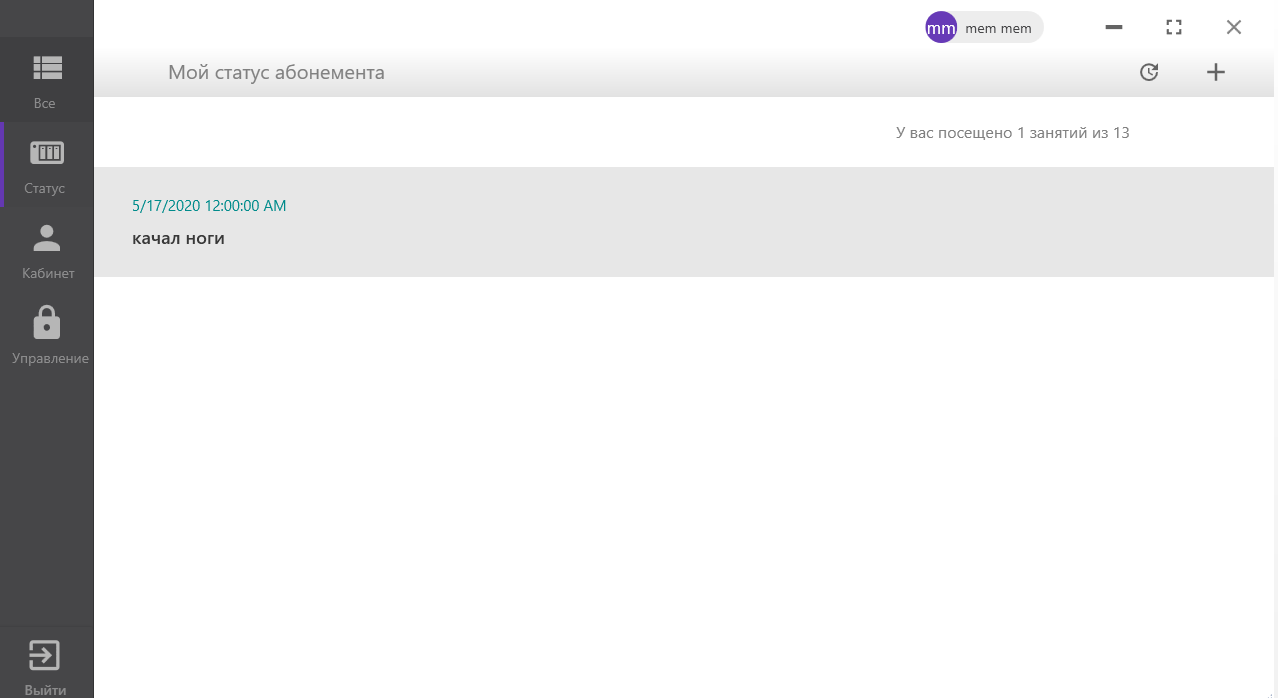


Рисунок 6.3.1 – «Страница Статус»

Если перейти на страницу статуса имея привилегию модератора, то получим возможность выдачи абонементов пользователям выбрав пользователя из выпадающего меню, количество тренировок и дату окончания абонемента (рисунок 6.3.2)

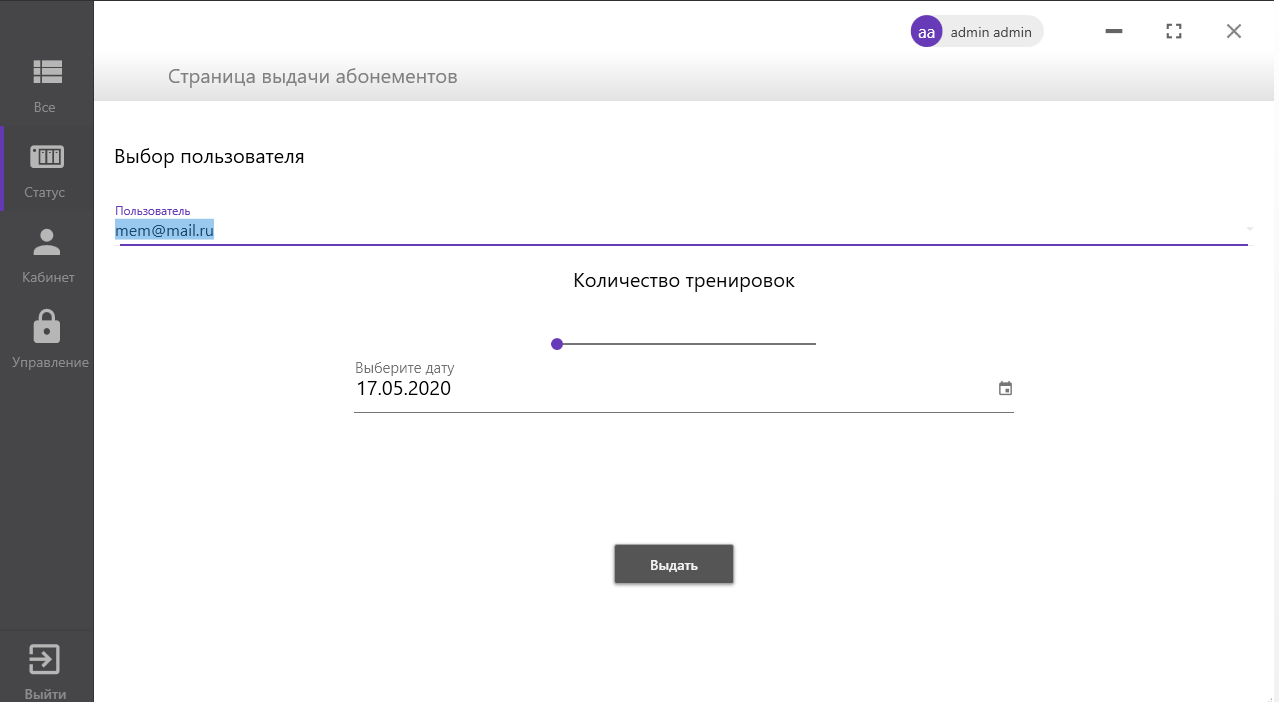


Рисунок 6.3.2 – «Окно добавления объявления»

Ниже, на рисунке 6.4, представлена страница «Кабинет». В которой есть возможность изменять данные о себе или совсем удалить себя из базы данных.

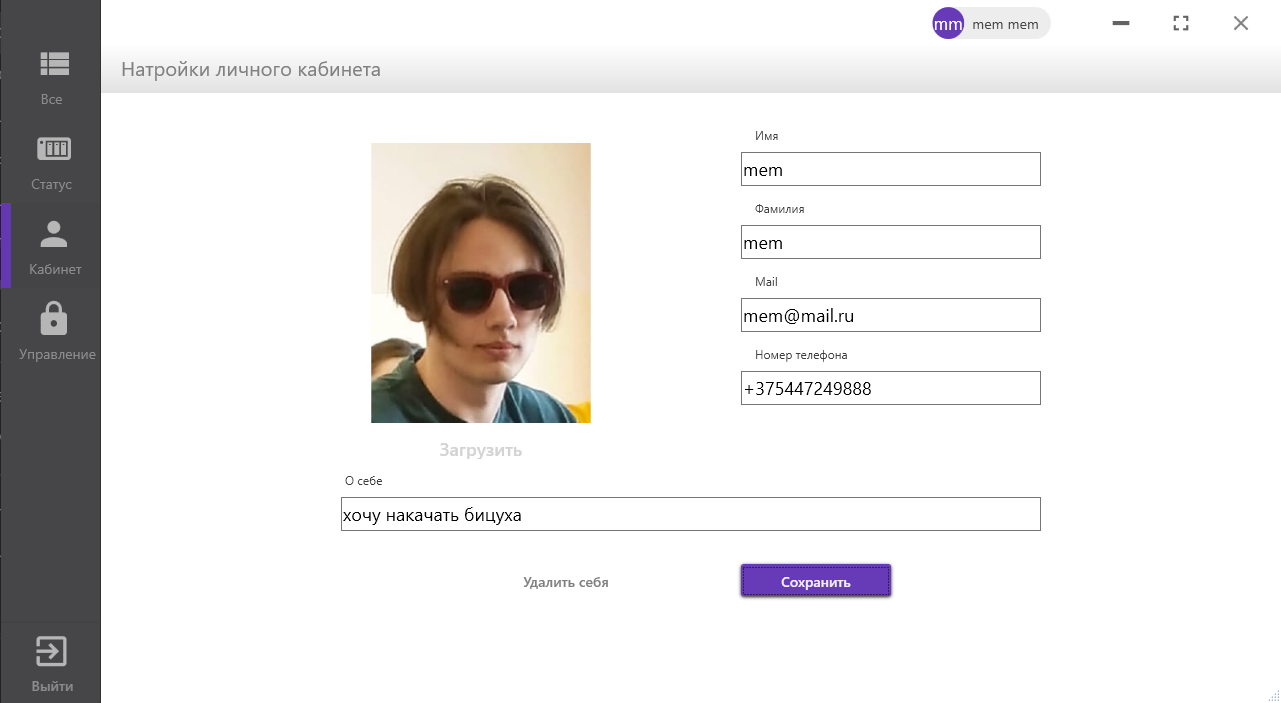


Рисунок 6.4 – «Страница Кабинет»

На рисунке 6.5 показана страница администратора/модератора.

На этой странице можно совершать действия над пользователями.

На рисунке 6.5 представлен скриншот страницы администратора/модератора.

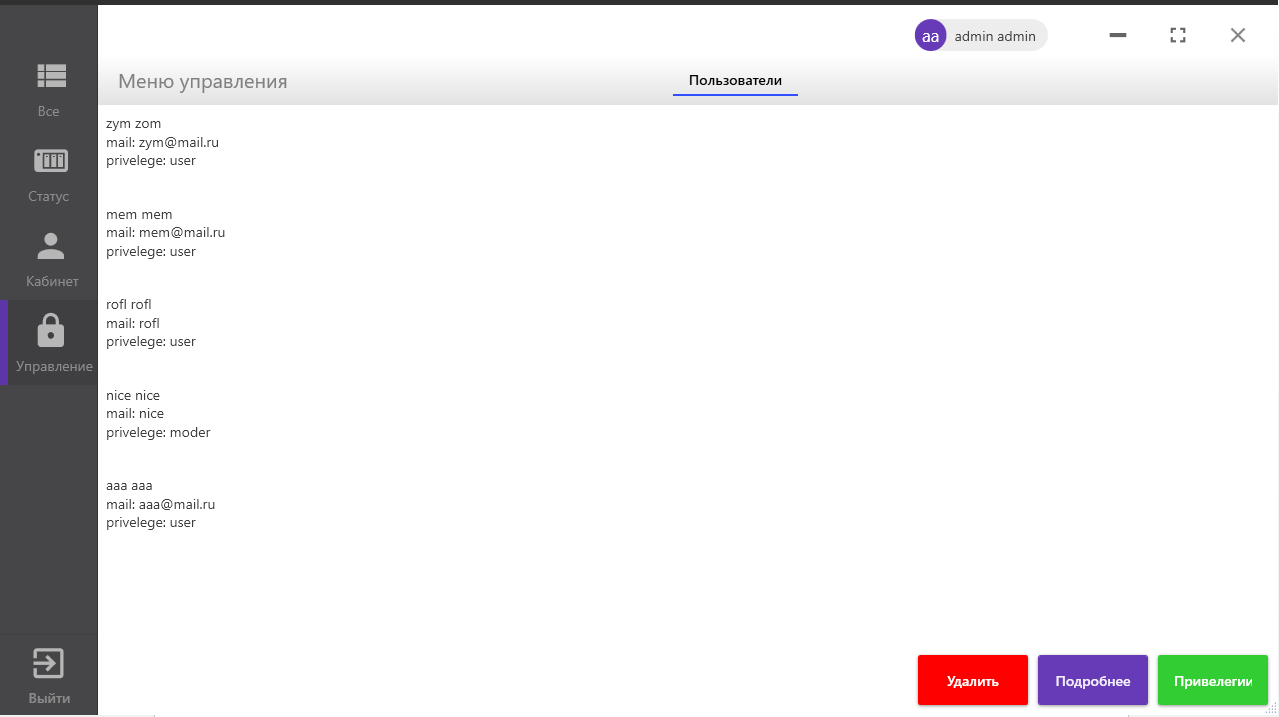


Рисунок 6.5– «Меню администратора»

# ЗАКЛЮЧЕНИЕ

В данном курсовом проекте было разработано программное средство «Тренировочный центр» при помощи языка программирования C#, API-интерфейса Windows Presentation Foundation, технологии Entity Framework и базы данных Microsoft SQL Server. При выполнении курсового проекта использовались принципы и приемы ООП.

Разработанное программное средство предоставляет пользователю следующие функциональные возможности:

* Идентификация пользователей
* Получение программы тренировок
* Выборка программы тренировок по различным критериям
* Получение информации об актуальном абонементе
* Добавление личных отметок о тренировках
* Настройка информации в личном профиле

Модератор:

* Выдача программы тренировок
* Выдача абонементов
* Получение информации о пользователе

Администратор может всё то же, что и модератор, и плюс ко всему:

* Удаление пользователей любого типа
* Изменение привилегий у любого пользователя

Проектирование осуществлялось по паттерну проектирования MVVM.

Для удобной работы с данными, был реализован паттерн Repository.

Приложение хорошо работает с базой данных. В ходе тестирования, не было выявлено нарушений в работе приложения с базой данных.

Для разработки дизайна использовалась библиотека MaterialDesign, которая имеет огромное количество красивых и удобных элементов управления.

Приложение было успешно протестировано, что означает пригодность этого приложения для своей цели.

Для хранения исходного кода и удобства контроля версий проекта использовался крупнейший веб-сервис для совместной разработки – GitHub. Ссылка на проект: (<https://github.com/zym43210/TrainCenter>).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

[1]. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2018. – 175 с.

[2]. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с. [3]. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: http://msdn.microsoft.com/library/

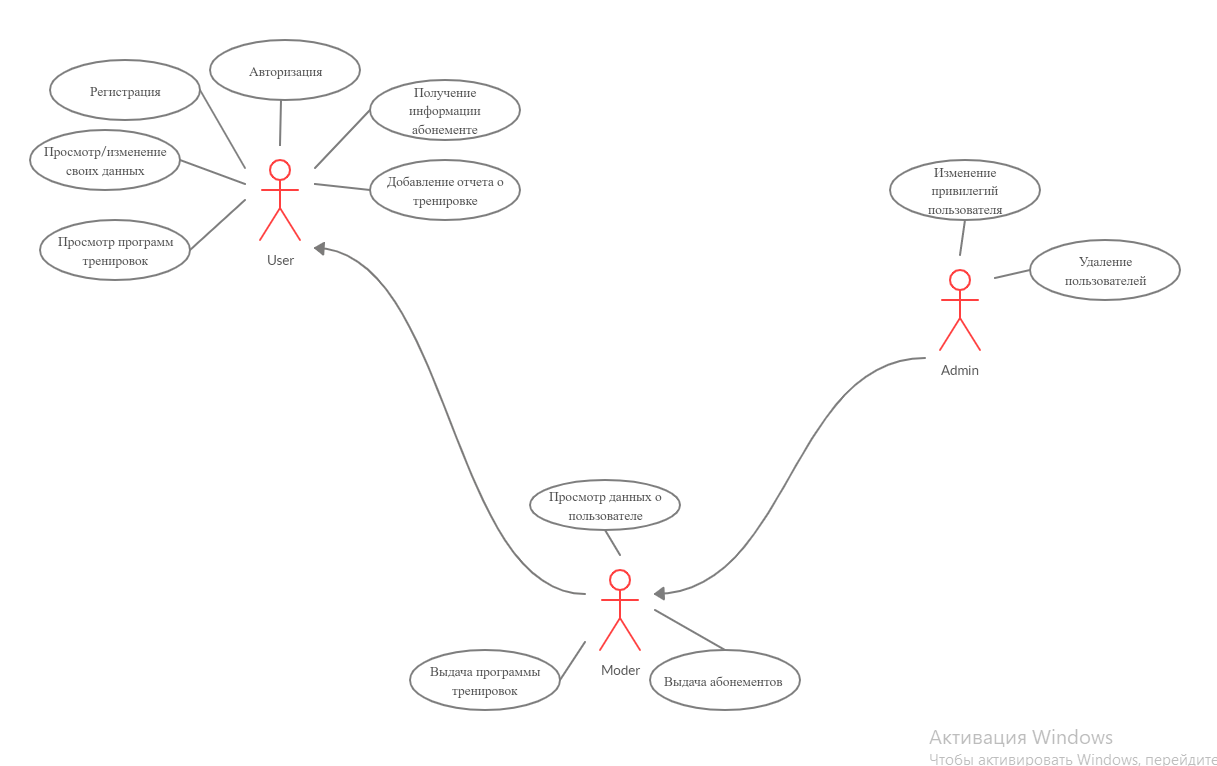
[4]. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: https://metanit.com

[5]. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: https://professorweb.ru

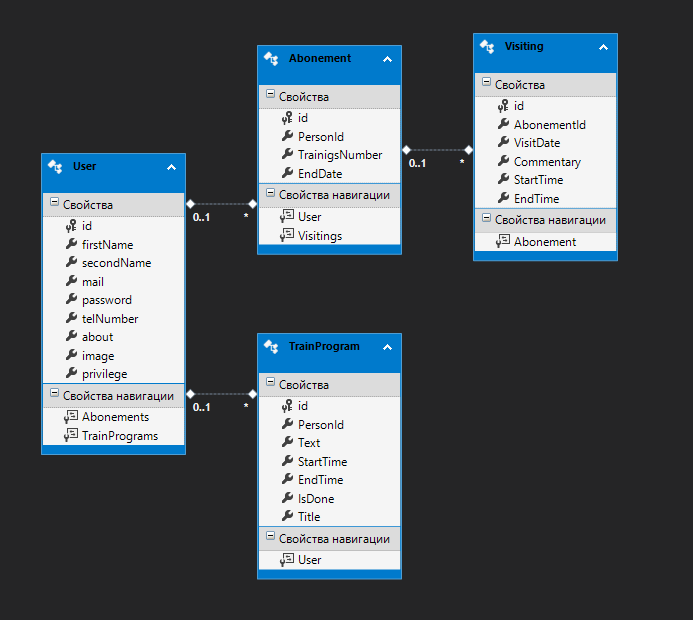
[6]. StackOverflow [Электронный ресурс] / Режим доступа: <https://stackoverflow.com>

[7]. Хабр [Электронный ресурс] / Режим доступа: <https://habr.com>

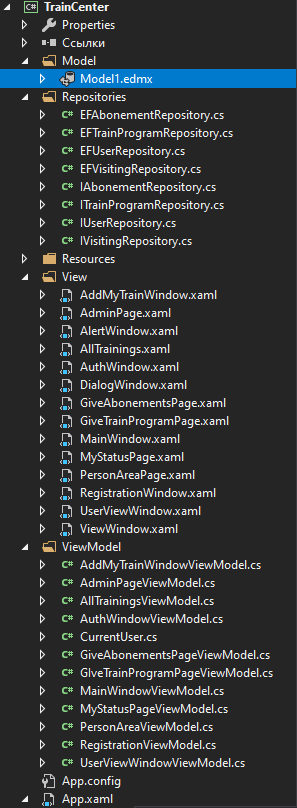
Приложение А



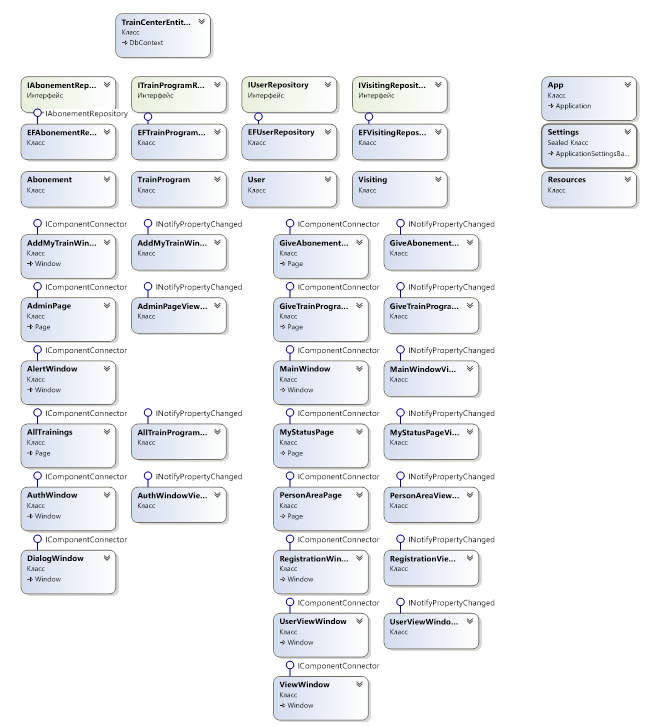
Приложение Б



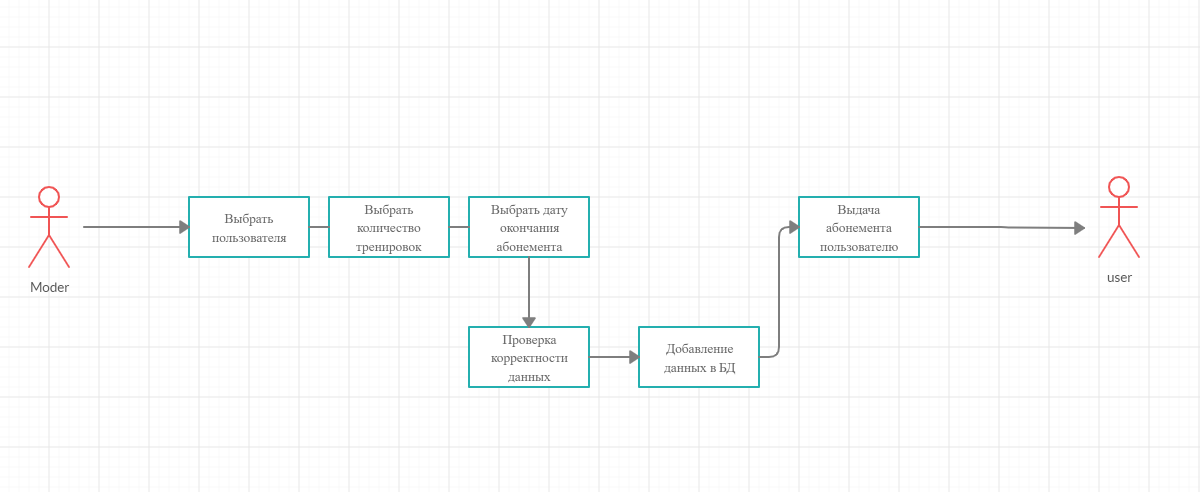
Приложение В



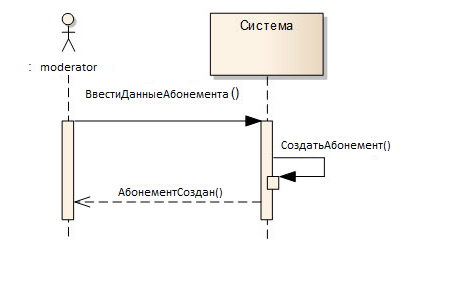
Приложение Г



Приложение Д

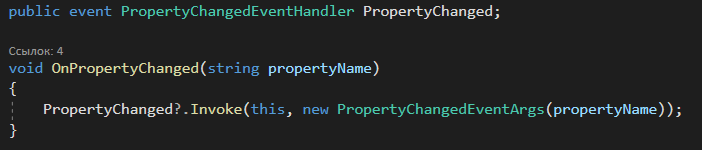


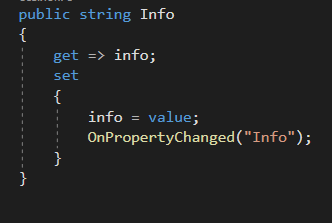
Приложение Е



Приложение Ж

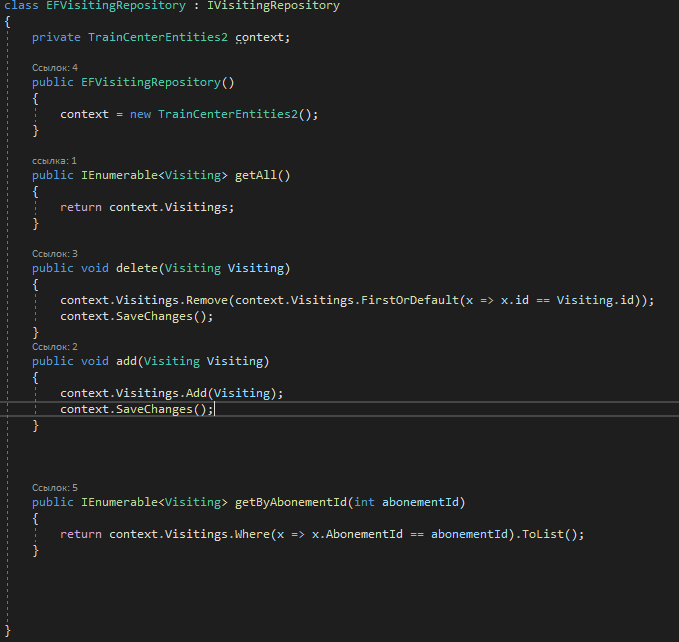
Листинг 1 – «INotifyPropertyChanged»

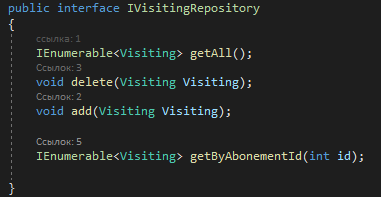




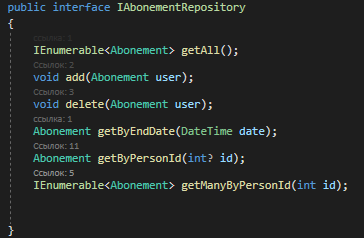
Приложение З

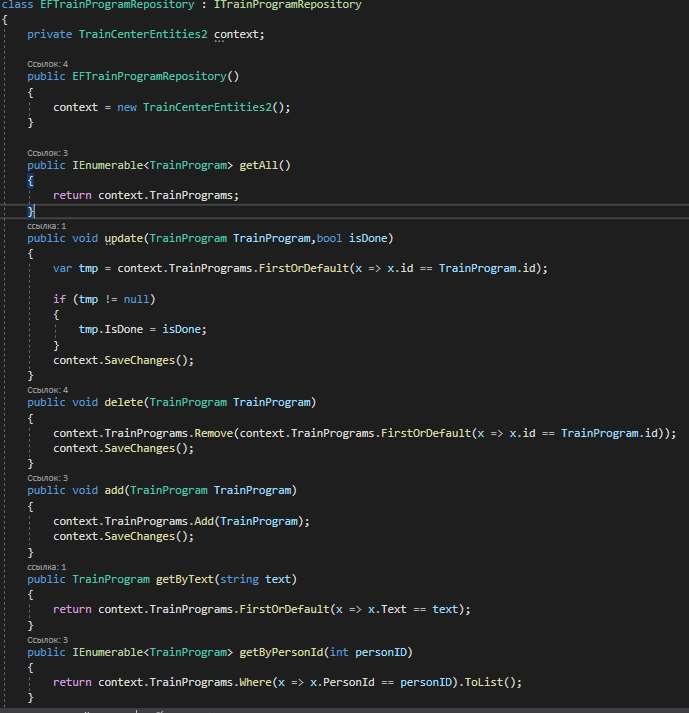
Листинг 2 – «Repositories»

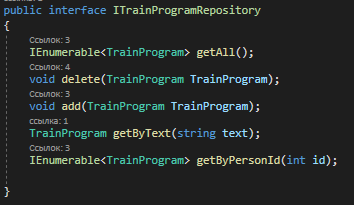


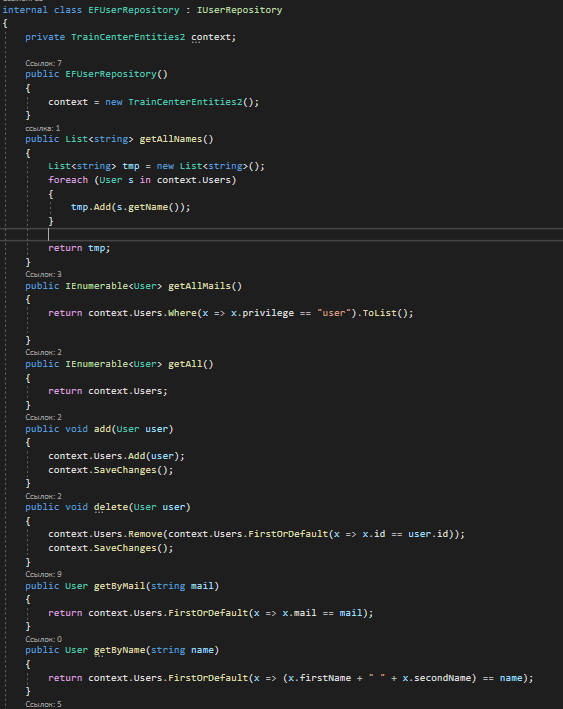


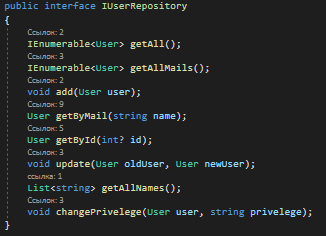






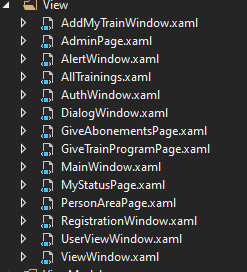






Приложение И

Листинг 5 – «Структура пакета View»



Приложение К

Листинг 5 – «Структура пакета ViewModel»

