**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| **ДП.09.02.03.23.191.12.ПЗ** | УТВЕРЖДАЮ  Зам. директора по УР, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_Е.А. Коробкова |

**ПРИЛОЖЕНИЕ**

**«ФИТНЕС-КЛУБ»**

|  |  |  |
| --- | --- | --- |
| Нормконтролер: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (В.А. Пролыгина) |
| Консультант по экономической части: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Рачкова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.Н. Филимонова) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.А. Иванова) |

Иркутск 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc103762663)

[1 Предпроектное исследование 6](#_Toc103762664)

[1.1 Описание предметной области 6](#_Toc103762665)

[1.2 Анализ инструментальных средств реализации 8](#_Toc103762666)

[2 Техническое задание на разработку программного продукта 10](#_Toc103762667)

[3 Проектирование 11](#_Toc103762668)

[3.1 Архитектура программного обеспечения 11](#_Toc103762669)

[3.2 Функциональное проектирование 12](#_Toc103762670)

[3.3 Проектирование базы данных 19](#_Toc103762671)

[3.4 Проектирование пользовательского интерфейса 24](#_Toc103762672)

[4 Реализация программного обеспечения 26](#_Toc103762673)

[4.1 Кодирование программного обеспечения 26](#_Toc103762674)

[4.2 Разработка базы данных 32](#_Toc103762675)

[4.3 Разработка программного продукта 33](#_Toc103762676)

[5 Документирование программного обеспечения 38](#_Toc103762677)

[5.1 Руководство пользователя 38](#_Toc103762678)

[6 Стоимость разработки и внедрения программного продукта 40](#_Toc103762679)

[6.1 Организационно-экономическое обоснование проекта 40](#_Toc103762680)

[6.2 Расчет затрат на разработку программного продукта 40](#_Toc103762681)

[6.3 Расчет затрат на внедрение программного продукта 44](#_Toc103762682)

[6.4 Основные выводы 44](#_Toc103762683)

[ЗАКЛЮЧЕНИЕ 45](#_Toc103762684)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#_Toc103762685)

[Приложение А – Листинг страницы создания расписания занятий 48](#_Toc103762686)

[Приложение Б – Листинг авторизация«login» 50](#_Toc103762687)

ВВЕДЕНИЕ

За последние несколько лет значительно возрос объем и оборот информации во всех сферах человеческой деятельности: экономической, финансовой, политической, духовной. В связи с увеличением информации возникает необходимость в использовании автоматических инструментов, позволяющих хранить, эффективно обрабатывать и распределять накопленные данные.

Преимуществом автоматизации является сокращение избыточности хранимых данных, а, следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции, увеличение степени достоверности информации и увеличение скорости обработки информации; излишнее количество внутренних промежуточных документов, различных журналов, папок, заявок и т.д., повторное внесение одной и той же информации в различные промежуточные документы. Также значительно сокращает время автоматический поиск информации, который производится из специальных экранных форм, в которых указываются параметры поиска объекта.

Основой задачей данного мобильного приложения является учет абонентов фитнес-клуба, подготовка документов для внешней среды (приказов, служебных записок, отчетов). Эти задачи имеют регулярный характер и выполняются непосредственными исполнителями (системными администраторами). Результаты выполнения операций регистрируются в соответствующих регистрах. Автоматизация этих процессов позволит хранить информацию в одной базе, в которую вводятся данные с помощью удобного интерфейса.

Целью данной дипломной работы является проектирование мобильного приложения учета клиентов для фитнес-клуба, приводящее к более эффективной работе.

В ходе дипломного проектирования были поставлены следующие задачи:

1. Провести предпроектное исследование;
   1. Провести исследование предметной области;
   2. Проанализировать инструменты, используемые в разработке программного обеспечения;
   3. Обосновать выбор программных продуктов для разработки;
2. Составить техническое задание на разработку программного продукта;
3. Провести проектирование программного продукта;
   1. Представить архитектуру программного обеспечения;
   2. Провести функциональное проектирование;
   3. Спроектировать базу данных;
   4. Спроектировать пользовательский интерфейс программного продукта;
4. Реализовать программный продукт;
5. Разработать документы для программного продукта;
6. Рассчитать стоимость разработки и внедрения программного продукта.
7. Предпроектное исследование
   1. Описание предметной области

В качестве предметной области дипломного проекта был выбран «Фитнес-клуб».

В настоящее время такие заведения получили большое признание среди общественности. Всё больше и больше людей стало посещать подобные заведения, направленные на создание, поддержание и укрепление здоровья для полноценного существования в обществе, создание материально-производственной базы, планирование, организация и проведение массовых коммерческих спортивно-оздоровительных мероприятий, пропаганда здорового образа жизни.

Может показаться, что фитнес-клуб состоит только из тренажерного зала! На самом деле он включает в себя очень много направленностей, таких как: бассейн, тренажерный зал, залы аэробики и различных групповых программ, массажные кабинеты и многое другое.

Обслуживание клиентов фитнес-клуба осуществляется таким образом:

1. Происходит выбор основных услуг:

* Тренажерный зал;
* Кардио-зона;
* Аэробные залы;
* Бассейн;
* SPA комплекс;
* Инструктажи в залах;

1. Далее выбор дополнительных услуг:

* Индивидуальные тренировки с тренером;
* Массаж;
* Мастер классы по ведущим направлениям аэробных занятий (Йога, Школа плавания, Пилатес и т.д)

1. Подборка подходящего абонемента, исходя из пожеланий клиента;
2. Проведение оплаты;
3. Назначение исполнителей;
4. Оказание соответствующих услуг;

Весь персонал сети фитнес-клуба можно разделить на группы, занимающиеся выполнением определенных функций:

* Руководитель;
* Специалисты;
* Исполнители;
* Обслуживающий персонал.

Таким образом, встает проблема регистрации и учета клиентов, и оптимизации работы с ними. Решением данной проблемы является создание удобной системы учета клиентов, которая способна выполнять те задачи, которые перед ней ставятся. Это позволит увеличить скорость и качество обслуживания клиентов, предоставит дополнительные возможности для развития компании и увеличит потенциально возможный поток клиентов.

* 1. Анализ инструментальных средств реализации

Для разработки данного продукта было рассмотрено несколько инструментальных средств разработки программного обеспечения. Такие как: C#, Python, JavaScript.

Python – это высокоуровневый язык программирования общего назначения. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами.

JavaScript – динамический скриптовый язык программирования высокого уровня. Он отличается мультипарадигменностью. Речь идет о поддержке функционального, императивного, событийно-ориентированного стилей. Чаще всего язык используется для создания интерактивных веб-страниц и приложений. Востребованность, сферы применения. Неизменно высокий интерес к JavaScript подтверждают специальные международные рейтинги.

C# – современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET.

Visual Studio – это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

Теперь сравним эти языки программирования между собой (таблица 1).

Таблица 1 – Сравнение языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии | Python | JavaScript | C# |
| Наличие библиотек | + | + | + |
| Инструменты для работы с БД | + | + | + |
| Объектно-ориентированные возможности | + | + | + |

Исходя из данной таблицы, было принято решение остановится на языке программирования C# потому, что он достаточно прост и понятен, что поможет реализовать мне данный продукт.

Разберем среды разработки. Выбор встал между Visual Studio и JetBrains Rider.

Visual Studio – линейка продуктов компании [Майкрософт](https://dic.academic.ru/dic.nsf/ruwiki/1018162), включающих [интегрированную среду разработки](https://dic.academic.ru/dic.nsf/ruwiki/940808) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](https://dic.academic.ru/dic.nsf/ruwiki/978372) [приложения](https://dic.academic.ru/dic.nsf/ruwiki/4631), так и приложения с [графическим интерфейсом](https://dic.academic.ru/dic.nsf/ruwiki/22252), в том числе с поддержкой технологии [Windows Forms](https://dic.academic.ru/dic.nsf/ruwiki/74841), а также [веб-сайты](https://dic.academic.ru/dic.nsf/ruwiki/1135216), [веб-приложения](https://dic.academic.ru/dic.nsf/ruwiki/35038), [веб-службы](https://dic.academic.ru/dic.nsf/ruwiki/80509) как в [родном](https://dic.academic.ru/dic.nsf/ruwiki/28842), так и в [управляемом](https://dic.academic.ru/dic.nsf/ruwiki/57362) кодах для всех платформ,

JetBrains Rider – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#.

Для выбора среды разработки была составлена таблица 2.

Таблица 2 – Сравнение сред разработки

|  |  |  |
| --- | --- | --- |
| Название IDE | Visual Studio | JetBrains Rider |
| Автосохранение | + | + |
| Автодополнение | + | + |
| Поиск по коду | + | + |

После сравнения были сделаны выводы, что для выбранного дипломного проекта лучшее всего подойдет Visual Studio потому, что данная программа имеет очень хороший функционал, что поможет в реализации данного программного продукта. Также для создания мобильного приложения на С# был выбран фреймворк Xamarin обладающий необходимым функционалом для успешной разработки.

Для работы приложения необходимы базы данных, поэтому необходимо также провести анализ для выбора средств реализации баз данных.

MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных. Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.

Преимущества:

* простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных;
* обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта;
* безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию;
* масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных;
* скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

SQLite – это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

Преимущества:

* файловая: вся база данных хранится в одном файле, что облегчает перемещение;
* стандартизированная: SQLite использует SQL; некоторые функции опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), однако, есть и некоторые новые;
* отлично подходит для разработки и даже тестирования: во время этапа разработки большинству требуется масштабируемое решение. SQLite, со своим богатым набором функций, может предоставить более чем достаточный функционал, при этом будучи достаточно простой для работы с одним файлом и связанной библиотекой.

PostgreSQL – это самая продвинутая РСУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO.

Преимущества:

* полная SQL-совместимость;
* сообщество: PostgreSQL поддерживается опытным сообществом 24/7;
* поддержка сторонними организациями: несмотря на очень продвинутые функции, PostgreSQL используется в многих инструментах, связанных с РСУБД;
* расширяемость: PostgreSQL можно программно расширить за счёт хранимых процедур;
* объектно-ориентированность: PostgreSQL – не только реляционная, но и объектно-ориентированная СУБД.

Сравним СУБД, чтобы определиться, какая больше подойдет для данной дипломной работы (таблица 3).

Таблица 3 – Сравнение средств реализации базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | MySQL | SQLite | PostgreSQL |
| Большое количество типов данных | + | - | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | - | + |
| Не требует удаленного сервера | - | + | - |
| Простота использования | + | + | - |
| Портативность | + | + | - |

При сравнении СУБД для данного дипломного проекта была выбрана MySQL потому, что с ней приятно и удобно работать, что поможет лучше реализовать данный продукт.

1. Техническое задание на разработку программного продукта

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 34.602-89.

Согласно ГОСТ 34.602-89 техническое задание должно включать следующие разделы:

1. общие сведения;
2. назначение и цели создания системы;
3. требования к системе в целом;
   1. требования к структуре и функционированию системы;
   2. требования к надежности;
   3. требования к безопасности;
   4. требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
4. требования к документированию;
5. состав и содержание работ по созданию системы.

Техническое задание на разработку приложения представлено отдельным документом.

1. Проектирование
   1. Архитектура программного обеспечения

Архитектура программного обеспечения – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Мобильное приложение «Таксопарк» использует клиент-серверную архитектуру (рисунок 1). **Архитектура «клиент-сервер» предусматривает разделение процессов предоставление услуг и отправки запросов на них на разных компьютерах в сети, каждый из которых выполняют свои задачи независимо от других.**

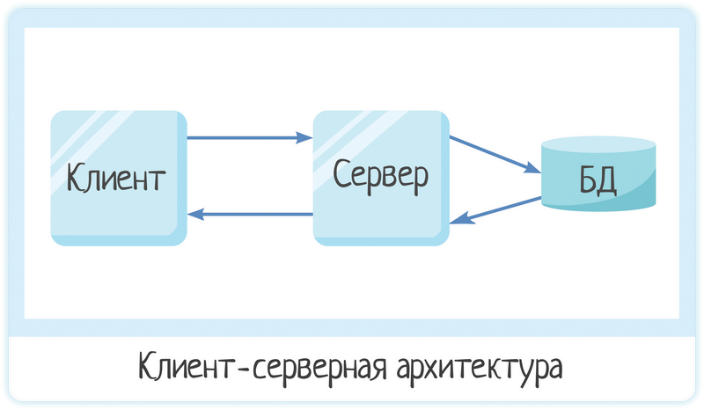


Рисунок 1 – Архитектура мобильного приложения «Таксопарк»

Сервер – специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Параметры, которые могут реализоваться на стороне сервера:

1. хранение, защита и доступ к данным;
2. работа с поступающими клиентскими запросами;
3. процесс отправки ответа клиенту.

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Параметры, которые могут реализоваться на стороне клиента:

1. площадка по предоставлению пользовательского графического интерфейса;
2. формулировка запроса к серверу и его последующая отправка;
3. получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

Таким образом, в архитектуре «клиент-сервер» клиент посылает запрос на предоставление данных и получает только те данные, которые действительно были затребованы.

* 1. Функциональное проектирование

Проектирование мобильного приложения происходит при помощи CASE средств, которые помогают обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов.

На рисунке 2 изображена структурная схема ИС, она описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования. Основными элементами диаграммы являются участник (actor) и прецедент (вариант).

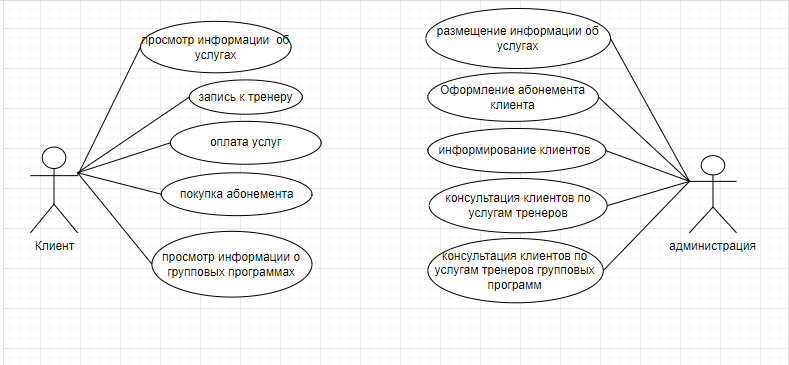


Рисунок 2 – Диаграмма прецедентов

Таким образом, представленные диаграммы демонстрируют взаимодействие основных объектов мобильного приложения и их действия.

На рисунке 3 изображена диаграмма деятельности. Применяется эта диаграмма для того, чтобы промоделировать последовательные и при необходимости параллельные шаги вычислительного процесса.

Сценарий работы с приложением показан на рисунке 3.

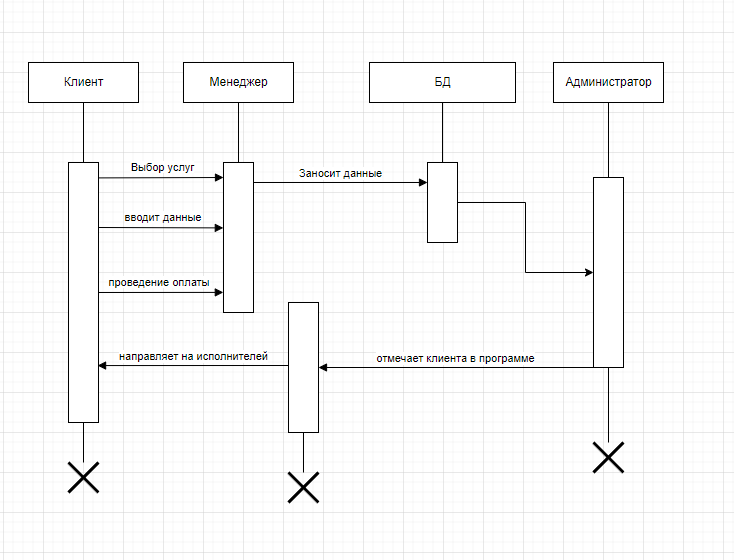


Рисунок 3 –Диаграмма последовательности

На рисунке 4 изображена диаграмма компонентов, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

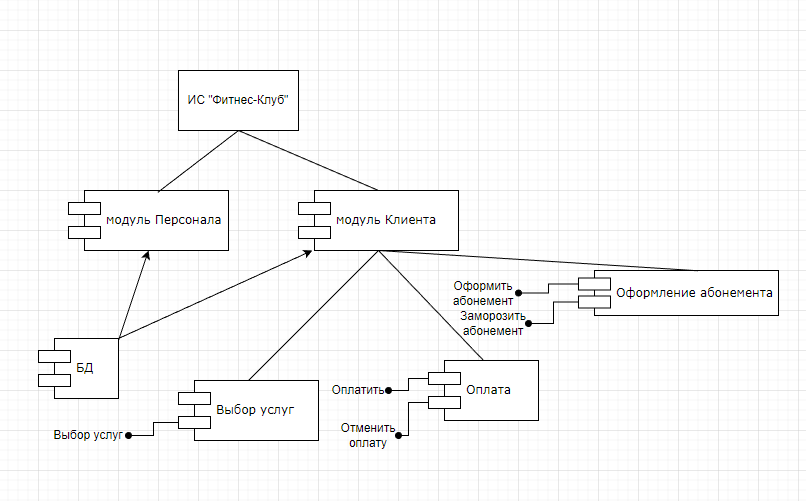


Рисунок 4 – Диаграмма компонентов

На рисунке 5 представлена диаграмма развертывания – это тип диаграммы, которая определяет физическое оборудование, на котором будет работать программная система. Он также определяет способ развертывания программного обеспечения на базовом оборудовании. Он отображает программные части системы на устройство, которое будет выполнять его.

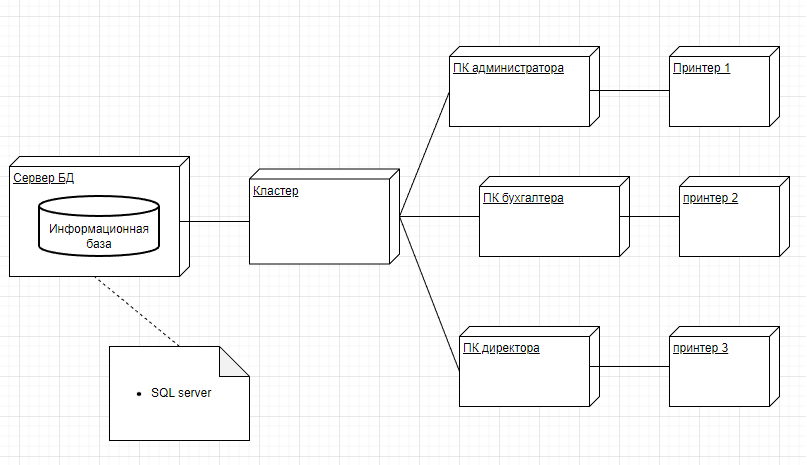


Рисунок 5 – Диаграмма развертывания

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На рисунке 6 изображена контекстная диаграмма, на которой показаны входные данные, управление, механизм, выходные данные и функция.

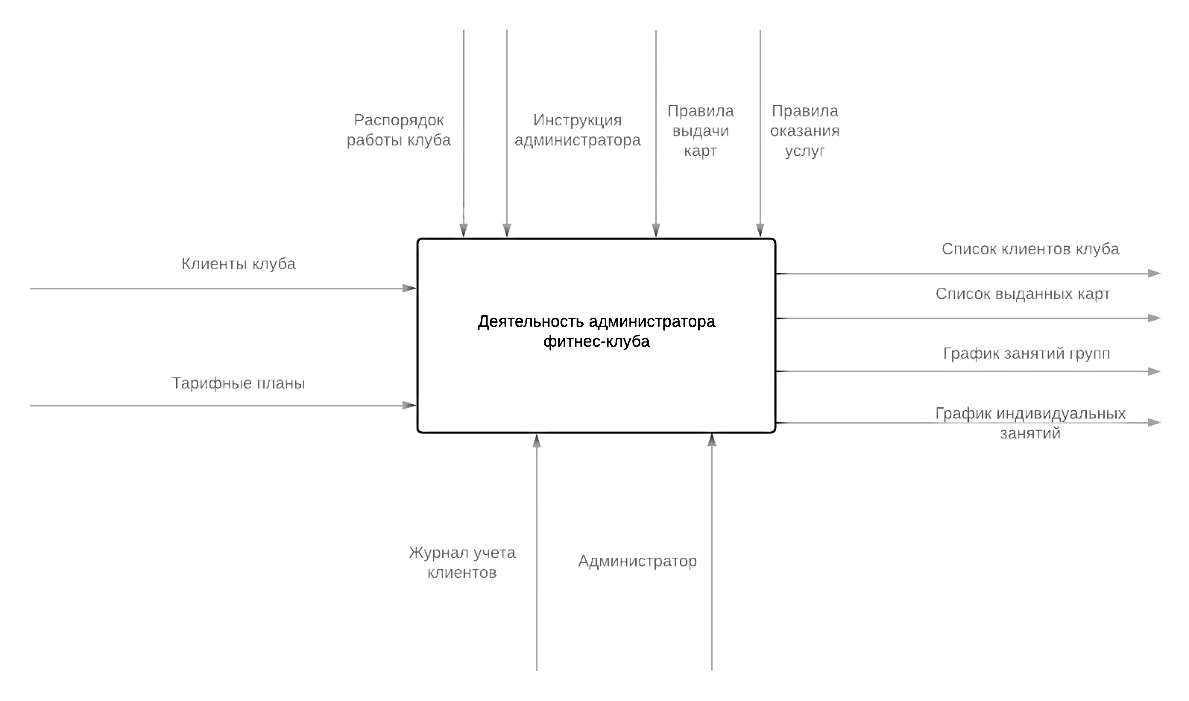


Рисунок 6 – Контекстная диаграмма

На рисунке 7 изображена диаграмма декомпозиции. Она показывает комплекс, процесс, организацию, предметную область данных или другой тип объекта, разбитый на более детализированные компоненты более низкого уровня. Например, диаграммы декомпозиции могут представлять организационную структуру или функциональную декомпозицию на процессы.

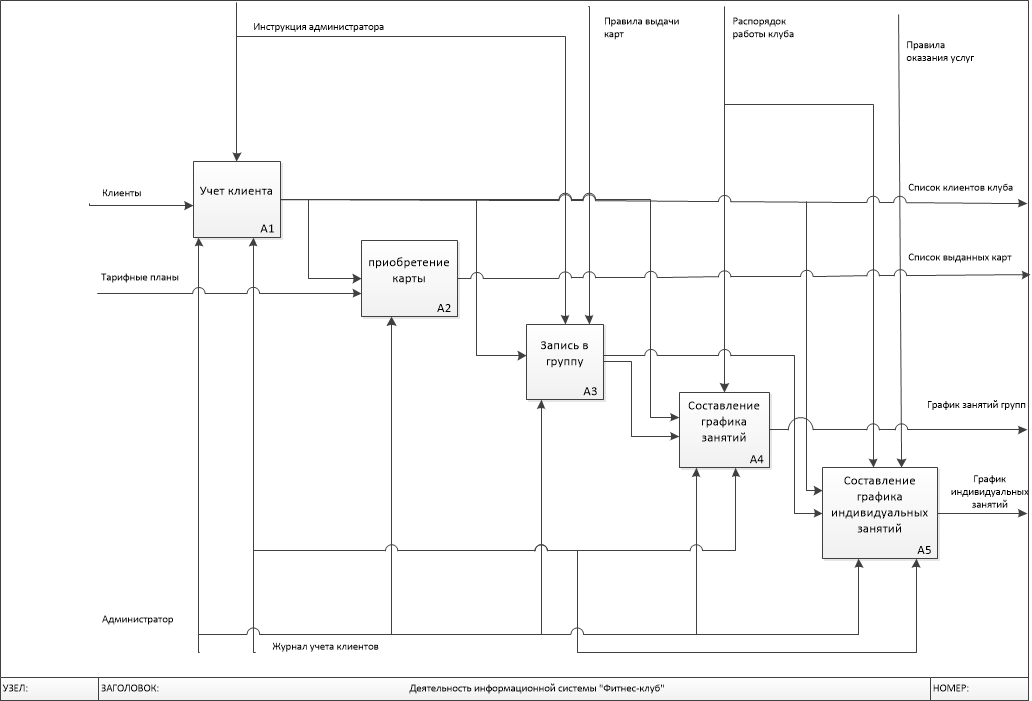


Рисунок 7 – Диаграмма декомпозиции

Также в ходе работы была разработана диаграмма классов, представленная на рисунке 8. Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

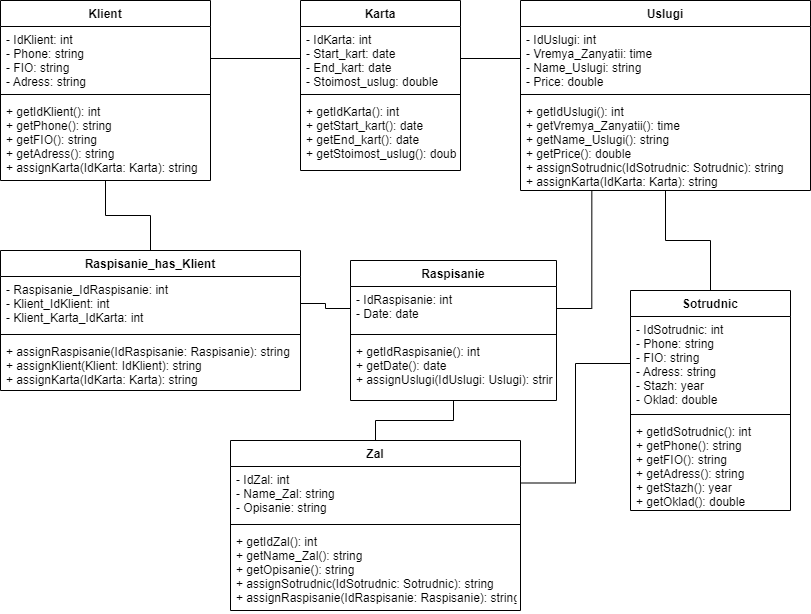
**

Рисунок 8 – Диаграмма классов

Диаграмма потоков данных, представленная на рисунке 9. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники, и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

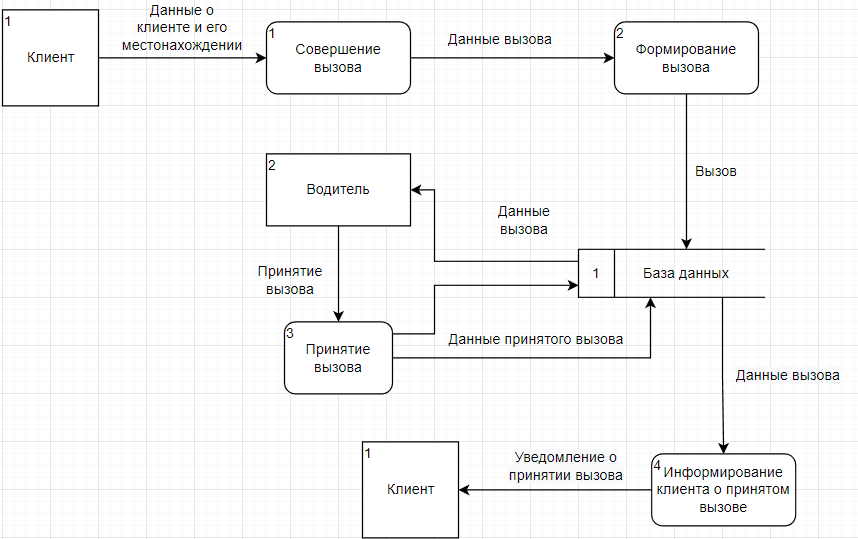


Рисунок 9 – Диаграмма потоков данных

Функциональное проектирование представляет наиболее общий подход к описанию систем. Определяются граничные условия и желательные входы, и выходы, составляется подробный перечень функций или операций, которые должны выполняться. При функциональном проектировании осуществляется синтез структуры и определяются основные параметры объекта и его составных частей (элементов), оцениваются показатели эффективности и качества процессов функционирования.

* 1. Проектирование базы данных

Перед разработкой базы данных есть необходимость в инфологическом моделировании. Результатом такого моделирования является инфологическая модель базы данных, изображенная на рисунке 10, которая наглядно показывает объект, его свойства и отношения между другими объектами.

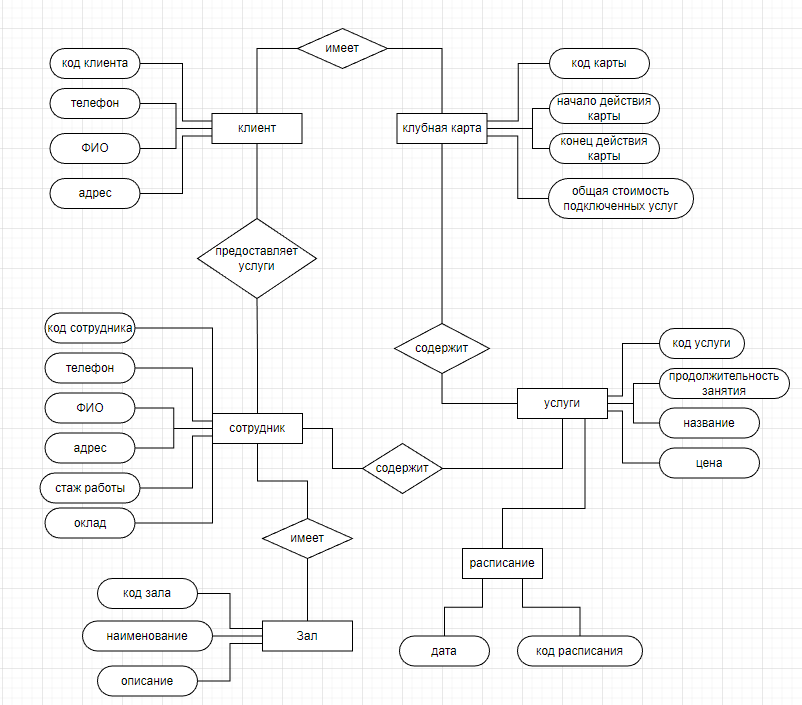


Рисунок 10 – Инфологическая модель БД

На рисунке 11 представлена даталогическая модель. Модель предполагает определение состава и взаимосвязей таблиц, отражающих содержание информационных сущностей инфологической модели в терминах конкретной СУБД.

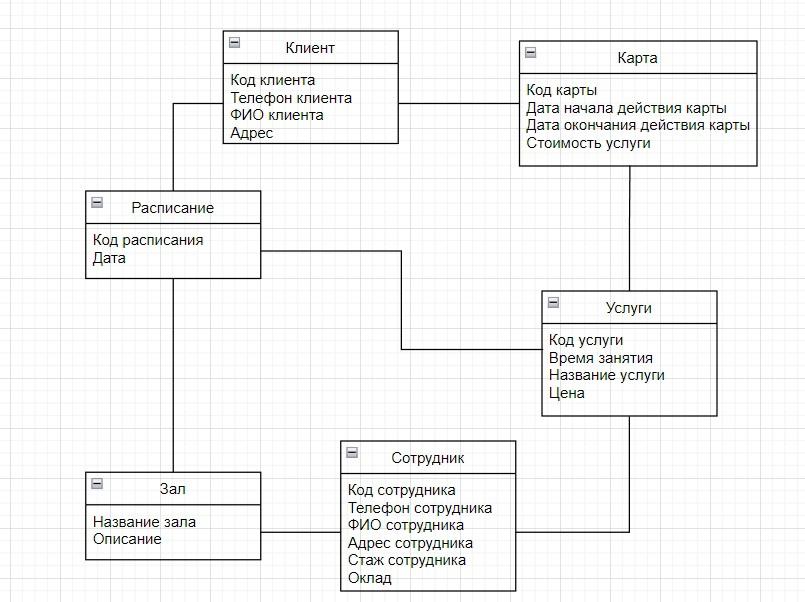


Рисунок 11 – Даталогическая модель

На рисунке 12 представлена ER–модель базы данных. Представлены таблицы, связи между ними и типы данных.

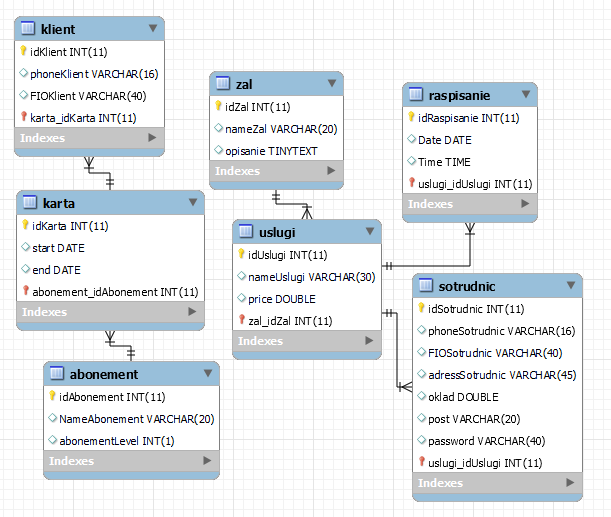


Рисунок 12 – ER-модель базы данных

Далее представим независимую сущность таблиц базы данных. Дальше будут отображены не все свойства сущностей, поскольку не являются необходимыми для функционирования дипломного проекта

Таблица 4 – Таблица «Клиент»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| idKlient | int | Код клиента |
| Phone | Varchar(45) | Телефон |
| FIO | Text(60) | ФИО |
| Adress | Varchar(60) | Адрес |
| Karta\_IdKarta | int | Код карты |

Таблица 5 – Таблица «Карта»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdKarta | int | Код карты |
| Start\_Kart | date | Начало абонемента |
| End\_Kart | date | Окончание абонемента |
| Stoimost\_uslug | Double | Стоимость абонемента |

Таблица 6 – Таблица «Услуги»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdUslugi | int | Код услуги |
| Vremya\_Zanati | Time | Время занятия |
| Name\_Uslugi | Text(50) | Название услуги |
| Price | Double | Цена услуги |
| Sotrudnic\_IdSotrudnic | int | Код сотрудника |
| Karta\_IdKarta | int | Код карты |

Таблица 7 – Таблица «Сотрудник»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdSotrudnic | int | Код сотрудника |
| Phone | Varchat(30) | Телефон сотрудника |
| FIO | Text(30) | ФИО сотрудника |
| Adress | Varchat(45) | Адрес сотрудника |
| Stazh | Year(3) | Стаж |
| Oklad | Double | Оклад |

Таблица 8 – Таблица «Зал»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdZal | int | Код зала |
| Name\_Zal | Text(20) | Название зала |
| Opisanie | Text(60) | Описание зала |
| Sotrudnic\_IdSotrudnic | int | Код сотрудника |
| Raspisanie\_IdRaspisanie | int | Код расписания |

Таблица 9 – Таблица «Расписание»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdRaspisanie | int | Код расписания |
| Date | Datetime | Дата расписания |
| Uslugi\_IdUslugi | int | Код услуги |

Таблица 10 – Таблица «Расписание клиента»

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| idRaspisanie | int | Код расписания |
| Date | Datetime(60) | Дата и время |
| Karta\_IdKarta | int | Код карты |

Для разработки даталогической был использован инструмент Draw.io.

Для разработки ER-модели был использован MySQL Workbench 8.0 CE.

После завершения разработки прототипа базы данных и таблиц, находящихся в ней, можно перейти к проектированию интерфейса системы.

* 1. Проектирование пользовательского интерфейса

Пользовательский интерфейс (UI) - это взаимодействие между пользователем и компьютерной системой, которое позволяет пользователю управлять функциональностью и настройками программного обеспечения с помощью графических элементов, таких как кнопки, текстовые поля, меню и диалоговые окна.

Разработка удобного пользовательского интерфейса – это один из важнейших этапов в процессе создания программного продукта.

В результате проектирование интерфейса будущей информационной системы были спроектированы прототипы трёх форм: Окно авторизации (рисунок 13), окно регистрации (рисунок 14), и окно аккаунта пользователя (рисунок 15).

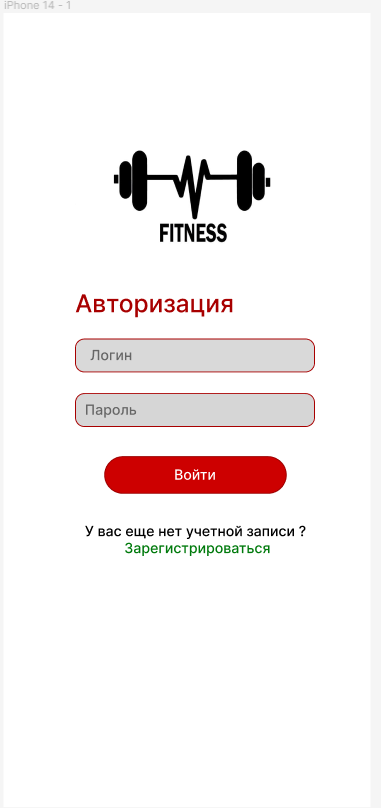


Рисунок 13 – Интерфейс авторизации

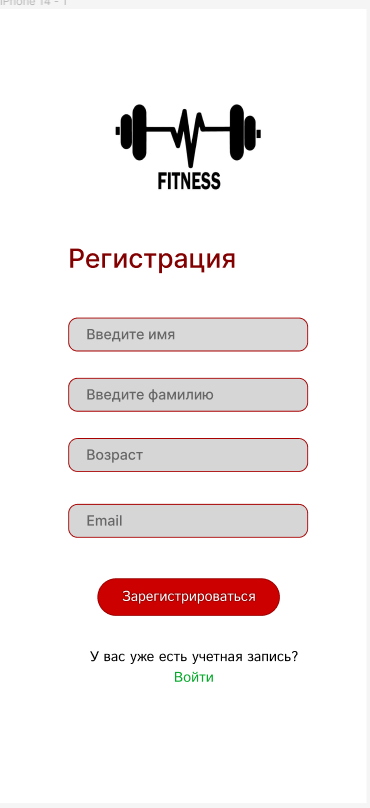


Рисунок 14 – Интерфейс регистрации



Рисунок 15 – Аккаунт пользователя