

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *компьютерных наук*

Кафедра *технологий обработки и защиты информации*
«Организация занятий в фитнес-центре»

Курсовая работа по дисциплине «Технологии программирования»
09.03.02 Информационные системы и технологии

Обучающиеся _____ Е.В. Кошелев, Д.Д. Козлов, В.В. Дмитриев

3 курс, *д/о*

Руководитель _____ преподаватель В.С. Тарасов __.__.2020

Воронеж 2020

Содержание

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	3
1. Анализ предметной области	4
1.1. Анализ существующих решений.	4
1.2. Схемы	5
1.2.1. Диаграмма прецедентов.....	5
1.2.2. Диаграмма классов.....	6
1.2.3. Диаграмма объектов.....	7
1.2.4. Диаграмма последовательностей.....	8
1.2.5. Диаграмма взаимодействий.....	9
1.2.6. Диаграмма состояний.....	10
1.2.7. Диаграмма активностей.....	12
1.2.8. Диаграмма развертывания.....	14
1.3. Глоссарий проекта.	14
2. Постановка задач.....	15
3. Требования к приложению	16
3.1 Требования к системе в целом.....	16
3.2 Требования к структуре и функционированию системы	16
3.3 Требования к программному обеспечению сайта	16
4. Основная часть	18
4.1. Основные этапы выполнения работы	18
4.2. Структура приложения	19
4.3 Тестирование приложения.....	26
4.3.1 Сценарии воронок конверсии.....	26
4.3.2 Результаты тестирования функционала	27
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	31

Введение

Зачастую людям необходимо узнать расписание и загруженность в фитнес центре на удобное время, а также быстро записаться на занятия. Этот процесс может осуществляться и по приходу в фитнес-центр, но это не всем удобно и занимает много времени. Но с развитием современных технологий на первое место выходит сеть Интернет. Именно в сети можно быстро и легко узнать расписание в интересующем центре, загруженность тренеров, а также можно без особых проблем записаться на удобное время к понравившемуся тренеру.

Наше приложение рассчитана не только на клиентов, но и на сотрудников клуба. Тренера смогут, имея доступ к интернету узнать записавшихся клиентов на их занятия. Администратор удаленно от центра добавить нового сотрудника.

Решение к поставленной задаче представляет из себя веб – приложение.

Были проанализированы существующие приложения В ходе анализа было выявлено то, что существующие решения требуют установки дополнительного программного обеспечения на компьютеры сотрудников.

В нашем проекте анализ происходит на стороне сервера, от сотрудника требуется только авторизоваться на сайте.

1. Анализ предметной области

1.1. Анализ существующих решений.

По информации сервиса Яндекс. Подбор слов ежемесячно ключевое слово “фитнес клуб” вводится в поисковый запрос более чем 600 тысяч раз за один месяц. Если добавить к этому результаты поиска в конкретной области и работы в конкретном городе или области, то количество запросов будет в разы выше. Не стоит забывать и о других поисковых системах, которыми пользуются в нашей стране. Это говорит о том, что получение информации и запись на занятия в фитнес-центры с помощью средств сети Интернет очень популярен в РФ.

Проведем анализ существующих решений.

Для анализа фитнес клубов в РФ я выбрал несколько самых популярных центров: Encore(<https://encorefitness.ru/>), X-Fit(<https://www.xfit.ru/>) и Worldclass Метрополис(<https://www.worldclass.ru/>).

Данные фитнес-центры имеют свои сайты. Рассмотрим подробнее.

Преимущества данных системы:

- возможность регистрации и авторизации на сайте клиентов
- возможность узнать расписание тренировок и выбрать тренера
- возможность записи на индивидуальные и групповые тренировки
- возможность просмотра информации о фитнес-центре

Как показал анализ, сайты этих фитнес центров весьма многофункциональны, но у них всех есть главный недостаток - все они ориентированы только на работу с клиентами. Это мы собираемся изменить в нашем приложении, добавив функционал для работы персоналу клуба.

1.2. Схемы

1.2.1. Диаграмма прецедентов

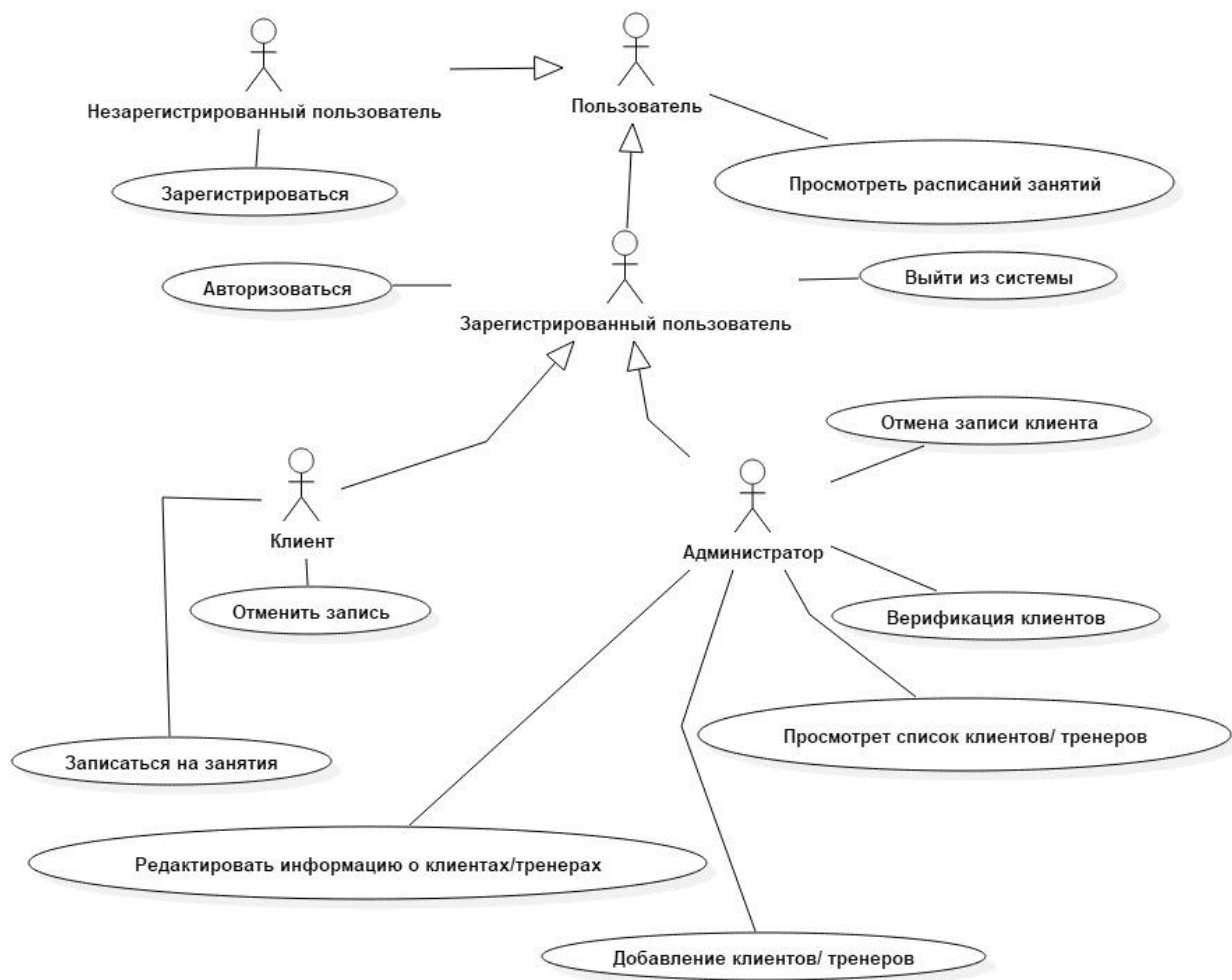


Рисунок 1 - Диаграмма прецедентов

1.2.2. Диаграмма классов

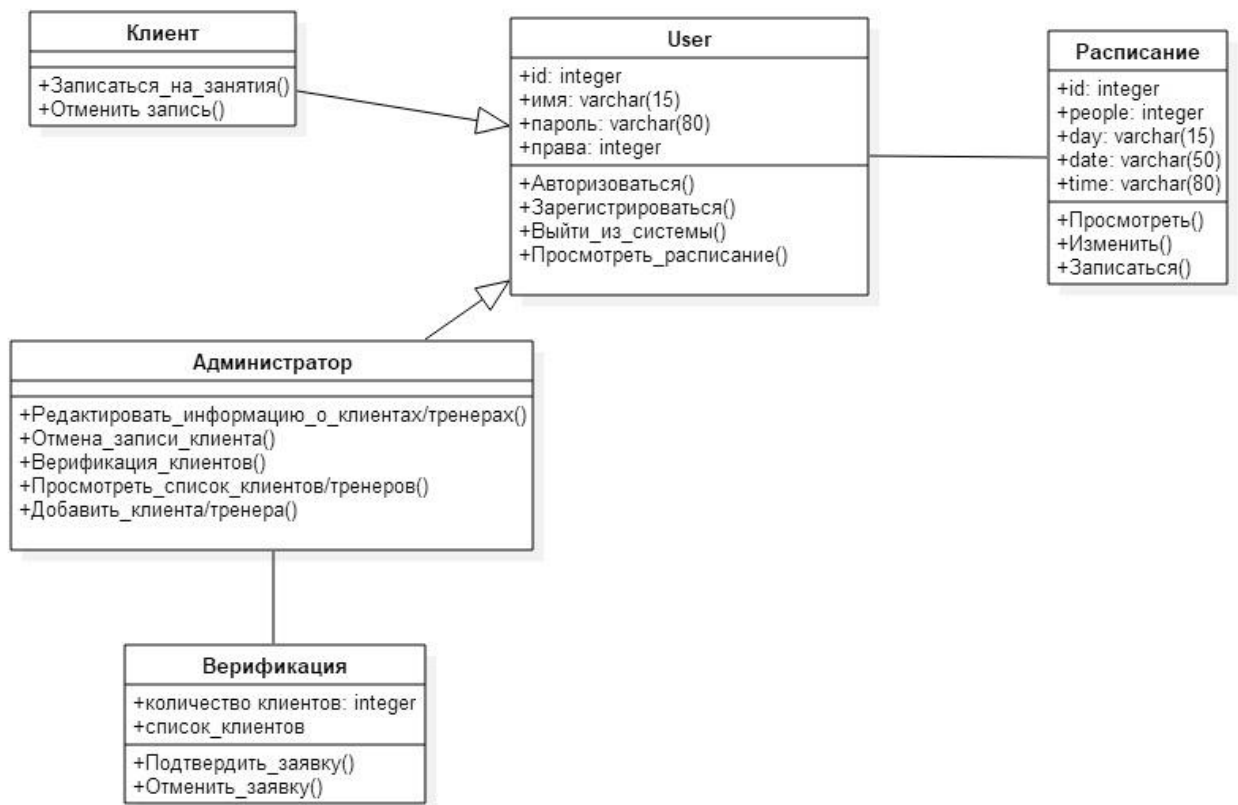


Рисунок 2 - Диаграмма классов

1.2.3. Диаграмма объектов

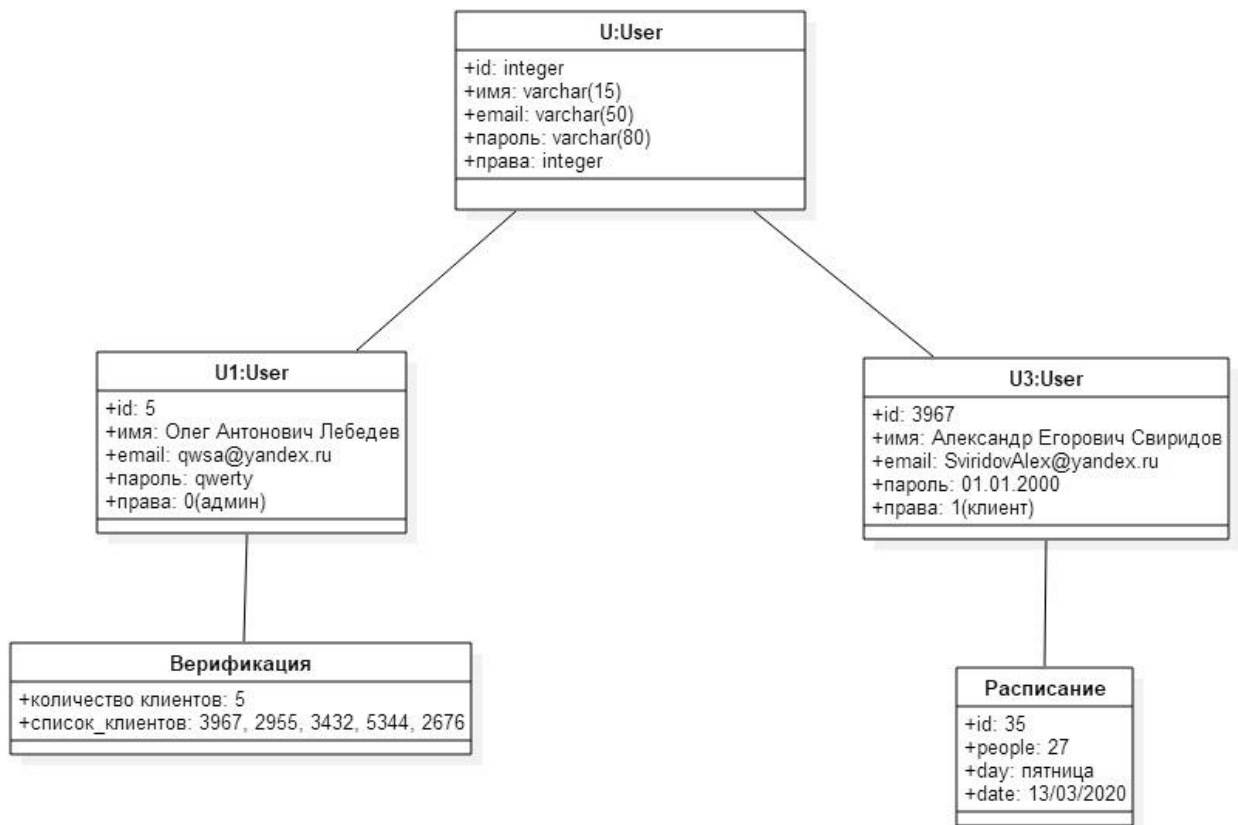


Рисунок 3 - Диаграмма объектов

1.2.4. Диаграмма последовательностей

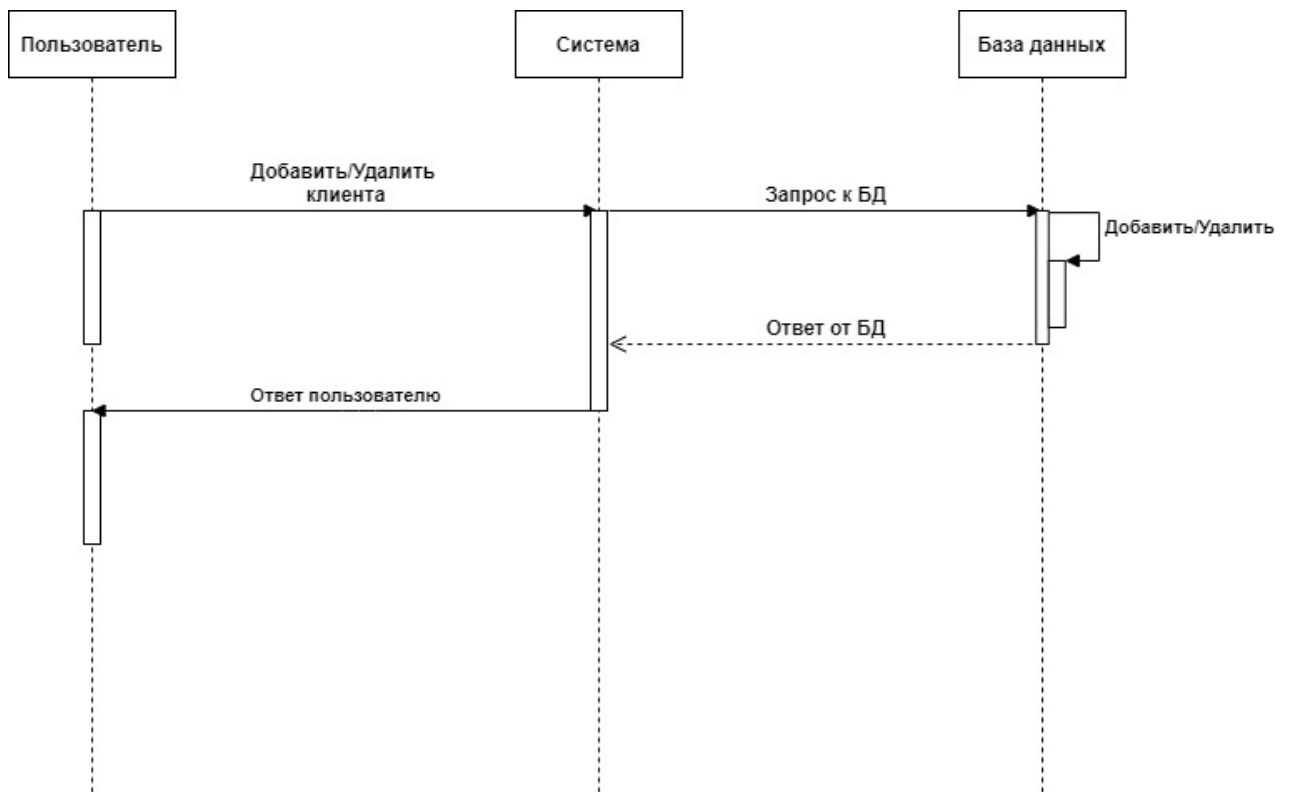


Рисунок 4 - Диаграмма последовательностей
«Добавление/удаление пользователя»



Рисунок 5 - Диаграмма последовательностей «Авторизация»

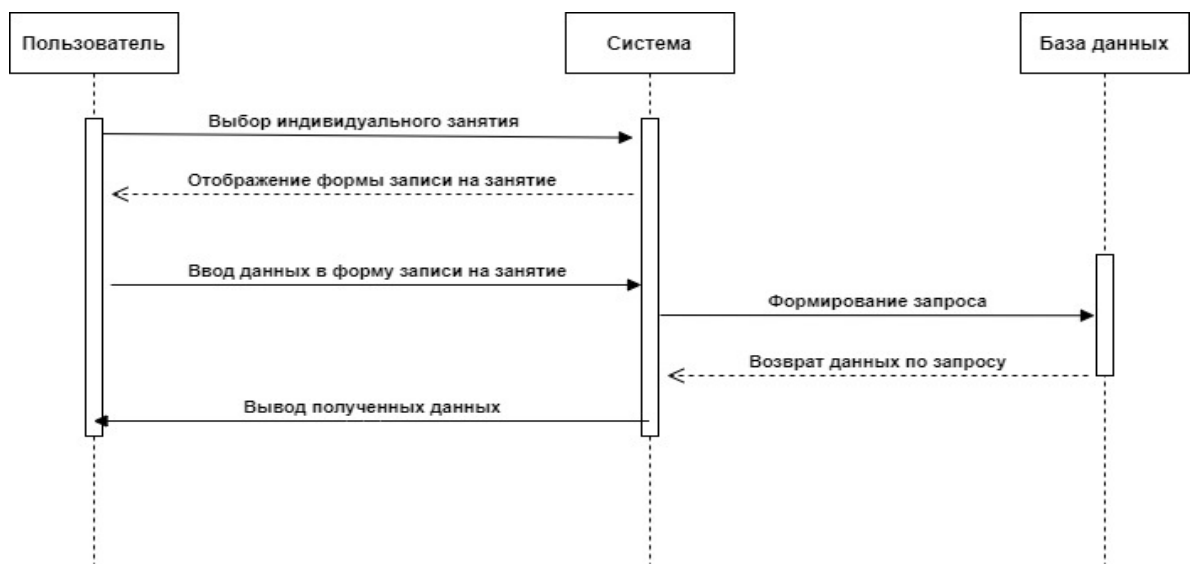


Рисунок 6 - Диаграмма последовательностей «Запись на занятия»

1.2.5. Диаграмма взаимодействий

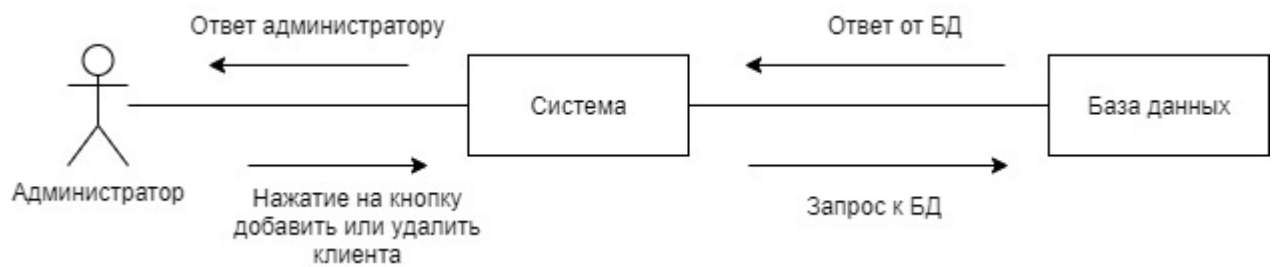


Рисунок 7 - Диаграмма взаимодействий «Добавить/удалить пользователя»

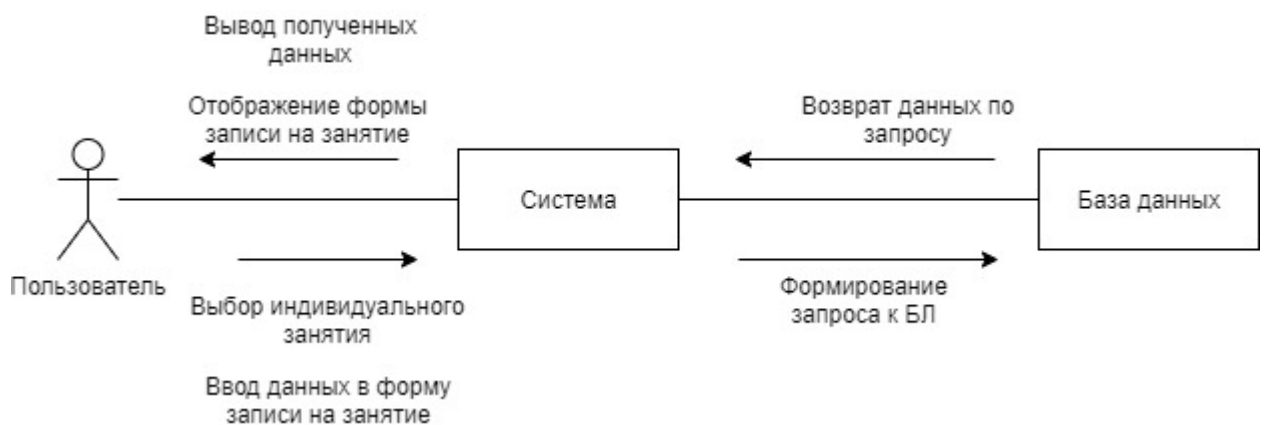


Рисунок 8 - Диаграмма взаимодействий «Выбор занятия»

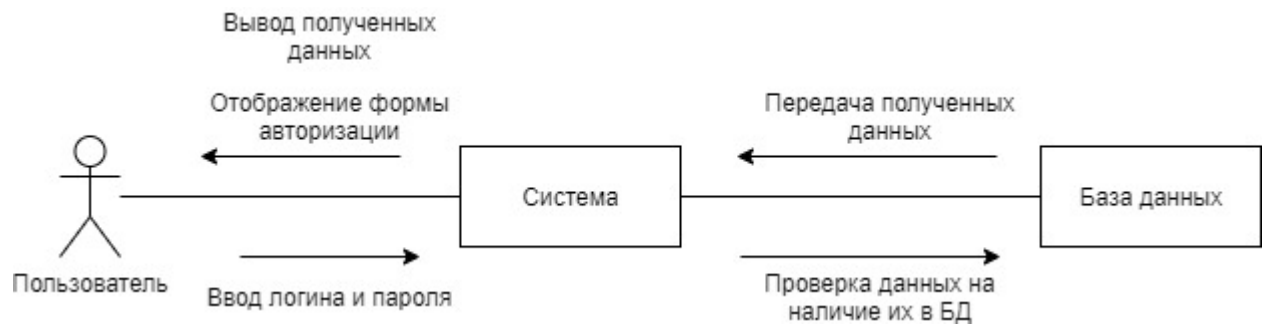


Рисунок 9 - Диаграмма взаимодействий «Авторизация»

1.2.6. Диаграмма состояний



Рисунок 10 - Диаграмма состояний «Добавление клиента»

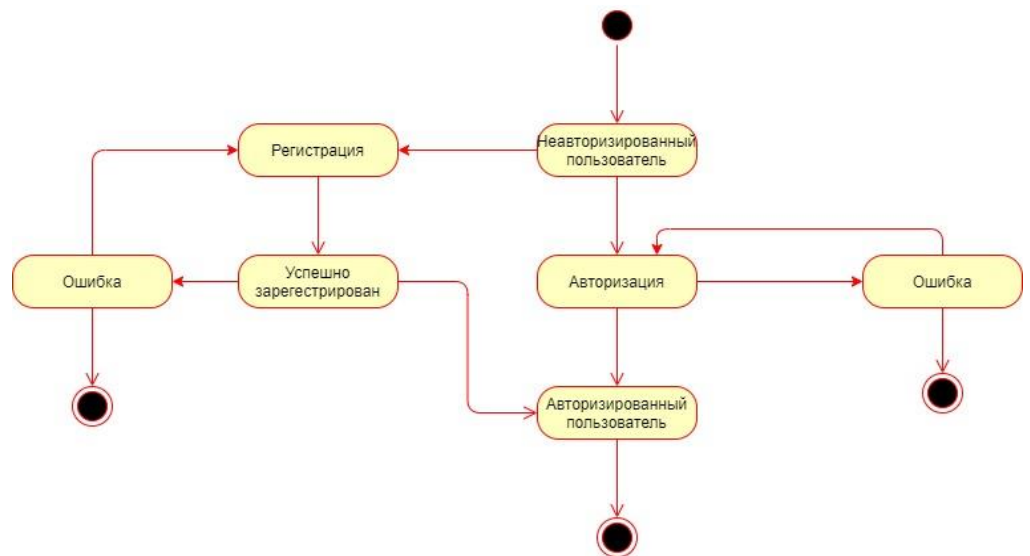


Рисунок 11- Диаграмма состояний «Авторизация»



Рисунок 12 - Диаграмма состояний «Удаление клиента»

1.2.7. Диаграмма активностей

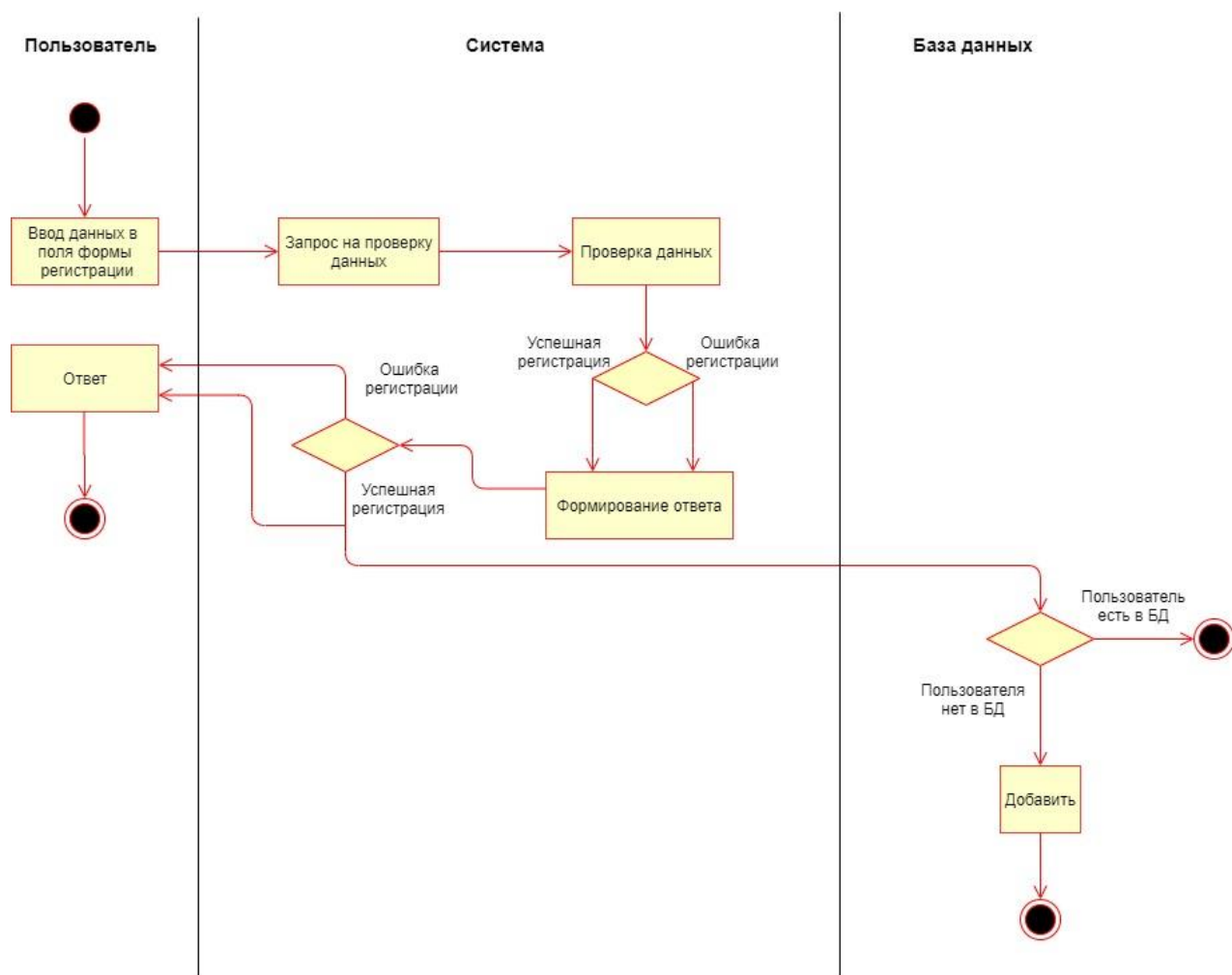


Рисунок 13 - Диаграмма активностей «Регистрация и добавление нового пользователя»

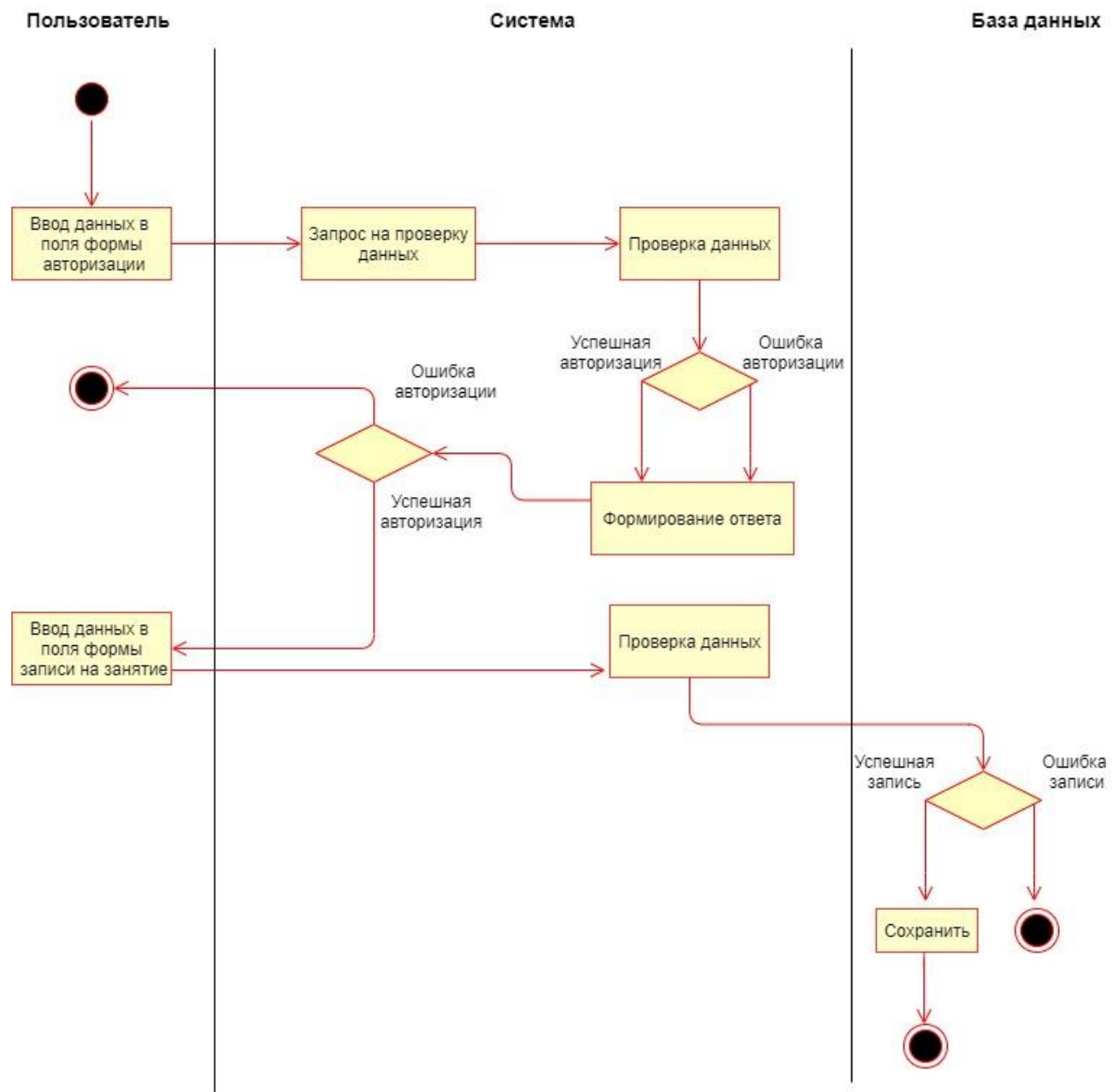


Рисунок 14 - Диаграмма активностей «Авторизация и запись на занятия»

1.2.8. Диаграмма развертывания

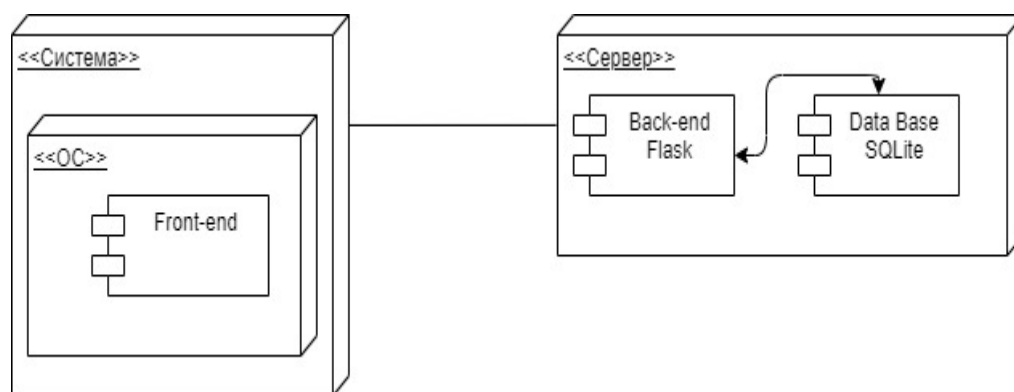


Рисунок 15 - Диаграмма развертывания

1.3. Глоссарий проекта.

Пользователь - человек, который использует действующую систему для выполнения конкретной функции.

Аккаунт (учетная запись) - хранимая в системе совокупность данных о пользователе, необходимая для его опознавания и предоставления доступа к его личным данным и настройкам.

Интерфейс – средства взаимодействия человека с системой

Web-приложение – приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

2. Постановка задач

Цель выполнения курсового проекта: создать веб-приложение, которое позволит:

- регистрацию и авторизации на сайте клиентов
- авторизации на сайте администратора
- клиентам узнать расписание тренировок и выбрать тренера, записаться на индивидуальные и групповые тренировки, отслеживать загруженность тренера на выбранное время
- администраторам добавлять новых клиентов и сотрудников, а также редактировать их профили, просматривать расписание занятий, отмечать зарегистрированных пользователей как верифицированных

3. Требования к приложению

3.1 Требования к системе в целом

Данная система должна удовлетворять следующим требованиям:

- Система должна работать в популярных браузерах, таких как: Google Chrome, Яндекс браузер, Opera.
- Выполнять функции, изложенные выше.
- Обладать простым для пользователя интерфейсом.

3.2 Требования к структуре и функционированию системы

Система должна иметь разделение на back-end и front-end.

3.3 Требования к программному обеспечению сайта

Данная система будет реализована с помощью следующий технических средств

- Языки гипертекстовой разметки: HTML и CSS
- Языки программирования: Python (Фреймворк Flask), JavaScript
- СУБД: SQLite

Причины выбора фреймворка Flask:

- Flask превосходит для разработки простых веб-приложений.
- Flask великолепен для построения простых сайтов со статическим контентом.
- Он обеспечивает весь нужный функционал и позволяет кастомизацию в больших объемах.

Причины выбора СУБД SQLite:

- Файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины.
- Отсутствие необходимости настройки сервера СУБД.
- Очень экономичная, в плане ресурсов, архитектура.

4. Основная часть

4.1. Основные этапы выполнения работы

Процесс выполнения работы включает в себя следующие основные этапы:

1. Разработка документации к приложению
2. Разработка приложения
3. Тестирование приложения

Каждый из этапов содержит в себе множество подзадач. Рассмотрим этапы по порядку.

Этап разработки документации к приложению включает в себя:

1. Постановка задач
2. Установка сроков реализации проекта и документации
3. Проведение анализа предметной области
4. Создание диаграмм приложения
5. Создание технического задания
6. Создание отчетного документа по ролям
7. Создание курсового проекта
8. Создание проекта в системе Git

Этап разработки приложения включает в себя:

1. Подготовку используемых для разработки средств к работе
2. Проектирование Базы Данных
3. Подключение Базы Данных
4. Создание скелета приложения
5. Полное создание приложения

Этап тестирования приложения включает в себя:

1. Создание сценариев
2. Проведение тестирования модулей
3. Выявление ошибок
4. Создание документа по результатам тестирования
5. Принятие решения о соответствии приложения предъявленным требованиям

4.2. Структура приложения

В результате работы над курсовым проектом была реализована клиентская и серверная часть.

На рисунке ниже показан главный экран нашего приложения.

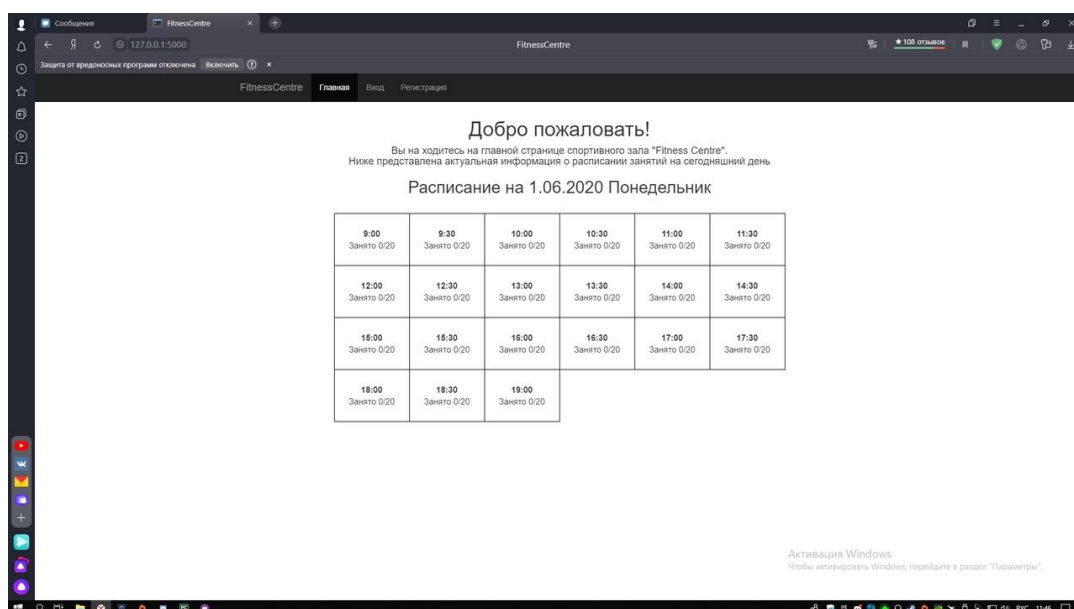


Рисунок 16 – Главный экран

Обычно в подобных приложениях при входе дается куча информации, что усложняет поиск нужной. Поэтому мы максимально упростили главный экран и вывели на него только самую востребованную информацию.

При входе пользователь увидит актуально расписание на текущую дату, а также сколько человек на какое время записалось, после чего сможет определиться какое время ему больше подходит.

Для того чтобы записаться на занятия или получить больше информации клиенту стоит пройти регистрацию.

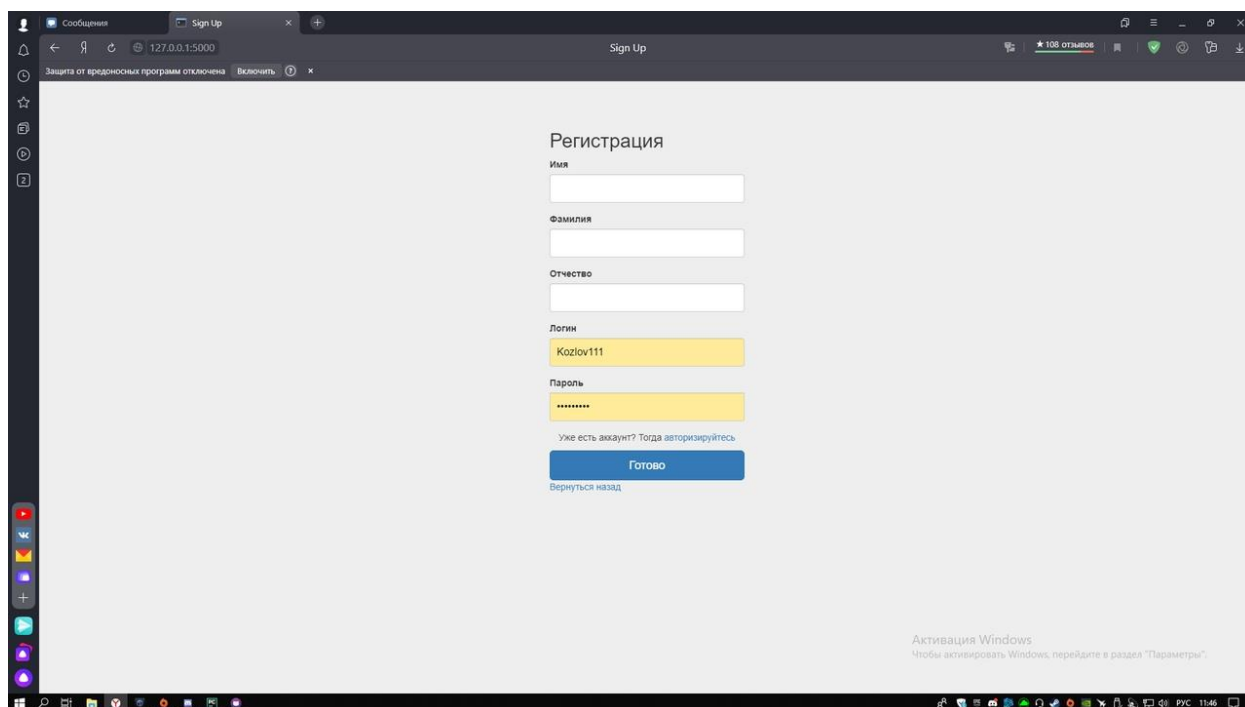


Рисунок 17 – Окно регистрации

В окне регистрации клиенту нужно ввести свое имя, фамилию, отчество, логин и пароль. После верификации клиент получит права пользователя.

Если он уже зарегистрирован, тогда для получения функций пользователя ему необходимо авторизоваться.

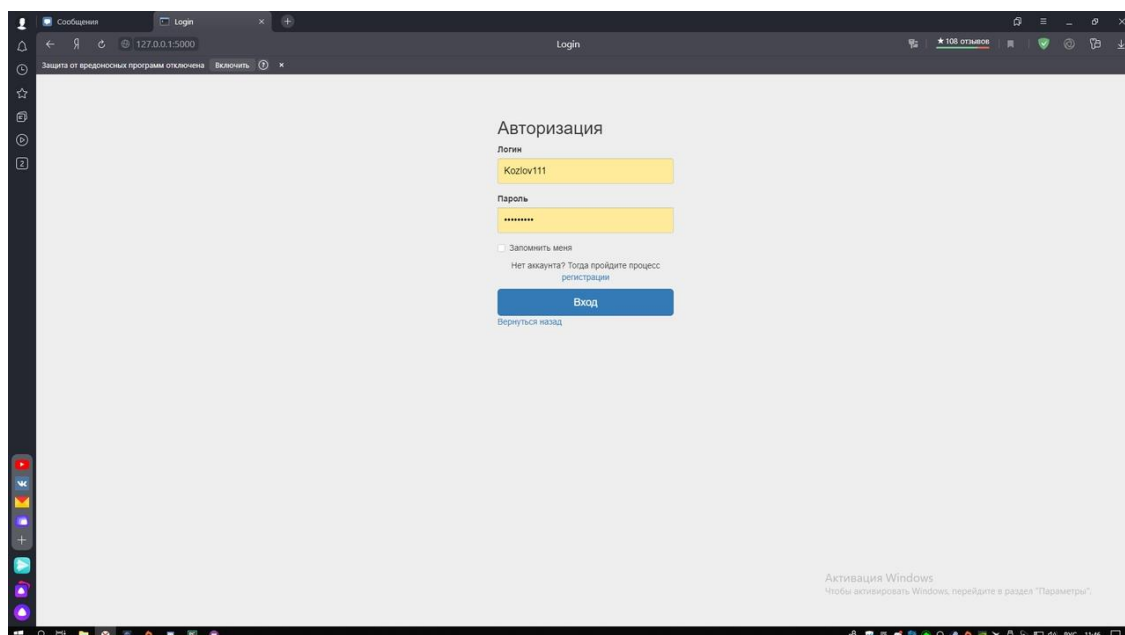


Рисунок 18 – Окно авторизации

Для авторизации клиенту потребуется только логин, который он использовал при регистрации и пароль.

После авторизации клиент попадает на домашнюю страницу.

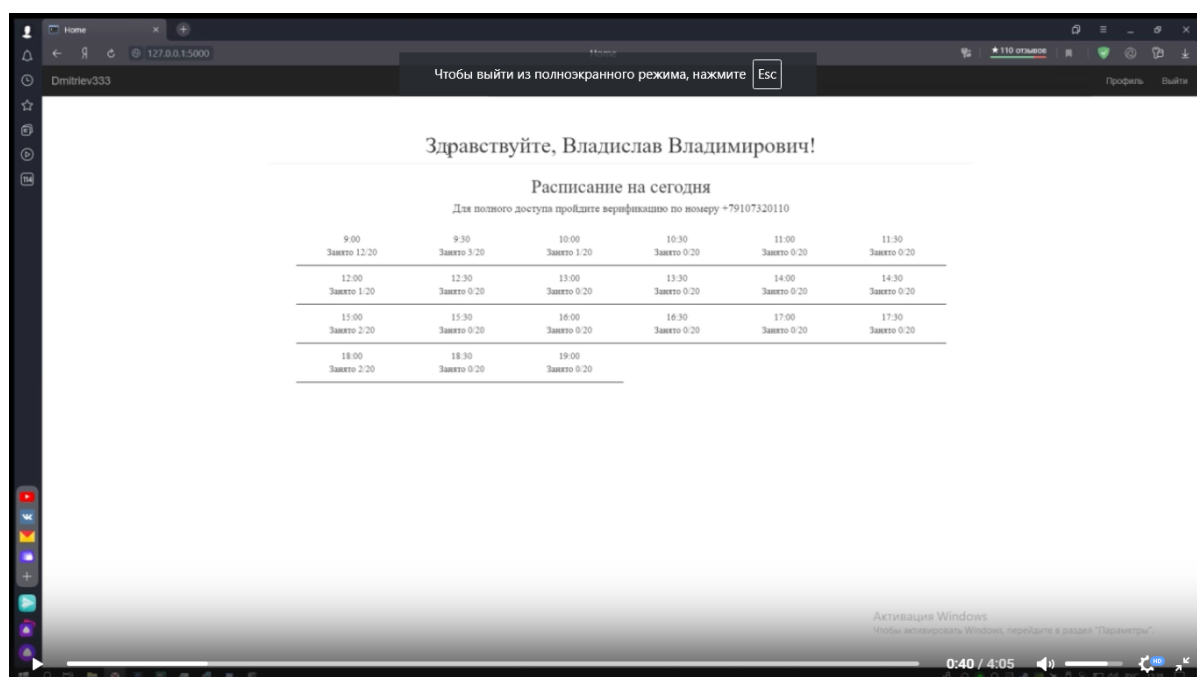


Рисунок 19 – Домашняя страница не верифицированного пользователя

Однако, чтобы получить права пользователя клиент должен пройти верификацию. После верификации у него появится окно, где он может записаться на занятия на удобное для него дату.

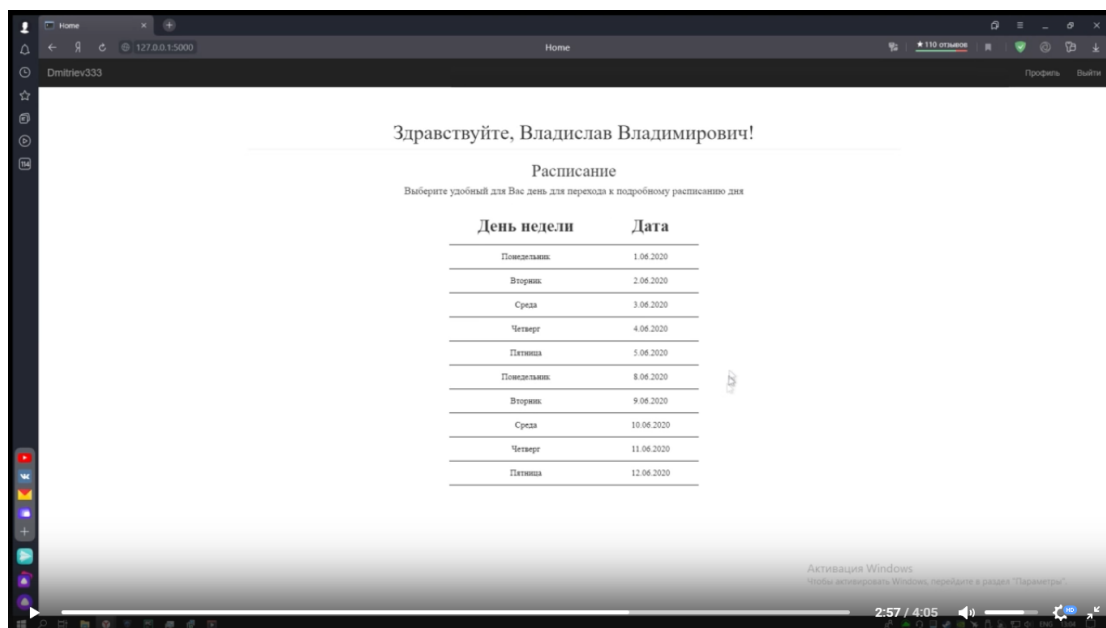


Рисунок 20 – Домашняя страница пользователя

После выбора дня перед клиентом появится список всех занятий.

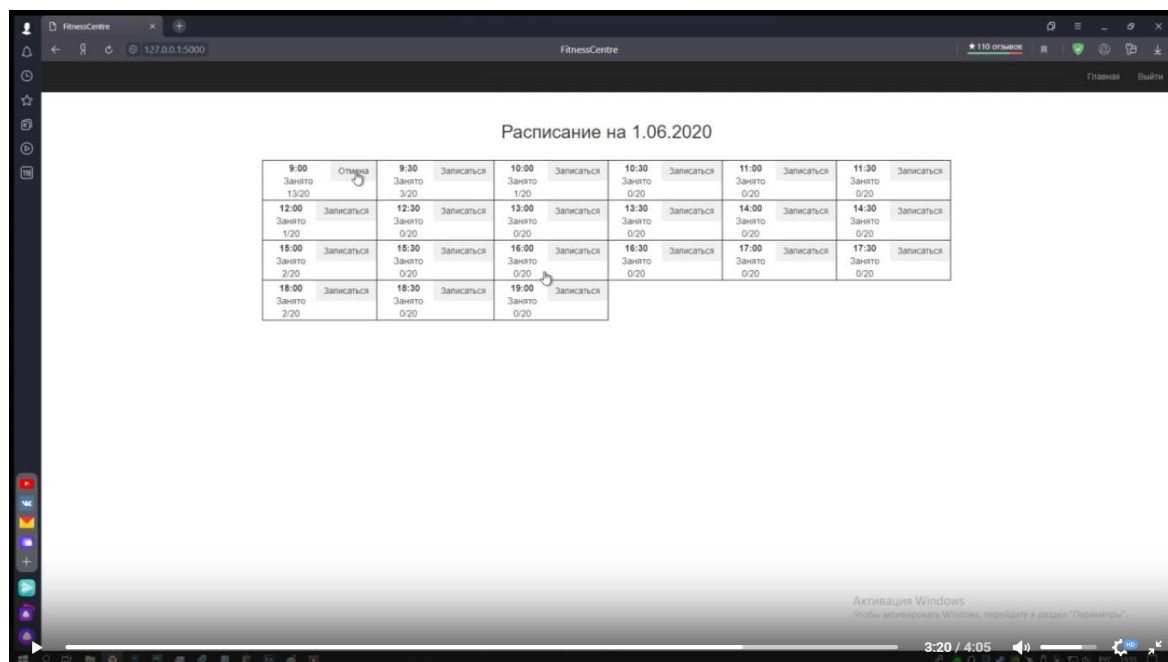


Рисунок 21 – Запись/отмена

В этом окне пользователь может посмотреть загруженность каждого занятия и записать на удобное время или отменить свою запись.

В личном кабинете пользователю доступны просмотр его данных.

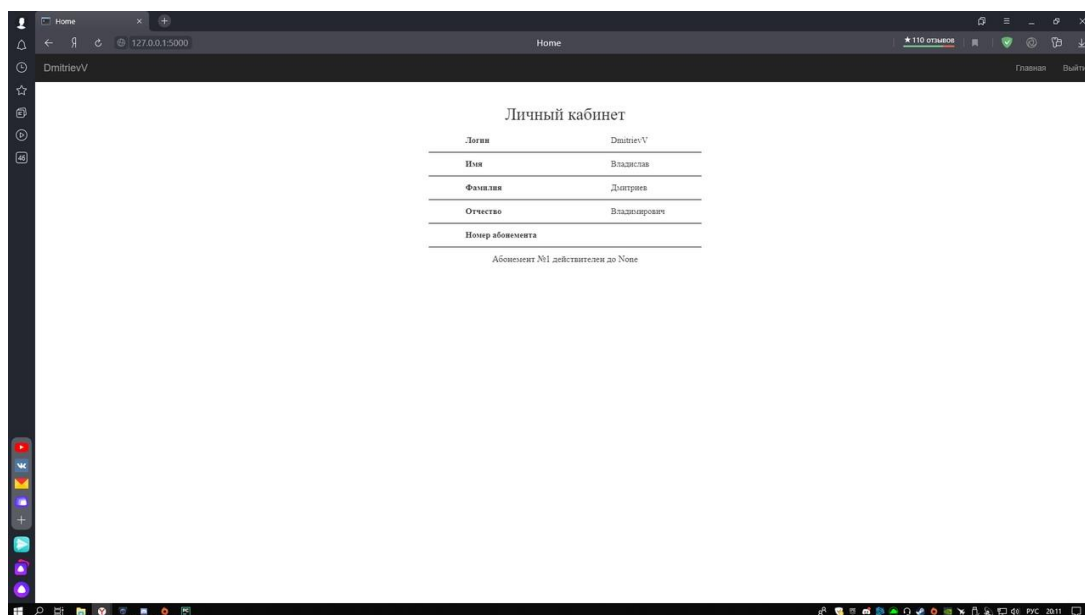


Рисунок 22 – Личный кабинет

Администратор при авторизации попадает на свою домашнюю страницу.

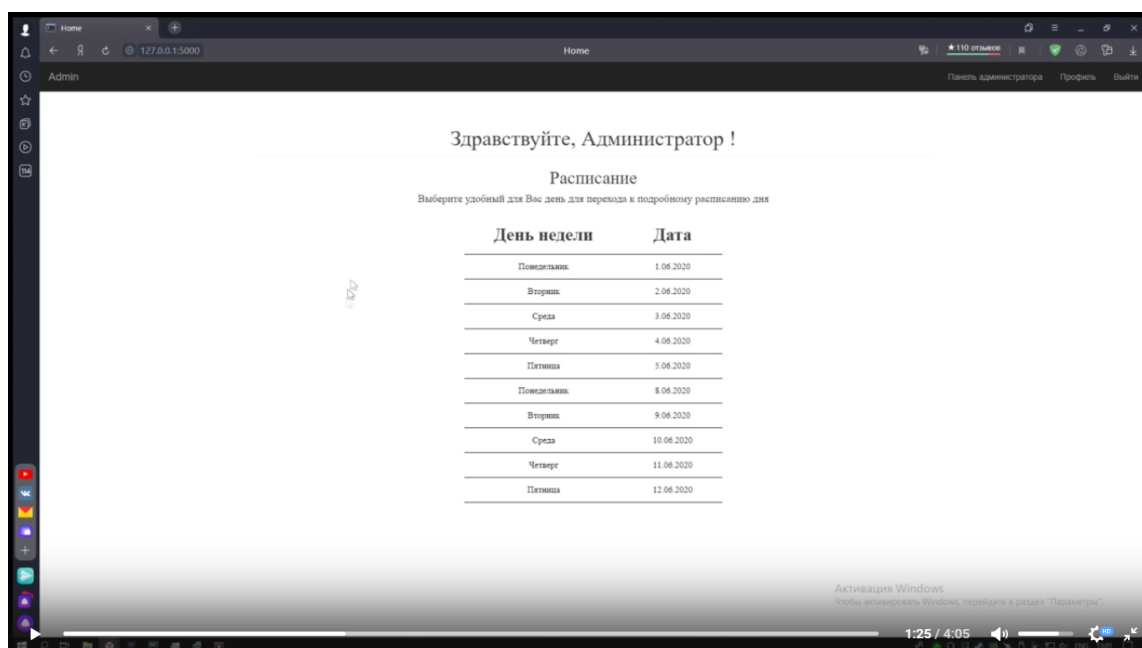


Рисунок 23 – Домашняя страница администратора

На ней он видит полную версию расписания, а также панель администратора.

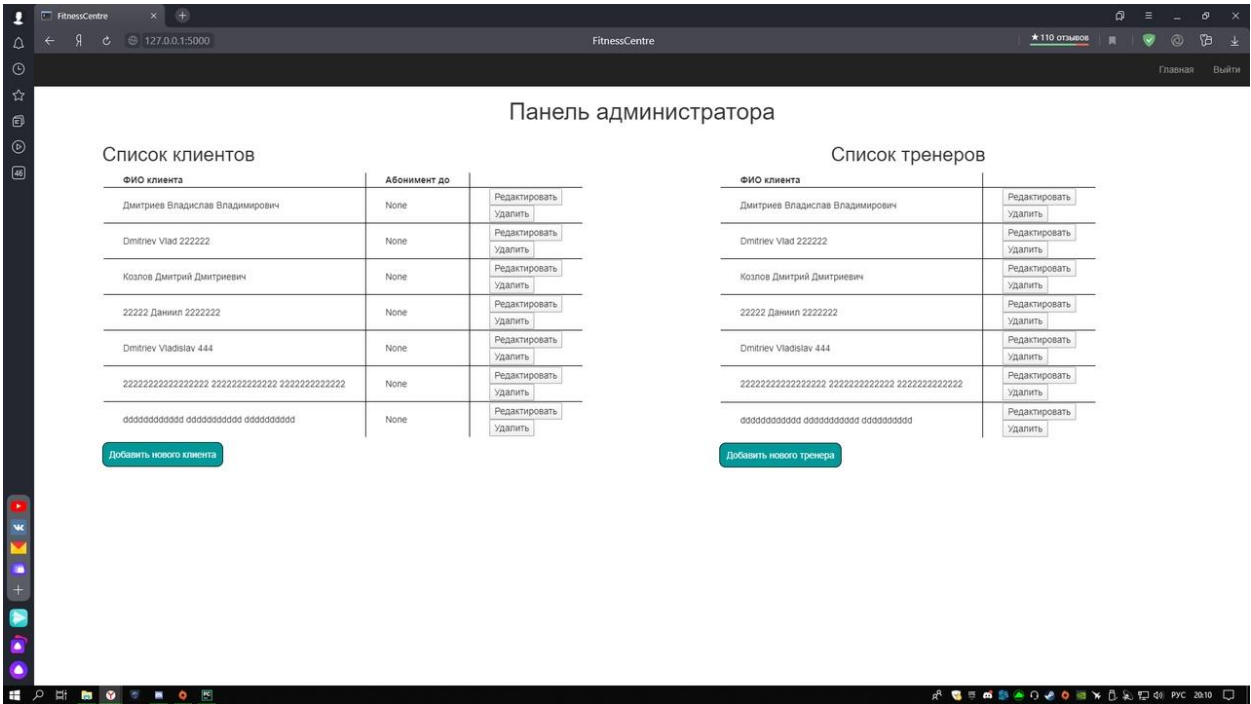


Рисунок 24 – Панель администратора

В этой панели ему доступен просмотр списков клиентов и тренеров, возможность добавить новых и редактировать информацию уже добавленных.

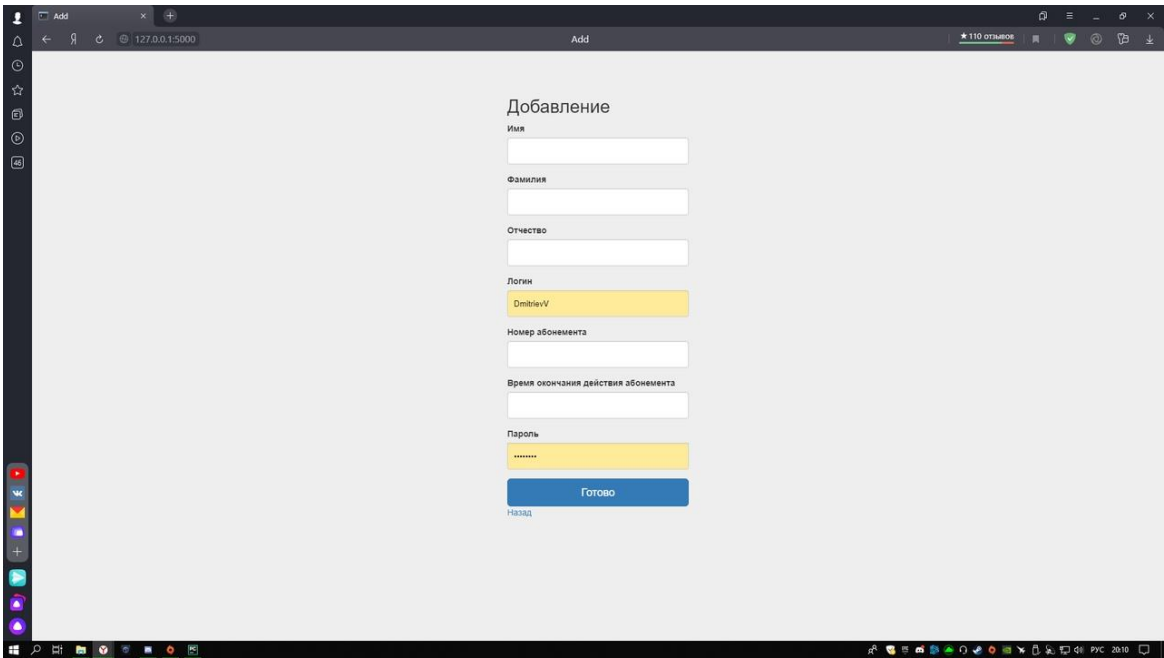


Рисунок 25 – Добавление клиента

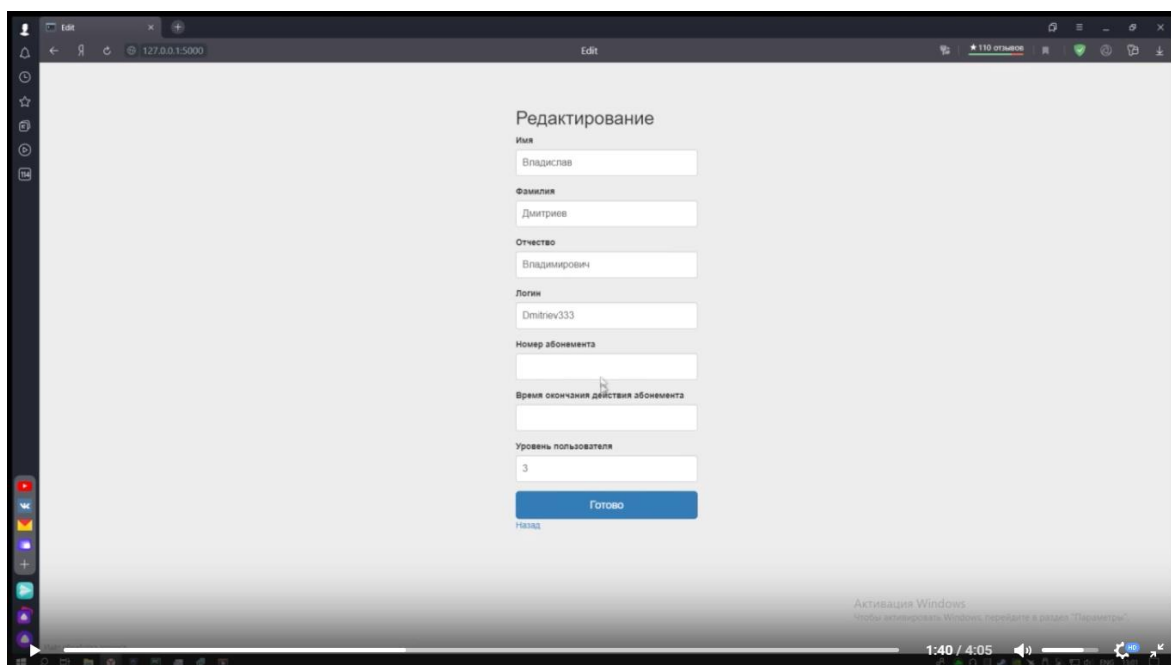


Рисунок 26 – Редактирование информации клиента

4.3 Тестирование приложения

4.3.1 Сценарии воронок конверсии

1) Неавторизованный пользователь:

1. Зайти на главную страницу - авторизовался.
2. Зайти на главную страницу - зарегистрировался
3. Зайти на главную страницу - вышел с сайта
4. Зайти на главную страницу - просмотрел расписание - зарегистрировался - вышел с сайта
5. Зайти на главную страницу - просмотрел расписание - авторизовался - вышел с сайта

2) Авторизованный пользователь:

1. Авторизовался - вышел с сайта/аккаунта
2. Авторизовался - просмотрел расписание - вышел с сайта/аккаунта

3) Авторизованный верифицированный пользователь

1. Авторизовался - просмотрел расписание - записался на занятия(в группе)/ к тренеру(индивидуальные) - вышел с сайта/аккаунта
2. Авторизовался - просмотрел расписание - записался на занятия(в группе)/ к тренеру(индивидуальные) - отменил запись - вышел с сайта/аккаунта
3. Авторизовался - отменил запись - вышел с сайта/аккаунта
4. Авторизовался - просмотрел расписание - отменил запись - вышел с сайта/аккаунта
5. Авторизовался - просмотрел расписание - отменил запись - записался на другое время - вышел с сайта/аккаунта

4) Администратор:

1. Авторизовался - просмотрел расписание - отредактировал расписание - вышел с сайта/аккаунта
2. Авторизовался - просмотрел список тренеров - добавил нового тренера - вышел с сайта/аккаунта
3. Авторизовался - просмотрел список тренеров - вышел с сайта/аккаунта

4. Авторизовался - открыл список не верифицированных пользователей - верифицировал пользователей - вышел с сайта/аккаунта
5. Авторизовался - открыл список не верифицированных пользователей - вышел с сайта/аккаунта
6. Авторизовался - открыл список пользователей - открыл информацию пользователя - удалил/отредактировал данные пользователя - вышел с сайта/аккаунта
7. Авторизовался - открыл список пользователей - открыл информацию пользователя - закрыл окно информации - вышел с сайта/аккаунта

4.3.2 Результаты тестирования функционала

Ход тестирования:

1. Регистрация нового пользователя:

- Открыть сайт
- Нажать на кнопку регистрации
- Заполнить все поля
- Нажать на кнопку "Готово"

Проверка: успешно.

2. Авторизация

- Нажать на кнопку "Авторизоваться"
- Ввести логин
- Ввести пароль
- Нажать на кнопку "Вход"

Проверка: успешно.

3. Запись на занятие

- Авторизоваться в системе
- Выбрать нужный день и нажать на него
- На появившейся таблице выбрать нужный день и нажать кнопку "Записаться"

Проверка: успешно.

4. Выход из системы

- Авторизоваться в системе
- Нажать на кнопку "Выйти"

Проверка: успешно.

5. Удаление пользователя

- Авторизоваться в системе
- Нажать на кнопку "Панель администратора"
- Выбрать соответствующего пользователя и нажать на кнопку "Удалить" рядом с пользователем

Проверка: успешно.

6. Добавление пользователя

- Авторизоваться в системе
- Нажать на кнопку "Панель администратора"

- Нажать на кнопку "Добавить нового клиента"

Проверка: успешно.

7. Редактирование информации о клиентах

- Авторизоваться в системе
- Нажать на кнопку "Панель администратора"
- Нажать на нужного клиента
- В появившейся форме изменить необходимые данные
- Нажать на кнопку "Готово"

Проверка: успешно.

Результаты тестирования показали, что все основные функции системы успешно работают.

Заключение

В заключение, итогом проделанной работы является реализованное web-приложение для фитнеса центра для упрощения процесса записи на занятия/тренировки, внесения изменений в расписания занятий, упрощение отслеживания записавшихся. В результате тестирования и в соответствии с прописанными в техническом задании требованиями приложение работает.

Список используемой литературы

- 1) Р. Никсон Создаем динамические веб-сайты

<https://booster.by/files/oeu.pdf>

(дата обращения 03.06.2020)

[Электронный ресурс]

- 2) Flask

<https://flask.palletsprojects.com/en/1.1.x/>

[Электронный ресурс]

- 3) Python

<https://www.python.org/>

[Электронный ресурс]