

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему: «Разработка автоматизированного администратора клуба с биометрической  
идентификацией»

Обучающийся

И.А. Бутрим

(Инициалы Фамилия)

(личная подпись)

Руководитель

С.В. Митин

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент, С.А. Гудкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

## Аннотация

Тема бакалаврской работы: «Разработка автоматизированного администратора клуба с биометрической идентификацией».

В работе описан практический способ применения системы с биоидентификацией в фитнес зале ООО «НЕО» в технопарке в сфере высоких технологий «Жигулевская долина», расположенному в г. Тольятти, ул. Южное шоссе, 165А.

Проведено описание предметной области.

Рассмотрены способы и методы идентификации по фото, выбран метод гибкого сравнения на графах.

Созданы логическая и физическая модели базы данных по методологии ER.

Выполнен технико-экономический анализ, обоснована возможность замены администратора зала.

Проведен анализ нынешнего процесса управления и сравнение аспектов «Как было» и «Как будет».

Проведена оценка экономической эффективности проекта.

Сформулированы требования к программно-аппаратному обеспечению сервера, а также аппаратного комплекса для распознавания клиентов.

Описана актуальность проекта и сформулированы требования к программно-аппаратному комплексу.

Выбрана среда веб-разработки и система управления базами данных, а также реализован и протестирован пользовательский интерфейс.

Разработан и протестирован программный модуль для идентификации.

Бакалаврская работа состоит из 63 страниц текста, 43 рисунков, 9 таблиц и 22 источников.

## **Abstract**

The theme of the bachelor's work is the development of an automated club administrator with biometric identification.

The paper describes a practical way of application of the system with bioidentification in the fitness hall of LLC "NEO" in the technopark in the sphere of high technologies "Zhigulevskaya Dolina", located in Togliatti, Yuzhnoe shosse str, 165A.

The description of the subject area is carried out.

Methods and techniques of identification by photo are considered, the method of flexible comparison on graphs.

Logical and physical database models were created using ER methodology.

A feasibility analysis was performed, the possibility of replacing the hall administrator.

Analyzed the current management process and compared aspects of "As it was" and "As it will be" aspects.

The cost-effectiveness of the project was evaluated.

The requirements to the software and hardware of the server as well as hardware complex for client recognition.

The relevance of the project is described and requirements to the hardware and software complex are formulated.

A web development environment and a database management system are selected, and the user is realized and tested.

The user interface was realized and tested.

A program module for identification is developed and tested.

The bachelor's thesis consists of 60 pages of text, 39 figures, 9 tables and 22 sources.

## Оглавление

Введение.....	5
Глава 1 Функциональное моделирование предметной области.....	7
1.1 Техничко-экономическая характеристика предметной области .....	7
1.2 Анализ нынешнего процесса управления административными задачами .....	8
1.3 Постановка задачи на разработку программно-аппаратного комплекса .....	9
1.4 Улучшения бизнес-процессов управления административными задачами .....	11
Глава 2 Логическое проектирование программно-аппаратного комплекса .....	13
2.1 Выбор технологии логического проектирования .....	13
2.2 Логическая модель программно-аппаратного комплекса .....	13
2.3 Проектирование базы данных .....	20
2.4 Требования к программному обеспечению .....	21
2.5 Требования к аппаратному обеспечению модуля считывания и обработки биометрических данных в реальном времени.....	22
Глава 3 Реализация программно-аппаратного комплекса.....	23
3.1 Выбор архитектуры .....	23
3.2 Выбор средства реализации программно-аппаратного комплекса ..	25
3.3 Разработка физической модели данных .....	30
3.4 Аппаратное обеспечение комплекса .....	31
3.5 Описание интерфейса пользователя .....	33
3.6 Описание роли «Администратор».....	46
3.7 Описание роли «Модератор» .....	51
3.8 Тестирование программно-аппаратного комплекса .....	52
3.9 Оценка экономической эффективности разработки.....	56
Заключение.....	59
Список используемой литературы и используемых источников.....	61

## Введение

В современном мире технологии активно проникают в различные сферы жизни, в том числе в область фитнес-индустрии. С постоянным ростом спроса на здоровый образ жизни и фитнес-услуги, становится важным обеспечить эффективное управление и автоматизацию процессов в фитнес-залах.

В связи с этим актуальной становится задача разработки программно-аппаратного комплекса, который бы автоматизировал работу администратора фитнес-зала, включая процесс идентификации клиентов с использованием биометрических данных и нейронных сетей.

Объектом исследования является процесс автоматизации работы фитнес-зала на основе программно-аппаратного комплекса.

Предметом исследования является разработка программно-аппаратного комплекса, обеспечивающего биометрическую идентификацию клиентов фитнес-зала на основе компьютерного зрения и нейронных сетей.

Целью данной работы является создание функционального программно-аппаратного комплекса, способного автоматизировать процессы регистрации и инновационной идентификации клиентов фитнес-зала, сокращая человеческий фактор и повышая эффективность управления.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить работу компании ООО «НЕО ЦЕНТР» (изучить направления деятельности компании, рассмотреть основные процессы, построить организационную структуру);
- спроектировать концептуальную модель предметной области;
- спроектировать логическую модель программно-аппаратного комплекса;
- разработать программное обеспечение в виде информативного одностраничного сайта с личными кабинетами и автоматизированной информационной системы для загрузки,

- обработки и идентификации лиц по Face ID на основе нейронных сетей, ведения базы данных клиентов, покупки абонементов;
- разработать панели администратора и модератора для удобства разбиение управления фитнесом менеджером при необходимости;
  - определить требования к аппаратной части комплекса, обеспечивающую считывание и обработку биометрических данных в реальном времени;
  - интегрировать программную и аппаратную часть в единый функциональный комплекс;
  - интегрировать модуль считывания и обработки биометрических данных в реальном времени в единый функциональный комплекс;
  - провести тестирование разработанного модуля считывания и обработки биометрических данных в реальных условиях фитнес-зала;
  - оценить экономическую эффективность разработанного программно-аппаратного комплекса.

Выпускная квалификационная работа состоит из введения, трёх глав, заключения, списка используемой литературы и приложения.

В первой главе проводится анализ и моделирование предметной области, исследуются проблемы и их решения. Построены функциональные модели «Как есть» и «Как должно быть».

Во второй главе проводится логическое проектирование программно-аппаратного комплекса. Определяется внешний вид и структура. Проводится анализ функций и разработка требований к информационной системе.

В третьей главе выполнена разработка программного и аппаратного обеспечения программно-аппаратного комплекса, проведены результаты тестирования и оценка экономической эффективности.

## **Глава 1   Функциональное моделирование предметной области**

### **1.1   Технико-экономическая характеристика предметной области**

#### **1.1.1   Характеристика ООО «НЕО ЦЕНТР»**

Основная деятельность компании направлена на оказание деятельности фитнес-центров (ОКВЭД: 93.13).

Компания расположена в технопарке в сфере высоких технологий «Жигулевская долина». Благодаря удобному расположению в технопарке, является популярным местом не только для профессиональных спортсменов, но так и для резидентов и гостей технопарка.

ООО «НЕО ЦЕНТР» предоставляет современные и удобные тренажеры, обеспечивающие клиентам удобство в области фитнеса и тренировок.

Основной целью компании является внедрение современных инновационных технологий, для предоставления клиентам наилучшего опыта тренировок, чтобы сделать их более доступными и привлекательными для широкой аудитории.

#### **1.1.2   Организационная структура**

Генеральный директор – высший руководитель, отвечающий за общее управление и принятие стратегических решений.

Менеджер зала (управляющий) ответственен за операционную деятельность фитнес-зала, включая координацию работы отделов и обеспечение высокого уровня обслуживания клиентов.

Отдел ресепшена включает в себя администраторов ресепшена, которые осуществляют прием посетителей, выдачу абонементов и оказание первичной информационной поддержки.

В тренажерном зале работают тренеры, ответственные за проведение индивидуальных и групповых тренировок, разработку программ тренировок и контроль за безопасностью занятий.

Обслуживающий персонал состоит из уборщицы, ответственной за поддержание чистоты и порядка в помещениях фитнес-зала, обеспечивая комфортную атмосферу для посетителей.

Бухгалтерия занимается ведением финансовой отчетности, расчетом заработной платы сотрудников, контролем финансовых потоков и налогового учета компании.

Организационная структура компании представлена на рисунке 1.

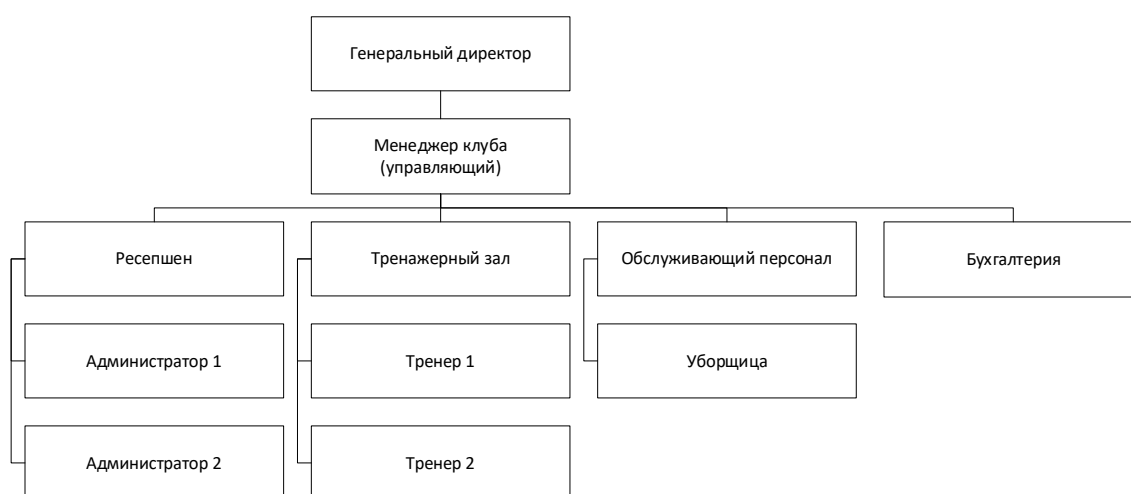


Рисунок 1 – Организационная структура компании

Такая структура организует работу компании и обеспечивает обслуживание клиентов в фитнес-центре.

## **1.2 Анализ нынешнего процесса управления административными задачами**

### **1.2.1 Описание предметной области**

В рамках текущего процесса управления административными задачами в фитнес-зале администратор принимает на себя ряд обязанностей, начиная от приема новых клиентов и регистрации их в системе до продажи абонементов и выдачи физических карт для доступа в зал.



Клиенты должны лично посетить фитнес-зал для прохождения этих процедур, что может привести к длительным ожиданиям в очереди и потере времени. Кроме того, администратору приходится обрабатывать большой объем информации вручную, что может привести к ошибкам и задержкам в обслуживании.

### **1.2.2 Обоснование необходимости автоматизированного варианта решения**

Нынешний процесс управления административными задачами в фитнес-зале предполагает наличие администратора, который регистрирует клиентов, продает абонементы и выдает физические карты для доступа в зал. Этот процесс подвержен ряду ограничений, включая необходимость наличия квалифицированного персонала на ресепшене, риск возникновения ошибок при ручной регистрации и длительное ожидание в очередях для новых клиентов. Также, администратор должен выполнять ряд рутинных задач, что может снижать его производительность и внимание к клиентам.

Внедрение инновационной и современной системы управления клиентами с использованием биометрической идентификации позволит решить эти проблемы. Она предоставит клиентам возможность самостоятельно регистрироваться, покупать абонементы и получать доступ в зал через онлайн-платформу, используя биометрические данные для идентификации. Это сократит время на административные процедуры, уменьшит риск возникновения ошибок и повысит удобство обслуживания для клиентов. Кроме того, инновационный метод доступа в зал с помощью биометрической идентификации предоставит дополнительный уровень безопасности и защиты данных клиентов.

### **1.3 Постановка задачи на разработку программно-аппаратного комплекса**

Цели программно-аппаратного комплекса:

- автоматизация процессов управления фитнес-залом, включая регистрацию клиентов, продажу абонементов и контроль доступа в зал;
- обеспечение безопасного и удобного доступа клиентов в зал с использованием биометрической идентификации.

Требования функциональности:

- интерфейс клиента, включая регистрацию личного кабинета с помощью информативного одностраничного сайта через смс-сервис;
- оплата абонементов в личном кабинете онлайн;
- добавление фотографии, на основе которой аппаратная часть идентифицирует клиента по биометрии в части лица по средствам нейронной сети, позволяя входить и выходить из зала активного отдыха через дверь с электромагнитным замком;
- интерфейс возможности администрирования, включая просмотр и изменения данных клиентов, тренеров на сайте, цен на абонементы, просмотр всех смс, мониторинг посещений входа-выхода в зал.
- интерфейс возможности модератора, включая редактирование списка тренеров, включение и отключение СМС, изменения дополнительных страниц ознакомления

Требования к архитектуре и реализации:

- гибкость и масштабируемость системы, чтобы она могла адаптироваться к изменяющимся потребностям и объему клиентов фитнес-зала;
- высокий уровень безопасности и защиты данных, чтобы гарантировать конфиденциальность информации клиентов;
- использование современных технологий биометрической идентификации для обеспечения точности и надежности процесса.

## 1.4 Улучшения бизнес-процессов управления административными задачами

Сравнение процессов и функций до и после внедрения программно-аппаратного комплекса представлено в таблице 1. Она иллюстрирует, какие улучшения принесла автоматизация работы фитнес-зала с использованием новой технологии [3], [8], [11].

Таблица 1 – Сравнение аспектов бизнес-процессов «Как было» и «Как будет»

Аспект	Как было	Как будет
Регистрация клиентов	Ручная регистрация клиентов администратором на ресепшене, что требует наличия квалифицированного персонала и может приводить к длительным очередям и ошибкам.	Самостоятельная регистрация клиентов через онлайн-платформу. Клиенты заполняют регистрационную форму, указывая необходимые данные, что сокращает время на административные процедуры и уменьшает риск ошибок.
Покупка абонементов	Покупка абонементов осуществляется на ресепшене через администратора.	Онлайн-покупка абонементов через интернет-платформу с использованием биометрической идентификации. Клиенты могут выбрать подходящий абонемент и оплатить его онлайн, что упрощает процесс и делает его доступным 24/7.
Доступ в зал	Выдача физических карт для доступа в зал.	Доступ в зал осуществляется посредством биометрической идентификации через систему распознавания лиц, что обеспечивает быстрый и безопасный доступ без необходимости использования физических карт и минимизирует риск утери или кражи.

Продолжение таблицы 1

Аспект	Как было	Как будет
Управление клиентской базой данных	Управление клиентской базой данных и модерация технических моментов осуществляется администратором с использованием ручных методов, что может быть трудоемким и требует дополнительного времени и усилий.	Управление клиентской базой данных осуществляется через панель администратора. Администратор может легко добавлять, редактировать и удалять записи клиентов, а также отслеживать их активность, что повышает эффективность и точность управления данными.
Управление модерацией технических моментов		Управление модерацией технических моментов осуществляется через панель модератора. Модератор может легко добавлять, редактировать и удалять тренеров, включать или отключать смс-сервис, а также изменять дополнительные страница для ознакомления клиента.

В первой главе была представлена основная информация о компании ООО "НЕО ЦЕНТР", определена её организационная структура. Также были выявлены существующие недостатки и определен объект для дальнейшей автоматизации с учетом выявленных проблем. На основе проведенного анализа определены функциональные и архитектурные требования к программно-аппаратному комплексу.

## **Глава 2 Логическое проектирование программно-аппаратного комплекса**

### **2.1 Выбор технологии логического проектирования**

Для логического проектирования используются два типа моделей: UML и ER диаграммы.

UML – это объектно-ориентированный язык для описания бизнес-процессов, а ER диаграммы представляют собой структурно-функциональную модель для описания баз данных [1], [9], [16].

Для реализации логического проектирования применяются следующие инструменты: StarUML и Erwin.

StarUML – это проект с открытым кодом, предназначенный для создания гибких и функциональных моделей на платформе UML/MDA для операционных систем Windows. Он обладает множеством возможностей моделирования, простым графическим интерфейсом и поддержкой плагинов.

Erwin – это прикладная программа, специализирующаяся на проектировании и документировании баз данных. Она основана на методологии IDEF1X и предназначена для создания как физических, так и логических моделей данных с использованием ER диаграмм. Однако, Erwin также имеет свои недостатки, включая сложность изучения и внедрения, а также избыточность языка, что делает его критикуемым как слишком большой и сложный инструмент.

### **2.2 Логическая модель программно-аппаратного комплекса**

Для определения ролей пользователей и их взаимодействия с системой используется диаграмма претендентов, которая представлена на рисунке 2.

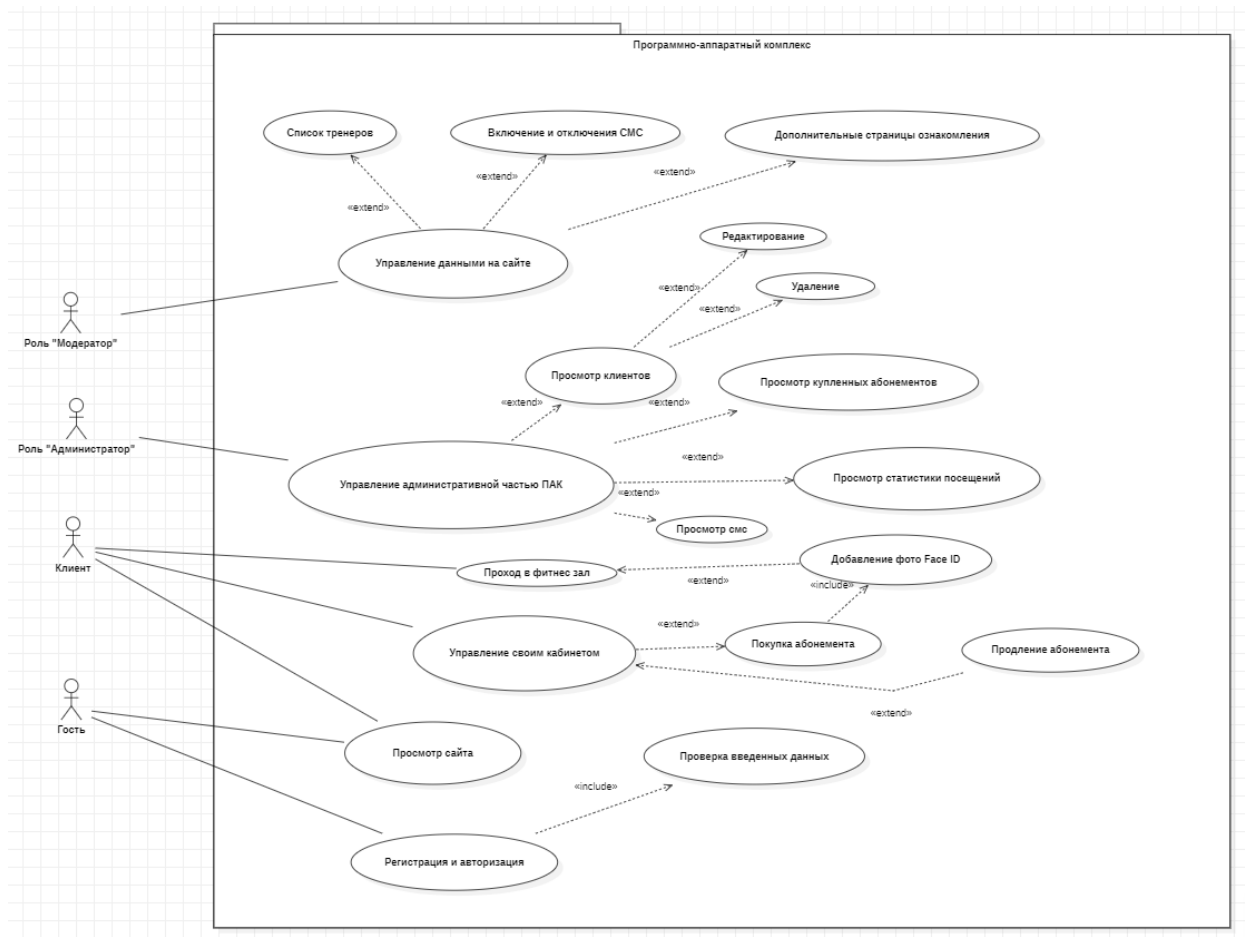


Рисунок 2 – Диаграмма прецедентов, показывающая роли пользователей и их взаимодействие с системой

В таблице 2 представлены спецификации диаграммы прецедентов.

Таблица 2 – Описательные спецификации прецедентов «Управление данными на сайте», «Управление административной частью ПАК», «Проход в фитнес зал», «Управление своим кабинетом», «Просмотр сайта», «Регистрация и авторизация»

Прецедент	Субъект	Предусловие	Основной поток	Постусловие
Управление данными на сайте	Роль «Модератор»	Потребность в изменениях данных на сайте	Редактирование списка тренеров, включение и отключение СМС, изменения дополнительных страниц ознакомления	Изменения успешно сохранены

Продолжение таблицы 2

Прецедент	Субъект	Предусловие	Основной поток	Постусловие
Управление административной частью ПАК	Роль «Администратор»	Потребность в изменениях административных данных на сайте	Просмотр клиентов, просмотр купленных абонементов, просмотр статистики, просмотр СМС	Изменения успешно сохранены
Проход в фитнес зал	Клиент	Потребность в занятиях в зале	Покупка абонемента и добавление фото в Face ID	Попадает в фитнес зал
Управление своим кабинетом	Клиент	Действие с данными в личном кабинете	Покупка абонемента, добавление фото в Face ID, продление абонемента	Требуемые операции успешно выполнены
Просмотр сайта	Клиент, гость	Потребность в изучении фитнес зала	Поиск нужной информации	Нужная информация получена
Регистрация и авторизация	Гость	Потребность в занятиях в фитнес зале	Создание нового аккаунта, авторизация в существующем	Идентификация пользователя в системе

Для иллюстрации функционирования программно-аппаратного комплекса используются диаграммы деятельности.

Диаграмма деятельности – это диаграмма, в которой показано разбиение процесса на его составные части. Она используется для описания исполняемого поведения через последовательное и параллельное выполнение различных элементов.

Диаграмма, описывающая процесс «Регистрация» представлена на рисунке 3.

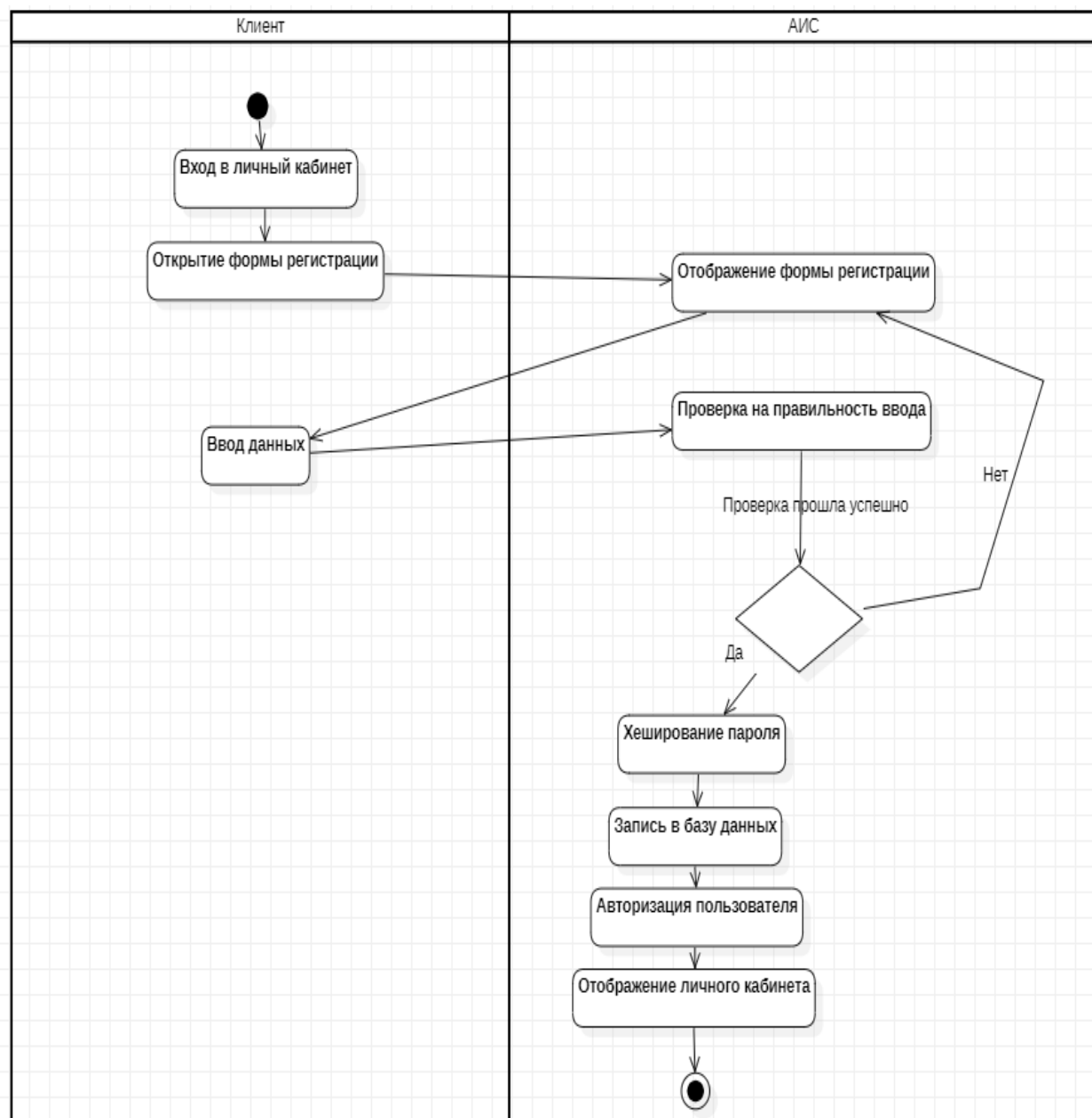


Рисунок 3 – Диаграмма деятельности процесса «Регистрация»

В процессе регистрации происходит проверка правильности введенных данных, включая заполнение всех обязательных полей и проверку уникальности номера телефона в базе данных, она занимает до нескольких секунд.

После успешной проверки пароль усложняется путем его хеширования с использованием соли.

Соль – это случайно сгенерированная строка, которая добавляется к паролю перед его хешированием. Хеширование представляет собой процесс преобразования произвольных данных в строку фиксированной длины,



состоящую из букв и цифр, с помощью математического алгоритма. Даже если два пользователя используют одинаковые пароли, но с разными солями, результат хэширования будет уникальным.

Встроенная функция `make_password()` фреймворка Django автоматически генерирует уникальную соль для каждого пользователя. Пароль пользователя и сгенерированная соль объединяются в одну строку. Объединенная строка (пароль + соль) хешируется с использованием алгоритма хеширования SHA256. Полученный хеш пароля вместе с использованной солью сохраняется в базе данных.

Пример хранения одинаковых паролей от разных пользователей при условии персонально сгенерированной соли представлен в таблице 3.

Таблица 3 – Пример хранения одинаковых паролей от разных пользователей при условии персонально сгенерированной соли

Имя пользователя	Пароль	Значение соли	SHA256 (пароль + значение соли)
user1	qwerty123	j12NcVO5uFxy8ZL xh5b94w==	EF92B778BAFE771E89245B89ECBC08A44A4E16 6C06659911881F383D4473E94F
user2	password456	Yz8gXlr+2emDOqD VvqC+Vw==	82AB28C6ACBC7A30100F5A1774C75CDB4F832 F47A38CACCB049F46DC5A839BC

Таким образом, в случае получения прямого доступа в базу данных злоумышленником, он не сможет получить пароли клиентов.

Диаграмма, описывающая процесс «Покупка абонемента» представлена на рисунке 4.

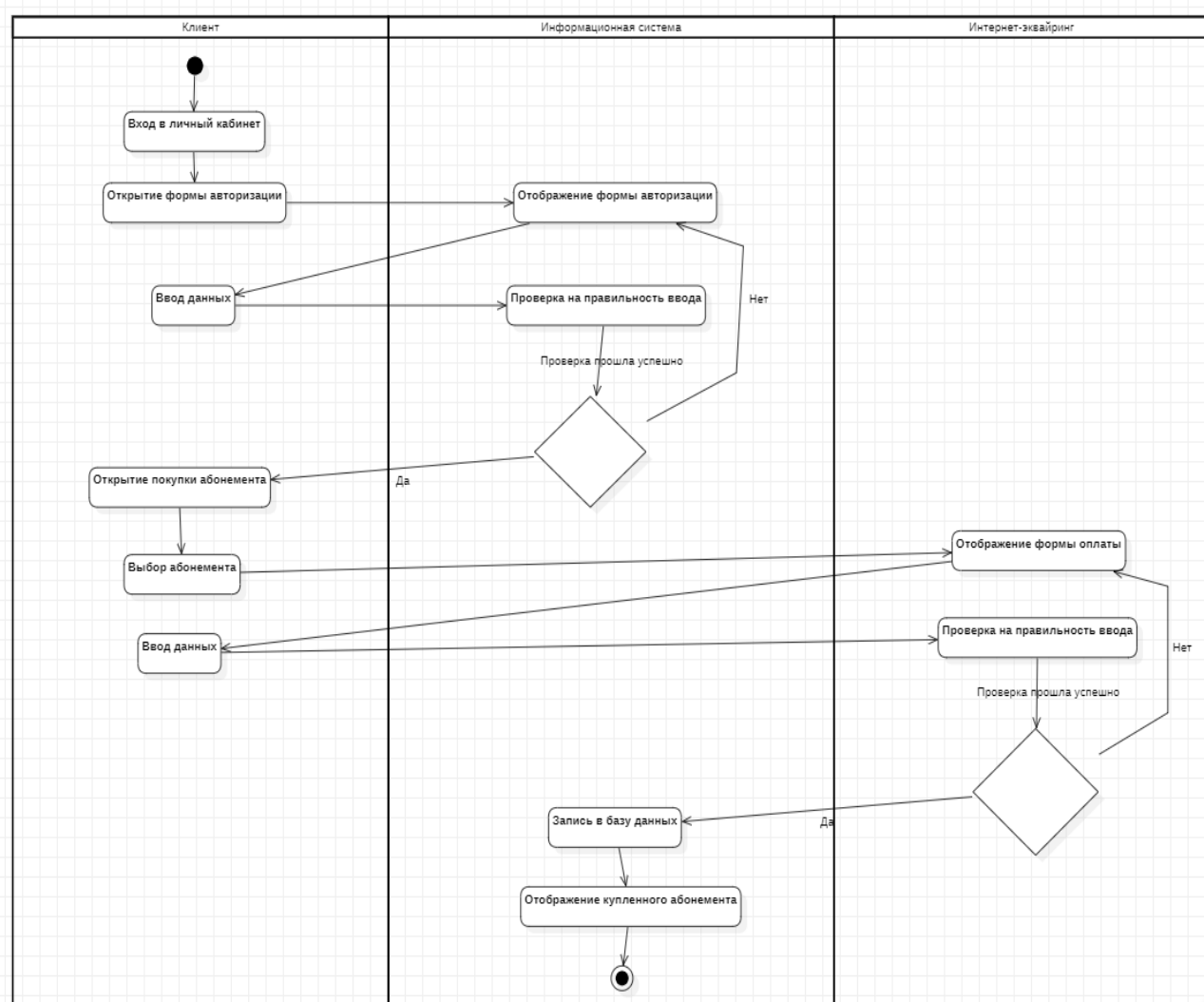


Рисунок 4 – Диаграмма деятельности процесса «Покупка абонемента»

В процессе «Покупка абонемента» есть две проверки. Первая проверка правильности заполненных и введенных данных происходит при авторизации, включая номера телефона и пароля, она занимает до нескольких секунд.

Вторая проверка правильности введенных данных происходит в Интернет-эквайринге. После успешной проверки купленный абонемент заносится в базу данных, она занимает до нескольких секунд.

Диаграмма, описывающая процесс «Загрузка фото Face ID» представлена на рисунке 5.

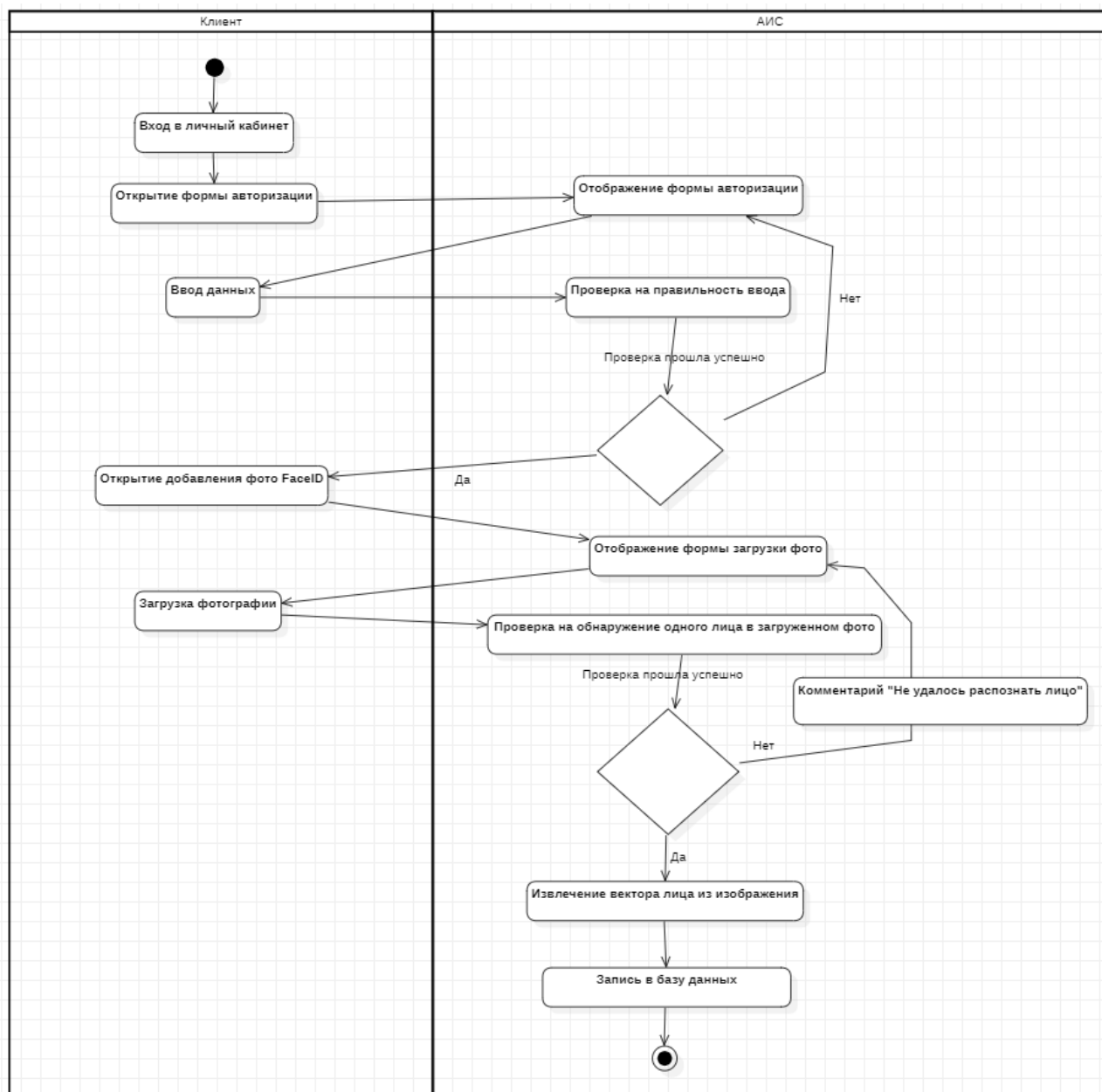


Рисунок 5 – Диаграмма деятельности процесса «Загрузка фото Face ID»

В процессе «Загрузка фото Face ID» есть две проверки. Первая проверка правильности заполненных и введенных данных происходит при авторизации, включая номера телефона и пароля, она занимает до нескольких секунд.

Вторая проверка правильности загружаемой фотографии. После успешной проверки происходит процесс извлечения вектора лица из изображения - массива чисел фиксированной длины, описывающих форму лица через ключевые структуры, и заносится в базу данных. Проверка занимает до нескольких секунд.

При неудачной проверке клиенту пишется модальное окно, что не удалось распознать лицо.

## 2.3 Проектирование базы данных

Для проектирования базы данных логической модели данных используется программа ERwin, представленная на рисунке 6. Она представляет собой описание структуры базы данных, включая таблицы, их атрибуты, ключи и отношения, независимо от конкретной системы управления базами данных.

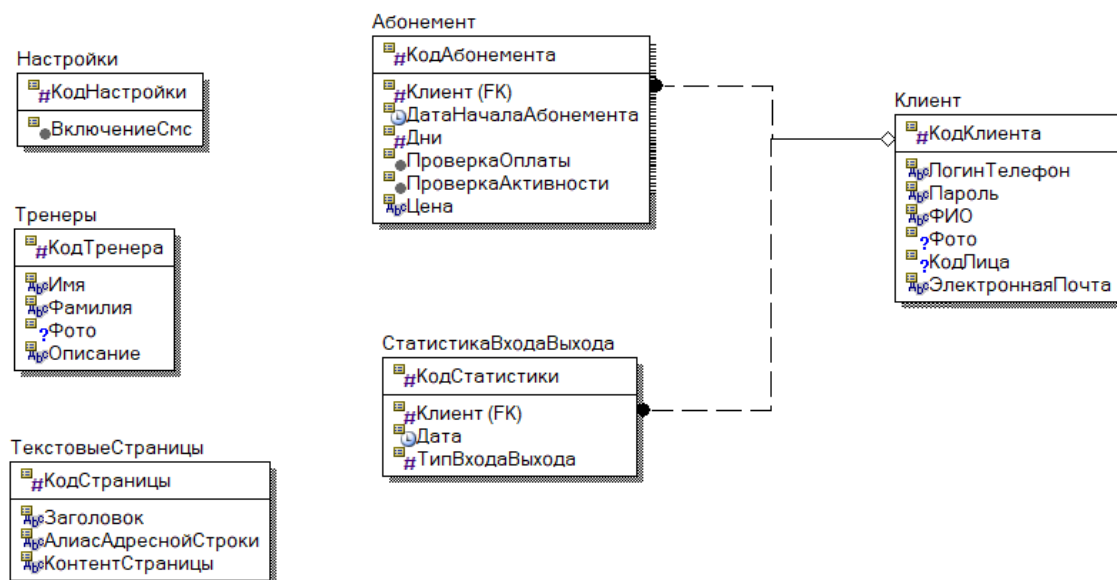


Рисунок 6 – Логическая модель базы данных

Данная база данных состоит из 6 таблиц:

- клиент – содержит информацию для идентификации клиента посредством авторизации, номер телефона, электронную почту, фамилию, имя, отчество, дату рождения, фотографию в личном кабинете, фотографию для Face ID, а также все его купленные абонементы;
- абонемент – содержит информацию о ценах и условиях абонементов;

- статистика входа выхода – содержит информацию о том, какой клиент во сколько зашел и вышел;
- настройки – содержит настройку включения или отключения смс;
- тренеры – содержит информацию о тренерах фитнес-зала, их имя, фамилию, должность и фото, которое показывается на главной странице сайта;
- текстовые страницы – содержит описание заголовка и текста страницы для информативности клиента на сайте.

Связи между таблицами представлены в таблице 4.

Таблица 4 – Связи информационных объектов

Ключ связи	Главный информационный объект	Подчинённый информационный объект	Тип связи
Id_customer	Клиент	Абонемент	Один ко многим
Id_customer	Клиент	Статистика входа выхода	Один ко многим

## 2.4 Требования к программному обеспечению

Для эффективной работы программного комплекса сервер должен соответствовать следующим техническим характеристикам:

- процессор с 4 ядрами;
- тактовая частота процессора не менее 2,5 ГГц;
- объем оперативной памяти 4 Гб;
- доступное место на жестком диске не менее 100 Гб;
- скорость интернет-соединения не менее 100 Мбит/С.

Для комфортного использования программного комплекса пользовательское оборудование должно соответствовать следующим требованиям:

- процессор с тактовой частотой не менее 2,2 ГГц;

- объем оперативной памяти не менее 2 Гб;
- доступное место на жестком диске не менее 16 Гб;
- скорость интернет-соединения не менее 1 Мбит/С;
- камерой не менее 2МР;
- наличие совместимого веб-браузера.

## **2.5 Требования к аппаратному обеспечению модуля считывания и обработки биометрических данных в реальном времени**

Для эффективной работы аппаратный комплекс должен соответствовать следующим техническим характеристикам:

- процессор с 4 ядрами;
- тактовая частота процессора не менее 1,5 ГГц;
- объем оперативной памяти 4 Гб;
- доступное место на карте памяти не менее 32 Гб;
- скорость беспроводного интернет-соединения не менее 10 Мбит/С;
- USB камерой фокусом 3,6 мм и качеством не менее 720p;
- Камерой ночного видения 5MP OV5647 для Raspberry Pi;
- HDMI модуль с экраном для Raspberry PI.

Таким образом, при проектировании были рассмотрены все важные аспекты будущего программно-аппаратного комплекса, которые нужно учитывать при разработке.

С использованием UML были разработаны диаграммы прецедентов и деятельности.

С помощью ER была построена логическая модель базы данных, включающая в себя структуру таблиц и связи между ними. Также были определены необходимые требования к программно-аппаратному комплексу, необходимому для функционирования.

## Глава 3 Реализация программно-аппаратного комплекса

### 3.1 Выбор архитектуры

После проведения исследования и анализа предметной области была построена UML-диаграмма, которая представлена на рисунке 7, в соответствии с рекомендациями и методологиями, описанными в руководствах по проектированию программных систем [13].

Данные между блоками передаются в формате JSON используя протокол HTTPS.

В результате применения этих методологий была разработана структура диаграммы, которая включает все необходимые компоненты и взаимосвязи между ними. Это позволяет лучше понять структуру и логику работы комплекса и обеспечить его эффективное функционирование.

Для построения архитектуры программного комплекса используется архитектура «клиент-сервер».

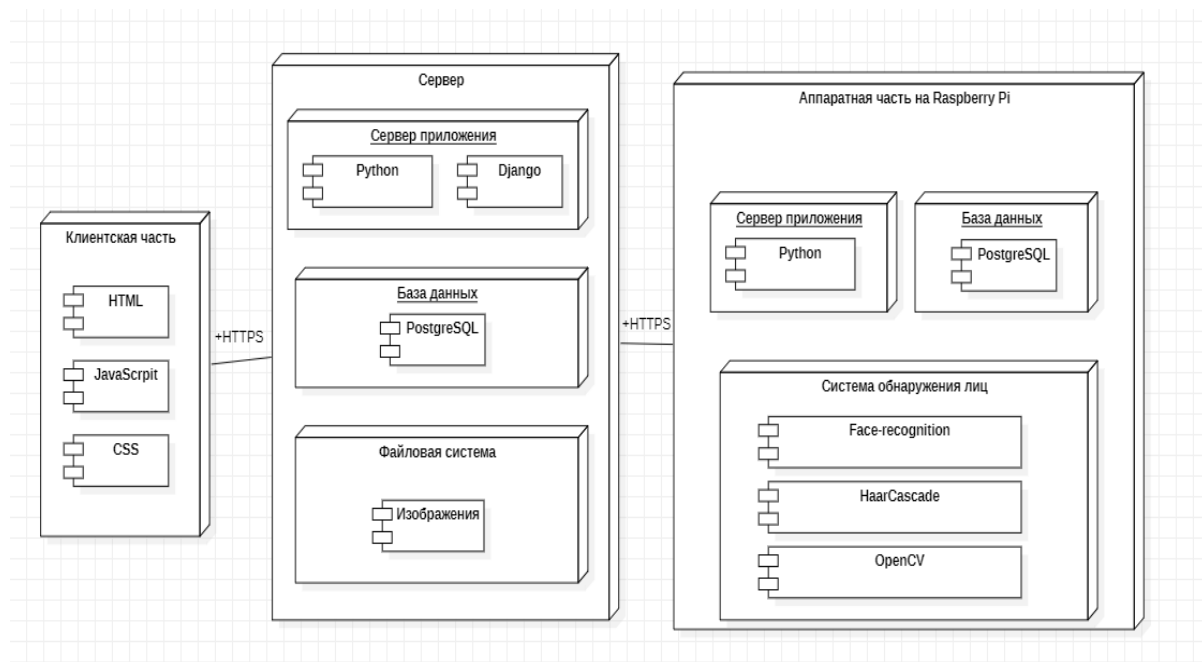


Рисунок 7 – Диаграмма компонентов разрабатываемого программного комплекса

Программное обеспечение серверной части программно-аппаратного комплекса должно соответствовать следующим требованиям:

- операционная система Linux Debian 9 или выше;
- СУБД PostgreSQL версии 12 или выше;
- веб-сервер Nginx версии 1.18.0 или выше;
- язык программирования Python 3 или выше;
- фреймворк Django 3.12 или выше.

Комплекс состоит из трёх основных частей.

Первая часть комплекса представлена клиентской частью, которая включает веб-страницы, разработанные с использованием HTML, JavaScript и CSS. Этот блок отвечает за отображение пользовательского интерфейса и обеспечивает интерактивное взаимодействие с системой. Пользователи могут регистрироваться, выбирать абонементы, загружать фотографии и выполнять другие действия на веб-страницах.

Вторая часть комплекса представлена сервером, который работает на языке программирования Python с использованием фреймворка Django. Этот сервер отвечает за обработку запросов от клиентов, управление данными и логику приложения. Он взаимодействует с базой данных PostgreSQL для хранения и извлечения информации, а также управляет передачей данных между клиентской и аппаратной частями.

Третья часть комплекса представлена аппаратной частью, которая включает физический блок на базе Raspberry Pi с камерой и монитором дисплея. Этот блок функционирует как система обнаружения лиц в реальном времени с использованием алгоритмов распознавания лиц, таких как Face recognition, HaarCascade и OpenCV. Алгоритмы Face recognition и OpenCV широко известны и широко применяются в области компьютерного зрения и распознавания лиц. Алгоритм HaarCascade является одним из классических методов для детектирования объектов на изображениях и также широко используется для распознавания лиц. Аппаратная часть должна соответствовать минимальным



требованиям к оборудованию для обеспечения эффективной работы системы обнаружения лиц в реальном времени.

Сервер взаимодействует с клиентской и аппаратной частями через защищенный протокол передачи данных HTTPS. При этом происходит обмен информацией между клиентскими запросами и сервером, а также между сервером и аппаратной частью, в основном по принципу запрос-ответ. Это обеспечивает безопасную и эффективную передачу данных в рамках данного комплекса.

## **3.2 Выбор средства реализации программно-аппаратного комплекса**

### **3.2.1 Выбор технологий разработки программно-аппаратного комплекса**

HTML (HyperText Markup Language) используется для создания структуры веб-страниц и отображения их содержимого.

CSS (Cascading Style Sheets) используется для стилизации элементов HTML, что позволяет придать веб-страницам желаемый внешний вид.

JavaScript используется для создания интерактивных элементов на веб-страницах, обеспечивая динамическое взаимодействие с пользователем.

Python используется для разработки серверной части приложения [7], [19], [20].

Фреймворк Django обеспечивает эффективное управление данными и общую логику работы системы. Django обладает богатым функционалом, встроенным механизмом безопасности, возможностями масштабирования и хорошо подходит для создания веб-приложений [2], [10]. Его широкое сообщество разработчиков поддерживает активную экосистему плагинов и расширений, а встроенная административная панель делает управление данными более удобным. Все это делает Django превосходным выбором для создания веб-приложений, особенно в условиях, когда требуется быстрое развертывание безопасных и масштабируемых приложений.

### **3.2.2 Выбор метода для распознавания лиц**

Основная методология и основные принципы распознавания лиц используются с помощью гибкого сравнения на графах.

Он работает на основе сравнения графов, которые представляют лица в виде сети точек с весами и связями между ними. Во время распознавания один из этих графов остается неизменным (эталонным), а другой изменяется таким образом, чтобы наилучшим образом соответствовать эталонному.

Графы могут быть либо прямоугольными решетками, либо структурами, созданными из ключевых точек на лице. В каждой точке графа вычисляются значения определенных признаков, часто используются комплексные значения фильтров Габора или их комбинации. Эти фильтры помогают выделить характерные особенности лица.

Весы на ребрах графа определяются расстояниями между соседними точками. Для измерения различий между графами применяется функция деформации, которая учитывает, насколько один граф нужно изменить, чтобы совпадать с другим.

### **3.2.3 Выбор подходящего алгоритма для задачи идентификации лиц**

OpenCV – это библиотека с открытым исходным кодом, которая используется для обработки изображений и видео. Она используется в аппаратной части для обнаружения лиц на изображениях, а также для реализации функций распознавания лиц [21].

HaarCascade – это метод обнаружения объектов на изображениях, основанный на использовании признаков Хаара. Этот метод позволяет обнаруживать объекты, имеющие определенные характеристики на изображениях. HaarCascade интегрируется в аппаратную часть комплекса и используется вместе с библиотекой OpenCV для обнаружения лиц в реальном времени [21].

Библиотека `face_recognition` является популярным инструментом для распознавания лиц в Python, основанным на библиотеках `dlib` и `OpenCV`. [18]. Она

предоставляет простой и понятный API для работы с распознаванием лиц, что делает ее доступной даже для начинающих разработчиков, обеспечивает высокую точность распознавания лиц благодаря использованию передовых алгоритмов нейронных сетей, а также обладает хорошей производительностью даже на стандартных компьютерах благодаря оптимизированным алгоритмам и возможности использования GPU.

### 3.2.4 Сравнение способов и алгоритмов распознавания изображений

Сравнение Face Recognition с Microsoft Face API и Amazon Rekognition представлено в таблице 5.

Таблица 5 – Сравнение Face Recognition с аналогами.

	Face Recognition	Microsoft Face API	Amazon Rekognition
Точность	Высокая	Высокая	Высокая
Скорость	Высокая	Высокая	Высокая
Поддержка платформ	Веб, мобильные устройства	Веб, мобильные устройства, устройства Windows	Веб, мобильные устройства
Распознавание лиц в реальном времени	Поддерживается	Поддерживается	Поддерживается
Стоимость	Бесплатно	Бесплатно с ограничениями запросов	Платная услуга

Face Recognition выделяется благодаря своей точности, скорости и простоте использования, а главным её плюсом является бесплатное использование.

### 3.2.5 Тестовый пример алгоритма идентификации

Фрагмент кода для обнаружения тестового лица с помощью метода обнаружения HaarCascade и библиотеки face-recognition имеет вид, представленный на рисунке 8.

```

import cv2
import face_recognition

# Загрузка предварительно обученной модели для распознавания лиц с помощью Haar Cascade
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

# Загрузка известных лиц и их имен
known_faces_encodings = []
known_faces_names = []

# Загрузка изображения для распознавания
image = cv2.imread('1.webp')

# Преобразование изображения в оттенки серого для Haar Cascade
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Распознавание лиц с помощью Haar Cascade
faces_haar = face_cascade.detectMultiScale(gray_image, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Распознавание лиц на изображении с помощью face_recognition
face_locations = face_recognition.face_locations(image)
face_encodings = face_recognition.face_encodings(image, face_locations)

# Определение имен лиц на изображении
for face_encoding in face_encodings:
    matches = face_recognition.compare_faces(known_faces_encodings, face_encoding)
    name = "Butrim Ivan"

    # Если нашлись совпадения, то определяем имя
    if True in matches:
        first_match_index = matches.index(True)
        name = known_faces_names[first_match_index]

# Отрисовка прямоугольников вокруг обнаруженных лиц с помощью Haar Cascade и добавление имен
for (x, y, w, h) in faces_haar:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.putText(image, name, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

# Отрисовка прямоугольников вокруг обнаруженных лиц с помощью face_recognition
for top, right, bottom, left in face_locations:
    cv2.rectangle(image, (left, top), (right, bottom), (255, 0, 0), 2)

# Вывод результатов
cv2.imshow('Face Recognition', image)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Рисунок 8 – Код, реализующий обнаружение тестового лица с помощью метода обнаружения HaarCascade и библиотеки face-recognition.

Результат отображаемого тестового изображения после обнаружения HaarCascade и библиотеки face-recognition представлен на рисунке 9.

В результате эксперимента на различных наборах данных, охватывающих разнообразные условия освещения, углы обзора, эмоциональные выражения лиц и прочие факторы была достигнута высокая точность распознавания лиц, что подтверждается низким уровнем ложных срабатываний и высоким уровнем обнаружения реальных лиц [17].

Помимо точности, система обладает высокой скоростью работы, что особенно важно для реального времени.

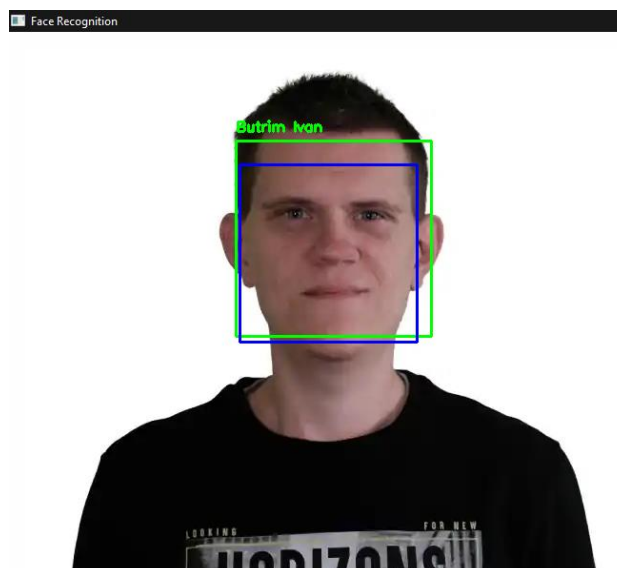


Рисунок 9 – Отображаемое тестовое изображение после обнаружения HaarCascade и библиотеки face-recognition

Таким образом, сочетание быстрого обнаружения лиц с использованием Haar Cascade и точного определения имен с помощью библиотеки face\_recognition обеспечивает оптимальное сочетание скорости и точности в системе распознавания лиц. В результате получается преимущество эффективного обнаружения лиц в реальном времени при высокой надежности определения их личности.

### 3.2.6 Выбор системы управления базами данных

Для управления данными в программно-аппаратном комплексе, разрабатываемом для автоматизации работы администратора фитнес-зала, выбрана PostgreSQL [6], [15].

PostgreSQL — это мощная объектно-реляционная система управления базами данных с открытым исходным кодом. Она предоставляет широкие возможности для хранения, организации и обработки данных.

Выбор PostgreSQL обусловлен следующими преимуществами:

- обладает надежной системой безопасности, обеспечивающей защиту данных от несанкционированного доступа и потери;

- обеспечивает высокую скорость работы и эффективное выполнение запросов;
- обладает возможностью работать с большими объемами данных и поддерживать высокую производительность даже при увеличении нагрузки;
- с открытым исходным кодом, что обеспечивает гибкость в настройке и расширении функциональности по мере необходимости;
- поддерживает различные типы данных, что позволяет эффективно хранить и обрабатывать разнообразные данные фитнес-зала, включая информацию о клиентах, абонементы, расписаниях и т.д.

Эти преимущества делают PostgreSQL оптимальным выбором для создания надежной и эффективной системы управления данными в фитнес-зале, обеспечивая высокий уровень безопасности, производительности и гибкости.

### **3.3 Разработка физической модели данных**

Физическая модель базы данных представляет собой конкретную реализацию логической модели, адаптированную для определенной системы управления базами данных (СУБД) [5], [14].

Она описывает структуру базы данных на уровне таблиц, столбцов, типов данных и ограничений, необходимых для создания и эффективного функционирования базы данных.

В базе данных задействованы таблицы: «Customer», «Abonement», «TableEntering», «Settings», «Couch» и «TextPage».

Данная модель, ориентированная на СУБД PostgreSQL представлена на рисунке 10.

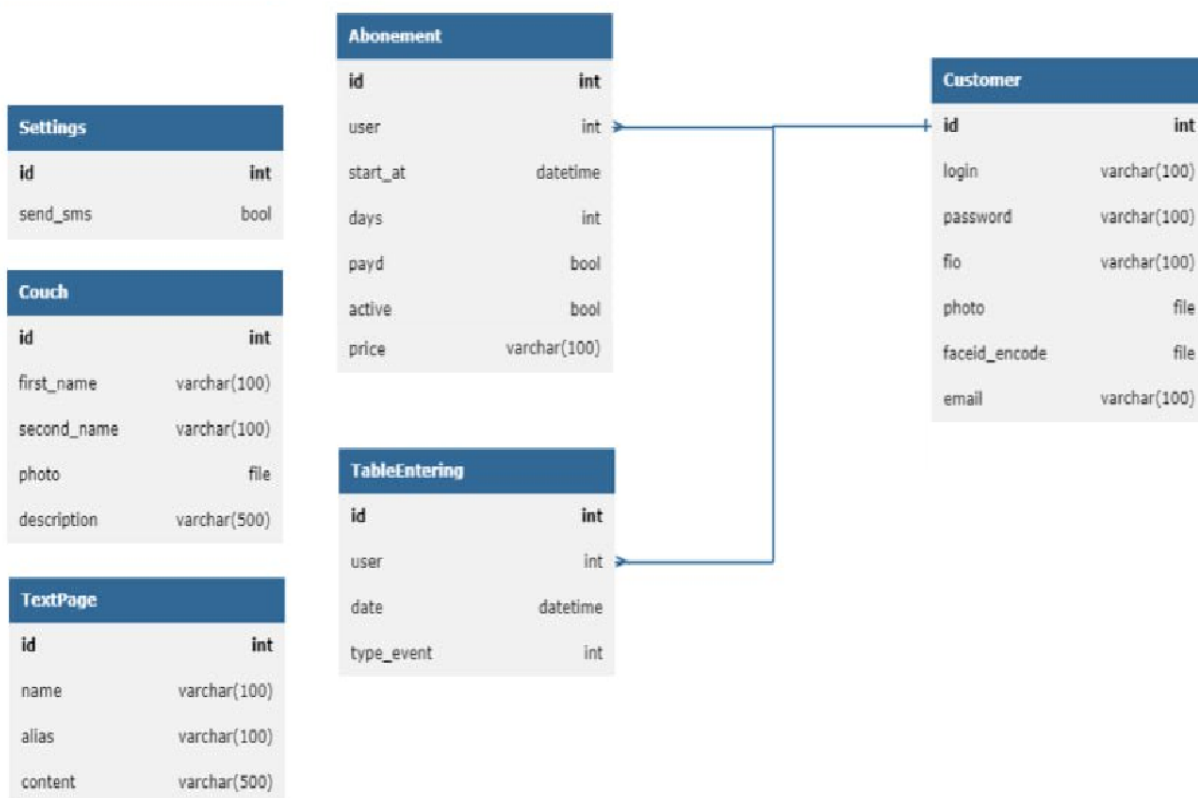


Рисунок 10 – Физическая модель базы данных

Физическая модель базы данных обеспечивает четкую структуру и организацию данных, что позволяет эффективно управлять информацией и выполнять запросы.

### 3.4 Аппаратное обеспечение комплекса

Основой аппаратной части является мини-компьютер Raspberry Pi 4 ModelB 4Gb, который представляет собой небольшое устройство с высокой вычислительной мощностью [22].

Raspberry Pi используется для управления аппаратной частью комплекса и выполнения различных вычислительных задач. На Raspberry Pi устанавливается программное обеспечение, которое обеспечивает функционирование системы обнаружения лиц в реальном времени.

Для обеспечения беспроводного подключения и доступа к сети интернет используется Wi-Fi модуль. Он позволяет Raspberry Pi подключаться к интернету для передачи данных и взаимодействия с сервером.

Для обнаружения лиц в реальном времени используются алгоритмы компьютерного зрения и нейронные сети. В аппаратной части комплекса реализуется система, которая анализирует видеопоток с камеры, обнаруживает лица и передает информацию на сервер для идентификации.

В состав аппаратной части входит камера, которая используется для захвата изображений в реальном времени. Камера подключается к Raspberry Pi и предоставляет потоковое видео, которое обрабатывается системой обнаружения лиц, с обеих сторон (на вход и на выход) аппаратного комплекса.

Для визуализации процесса обнаружения лиц и взаимодействия с системой устанавливается дисплей. Дисплей позволяет показать пользователю изображение, передающее из камеры.

Библиотека RPi.GPIO используется для управления замком двери, который открывается в случае успешного распознавания лица и выдачи соответствующего сигнала.

Для обеспечения безопасного удаленного доступа к системе используется сервис PiTunnel, который обеспечивает безопасное удаленное управление и доступ к Raspberry Pi.

На рисунке 11 представлен собранный аппаратный комплекс на рабочем месте.





Рисунок 11 – Собранный аппаратный комплекс на рабочем месте

Такая конфигурация аппаратной части обеспечивает эффективную работу системы

### **3.5 Описание интерфейса пользователя**

В итоге проделанной работы разработан информативный одностраничный сайт с личными кабинетами клиентов [12].

Интерфейс пользователя разработан с учетом пожелания заказчика, удобства и интуитивной понятности для обеспечения комфортного взаимодействия клиентов с системой.

Основное меню навигации по одностраничному сайту – это шапка, по которой можно перемещаться по основным блокам.

При нажатии на логотип «VK» переходит в группу ВКонтакте фитнес-зала.

При нажатии на номер телефона можно позвонить по номеру телефону фитнес зала.

При нажатии на кнопку «Войти в аккаунт» можно зайти в личный кабинет клиента, который представлен на рисунке 18.

При наведении на объекты в шапке они подсвечиваются желтым.

Шапка и стартовый экран при заходе на сайт представлен на рисунке 12.

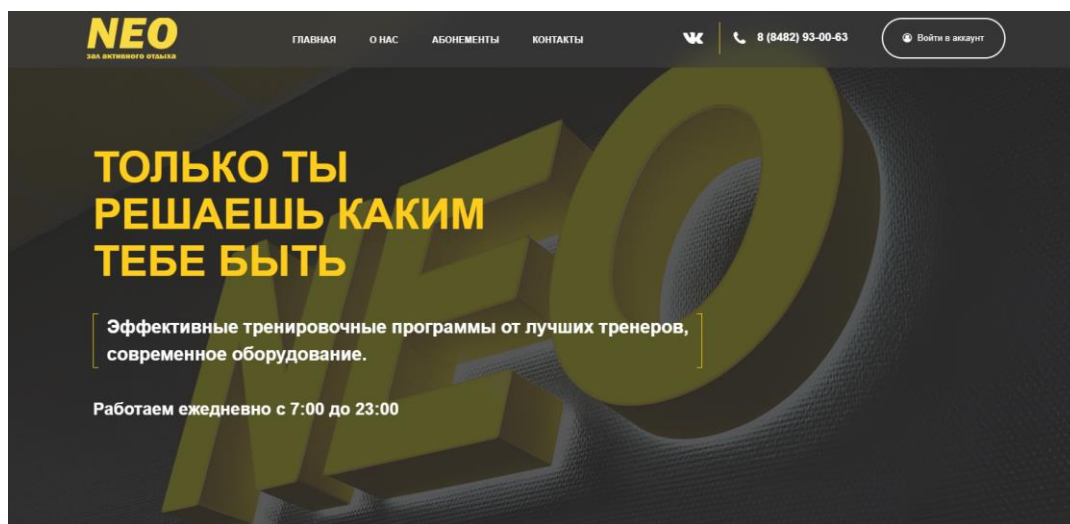


Рисунок 12 – Экранная форма шапки и стартового экрана сайта

На рисунке 13 представлен блок с фотографиями помещений фитнес-зала. При нажатии на стрелочки можно переключать слайдеры с фотографиями.

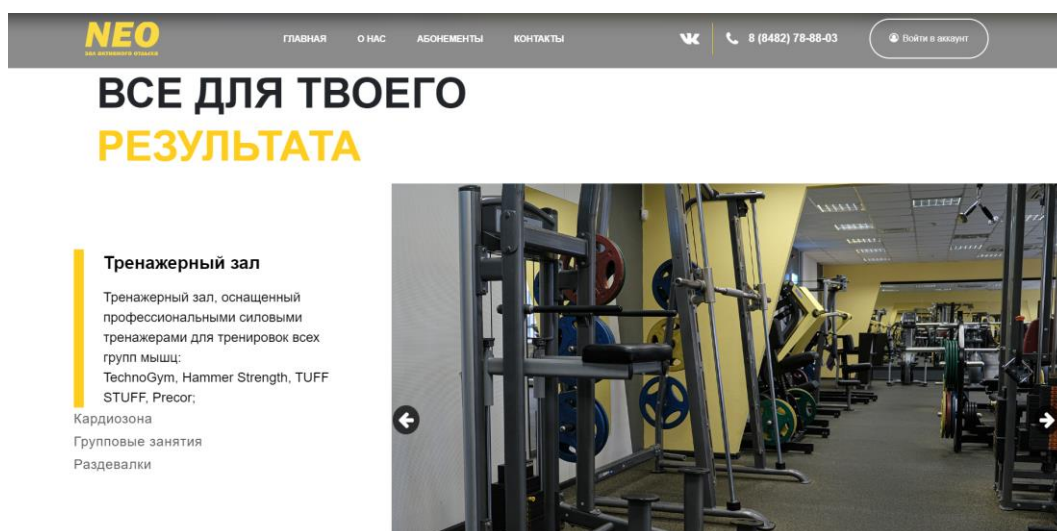


Рисунок 13 – Экранная форма блока с фотографиями помещений фитнес-зала

На рисунке 14 представлен список тренеров. При отсутствии фотографии отображается заглушка с рисунком человечка.

## НАШИ ТРЕНЕРЫ

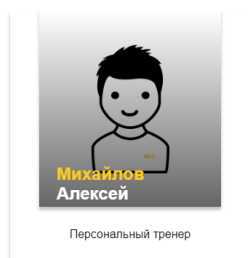


Рисунок 14 – Экранная форма списка тренеров

При нажатии на тренера открывается модальное окно с номером телефона, его фотографией и описанием тренера.

Модальное окно тренера представлено на рисунке 15.

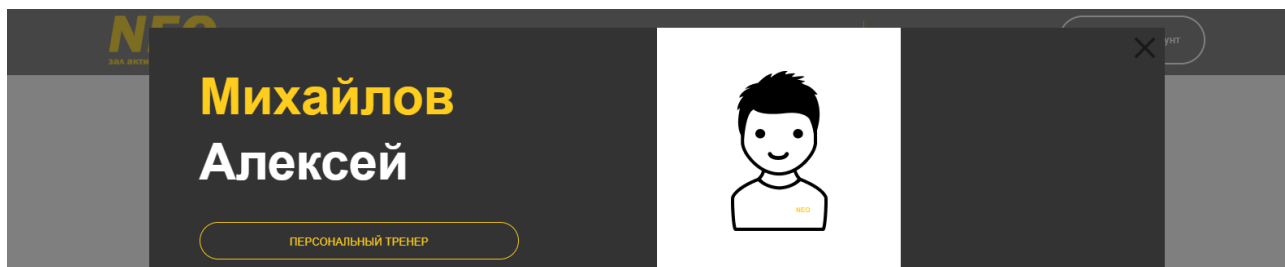


Рисунок 15 – Экранная форма модального окна тренера

Действующие цены на абонемент фитнес-зала представлены на рисунке 16. При наведении на абонемент блок цены сменяется на желтый, а кнопка «Выбрать» сменяется на черный. При нажатии на кнопку «Выбрать» открывается страница с авторизацией клиента, представленная на рисунке 18.

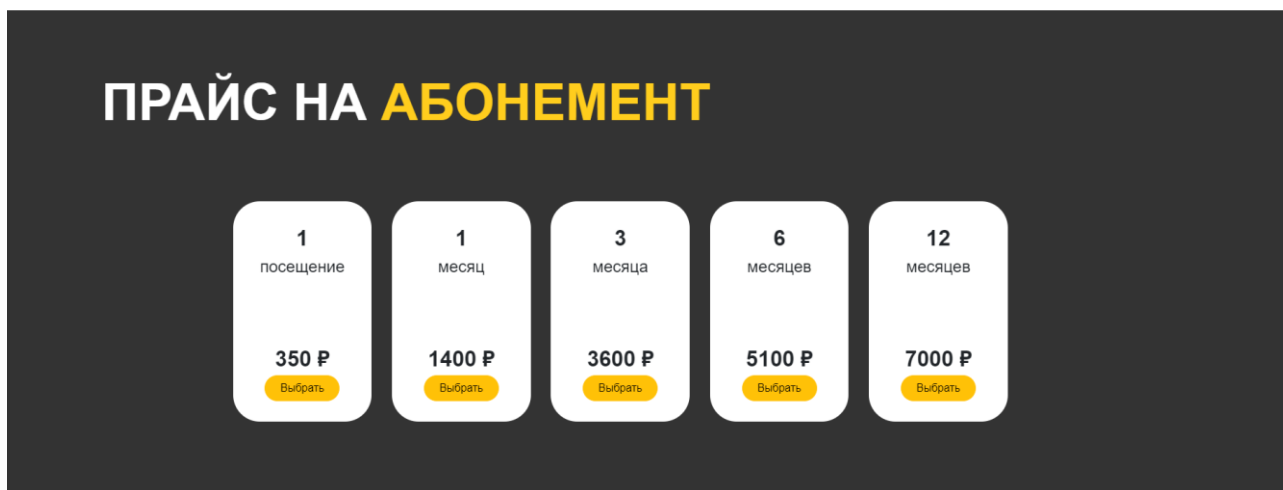


Рисунок 16 – Экранная форма с ценами на абонементы

На рисунке 17 представлен подвал сайта, в котором находятся контактные данные фитнес-зала. В левой части подвала указана геолокация через сервис «Яндекс Карты». В правой части указан режим работы, группа ВКонтакте и номер телефона.

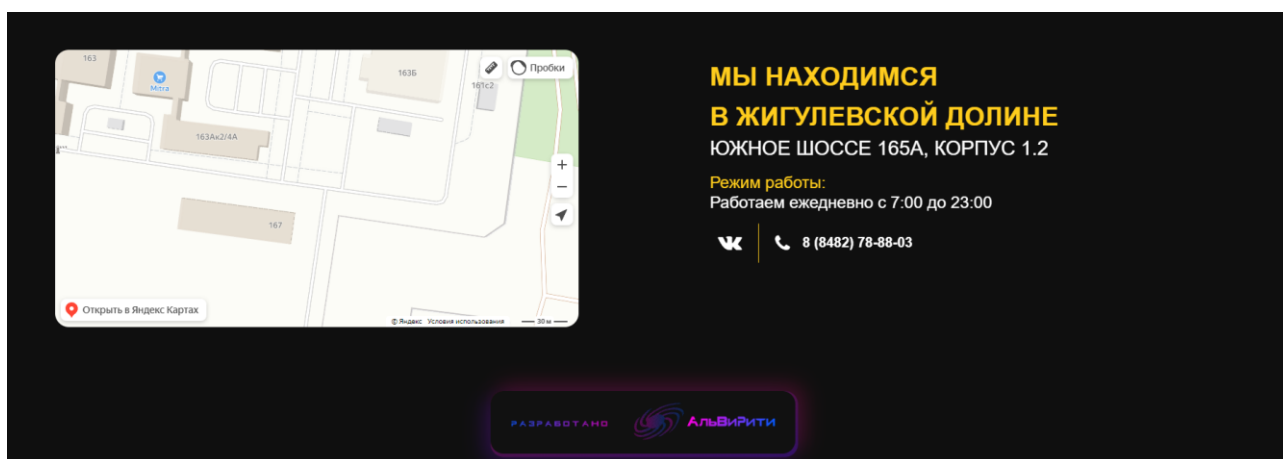


Рисунок 17 – Экранная форма подвала сайта

Авторизация в личном кабинете представлена на рисунке 18. Вход в аккаунт происходит по форме номера телефона и пароля. При наведении на поле с номером телефона появляется подсказка в виде маски в каком виде необходимо ввести данные. При нажатии на поле номера телефона или ввода пароля, оно подсвечивается оранжевым. При наведении на «Зарегистрироваться» надпись

подчеркивается и переходит в форму регистрации, которая представлена на рисунке 19. При наведении на «Забыли пароль?» надпись подчеркивается и переходит в форму восстановления пароля, которая представлена на рисунке 32.

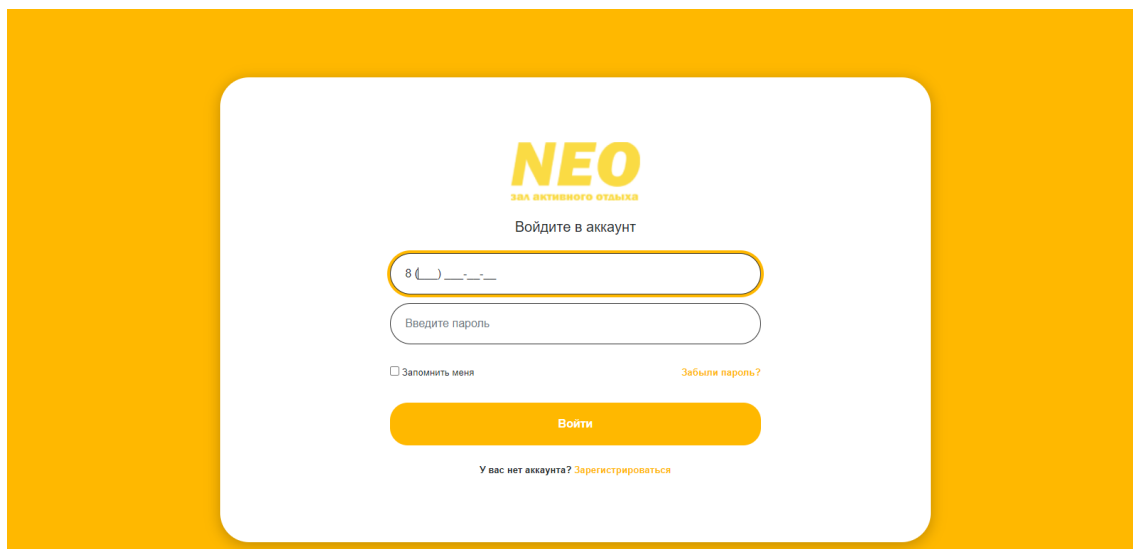


Рисунок 18 – Экранная форма авторизации в личном кабинете

Форма регистрации представлена на рисунке 19.

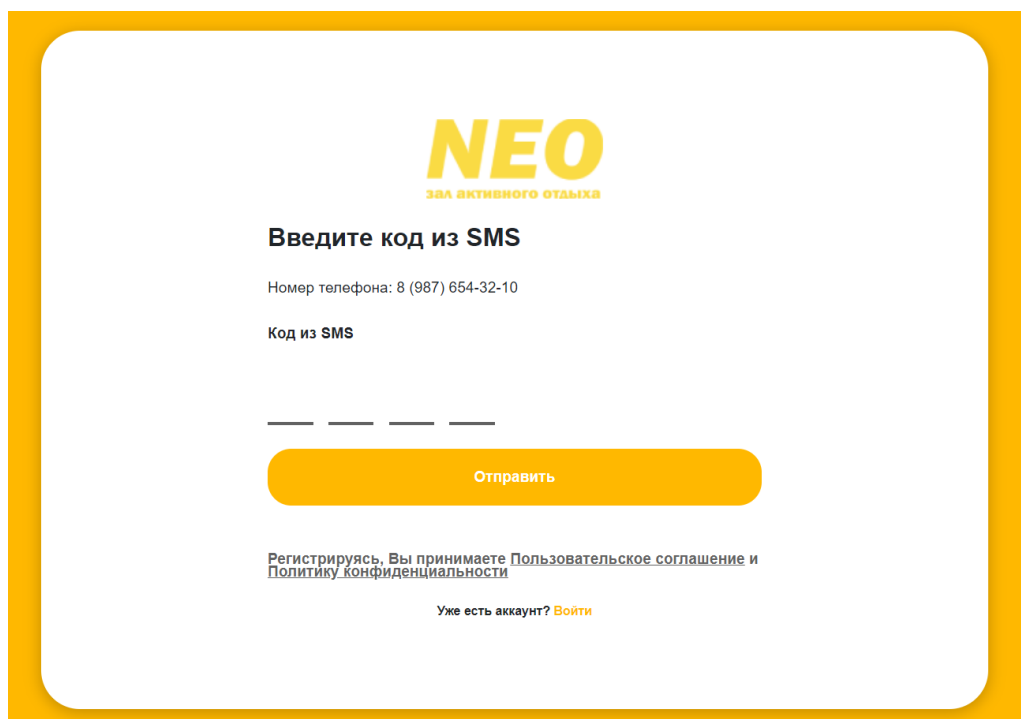


Рисунок 19 – Экранная форма регистрации

После ввода номера телефона через сервис «SMSЦЕНТР» клиенту приходит СМС-рассылка на указанный номер, пример которой указан на рисунке 20. Каждый код генерируется случайным образом.

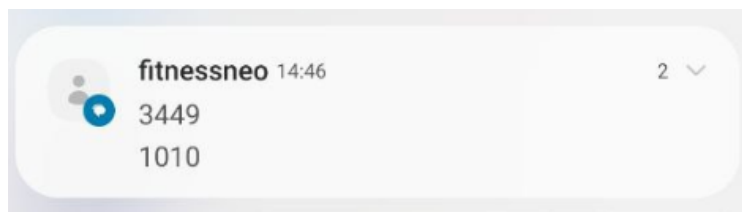


Рисунок 20 – Экранная форма получения СМС клиенту

После правильного введенного номера телефона, клиент попадает на заполнение формы регистрации, которая представлена на рисунке 21. При регистрации необходимо заполнить поля с именем и фамилией, выбрать свой пол, заполнить поле с паролем и повторить его. При нажатии на поле с паролем клиенту пишется подсказка «Не менее 8 символов. Не используйте личные данные, последовательности (123456, qwerty) и популярные пароли (password)».

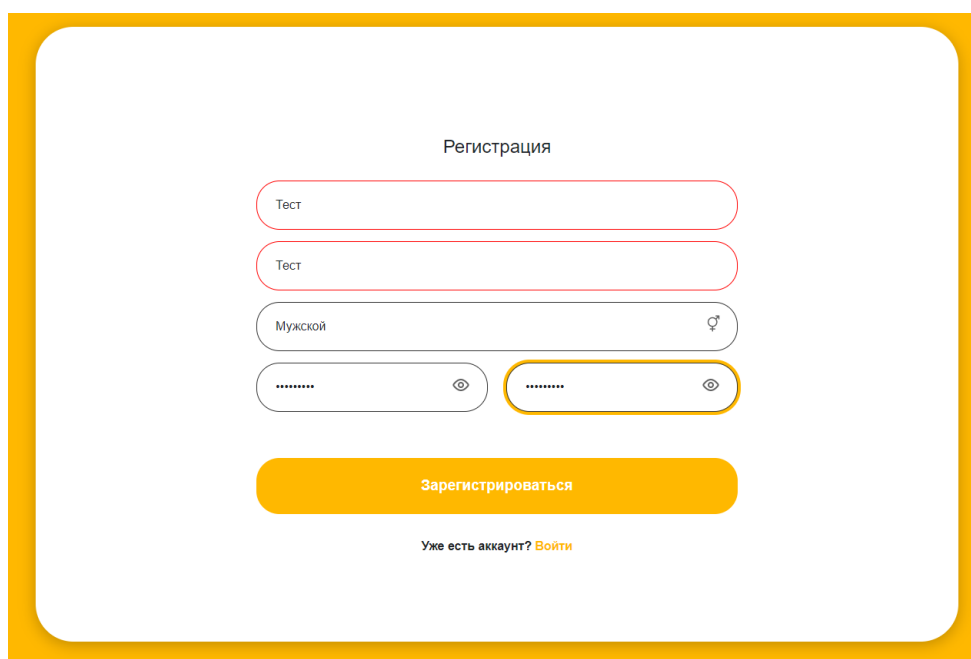


Рисунок 21 – Экранная форма заполнения полей регистрации

После заполнения полей регистрации клиент попадает в свой личный кабинет, где может купить абонемент, посмотреть историю своих посещений, добавить Face ID. Меню личного кабинета расположено в левой части экрана. Изначально на фотографии клиента отображается заглушка с рисунком человечка, при необходимости клиент может поменять фотографию, нажав на нее, и выбрать свою. Страница с выбором абонемента представлена на рисунке 22.

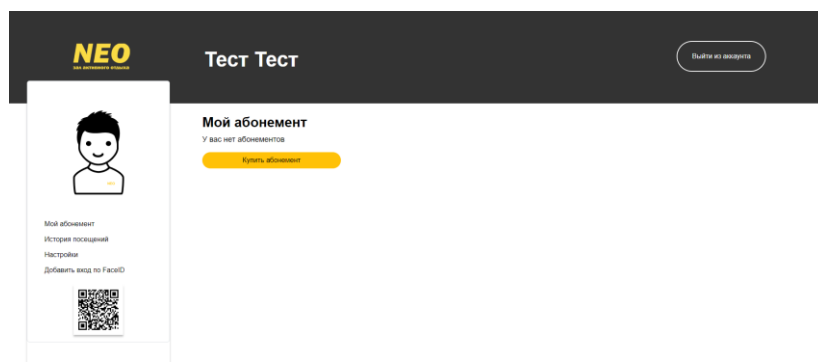


Рисунок 22 – Экранная форма страницы с выбором клиента

При нажатии на кнопку «Купить абонемент» клиент попадает на страницу с выбором абонемента, которая представлена на рисунке 23. После выбора абонемента он выбирает 1-ый день тренировки. С этой даты будет активен абонемент и отсчитывается период выбранного абонемента.

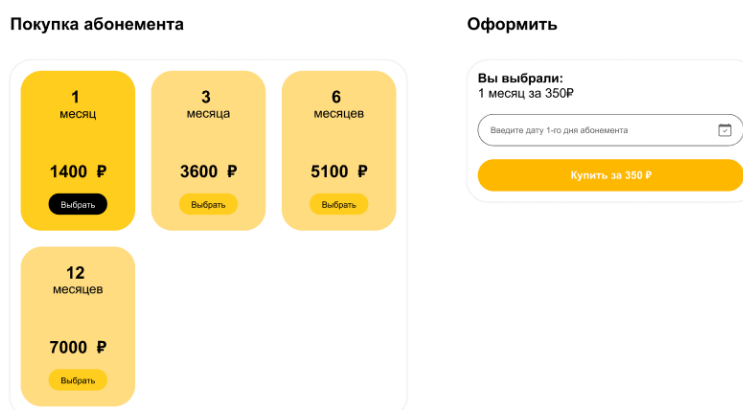


Рисунок 23 – Экранная форма покупки абонемента

Фрагмент кода, который обрабатывает запрос на оплату, получает данные о цене и сроке абонемента, определяет даты начала и окончания абонемента, создает запись об абонементе в базе данных, вычисляет сумму платежа, инициирует платеж через банковский сервис, и возвращает URL для перенаправления пользователя на страницу оплаты, при возникновении ошибок выдает исключение, имеет вид, представленный на рисунке 24.

```
def pay(request):
    try:
        price = request.POST['price_subs']
        month = request.POST['month']
        od = request.POST['od']
        days = ''
        if 'result_sub_day_subs' in request.POST:
            days = request.POST['result_sub_day_subs']
        if 'first_day' in request.POST:
            start_at = request.POST['first_day']
        else:
            start_at = str(datetime.date.today())
        start_at_date = datetime.datetime.strptime(start_at, '%Y-%m-%d')
        if od:
            start_end = start_at_date + datetime.timedelta(days=1)
        else:
            start_end = add_months(start_at_date, int(month))
        client_id = request.user.pk
        if not month == 'None':
            abon = ClientAbon(item=Client.objects.get(userauth__id=client_id), month=month,
                              price=price, days=days, zero=request.POST['zero'], start_at=start_at,
                              start_end=start_end)
        else:
            abon = ClientAbon(item=Client.objects.get(userauth__id=client_id), month='Одноразовое посещение',
                              price=price, days=days, zero=request.POST['zero'], start_at=start_at,
                              start_end=start_end)

        abon.save()
        # Сумма в рублях
        amount = abon.price
        # Уникальный ID пользователя, используется для привязки карт
        # Если None, пользователь не сможет выбрать ранее привязанную карту
        # или привязать карту в процессе оплаты
        svc = BankService('p6320049775')
        # url - адрес, на который следует перенаправить пользователя для оплаты
        # payment - объект Payment из БД, содержит информацию о платеже
        # description - назначение платежа в веб-форме банка
        # params - произвольные параметры, которые можно привязать к платежу
        payment, url = svc.pay(amount, params={'aboniment': abon.id}, client_id=client_id,
                               description="Оплата заказа №{}".format(abon.id), request=request)
        return HttpResponseRedirect(url)
    except Exception as exc:
        # Что-то пошло не так
        raise
```

Рисунок 24 – Экранная форма покупки абонемента



После нажатия на кнопку «Купить», клиент попадает на страницу оплаты, реализованной через Интернет-эквайринг Сбербанка, которая представлена на рисунке 25.

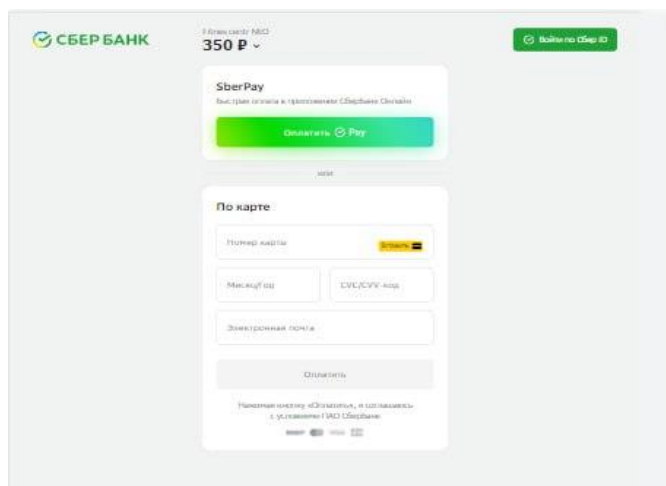


Рисунок 25 – Экранная форма оплаты абонемента

Фрагмент кода, который проверяет статус платежа в системе банка по его уникальному идентификатору, имеет вид, представленный на рисунке 26. Он сначала ищет запись о платеже в базе данных, затем отправляет запрос к банковскому сервису для получения статуса заказа. В зависимости от полученного статуса, он обновляет статус платежа в системе и возвращает его.

```
def check_status(self, payment_uid):
    try:
        payment = Payment.objects.get(pk=payment_uid)
    except Payment.DoesNotExist:
        raise PaymentNotFoundException()
    data = {'orderId': str(payment.bank_id)}
    response = self.execute_request(data, "rest/getOrderStatusExtended", payment)
    if response.get('orderStatus') == 2:
        payment.status = Status.SUCCEEDED
        payment.details.update({"pan": response['cardAuthInfo']['pan']})
    elif response.get('orderStatus') in [3, 6]:
        payment.status = Status.FAILED
    elif response.get('orderStatus') == 4:
        payment.status = Status.REFUNDED
    payment.save(update_fields=['status', 'details'])
    return payment
```

Рисунок 26 – Код, проверяющий статус платежа в системе банка по его уникальному идентификатору

После оплаты клиент возвращается в свой личный кабинет, который представлен на рисунке 27, и видит до какого дня действует абонемент. При необходимости может продлить абонемент.

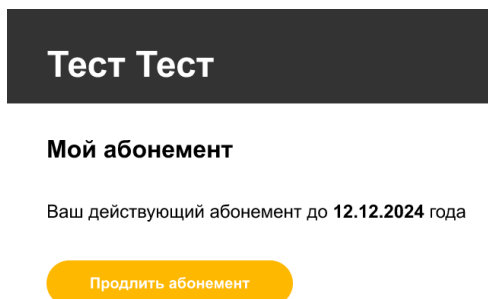


Рисунок 27 – Экранная форма купленного абонемента

Фрагмент кода, который сначала приводит номер телефона к стандартному формату и находит клиента по этому номеру в базе данных, имеет вид, представленный на рисунке 28. Затем он обрабатывает изображение, полученное в кодировке base64: сохраняет его, загружает изображение лица клиента из базы данных и вычисляет его уникальное представление.

После этого происходит сравнение энкодинга лица клиента с энкодингом загруженного изображения. Если обнаружено совпадение, пользователь аутентифицируется и выполняется вход в систему, иначе возвращается сообщение об ошибке.

```

phone = request.POST['phone']
if phone[:1] == '8':
    phone = '7{}'.format(phone[1:])
if phone[:2] == '+7':
    phone = '7{}'.format(phone[2:])
client = Client.objects.get(userauth__username=phone)
image_data = request.POST['img']
format, imgstr = image_data.split(';base64,')
ext = format.split('/')[1]
data = ContentFile(base64.b64decode(imgstr))
file_name = "{}1".format(client.phone) + ext
fs = FileSystemStorage()
try:
    picture_of_me = face_recognition.load_image_file("{}".format(client.photo_faceid.path))
    my_face_encoding = face_recognition.face_encodings(face_image=picture_of_me, num_jitters=5, model='large')[0]
    fs.save(file_name, data)
    unknown_picture = face_recognition.load_image_file("media/{}".format(file_name))
    unknown_face_encoding = face_recognition.face_encodings(face_image=unknown_picture, num_jitters=5, model='large')[0]
    results = face_recognition.compare_faces([my_face_encoding], unknown_face_encoding)
    fs.delete(file_name)
    if results[0] == True:
        user = authenticate(request, username=client.userauth.username)
        login(request, user)
        return JsonResponse({'result': 'ok'})
    else:
        return JsonResponse({'result': 'error'})
except:
    fs.delete(file_name)
    return JsonResponse({'result': 'error'})

```

Рисунок 28 – Код аутентификации клиента по номеру телефона и изображению лица

При нажатии в меню на кнопку «Добавить вход по FaceID» клиент переходит на страницу добавления своей фотографии. Фотография делается в режиме реального времени через сайт. Страница с выбором разрешения на использование камеры представлена на рисунке 29.

При добавлении лица в базу данных, происходит процесс извлечения вектора лица из изображения - массива чисел фиксированной длины, описывающих форму лица через ключевые структуры (глаза, брови, нос, губы и т. д.).

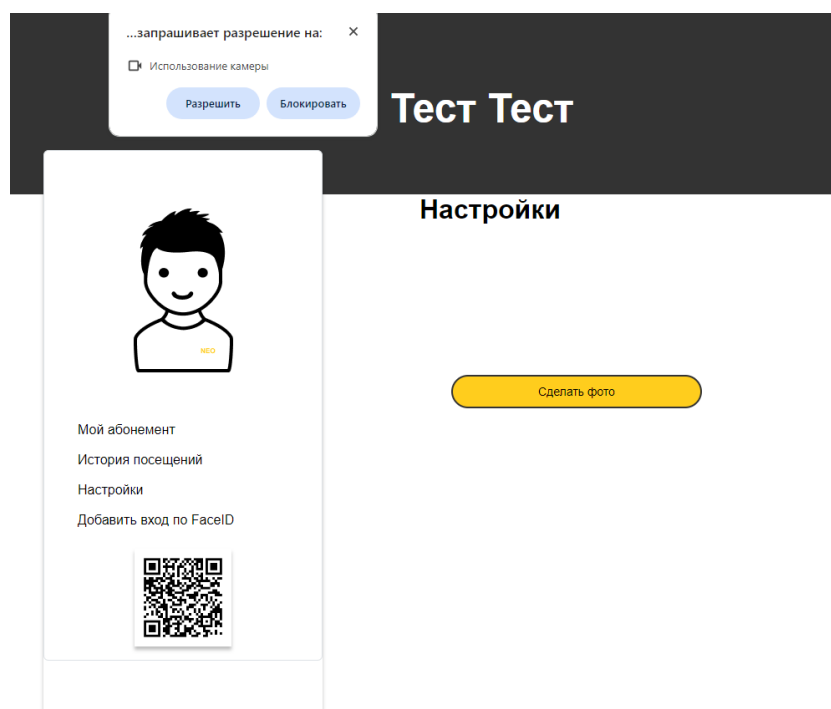


Рисунок 29 – Экранная форма добавления фотографии по Face ID

При нажатии в меню на кнопку «История посещений» клиент переходит на страницу с историей своих посещений. Посещения записываются в историю даты и времени прохождения через систему Face ID. История посещений представлена на рисунке 30.

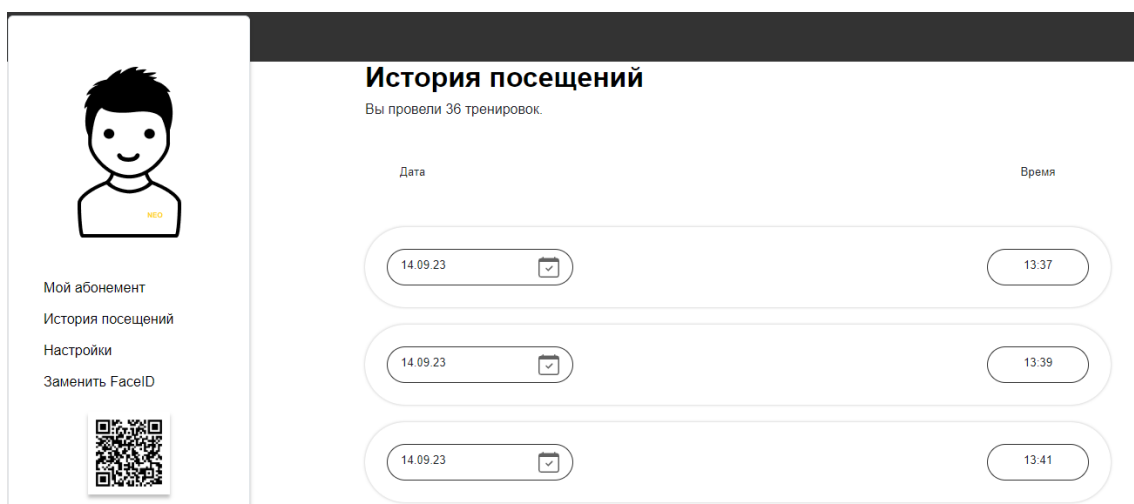


Рисунок 30 – Экранная форма историй посещений клиента

При нажатии в меню на кнопку «Настройки» клиент переходит на страницу со своими данными, представленную на рисунке 31. В настройках он может поменять номер телефона, электронную почту, дату рождения, пол и пароль.

The screenshot shows the 'Настройки' (Settings) page. At the top left is the NEO logo with the tagline 'ЗА АКТИВНОГО ОТДЫХА'. The page title is 'Тест Тест'. In the top right corner, there is a button 'Выйти из аккаунта'. On the left, a profile card displays a user icon, the text 'Мой абонемент', 'История посещений', 'Настройки', and 'Добавить вход по FaceID', along with a QR code. The main settings area includes: 'Телефон' (79876543210), 'Электронная почта' (placeholder: 'Введите электронную почту'), 'Дата рождения' (25.03.2024), 'Пол' (radio buttons for 'Мужской' and 'Женский'), and 'Пароль' section with fields for 'Введите старый пароль', 'Введите новый пароль', and 'Подтвердите пароль', plus a toggle for 'Изменить пароль'. A yellow 'Сохранить' button is at the bottom.

Рисунок 31 – Экранная форма страницы настроек в личном кабинете клиента

На рисунке 32 представлено восстановление пароля в случае, когда клиент забыл свой пароль. Он указывает свой номер телефона, пишет новый пароль и подтверждает его.

The screenshot shows the password recovery page. At the top is the NEO logo with the tagline 'ЗА АКТИВНОГО ОТДЫХА'. Below it is the heading 'Восстановление аккаунта'. The instructions read: 'Укажите номер телефона, который вы использовали при создании учетной записи'. The form contains three input fields: '8 (111) 111-11-11', 'Введите пароль', and 'Повторите пароль'. A yellow 'Отправить' button is at the bottom. Below the button is a link: 'У вас нет аккаунта? [Зарегистрироваться](#)'.

Рисунок 32 – Экранная форма восстановления пароля

### 3.6 Описание роли «Администратор»

Через роль «Администратор» менеджер фитнес-зала смотрит основной функционал системы: настройка абонементов, клиентов, просмотр отправленных смс кодов, статистика вдоха и выхода клиентов через функцию Face ID, а также для полного удаления клиента из системы раздел «Пользователи».

В разделе «Абонементы» находится страница с тарифами абонементов, в которых можно указывать абонемент, цену в рублях и сортировку их размещения в системе. Раздел «Абонементы» представлен на рисунке 33.

АБОНЕМЕНТ	СОРТИРОВКА
<input type="checkbox"/> Одноразовое посещение	0
<input type="checkbox"/> 1 месяц	1
<input type="checkbox"/> 3 месяц	2
<input type="checkbox"/> 6 месяц	3
<input type="checkbox"/> 12 месяц	4

Рисунок 33 – Экранная форма раздела «Абонементы»

При добавлении нового абонемента или по нажатию на текущий открывается страница, представленная на рисунке 34.

В поле «На сколько месяцев» указывается то число месяцев, на сколько будет действовать купленный абонемент (1 месяц абонемента равняется 30 календарным дням).

В поле «Цена в рублях» указывается та сумма, сколько клиент заплатить на использование текущего абонемента.

В поле «Сортировка» указывается очередность этого абонемента в размещении системы.

Изменить Абонемент

1 месяц ИСТОРИЯ

На сколько месяцев:

Цена в рублях:

Сортировка:

Одноразовое посещение?

Удалить Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Рисунок 34 – Экранная форма настройки абонемента

В разделе «Клиенты» находится страница с клиентами, в которых можно указывать телефон, электронную почту, фамилию и имя, дату рождения, пол, фото для страницы личного кабинета на сайте, фото для сканирования по Face ID, абонементы клиента и его занятия.

В поле «Телефон» указывается номер телефона, который указал клиент при регистрации в системе.

В поле «Email» указывается электронная почта, которую пользователь указывает в настройках своего личного кабинета.

В поле «ФИО» указывается фамилия и имя клиента, которую он указал при регистрации в системе.

В поле «Дата рождения» указывается дата рождения, которую клиент указывает в настройках своего личного кабинета.

В поле «Пол» указывается выбранный гендерный пол клиентом.

В поле «Фото» отображается фото, которое клиент загрузил в своем личном кабинете на сайте.

В поле «Фото faceid» отображается фото клиента, которое он загрузил для прохождения через функцию FaceID.

Чекбокс «Было обновление» указывается для синхронизации с аппаратной частью выбранного клиента при ручном добавлении абонемента клиенту. После синхронизации

Чекбокс «Было блокирование входа» ставится только для удаления клиента с аппаратной части.

Галки в чекбоксах снимаются автоматически, когда устройства 1 раз в 5 минут делает запрос на сервер.

В разделе «Абонемент клиента» указываются все его абонементы в системе.

В поле «Абонемент» указывается выбранный абонемент пользователем. В поле «Цена» указывается та цена, которая была указана абонементу. В поле «Дата начала» указывается с какой даты начинает действовать абонемент. В поле «Дата окончания» указывается до какой даты действует абонемент. В поле «Время публикации» показывается та дата, когда был выбран абонемент. В поле «Оплачено» отмечается галочкой, если клиент покупал абонемент, в том случае если клиент не стал покупать, то поле будет без галочки. В поле «Активный» отмечается галочкой, если в текущее время просмотра действует абонемент, когда абонемент заканчивается, то галочка убирается. Абонемент действует только в том случае, если стоят галочки «Оплачено» и «Активный». Чтоб удалить из системы абонемент, необходимо поставить галочку в «Удалить» и нажать кнопку «Сохранить».

Страница клиента в роли администратора представлена на рисунке 35.

абонемент	цена	дата начала	дата окончания	время публикации	оплачено	активный	удалить
1	1400	10.07.2023	11.12.2023	10 июля 2023	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	1400	09.10.2023	09.11.2023	9 октября 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	1400	10.11.2023	10.12.2023	10 ноября 2023	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 35 – Экранная форма страницы клиента со стороны администратора



В разделе «Отправка СМС», который представлен на рисунке 36, находится страница с кодами отправленных на телефоны, которые указывали для авторизации аккаунта.

<input type="checkbox"/> ТЕЛЕФОН	КОДИЗ SMS
<input type="checkbox"/> 8 (987) [REDACTED]	8068
<input type="checkbox"/> 8 (987) [REDACTED]	9639
<input type="checkbox"/> 8 (927) [REDACTED]	4192
<input type="checkbox"/> 8 (903) [REDACTED]	1629
<input type="checkbox"/> 8 (987) [REDACTED]	3318

Рисунок 36 – Экранная форма всех отправленных смс

В разделе «Статистика входа-выхода» указывается фамилия и имя клиента, время во сколько он прошел через Face ID и событие. Событие «Зашел» указывается в том случае, когда клиент прошел через входную дверь с использованием Face ID. Событие «Вышел» указывается в том случае, когда клиент прошел через выходную дверь с использованием Face ID.

Статистика со входами и выходами представлена на рисунке 37.

Выберите Статистика входа-выхода для изменения ДОБАВИТЬ СТАТИСТИКА ВХОДА-ВЫХОДА +

Действие:   Выбрано 0 объектов из 9

<input type="checkbox"/> КЛИЕНТ	ВРЕМЯ	СОБЫТИЕ
<input type="checkbox"/> Бутрим Ваня	14 сентября 2023 г. 13:42	Вышел
<input type="checkbox"/> Бутрим Ваня	14 сентября 2023 г. 13:41	Вышел
<input type="checkbox"/> Бутрим Ваня	14 сентября 2023 г. 13:41	Зашел
<input type="checkbox"/> Бутрим Ваня	14 сентября 2023 г. 13:39	Зашел

Рисунок 37 – Экранная форма статистики входа-выхода

В разделе «Пользователи» находится страница с телефонами клиентов. Она используется для полного удаления клиента из системы, при необходимости.

Удалить клиента можно через отмеченной галочки в списке из всех клиентов, с выбранным действием «Удалить выбранные пользователи» и нажатием на кнопку «Выполнить», либо же через нажатие номера телефона

клиента и нажатием на красную кнопку «Удалить». На рисунках 38-39 представлены варианты удаления клиента.

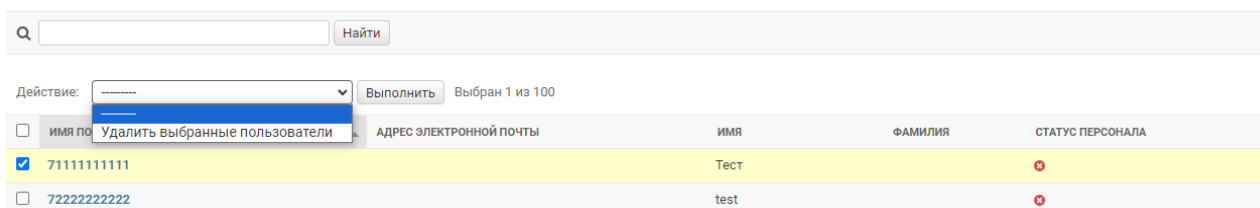


Рисунок 38 – Экранная форма удаления выбранных клиентов

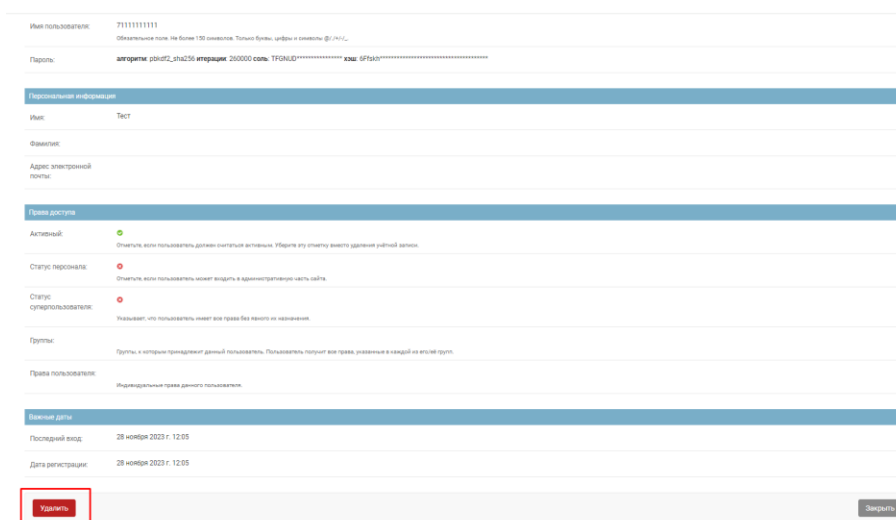


Рисунок 39 – Экранная форма удаления конкретного клиента

Раздел «Платежи» представлен на рисунке 40. В нем указываются данных сделанных платежей через Интернет-эквайринг.

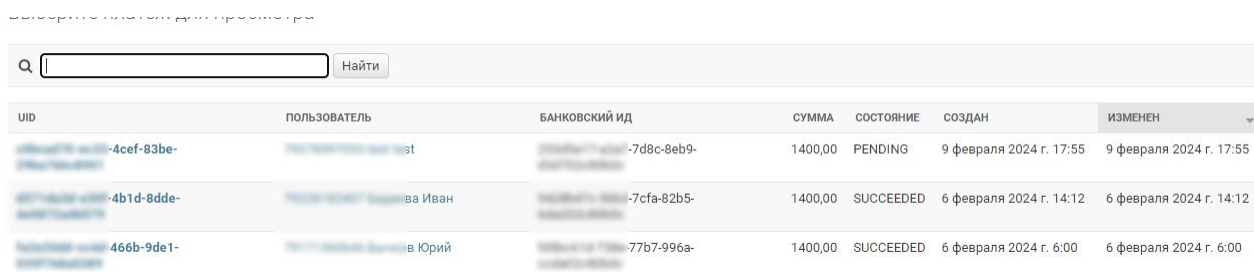


Рисунок 40 – Экранная форма страницы платежей

### 3.7 Описание роли «Модератор»

Через роль «модератор» настраивается информация на основном сайте.

В разделе «Сайты» можно включать или отключать СМС оповещения для клиентов. При отключенном чекбоксе СМС будут приходить только в панели администратора. Настройка включения и отключения СМС представлена на рисунке 41.

Изменить Сайт ИСТОРИЯ

1

Домен:  ✎ + ✕

Имя шаблона:

Отправлять смс?

Удалить Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Рисунок 41 – Экранная форма настройки включения и отключения СМС

В настройке дополнительных страниц сайта пишется описание текста для информативности клиента на сайте, по типу политики конфиденциальности, которая представлена на рисунке 42.

Изменить Страницы ИСТОРИЯ

**Политика конфиденциальности**

Описание преимущества:

url:

Статья:

Формат... | Шрифт | **B** *I* U | | Источник

**1.1 Цели**

Настоящим в соответствии с Федеральным законом № 152-ФЗ «О персональных данных» от 27.07.2006 года Вы подтверждаете свое согласие на обработку ООО "НЕО Центр" персональных данных: сбор, систематизацию, накопление, хранение, уточнение (обновление, изменение) Целью настоящей Политики является обеспечение защиты прав и свобод человека, и гражданина при обработке его персональных данных, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну.

ООО "НЕО Центр" гарантирует конфиденциальность получаемой информации. Обработка персональных данных осуществляется в целях эффективного исполнения заказов, договоров и иных обязательств, принятых ООО "НЕО Центр" в качестве обязательных к исполнению.

**1.2 Область применения**

Настоящая Политика действует в отношении всей информации, содержащей персональные данные субъектов персональных данных, которую Общество и/или связанные с ним юридические лица могут получить о субъекте персональных данных при осуществлении основной хозяйственной деятельности.

Рисунок 42 – Экранная форма текста политики конфиденциальности



Если в поле «Код из СМС» пользователь введет код не полностью, то система возвращает ошибку: заполните поле.

Если в поле «Имя» пользователь введет неверный код, то система возвращает ошибку: заполните поле.

Если в поле «Фамилия» пользователь введет код не полностью, то система возвращает ошибку: заполните поле.

Если в поле «Пол» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Если в поле «Пароль» пользователь пишет пароль меньше 8 символов, то система возвращает ошибку: ненадежный пароль.

Если в поле «Пароль» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Если в поле «Повторенный пароль» пользователь пишет пароль, отличающийся от введенного пароля в поле «Пароль» то система возвращает ошибку: пароли не совпадают.

Если в поле «Повторенный пароль» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Таблица 6 – Тестирование формы «Регистрация»

Название поля	Тестовый набор	Сообщение об ошибке
Номер телефона	8 (123) 456-78-90	Пользователь уже существует
Номер телефона	8 (987) 654-3_-__	Заполните поле
Номер телефона	8 (987) 654-32-21	-
Код из смс	4 2 2 1	Неверный код
Код из смс	4 2 2 _	Заполните поле
Код из смс	4 2 2 9	-
Имя	-	Заполните поле
Имя	Иван	-
Фамилия	-	Заполните поле
Фамилия	Иванов	-
Пол	-	Заполните поле
Пол	мужской	-
Пароль	qwerty	Ненадежный пароль
Пароль	-	Заполните поле

Продолжение таблицы 6

Название поля	Тестовый набор	Сообщение об ошибке
Пароль	Ghwey26723!	-
Повторенный пароль	Ghwey2672!	Пароли не совпадают
Повторенный пароль	Ghwey26723!	-

В таблице 7 реализовано тестирование формы «Авторизация».

Если в поле «Номер телефона» пользователь введет номер телефона не полностью или оставит поле пустым, то система возвращает ошибку: заполните поле.

Если в поле «Пароль» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Если в полях «Номер телефона» или «Пароль» ввести неверные данные, то система возвращает ошибку: ваш номер телефона или пароль не совпадают. Пожалуйста, попробуйте еще раз.

Таблица 7 – Тестирование формы «Авторизация»

Название поля	Тестовый набор	Сообщение об ошибке
Номер телефона	8 (987) 654-3_-__	Заполните поле
Пароль	-	Заполните поле
Номер телефона	8 (987) 654-32-21	Ваш номер телефона или пароль не совпадают. Пожалуйста, попробуйте еще раз.
Пароль	Ghwey2161611!	Ваш номер телефона или пароль не совпадают. Пожалуйста, попробуйте еще раз.
Название поля	Тестовый набор	Сообщение об ошибке
Номер телефона	8 (987) 654-32-21	-
Пароль	Ghwey26723!	-

В таблице 8 реализовано тестирование формы «Восстановление пароля»

Если в поле «Номер телефона» пользователь введет номер телефона не полностью, то система возвращает ошибку: заполните поле.

Если в поле «Номер телефона» пользователь введет номер телефона, которого нет в базе данных, то система возвращает ошибку: заполните поле.

Если в поле «Пароль» пользователь пишет пароль меньше 8 символов, то система возвращает ошибку: ненадежный пароль.

Если в поле «Пароль» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Если в поле «Повторенный пароль» пользователь пишет пароль, отличающийся от введенного пароля в поле «Пароль» то система возвращает ошибку: пароли не совпадают.

Если в поле «Повторенный пароль» пользователь ничего не выберет, то система возвращает ошибку: заполните поле.

Таблица 8 – Тестирование формы «Восстановление пароля»

Название поля	Тестовый набор	Сообщение об ошибке
Номер телефона	8 (987) 654-3 _ -__	Заполните поле
Номер телефона	8 (987) 654-32-20	Пользователь не найден
Номер телефона	8 (987) 654-32-21	-
Пароль	qwerty	Ненадежный пароль
Пароль	-	Заполните поле
Пароль	Ghwey26723!	-
Повторенный пароль	Ghwey2672!	Пароли не совпадают
Повторенный пароль	Ghwey26723!	-

В таблице 9 представлено тестирование загрузки фотографии для распознавания лица Face ID.

Если загрузить фотографию, где отсутствует единственное лицо человека, то система возвращает ошибку: не удалось распознать лицо.

Таблица 9 – Тестирование загрузки фотографии для распознавания лица Face ID

Название поля	Тестовый набор	Сообщение об ошибке
Изображение	Стол	Не удалось распознать лицо
Изображение	Белый фон	Не удалось распознать лицо
Изображения	Группа из несколько человек	
Изображение	Лицо человека	-

### 3.9 Оценка экономической эффективности разработки

Для расчета экономической эффективности разработки системы и определения времени окупаемости следует рассчитать ежемесячную экономию.

Зарботная плата одного администратора фитнес зала составляет 25 000 руб./месяц.

Зарботная плата всех администраторов рассчитывается по формуле 1.

$$ЗП = ЗП_A \times K_A, \quad (1)$$

где ЗП – зарботная плата всех администраторов;

ЗП<sub>A</sub> – зарботная плата одного администратора;

K<sub>A</sub> – количество работающих администраторов.

Тогда:

$$ЗП = 25\,000 \times 2 = 50\,000 \text{ (руб./месяц)} .$$

Общие затраты за разработку программно-аппаратного комплекса составляют 150 000 рублей, включая затраты на аппаратную часть.

После разработки программно-аппаратного комплекса фитнес-зал тратит 10000 руб./месяц на ежемесячную поддержку и 5000 руб./месяц на сервер.

Затраты на обслуживание программно-аппаратного комплекса рассчитываются на по формуле 2.

$$C_{\text{ПАК}} = C_{\text{е.п.}} + C_{\text{серв}}, \quad (2)$$

где C<sub>ПАК</sub> – затраты на обслуживание программно-аппаратного комплекса;

C<sub>е.п.</sub> – стоимость ежемесячной поддержки программно-аппаратного комплекса;

C<sub>серв</sub> – ежемесячная стоимость сервера.



Тогда:

$$C_{\text{ПАК}} = 10\,000 + 5\,000 = 15\,000 \text{ (руб./месяц)}$$

Ежемесячная экономия использования программно-аппаратного комплекса рассчитывается по формуле 3.

$$\mathcal{E}_{\text{ежемесячная}} = \mathcal{ЗП} - C_{\text{ПАК}}, \quad (3)$$

где  $\mathcal{E}_{\text{ежемесячная}}$  – экономия использования программно-аппаратного комплекса в месяц.

Тогда:

$$\mathcal{E}_{\text{ежемесячная}} = 50\,000 - 15\,000 = 35\,000 \text{ (руб./месяц)}.$$

Время окупаемости можно рассчитать по формуле 4.

$$V_{\text{окуп}} = (P_{\text{ПАК}}) / \mathcal{E}_{\text{ежемесячная}}, \quad (4)$$

где  $V_{\text{окуп}}$  – время окупаемости программно-аппаратного комплекса;

$P_{\text{ПАК}}$  – стоимость разработки программно-аппаратного комплекса.

Тогда:

$$V_{\text{окуп}} = 150\,000 / 35\,000 \approx 4,28 \text{ месяца.}$$

Таким образом, система окупится через 5 месяцев после внедрения.

В результате проделанной работе разработан программно-аппаратный комплекс, состоящий из интерфейса пользователя в виде информативного

одностраничного сайта и личного кабинета клиента, административных панелей для администратора и модератора и аппаратной части.

Проведено тестирование разработанного модуля считывания и обработки биометрических данных в реальных условиях фитнес-зала.

Были протестированы четыре формы заполнения базы данных: регистрация, авторизация, восстановление пароля, добавления фотографии для Face ID, а также рассчитано время окупаемости после реализации проекта.

## Заключение

Выпускная квалификационная работа посвящена разработке автоматизации работы фитнес-зала на основе программно-аппаратного комплекса и выполнена на основе материалов преддипломной практики, проведенной в ООО «АльВиРити», которое является резидентом технопарка в сфере высоких технологий «Жигулевская долина».

Целью данной работы было создание функционального программно-аппаратного комплекса, способного автоматизировать процессы регистрации и инновационной идентификации клиентов фитнес-зала, сокращая человеческий фактор и повышая эффективность управления.

В результате проведенной работы были решены следующие задачи:

- изучена работа компании ООО «НЕО ЦЕНТР»;
- спроектирована концептуальная модель предметной области;
- спроектирована логическая модель программно-аппаратного комплекса;
- разработано программное обеспечение в виде информативного одностраничного сайта с личными кабинетами и автоматизированной информационной системы для загрузки, обработки и идентификации лиц по Face ID на основе нейронных сетей, ведения базы данных клиентов, покупки абонементов;
- разработаны панели администратора и модератора для удобства разбиение управления фитнесом менеджером при необходимости;
- разработана аппаратная часть комплекса, обеспечивающая считывание и обработку биометрических данных в реальном времени;
- интегрированы программная и аппаратная часть в единый функциональный комплекс;

- проведено тестирование разработанного программно-аппаратного комплекса в реальных условиях фитнес-зала;
- оценена экономическая эффективность разработанного программно-аппаратного комплекса.

Итогом бакалаврской работы является автоматизация бизнес-процессов компании, исключение человеческого фактора, сокращение фонда оплаты труда, укрепление инновационного имиджа и статуса организации, для которой реализован данный проект.

Разработанный программно-аппаратный комплекс прошел успешное тестирование и введен в эксплуатацию в зале активного отдыха ООО «НЕО» в технопарке в сфере высоких технологий «Жигулевская долина», расположенному в г. Тольятти, ул. Южное шоссе, 165А.

## Список используемой литературы и используемых источников

1. Буч, Гради Введение в UML от создателей языка / Гради Буч, Джеймс Рамбо, Ивар Якобсон. - М.: ДМК Пресс, 2015. - 496 с.
2. Дронов, В. Django 2.1 Практика создания веб-сайтов на Python / В. Дронов – СПб.: ВHV, 2019. – 672 с.
3. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы.
4. ГОСТ Р ИСО/МЭК 12119-2000 - Информационная технология. Пакеты программ. Требования к качеству и тестирование.
5. Королев, Е. Н. Администрирование СУБД: учебное пособие / Е. Н. Королев, Б. Н. Тишуков, А. В. Мандрыкин. — Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2021. — 156 с.
6. Лузанов, П. PostgreSQL для начинающих / П. Лузанов, Е. Рогов, И. Лёвшин. – М.: Компания Postgres Professional, 2017. – 118 с.
7. Лучано, Р. Python. К вершинам мастерства – ДМК Пресс, 2016. – 768с.
8. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: учебное пособие / Маглинец Ю.А. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 191 с.
9. Марка, Д.А. Методология структурного анализа и проектирования SADT / Д.А. Марка, К. МакГоуэн. – М.: МетаТехнология, 1993. – 243 с.
10. Меле, А. Django 2 в примерах / А. Меле; перевод Д. В. Плотникова. — Москва: ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [Электронный ресурс]. — URL: <https://www.iprbookshop.ru/126199.html>
11. Норенков И.П. Автоматизированные информационные системы: учебное пособие / Норенков И.П. — Москва: Московский

- государственный технический университет имени Н.Э. Баумана, 2011. — 341 с. — ISBN 978-5-7038-3446-6. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [Электронный ресурс]. — URL: <https://www.iprbookshop.ru/94730.html>
12. Петроченков, А. Идеальный Landing Page. Создаем продающие вебстраницы / А. Петроченков, Е. Новиков. – СПб.: Изд-во «Питер», 2016. – 320с.
  13. Раскин. Интерфейс: новые направления в проектировании компьютерных систем / Раскин, Джеф. - М.: Символ-плюс, 2017. - 272 с.
  14. Ревунков, Г. И. Проектирование баз данных: учебное пособие / Г. И. Ревунков, Н. А. Ковалева, Е. Ю. Силантьева. — Москва: Московский государственный технический университет имени Н.Э. Баумана, 2018. — 48 с.
  15. Ригс, С. Администрирование PostgreSQL 9. Книга рецептов / С. Ригс, Х. Кроссинг. – ДМК Пресс, 2013. – 368с.
  16. Фаулер, М. UML. Основы. Третье издание. / М. Фаулер. – М.: Символ-Плюс, 2006. – 192 с.
  17. Якименко А.А. Внедрение биометрической идентификации в системы контроля и управления доступом: учебное пособие / Якименко А.А., Вихман В.В. — Новосибирск: Новосибирский государственный технический университет, 2016. — 48 с. — ISBN 978-5-7782-3020-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [Электронный ресурс]. — URL: <https://www.iprbookshop.ru/91336.html>
  18. Face Recognition [Электронный ресурс]. URL: <https://face-recognition.readthedocs.io/en/latest/readme.html> (дата обращения 03.03.2024)
  19. Michael Learn. Learn Python programming / М. Learn. – Independently publisher, 2019. - 202 с.

20. Nat Dunn, Webucator. Python 3.8 / N. Dunn. – Webucator, 2020. - 554 с.
21. OpenCV: Cascade Classifier [Электронный ресурс]. URL: [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html) (дата обращения 03.03.2024)
22. Raspberry Pi Documentation [Электронный ресурс]. URL: <https://www.raspberrypi.com/documentation/computers/getting-started.html> (дата обращения 15.03.2024)