

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждения 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 приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и  
стандарты проектирования»

Тема Программное средство «Тренажёрный зал»

Исполнитель  
студент (ка) 3 курса группы 1 Велютич Дмитрий Игоревич  
(Ф.И.О.)

Руководитель работы ассистент Е. А. Гончар  
(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой \_\_\_\_\_  
Председатель \_\_\_\_\_  
(подпись)

Минск 2024

Утверждаю  
Заведующий кафедрой ИС  
\_\_\_\_\_ Смелов В. В.  
подпись инициалы и фамилия  
“ ” 2024г.

- Проектирование архитектуры проекта (структура модулей, классов).

- Разработка функциональной модели и модели данных ПС (выполняемые функции).
- Тестирование.
- Заключение.
- Список используемых источников.
- Приложения.

#### 4. Форма представления выполненного курсового проекта:

- Теоретическая часть курсового проекта должна быть представлена в формате docx. Оформление записки должно быть согласно выданным правилам.
- Листинги программы представляются в приложении.
- Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить диск, указанный преподавателем.

#### *Календарный план*

№ п/п	Наименование этапов курсового проекта	Срок выполнения этапов проекта	Примечание
1	Введение	19.09.2024	
2	Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования	12.10.2024	
3	Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных)	26.10.2024	
4	Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса	02.11.2024	
5	Кодирование программного средства	23.11.2024	
6	Тестирования и отладка программного средства	30.11.2024	
7	Оформление пояснительной записки	07.12.2024	
9	Защита проекта	20.12.2024	

5. Дата выдачи задания 17.09.2024

Руководитель \_\_\_\_\_  
(подпись)

Е. А. Гончар

Задание принял к исполнению \_\_\_\_\_ Д. И. Велютнич  
(дата и подпись студента)

Минск 2024

## Содержание

ВВЕДЕНИЕ .....	5
1 Аналитический обзор литературы и формирование требований .....	6
1.1 Анализ прототипов .....	6
1.1.1 GymEasy.....	6
1.1.2 FitSoft .....	7
1.1.3 SmartTimer .....	8
1.1.4 Альтернативный вариант .....	8
1.2 Требования к проекту .....	9
2 Анализ требований к программному средству и разработка функциональных требований.....	10
3 Проектирование программного средства.....	12
3.1 Общая структура .....	12
3.2 Разработка общей схемы приложения .....	14
3.3 Взаимоотношение между классами .....	16
3.4 Модель базы данных.....	16
3.5 Проектирование архитектуры приложения.....	18
4 Реализация программного средства .....	19
4.1 Основные классы программного средства .....	19
4.2 Авторизация.....	19
4.3 Регистрация.....	19
4.4 Просмотр услуг клуба.....	20
4.5 Заполнение формы оформления заказа .....	20
4.6 Просмотр забронированных заказов.....	20
4.7 Заполнение и просмотр отзывов.....	20
4.8 Реализация вспомогательных классов .....	21
5 Тестирование, проверка работоспособности и анализ полученных результатов.....	22
5.1 Тестирование авторизации и регистрации .....	22
6 Руководство по использованию .....	25
ЗАКЛЮЧЕНИЕ.....	29
Список использованных источников.....	30
ПРИЛОЖЕНИЕ А .....	31
ПРИЛОЖЕНИЕ Б .....	32
ПРИЛОЖЕНИЕ В.....	33
ПРИЛОЖЕНИЕ Г .....	34

## ВВЕДЕНИЕ

Тренажёрные залы становятся всё более популярными в разных странах, открывая отличные перспективы как для бизнеса, так и для людей, стремящихся вести здоровый образ жизни.

Многие выбирают спорт и фитнес как способ досуга, но не всегда хотят или могут покупать абонемент заранее или заниматься организацией своих тренировок. Поэтому удобнее посещать тренажёрный зал с возможностью бронирования.

Программное средство «Тренажёрный зал» — это приложение, позволяющее пользователям бронировать тренировки и оборудование в зале.

Приложение включает два типа пользователей: администратор и пользователь, каждый из которых имеет свои полномочия. Пользователи могут:

- бронировать тренажёры и время занятий;
- отслеживать список своих бронирований;
- сортировать данные по дате, времени начала, продолжительности или типу оборудования.

Администраторы обладают полными правами. Они могут:

- удалять или редактировать аккаунты;
- управлять расписанием;
- изменять настройки зала.

Для работы с приложением предусмотрена авторизация. Это значит, что перед бронированием тренажёра или времени тренировки пользователь должен зарегистрироваться. Все данные пользователей будут храниться в базе данных, с которой приложение будет взаимодействовать при входе или регистрации.

С помощью этого приложения пользователи смогут быстро получать информацию о доступном времени, тренажёрах и создавать комфортный график тренировок.

# 1 Аналитический обзор литературы и формирование требований

## 1.1 Анализ прототипов

Были проанализированы цели и задачи, поставленные в данном курсовом проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

### 1.1.1 GymEasy

Обычное программное средство для тренажерного зала — “GymEasy”[1], Главные преимущества — возможность бронирования тренажёров и занятий, а также инструменты для создания тарифных планов. Программное средство имеет простой дизайн и базовый функционал.

Однако, как и у многих других решений для тренажёрных залов, есть определённые недостатки. Например, система может включать большое количество тарифов с недостаточной информацией, что затрудняет выбор и сравнение. К тому же, приложение разрабатывалось давно и больше не поддерживается.

Интерфейс программного средства представлен на рисунке 1.1.

#### Создание тарифа

The screenshot shows the 'Создание тарифа' (Creating a tariff) interface. It includes a form for entering tariff details, a list of added and available game places, and a calendar view for selecting dates and prices.

**Описание тарифа**

Название: night

Часов в пакете: 10

Тип: Пакет

Тип пакета: ☒ Ограниченный ☐ Плавающий

Описание: Описание

Разрешить систему лояльности: ☒

Разрешить только подтвержденным: ☒

**Добавленные игровые места**

182.189  
17%

**Доступные игровые места**

0.217  
0.6  
0.150  
PS3  
0.100  
ps1

**Календарный период действия**

Начало: Конца: Цена: +

Нет данных

**С** Понедельник **До** Воскресенье 500 +

Начало: Конца: Цена: +

Нет данных

Рисунок 1.1 – Интернет-ресурс «GymEasy»

В целом, «GymEasy» представляет собой удобный и простой инструмент для управления тренировками и услугами тренажёрного зала.

### 1.1.2 FitSoft

Ещё одним альтернативным решением, которое можно использовать для управления бронированием в тренажёрных залах, является программное средство «FitSoft» [2]. Оно позволяет администраторам управлять бронированием тренажёров, предоставлять скидки, приостанавливать или сбрасывать бронирования, а также отслеживать занятость оборудования.

Основным преимуществом «FitSoft» является возможность гибкой настройки тарифов и просмотра текущей занятости тренажёров.

Однако у программы есть недостатки. Например, отсутствует мобильное приложение для клиентов, интерфейс имеет устаревший дизайн и сложен для освоения.

Интерфейс программного средства представлен на рисунке 1.2.



Рисунок 1.2 – Программное-средство «FitSoft»

«FitSoft» — полезное решение для администраторов, но, к сожалению, оно мало ориентировано на потребности клиентов.

### 1.1.3 SmartTimer

Другое программное средство, предлагающее управление доступом к тренажёрам, — это “SmartTimer” [3]. С его помощью администраторы могут настраивать тарифы, управлять расписанием занятий и создавать учетные записи с различными правами доступа. В свою очередь, пользователи могут бронировать тренажёры и время тренировок.

Однако есть и недостатки. Например, интерфейс программы не адаптируется под разные устройства, доступен только один язык интерфейса, а техническая поддержка отсутствует.

Интерфейс программного средства представлен на рисунке 1.3.

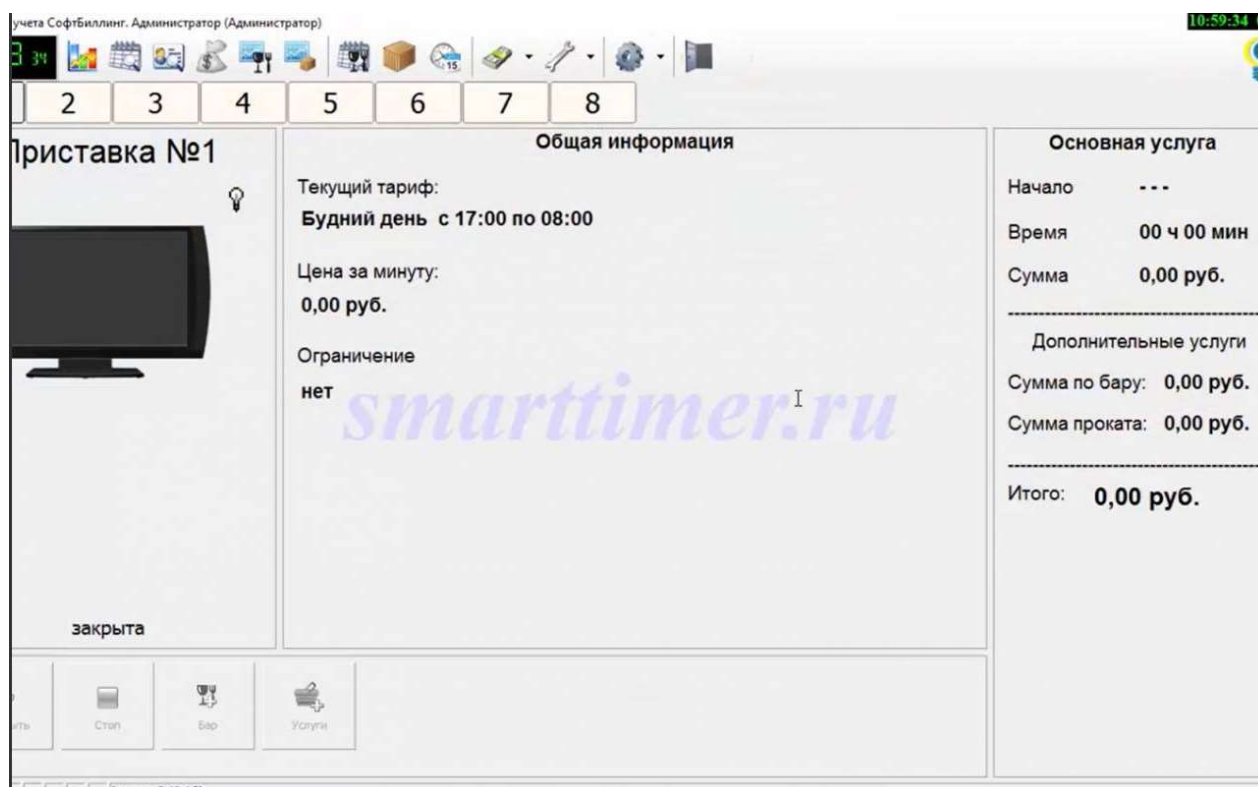


Рисунок 1.3 – Программное средство «SmartTimer»

В итоге, “SmartTimer” является удобным инструментом для управления тарифами и бронированиями, но нуждается в обновлении и улучшении пользовательского опыта.

### 1.1.4 Альтернативный вариант

Ещё одним возможным вариантом решения поставленных задач может быть обращение к специализированным фитнес-консультантам или сервисам по аренде оборудования. Основным преимуществом такого подхода является возможность личного ознакомления с услугами, их тестирования и выбора наиболее подходящих решений.



К недостаткам можно отнести ограниченный выбор предложений, сложность в поиске редких тренажёров или услуг, а также риск получить некачественное оборудование.

Несмотря на это, для любителей уникального фитнес-опыта данный вариант может стать интересной альтернативой.

## **1.2 Требования к проекту**

Целью данного курсового проекта является разработка приложения для тренажёрного зала, которое обеспечит эффективное управление и организацию деятельности клуба, а также повысит удобство использования его услуг для клиентов. Ниже приведены требования к функциональности и характеристикам приложения:

### **1.2.1 Функциональные требования:**

#### **1. Пользовательский интерфейс:**

- Приложение должно обладать интуитивно понятным и привлекательным пользовательским интерфейсом, обеспечивающим удобство использования и навигацию по функциональным возможностям.

- Необходима возможность регистрации и аутентификации пользователей с использованием логина и пароля.

#### **2. Управление клиентскими аккаунтами:**

- Приложение должно позволять создание и управление клиентскими аккаунтами, включая регистрацию, вход, просмотр и редактирование персональной информации.

- Пользователи должны иметь возможность просматривать свою историю посещений, бронировать услуги и получать уведомления о предстоящих мероприятиях.

#### **3. Управление услугами:**

- Клиенты должны иметь возможность бронировать услуги клуба через приложение.

## **2 Анализ требований к программному средству и разработка функциональных требований**

### **2.1 Анализ требований к программному средству**

В ходе анализа требований к программному средству для тренажёрного зала, были выявлены следующие основные потребности и задачи:

1. Управление доступом: Пользователям необходимо иметь возможность регистрироваться и входить в систему с помощью уникальных учетных записей. Кроме того, нужно предусмотреть возможность управления уровнем доступа для разных категорий пользователей, таких как администраторы и обычные пользователи.

2. Запись на тренировку: Пользователи должны иметь возможность записываться на тренировки на определенное время.

3. Управление услугами: Тренажёрный зал может предлагать различные услуги для посещения. Система должна предоставлять возможность управления услугами, включая покупку и просмотр подробной информации.

4. Управление отзывами: Пользователи должны иметь возможность оставлять отзывы и просматривать отзывы других пользователей.

### **2.2 Разработка функциональных требований**

На основе вышеуказанных анализированных требований, можно определить следующие функциональные требования к программному средству для тренажёрного зала:

1. Регистрация и аутентификация пользователей:
  - Пользователи должны иметь возможность создать новый аккаунт, указав необходимую информацию.
  - Пользователи должны иметь возможность войти в систему, используя свои учетные данные.
  - Система должна обеспечивать безопасную аутентификацию и защиту пользовательских данных.
2. Запись на тренировки:
  - Пользователи должны иметь возможность просмотреть доступные услуги и временные слоты для бронирования.
  - Пользователи должны иметь возможность выбрать тренировку и забронировать её на определенное время.
  - Система должна предотвращать двойное бронирование.
3. Учет времени использования:
  - Система должна отслеживать время, проведенное пользователями за тренировкой.
  - Администратор должен иметь доступ к отчетам о времени использования пользователей.
4. Управление услугами:

- Пользователи должны иметь возможность просматривать доступные услуги и их стоимость.

- Пользователи должны иметь возможность приобрести услугу через приложение.

#### 5. Управление отзывами:

- Пользователи должны иметь возможность создавать и просматривать отзывы на услуги тренажерного зала.

Эти функциональные требования являются основой для разработки приложения для тренажерного зала, и должны быть учтены в процессе проектирования и реализации приложения.

### 3 Проектирование программного средства

#### 3.1 Общая структура

Программное средство «Тренажерный зал» имеет следующую структуру, представленную на рисунке 3.1.

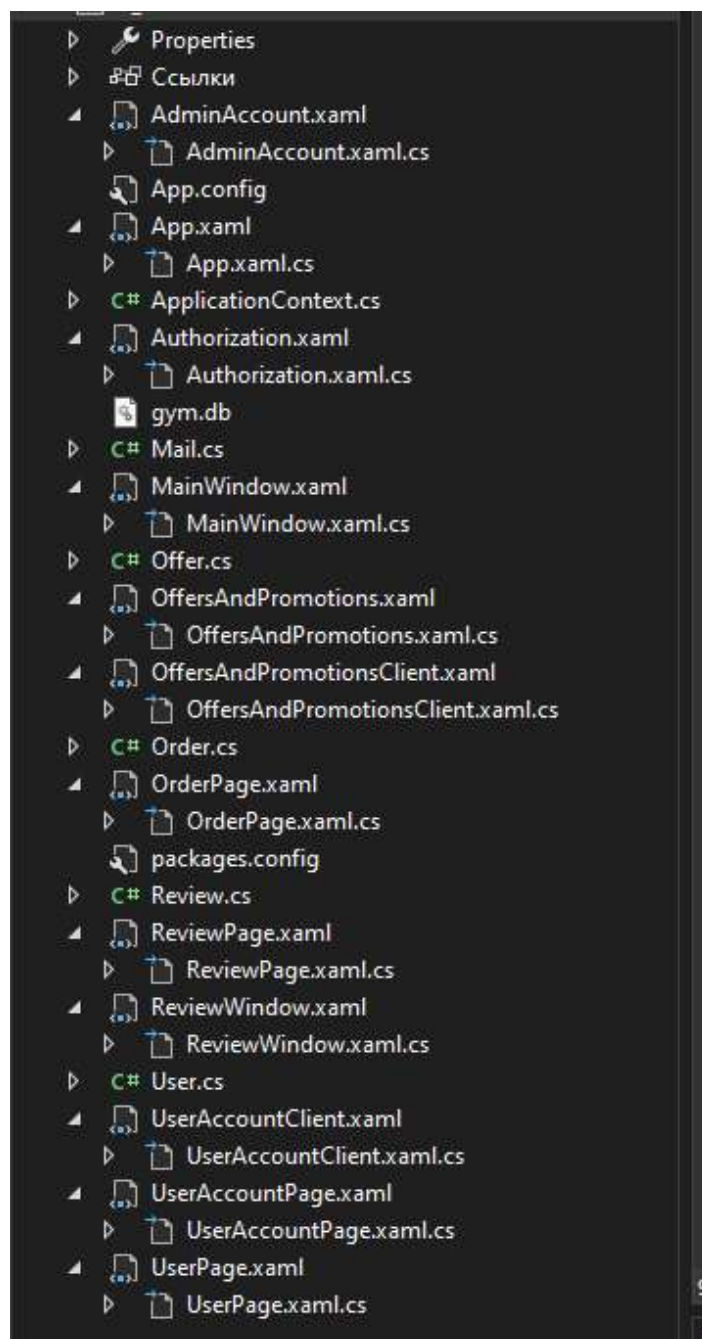


Рисунок 3.1 – Структура проекта

Таким образом, сформированная таблица помогает понять общую структуру проекта проектируемого программного средства, а также какие файлы используются для его работы. Эта информация будет крайне полезна в будущем.

Таблица 3.2 – Описание файлов проекта

Имя файла	Содержание
ApplicationContext.cs	Файл, содержащий коллекции DbSet и подключение к базе данных.
User.cs, Order.cs, Review.cs	Файлы для определения таблицы базы данных, содержащие поля и свойства, описывающие столбцы таблицы.
gym.db	Файл базы данных, где храниться информация о пользователях, заказах и отзывах об услугах клуба.
MainWindow.xaml, MainWindow.xaml.cs	Главный файл, представляющий из себя графический интерфейс для регистрации и программное описание к нему. Имеет возможность подключения к базе данных и внесения изменений.
Authorization.xaml, Authorization.xaml.cs	Файл, содержащий графический интерфейс для авторизации с программным описанием. Имеет возможность подключения к базе данных и получения информации.
UserPage.xaml, UserPage.xaml.cs	Файлы для роли администратора, содержащие список всех зарегистрированных аккаунтов с возможностью перехода на пользовательские страницы с возможностями администратора.
UserAccountPage.xaml, UserAccountPage.xaml.cs	Файлы реализации пользовательских аккаунтов с возможностями администратора. Администратору доступна подробная информация о заказах клиента, а также возможность блокировки пользовательского аккаунта.
UserAccountClient.xaml, UserAccountClient.xaml.cs	Файлы реализации пользовательских аккаунтов с возможностями клиента. Клиенту доступна некоторая информация о своих заказах, также есть возможность удаления и выхода из учётной записи, добавлены кнопки для написания отзывов и просмотра списка услуг тренажёрного зала.

Продолжение таблицы 3.2

Имя файла	Содержание
OfferAndPromotions.xaml, OfferAndPromotions.xaml.cs	Файлы для администраторской роли, содержащие список услуг клуба с подробной информацией, а также кнопку для просмотра отзывов авторизованных пользователей о клубе.
OfferAndPromotionsClient.xaml, OfferAndPromotionsClient.xaml.cs	Файлы клиентской роли, содержащие список услуг клуба с возможностью заказать понравившуюся услугу, а также прочитать отзывы других пользователей.
OrderPage.xaml, OrderPage.xaml.cs	Файлы для заполнения клиентом формы на оказание услуги. Есть возможность выбрать форму оповещения(почта, sms), выбрать способ оплаты, день, время.
AdminAccount.xaml, AdminAccount.xaml.cs	Файлы аккаунта администратора, где содержится полная информация обо всех заказах в данном приложении.
ReviewPage.xaml, ReviewPage.xaml.cs	Файлы отзывов, содержащие имя пользователя, комментарий, оценку и время отправления.
ReviewWindow.xaml, ReviewWindow.xaml.cs	Файл формы для создания отзыва. Есть возможность ввести имя, поставить оценку, написать комментарий и отправить.
Offer.cs	Файл, содержащий поля, свойства и конструктор с параметрами для реализации класса со списком услуг.
Mail.cs	Файл класса для реализации функций почты.

В целом, описание структуры проекта позволяет лучше понимать, как устроено программное средство и какие компоненты в нем присутствуют.

### 3.2 Разработка общей схемы приложения

Рассмотрим общую схему приложения «Тренажёрный зал». На рисунке 3.2 представлена общая схема работы приложения.

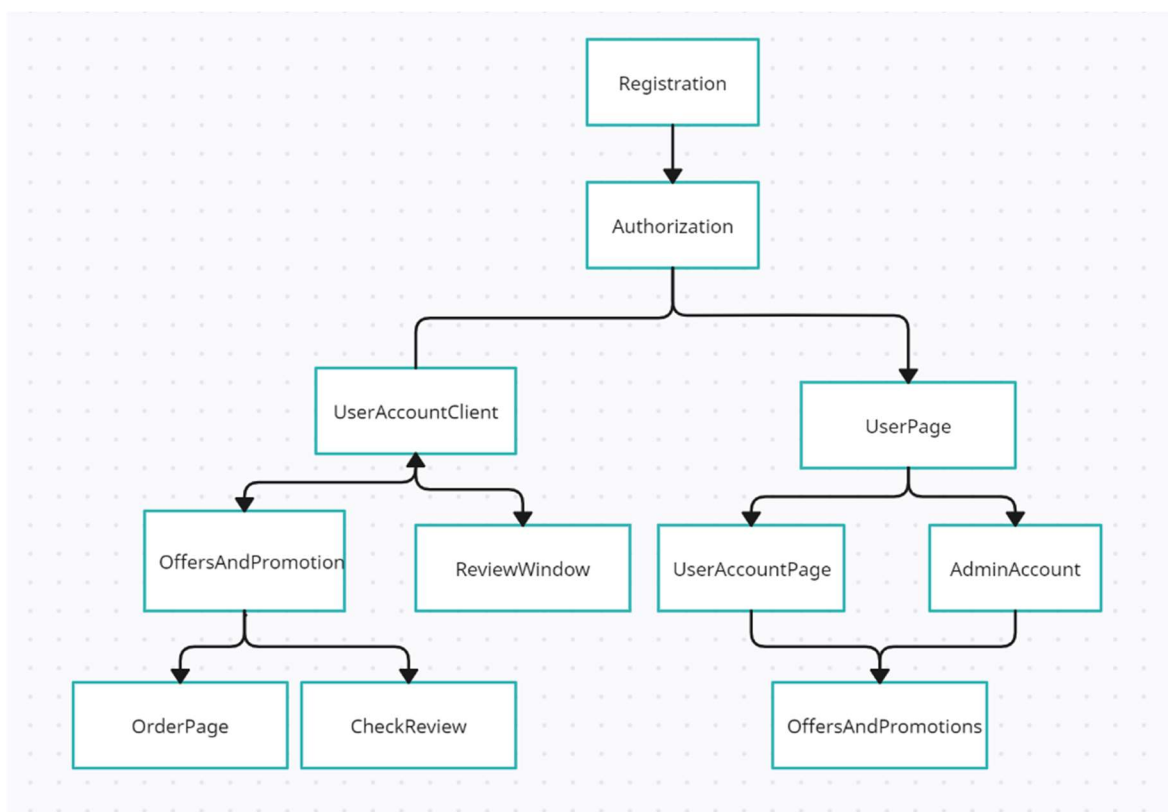


Рисунок 3.2 — Общая схема приложения

Из нее видно, что при запуске приложения у нас появляется окно регистрации, которое требует ввода логина и пароля для начала работы.

Если у пользователя есть аккаунт, он может нажать на кнопку «Log in» и авторизоваться. Далее в зависимости от вводимых данных, определяется роль администратор или клиент. При переходе на роль клиента, пользователь попадает на страницу своего аккаунта, откуда может попасть на страницы услуг и отзывов.

Если пользователь переходит в окно услуг, у него есть возможность перейти в окно заказов либо прочитав отзывы.

Если же была определена роль администратора, то переход осуществляется в окно аккаунтов, где расположены все аккаунты авторизованных пользователей.

Далее если администратор выбирает собственный аккаунт из списка, то он переходит в окно своего аккаунта, откуда может перейти в окно с услугами. Если же администратор выбирает другой номер, он может перейти в аккаунт пользователя в роли администратора, откуда также есть возможность перехода в окно услуг.

На рисунке 3.3. приведена диаграмма последовательности для авторизации.

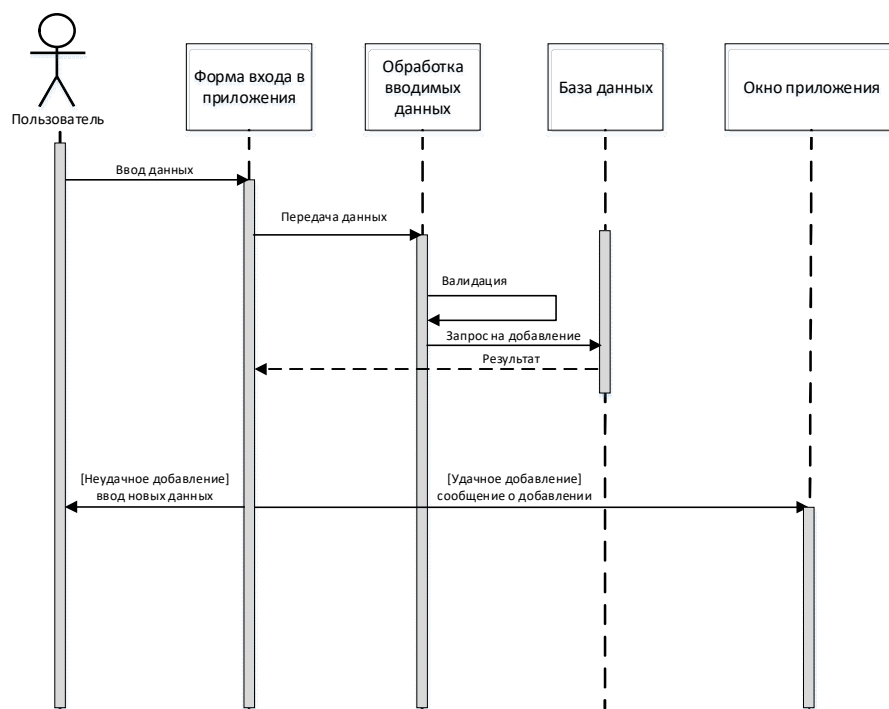


Рисунок 3.3 – Диаграмма последовательности для авторизации

### 3.3 Взаимоотношение между классами

Для визуализации взаимосвязей между классами используется диаграмма UML – графическое представление набора элементов, изображаемое чаще всего в виде связанного графа с вершинами (сущностями) и ребрами (отношениями).

Для представления внутренней структуры программы в виде классов и связей между ними используется диаграмма классов. Приложение спроектировано таким образом, что каждый класс выполняет свои функции и практически не зависит от других. Диаграмма классов представлена в приложении А.

### 3.4 Модель базы данных

Для реализации поставленной задачи была создана база данных gym.db. Для ее создания использовалась система управления реляционными базами данных SQLite. База данных состоит из таблиц, представленных на рисунке 3.4. Скрипт для создания базы данных представлен в приложении Г.



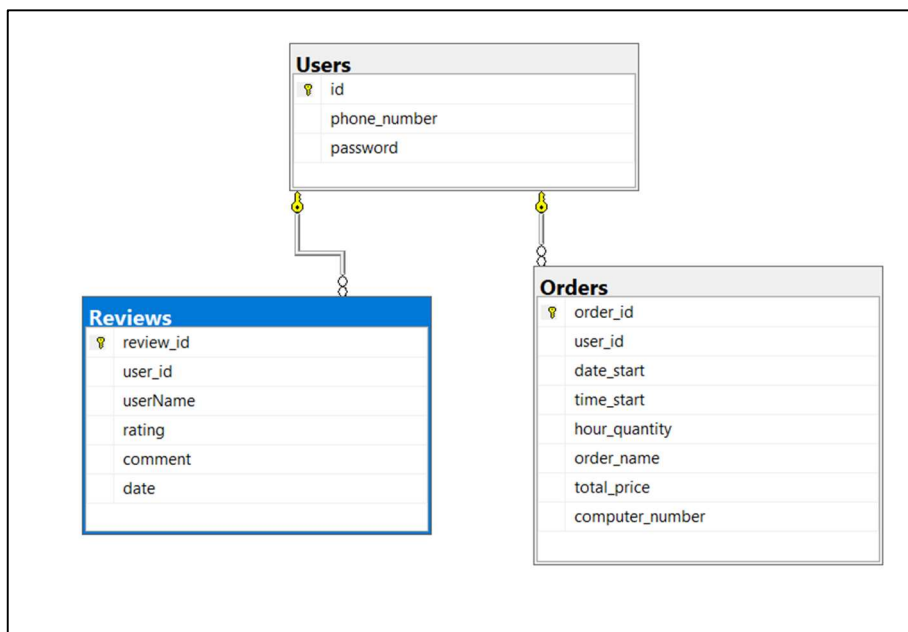


Рисунок 3.4 — gym.db

На рисунке 3.5 проиллюстрирована структура таблицы «Users», которая содержит информацию о пользователях. В данной таблице поле ID, представленное целочисленным значением, является первичным ключом. Поле phone\_number хранит номер телефона пользователя, password – пароль

Users	
id	INTEGER
phone_number	TEXT
password	TEXT

Рисунок 3.5 — Структура таблицы «Users»

На рисунке 3.6 изображена структура таблицы «Orders», содержащая информацию о заказах. Первичный ключ order\_id хранит уникальный идентификатор заказа, user\_id – идентификатор пользователя, date\_start – дата начала выполнения заказа, time\_start – время начала выполнения заказа, hour\_quantity – количество забронированных часов, order\_name – название заказа, total\_price – стоимость заказа.

Orders	
order_id	INTEGER
user_id	INTEGER
date_start	TEXT
time_start	TEXT
hour_quantity	INTEGER
order_name	TEXT
total_price	INTEGER

Рисунок 3.6 — Структура таблицы «Orders»

На рисунке 3.7 изображена структура таблицы «Review», в которой содержатся данные об отзывах. Первичный ключ – review\_id – номер отзыва, автоинкремент, далее user\_id – внешний ключ для таблицы “User” – id пользователя. В столбце userName хранятся никнеймы пользователей, которые они вводят при заполнении формы отзыва. rating содержит оценку от 1 до 5, comment – текстовое сообщение про клуб и его услуги. date хранит строковый формат даты.







 review_id	INTEGER
 user_id	INTEGER
 userName	TEXT
 rating	INTEGER
 comment	TEXT
 date	TEXT

Рисунок 3.7 — Структура таблицы «Review»

Создание базы данных позволит эффективно хранить и управлять информацией о пользователях, объявлениях и регионах, необходимой для функционирования тренажерного зала.

### 3.5 Проектирование архитектуры приложения

Для общего представления функционального назначения системы используется диаграмма использования, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. На диаграмме использования применяются два типа основных сущностей: варианты использования и группы пользователей.

Разные группы пользователей в диаграмме называются актёрами, и обозначают любые сущности, использующие систему. Любая функция системы называется вариантом использования. Каждый вариант использования обозначает набор действий, который может быть использован актёром для взаимодействия с системой, и определяет набор действий, выполняемых этой системой.

Диаграмма использования представлена в приложении Б.

## **4 Реализация программного средства**

### **4.1 Основные классы программного средства**

Для выполнения технических задач программного средства «Тренажёрный зал» должны быть реализованы следующие функции и соответствующие им классы и методы:

- авторизация;
- регистрация;
- просмотр услуг зала;
- заполнение формы оформления заказа;
- просмотр забронированных заказов;
- заполнение и просмотр отзывов.

Далее подробно рассмотрены каждые из необходимых для выполнения технических задач функции, а также созданные для их выполнения классы и методы и их функционал и реализация.

### **4.2 Авторизация**

Графическая часть формы авторизации описана в классе `Authorization`. После нажатия кнопки «Log in» происходит проверка введённых данных на корректность: соответствие номера белорусскому формату, наличие таких данных в базе данных и т.д.

При успешном прохождении всех проверок пользователь перенаправляется на главную страницу пользователя, а противном случае – получает сообщение о соответствующей ошибке.

Таким образом, в классе `Authorization` определены свойства, методы и события, которые обеспечивают взаимодействие с базой данных и обработку действий пользователя при авторизации.

### **4.3 Регистрация**

Для регистрации пользователем нового аккаунта реализован класс `MainWindow`, в котором описана графическая составляющая окна «Регистрации».

При переходе к окну регистрации с окна входа создается новый экземпляр класса `MainWindow`.

В начале все введённые пользователем данные проходят проверку: валидация осуществляется с помощью условных конструкций и регулярных выражений. После успешного прохождения вышеописанных проверок происходит попытка добавить пользователя в базу данных и перенаправление пользователя на главную страницу. Если номер телефона уже зарегистрирован в базе данных, то пользователь получает соответствующее уведомление об ошибке.

Таким образом, класс `MainWindow` содержит логику и данные, связанные с окном регистрации, и предоставляет методы для выполнения регистрации и уведомлений об изменении значений свойств.

#### 4.4 Просмотр услуг клуба

После успешной авторизации пользователь перенаправляется на основную страницу программного средства, графическая часть которой представлена классом `UserAccountClient`.

В конструкторе класса происходит загрузка всех уже имеющихся заказов, изображения пользователя, номера, кнопок и прочих данных.

Данный класс содержит `DataGrid` панель, где отображается информация о заказах пользователя. Есть привязка к событиям для различных функций.

#### 4.5 Заполнение формы оформления заказа

Заполнение формы для размещения объявления реализовано в классе `OrderPage`, который представляет графическое отображение окна.

Он использует базу данных, чтобы получить и сохранить данные.

Данный класс используется для оформления заказа. При заполнении формы у пользователя есть возможность выбрать способ оповещения: sms или почта. В зависимости от выбора пользователя, для ввода открывается либо поле номера телефона, либо поле почты соответственно.

С помощью слайдера пользователь может выбрать количество часов, которые он желает забронировать.

В форме есть поля для заполнения даты, времени, которые пользователь заполняет согласно выбранной услуге по своему усмотрению.

#### 4.6 Просмотр бронированных заказов

Графическая часть просмотра собственных бронированных заказов доступна в классах `AdminAccount`, `UserAccountPage`, `UserAccountClient`.

`AdminAccount` это аккаунт администратора, в котором отображается полная информация обо всех заказах пользователей.

`UserAccountPage` – администраторский аккаунт пользователей, где администратор может просмотреть заказы конкретного пользователя.

`UserAccountClient` – пользовательский аккаунт пользователей, где отображается информация об всех заказах данного пользователя.

Наиболее подробной информацией обладает администратор, который имеет доступ ко всей информации из таблицы “Orders”. В классе `UserAccountPage` представлена более детальная информация чем в классе `UserAccountClient`.

#### 4.7 Заполнение и просмотр отзывов

Для создания отзывов создан класс `ReviewWindow`, который связан с таблицей из базы данных «Reviews». Пользователь вводит свой никнейм, оценку и комментарий, далее это заносится в базу данных вместе с id пользователя и датой.

## 4.8 Реализация вспомогательных классов

Создан класс для работы с базой данных SQLite – `ApplicationContext`. В этом классе созданы 3 коллекции `DbSet<T>`, которые принимают в качестве параметра `T` классы, хранящие свойства, соответствующие полям таблиц из базы данных. Также в классе содержится строка подключения к базе данных. В любом приложении, работающем с БД через `Entity Framework Core`, необходимо использовать контекст (класс производный от `DbContext`) и набор данных `DbSet`, через который можно взаимодействовать с таблицами из БД.

## 5 Тестирование, проверка работоспособности и анализ полученных результатов

### 5.1 Тестирование авторизации и регистрации

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 авторизации, возможна такая ситуация, в которой пользователь ничего не ввёл. Обработка данного сценария приведена на рисунке 5.1.

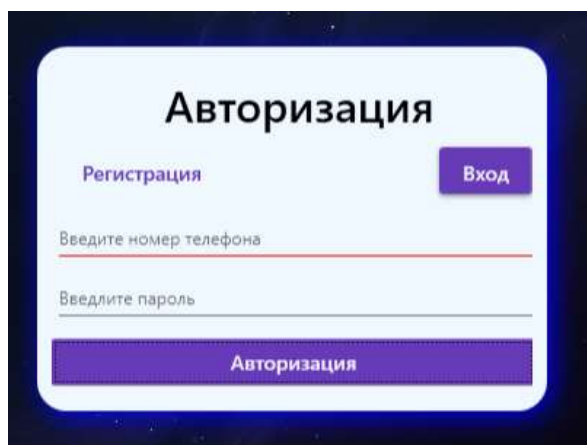


Рисунок 5.1 — Обработка пустых полей при авторизации

Как видно с рисунка, поле “Enter phone number” подчёркивается красным цветом.

При вводе некорректных данных, возникает ошибка, представленная на рисунке 5.2.

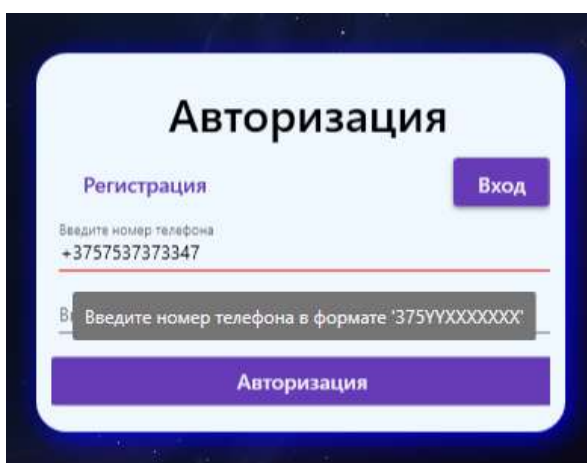


Рисунок 5.2 — Обработка некорректных данных

Как видно с рисунка, поле “Enter phone number” подчёркивается красным цветом.

Также возможен такие случаи:

Пользователь хочет зарегистрироваться, при этом не указав никаких данных (рисунок 5.3).

Пользователь указывает электронную почту, которая уже привязана к учётной записи (рисунок 5.4).

Пользователь, когда осуществляет ввод паролей, указывает несовпадающие пароли (рисунок 5.5).

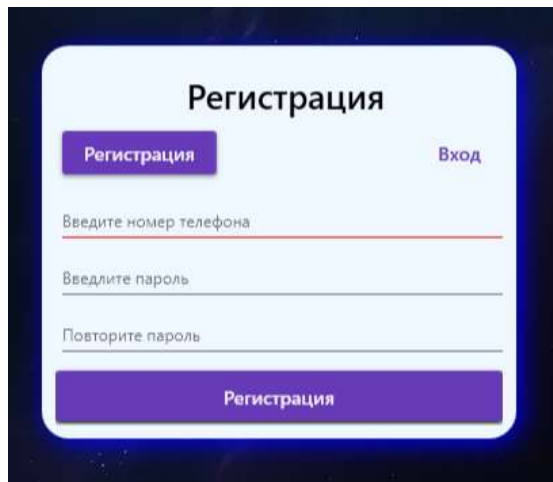
A screenshot of a registration form titled "Регистрация". At the top left is a blue button labeled "Регистрация", and at the top right is a link labeled "Вход". Below the title are three input fields: "Введите номер телефона", "Введите пароль", and "Повторите пароль". At the bottom is a large blue button labeled "Регистрация". All input fields are empty.

Рисунок 5.3 — Обработка пустых полей при регистрации

Как видно с рисунка, что поля сверху вниз подсвечиваются красным «Регистрация» не доступна для клика.

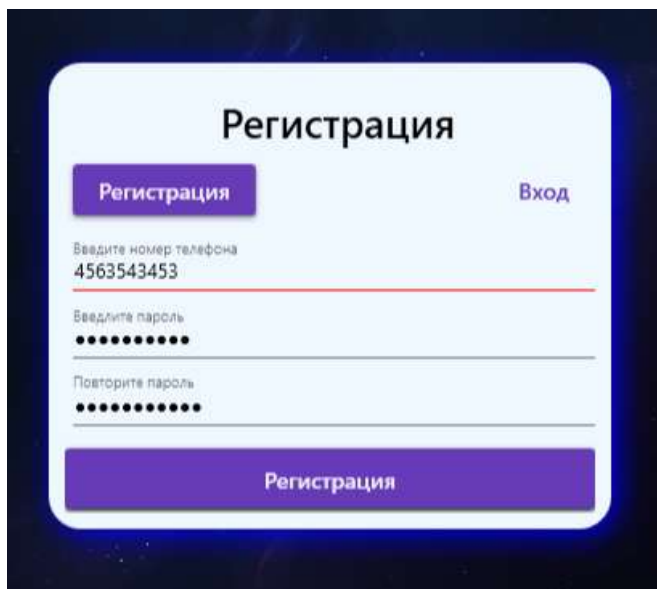
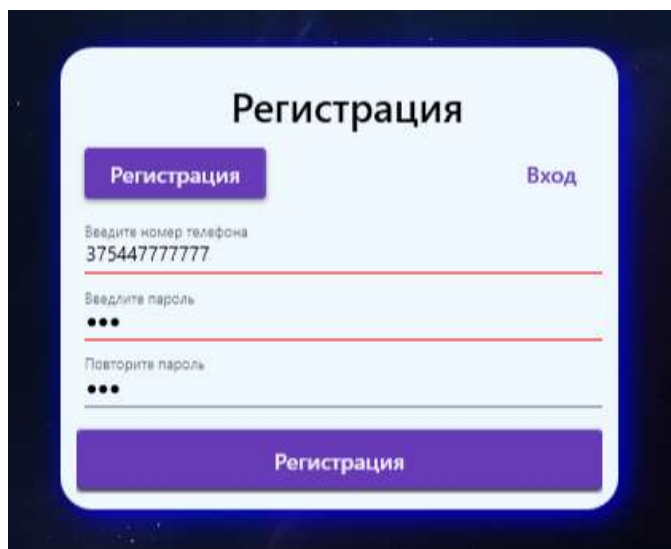
A screenshot of the same registration form. The "Введите номер телефона" field now contains the number "4563543453". The "Введите пароль" and "Повторите пароль" fields are filled with ten black dots each. The blue "Регистрация" button at the bottom remains visible.

Рисунок 5.4 — Обработка ввода уже зарегистрированного телефона

Как видно с рисунка, в статус-поле вывело сообщение.



The image shows a registration form titled "Регистрация" (Registration) in Russian. The form is set against a dark blue background. It features a light blue rounded rectangle containing the following elements: a title "Регистрация" at the top center; a purple button labeled "Регистрация" on the left and a link labeled "Вход" (Login) on the right; a label "Введите номер телефона" (Enter phone number) above a text input field containing "375447777777"; a label "Введите пароль" (Enter password) above a password input field with three dots; a label "Повторите пароль" (Repeat password) above another password input field with three dots; and a large purple button labeled "Регистрация" at the bottom.

Рисунок 5.5 — Обработка подтверждения пароля

Таким образом, в данном разделе были рассмотрены обработки основных исключений, которые могут быть вызваны пользователем при использовании программного средства.



## 6 Руководство по использованию

При запуске приложения у нас появляется окно регистрации, которое требует ввода логина, пароля и повтора пароля для начала работы (рисунок 6.1а).

После заполнения всех полей вам нужно будет перейти в окно авторизации (Рисунок 6.1б) по клику на кнопку «Log in» и ввести свои данные.

В случае, если вас есть своя учётная запись, можно просто авторизоваться, нажав на кнопку «Log in» в окне регистрации. Вы будете направлены в окно авторизации, где вам нужно будет заполнить все поля. После выполнения всех действий, в зависимости от вашей роли, вы будете адресованы на другое окно.

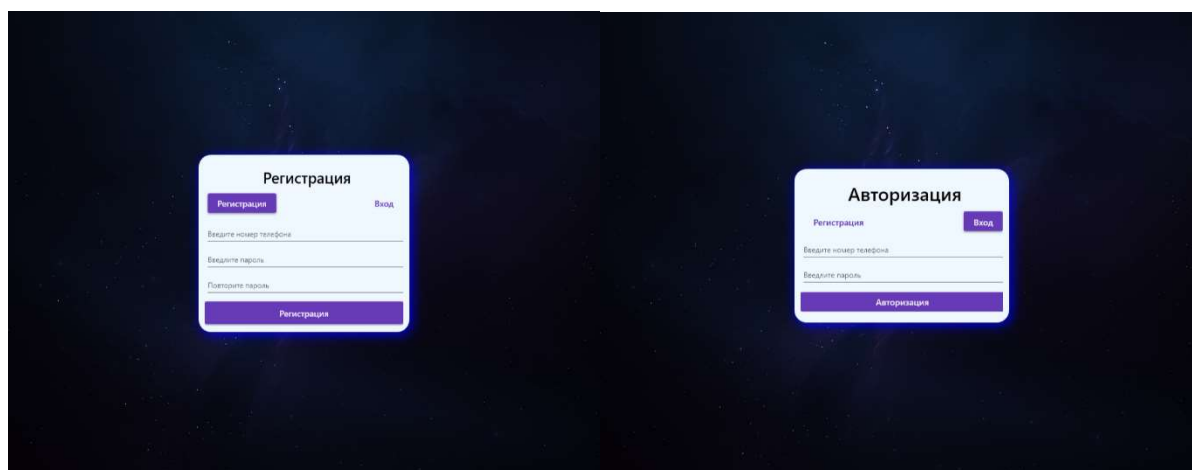


Рисунок 6.1а – Окно регистрации, 6.1б – окно авторизации

Если вы заходите как администратор, вы попадаете в окно со списком всех зарегистрированных пользователей (Рисунок 6.2).

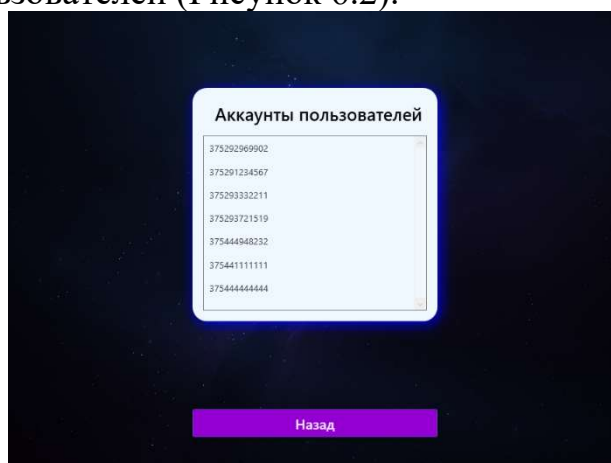


Рисунок 6.2 – Список аккаунтов

По нажатию на какой-либо номер происходит переадресация на другое окно. В случае если выбран номер аккаунта администратора, осуществляется переход в окно аккаунта администратора (Рисунок 6.3а), в любом другом – на окно администраторского аккаунта пользователя (Рисунок 6.3б).

Если вы заходите как пользователь, выполняется переход в аккаунт пользователя (Рисунок 6.4).

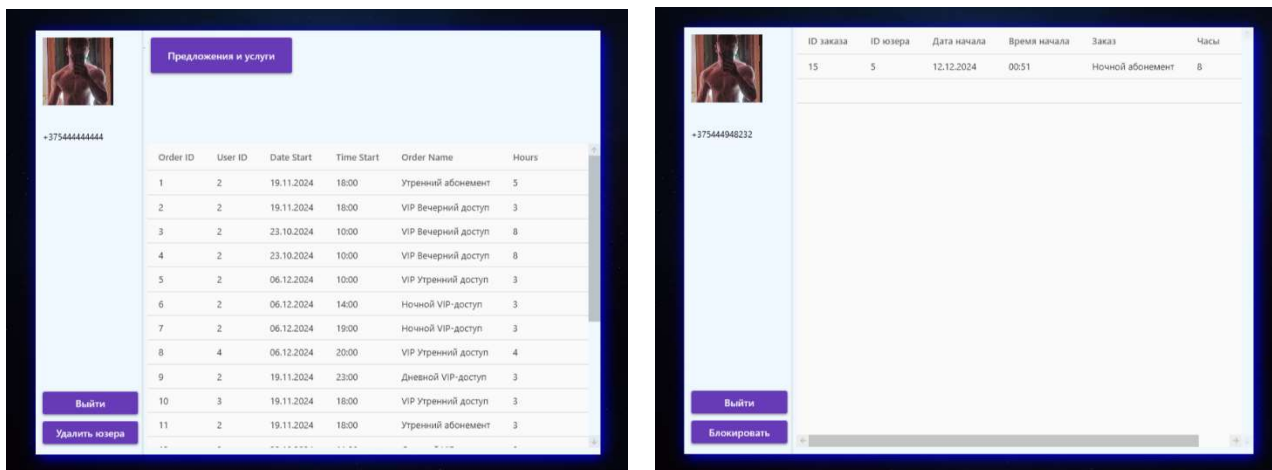


Рисунок 6.3а – аккаунт администратора, 6.3б – админ. аккаунт пользователя

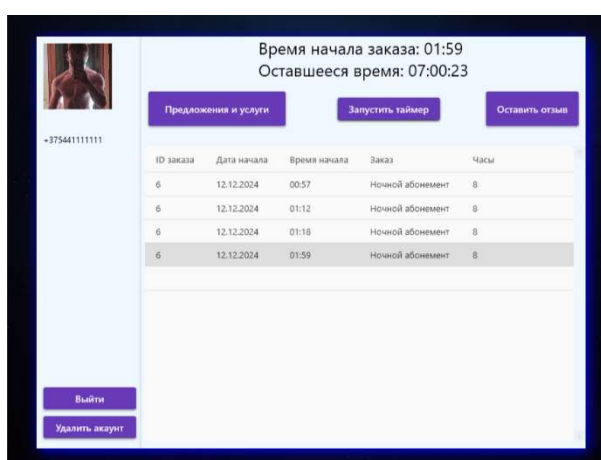


Рисунок 6.4 – польз. Аккаунт пользователя

Находясь в аккаунте администратора либо в администраторском аккаунте пользователя, вы можете перейти в окно «Предложений и акций» (Рисунок 6.5) по нажатию на кнопку «Offers and promotions».

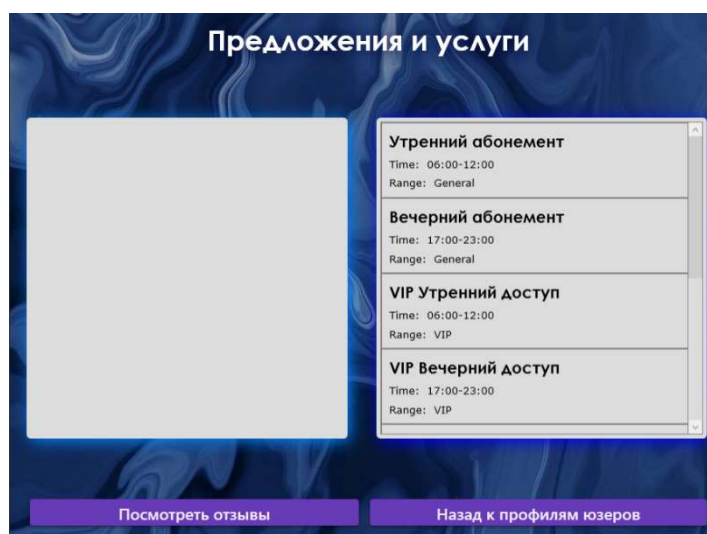


Рисунок 6.5– Окно услуг (администратор)

Находясь на аккаунте пользователя, вы также можете нажать на эту кнопку, однако вас переадресует в другое окно (Рисунок 6.6).

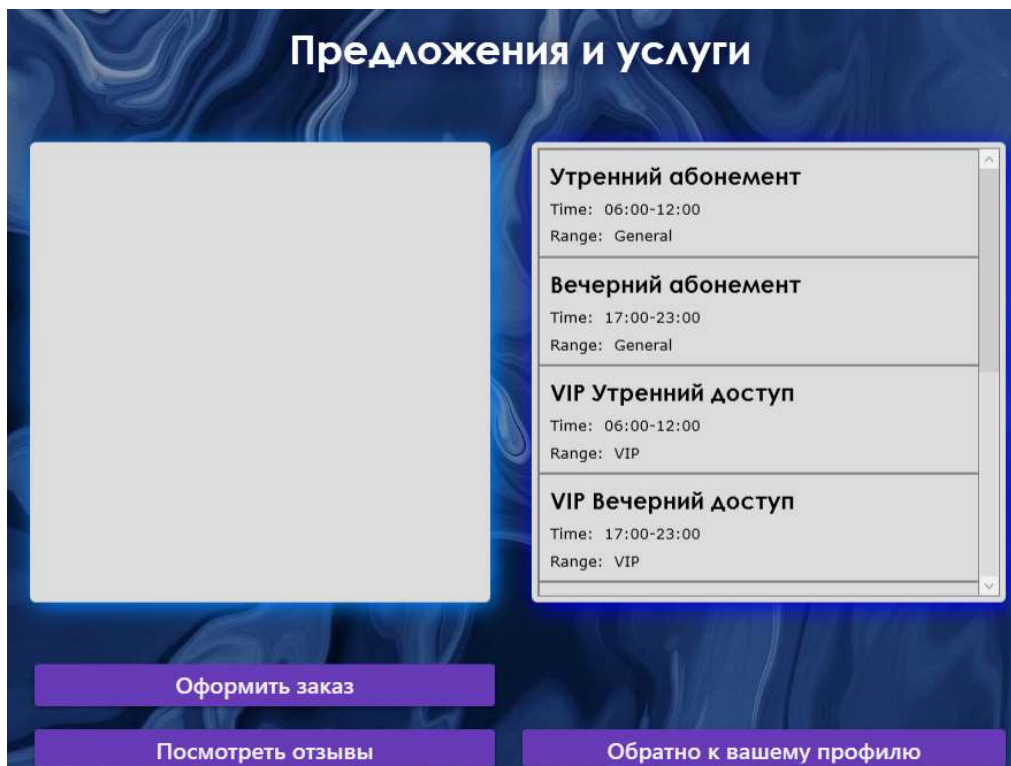


Рисунок 6.6 – Окно услуг (пользователь)

Здесь есть три кнопки: 1 – «Order», по нажатию на эту кнопку открывается окно бронирования, где можно оформить заказ. Первым заполняется поле выбора способа оповещения: при выборе способа оповещения «SMS», поле номера становится доступным, при выборе «Mail», доступным становится поле «Email».

Далее можно выбрать способ оплаты: «Cash» или «Card». С помощью слайдера выбирается время, которое пользователь желает забронировать. В последующие поля записывается дата, время.

## Записаться на тренировку

Введите ваш Email  
Dimasik425@yandex.ru

Введите ваш номер телефона  
375441111111

Выберите метод уведомлений  
Mail

Выберите метод оплаты  
Наличка

Ночной абонемент

12

Введите дату посещения  
12.12.2024

Введите время посещения  
03:00

Сейчас

Записаться

Рисунок 6.7 – Окно бронирования

В целом, данная инструкция предоставляет всю необходимую информацию, чтобы пользователи могли успешно использовать приложение, добавлять свои объявления, осуществлять поиск и управлять своими учетными записями, а администраторы имели полный контроль над системой и пользователями.

## ЗАКЛЮЧЕНИЕ

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

При разработке программного средства были выполнены все пункты из указанного списка предполагаемого основного функционала приложения, а именно:

Для пользователя:

- Выполнять регистрацию и авторизацию.
- Заполнять форму заказа по заданным формам.
- Оставлять отзывы.

Для администратора:

- Поддерживать работы с базой данных.
- Блокировать аккаунты пользователей.
- Бронировать время посещения.
- Оповещать клиента о выполнении заказа (почта, sms).
- Управлять временем доступа пользователей.

Тестирование программы показало, что она работает корректно и выполняет все свои функции.

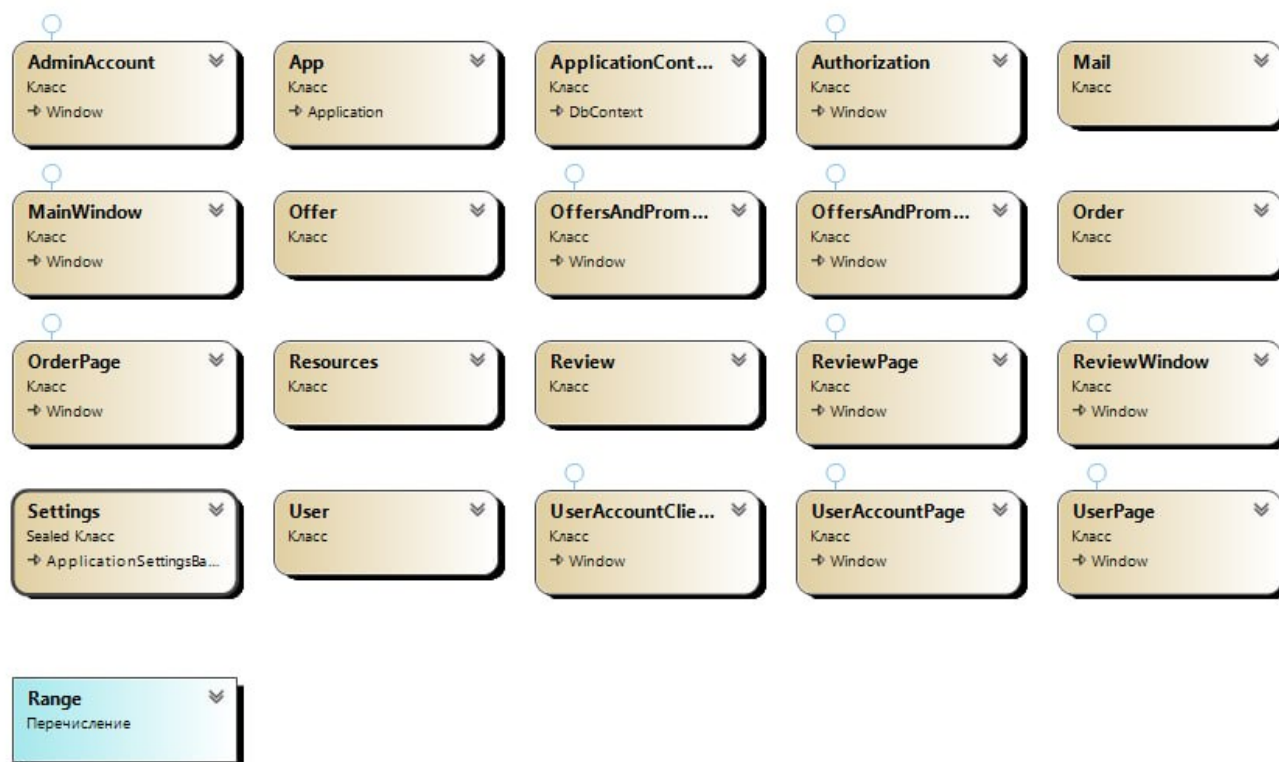
В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

### Список использованных источников

1. GymEasy [Электронный ресурс] – <https://GymEasy.com/> – Дата доступа 12.12.2024
2. FitSoft [Электронный ресурс] – [https://www. FitSoft.org/](https://www.FitSoft.org/) – Дата доступа 12.12.2024
3. SmartTimer [Электронный ресурс] – [https://www. SmartTimer.com/](https://www.SmartTimer.com/)– Дата доступа 12.12.2024
4. Microsoft Visual Studio [Электронный ресурс] – [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio) – Дата доступа 12.12.2024
5. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7. Режим доступа: <https://metanit.com/sharp/tutorial/> – Дата доступа: 12.12.2024
6. Пацей, Н.В. Курс лекций по языку программирования C# / Н. В. Пацей. – Минск: БГТУ, 2018. – 175 с.
7. Руководство по WPF // [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/wpf/> – Дата доступа: 12.12.2024
8. Руководство по XAML // [Электронный ресурс]. – Режим доступа: <https://www.tutorialspoint.com/xaml/index.htm>– Дата доступа: 12.12.2024
9. Работа с Entity Framework Core [Электронный ресурс] – <https://professorweb.ru/my/entity-framework/6/level1/> – Дата доступа 12.12.2024
10. Руководство по VkNet API // [Электронный ресурс]. – Режим доступа: <https://vknet.github.io/vk/> – Дата доступа: 12.12.2024
11. Блинова, Е.А. Курс лекций по Бадам данным / Е.А. Блинова. – Минск: БГТУ, 2019. – 175 с.

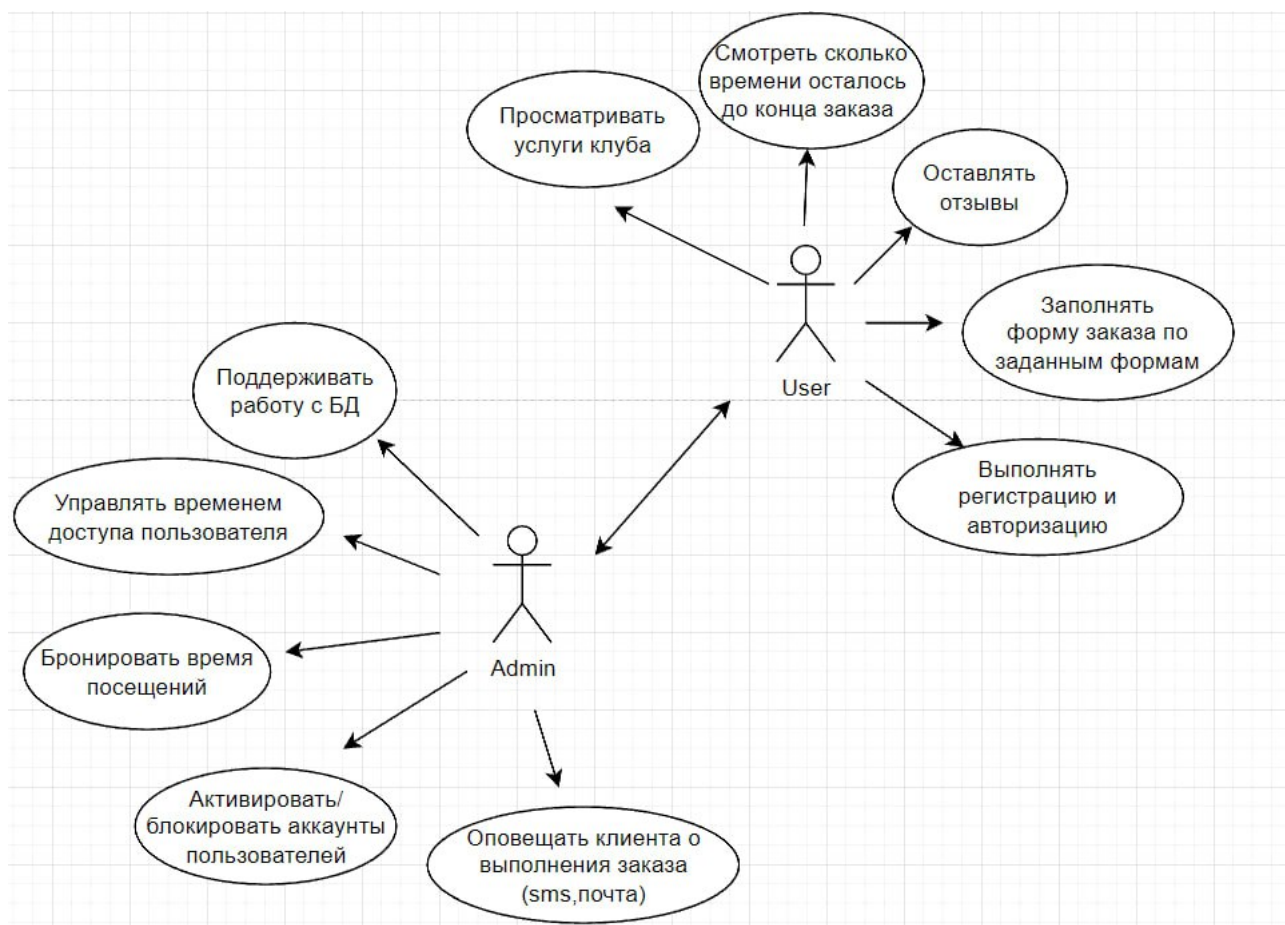
## ПРИЛОЖЕНИЕ А

## Диаграмма классов



## ПРИЛОЖЕНИЕ Б

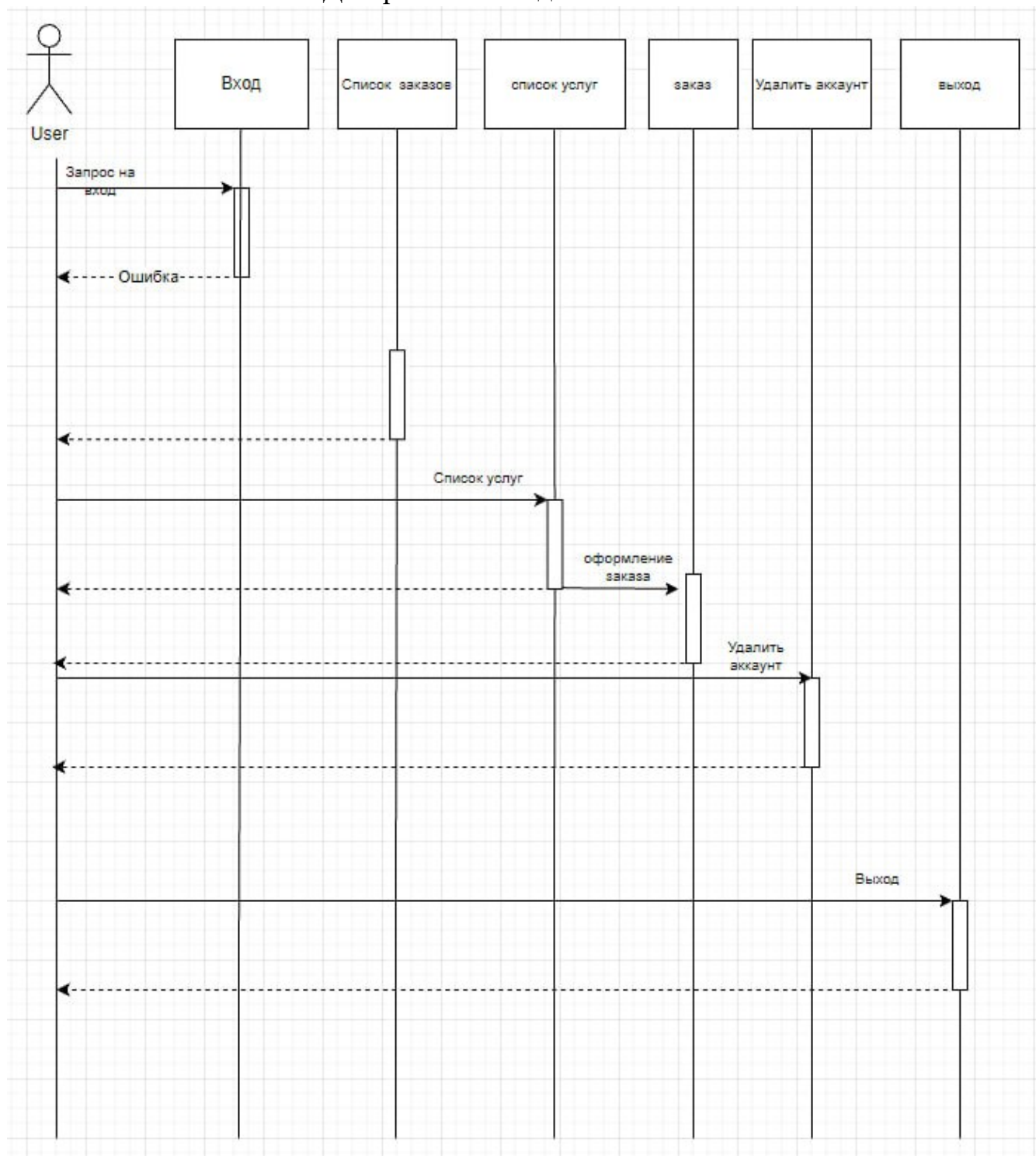
Диаграмма использований





## ПРИЛОЖЕНИЕ В

Диаграмма последовательностей



## ПРИЛОЖЕНИЕ Г

```
CREATE TABLE "Users" (  
    "id" INTEGER NOT NULL UNIQUE,  
    "phone_number" TEXT NOT NULL UNIQUE,  
    "password" TEXT NOT NULL,  
    PRIMARY KEY("id" AUTOINCREMENT)  
);  
CREATE TABLE "Orders" (  
    "order_id" INTEGER NOT NULL UNIQUE,  
    "user_id" INTEGER NOT NULL,  
    "date_start" TEXT NOT NULL,  
    "time_start" TEXT NOT NULL,  
    "hour_quantity" INTEGER NOT NULL,  
    "order_name" TEXT NOT NULL,  
    "total_price" INTEGER,  
);  
CREATE TABLE "Reviews" (  
    "review_id" INTEGER NOT NULL UNIQUE,  
    "user_id" INTEGER NOT NULL,  
    "userName" TEXT NOT NULL,  
    "rating" INTEGER NOT NULL,  
    "comment" TEXT NOT NULL,  
    "date" TEXT NOT NULL,  
    FOREIGN KEY("user_id") REFERENCES "Users"("id"),  
    PRIMARY KEY("review_id" AUTOINCREMENT)  
);
```