Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |  |
| --- | --- |
| Допускаю к защите |  |
| Руководитель | подпись |
|  | Л.С. Вахрушева |
|  | И.О. Фамилия |

|  |
| --- |
| Разработка цифрового медицинского помощника |

наименование темы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту по дисциплине

|  |  |  |
| --- | --- | --- |
| Технологии разработки программных комплексов | | |
|  | 1.24.00.00 - ПЗ |  |

обозначение документа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-21-1 |  |  |  | А.В. Собинов |
|  |  | шифр группы |  | подпись |  | И.О. Фамилия |
| Нормоконтроль |  |  |  |  |  | Л.С. Вахрушева |
|  |  |  |  | подпись |  | И.О. Фамилия |

Курсовой проект защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Иркутск 2024 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

ЗАДАНИЕ

НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

|  |  |  |  |
| --- | --- | --- | --- |
| По курсу | Технологии разработки программных комплексов | | |
| Студенту | Собинову А.В. | | |
|  | (фамилия, инициалы) | | |
| Тема проекта | Разработка цифрового медицинского помощника | | |
|  | | | |
| Исходные данные: | |  | |
| Разработка цифрового медицинского помощника | | | |
|  | | | |
| |  |  |  | | --- | --- | --- | | Рекомендуемая литература: |  | | | 1. Гутгарц Р.Д Проектирование автоматизированных систем обработки информации и управления: учебное пособие для академического бакалавриата. – М.: Издательство Юрайт, 2019. | | | 1. Проектирование АСОИУ [Электронный ресурс] : методические указания по выполнению курсового проекта: 09.03.01 «Информатика и вычислительная техника» для бакалавров по специальности «Автоматизированные системы обработки информации и управления» / Иркут. нац. исслед. техн. ун-т ; сост. Р. Д. Гутгарц. - Электрон. дан. - Иркутск : ИРНИТУ, 2018. | | | 1. Проектирование информационных систем : учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общ. ред. Д. В. Чистова. – М. : Издательство Юрайт, 2016. | | | 1. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления : учебное пособие для вузов / И. Д. Рудинский. - Москва : Горячая линия - Телеком, 2011. | | | | |  |

Графическая часть на \_\_\_\_\_\_\_\_\_\_\_\_\_\_ листах.

Дата выдачи задания « 20 » сентября 2024 г.

|  |  |  |  |
| --- | --- | --- | --- |
| Задание получил |  |  | А.В. Собинов |
|  | подпись |  | И.О. Фамилия |

Дата представления проекта руководителю 2024 г.

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель курсового проектирования |  |  | Л.С.Вахрушева |

**Содержание**

[Введение 4](#_Toc186238203)

[1 Анализ 5](#_Toc186238204)

[1.1 Важность своевременного отслеживания показателей здоровья 5](#_Toc186238205)

[1.2 Преимущества отслеживания показателей здоровья в цифровом виде 5](#_Toc186238206)

[1.3 Ключевые функции для медицинского помощника 5](#_Toc186238207)

[2 Обзор существующих решений 5](#_Toc186238208)

[2.1 Apple Health 5](#_Toc186238209)

[2.2 Google Fit 6](#_Toc186238210)

[2.3 Вывод 6](#_Toc186238211)

[3 Процесс AS IS vs TO BE 7](#_Toc186238212)

[3.1 Процесс AS IS 7](#_Toc186238213)

[3.2 Процесс TO BE 8](#_Toc186238214)

[4 Описание вариантов использования 8](#_Toc186238215)

[5 Выработка требований и постановка задачи 9](#_Toc186238216)

[6 Выбор и обоснование средств проектирования и реализации 9](#_Toc186238217)

[6.1 Средства проектирования 9](#_Toc186238218)

[6.2 Средства реализации 9](#_Toc186238219)

[7 Проектирование архитектуры приложения 10](#_Toc186238220)

[8 Проектирование хранилища данных 10](#_Toc186238221)

[8.1 Таблицы для хранения анкет 10](#_Toc186238222)

[8.2 Таблицы для хранения условий к вопросам 12](#_Toc186238223)

[8.3 Таблицы для хранения ответов анкеты 12](#_Toc186238224)

[8.4 Таблицы для хранения показателей здоровья 13](#_Toc186238225)

[9 Проектирование точек доступа для работы с базой данных и интеграции в приложения 14](#_Toc186238226)

[9.1 Точки доступа для работы с анкетой 14](#_Toc186238227)

[9.2 Точки доступа для работы с показателями здоровья 14](#_Toc186238228)

[10 Json-форматы для отправки данных в базу данных 15](#_Toc186238229)

[10.1 Json-формат для отправки показателей здоровья 15](#_Toc186238230)

[10.2 Json-формат для отправки результатов анкеты 15](#_Toc186238231)

[11 Разработка мобильного приложения 16](#_Toc186238232)

[Заключение 20](#_Toc186238233)

[Список использованных источников 21](#_Toc186238234)

# Введение

Проект «Разработка цифрового медицинского помощника» направлен на создание многофункционального программного комплекса, который станет важным инструментом для студентов медицинских вузов, преподавателей и сотрудников лечебных учреждений. Основная цель данного проекта – оптимизировать процессы обучения, диагностики и управления здоровьем пациентов посредством использования современных технологий.

Главной задачей разработки цифрового медицинского помощника является создание универсальной платформы со следующими функциями:

- Предоставление пользователям возможности самостоятельного ввода и отслеживания ключевых показателей здоровья.

- Анализ введенных данных и предоставление индивидуальных советов по поддержанию здоровья и предотвращению развития заболеваний.

- Реализация функционала для проведения анкет среди пациентов, что облегчит сбор статистической информации и оценку состояния здоровья групп людей.

Для успешной реализации проекта необходимо решить несколько важных задач:

1. Разработать базу данных для хранения анкет и показателей здоровья.
2. Разработать программные интерфейсы, которые позволят внешним приложениям и системам безопасно взаимодействовать с базой данных. Это обеспечит интеграцию цифрового медицинского помощника с различными информационными.
3. Разработать мобильное приложения.
4. Реализовать возможности внесения данных пользователями.
5. Разработать алгоритмы, которые будут анализировать введённые пользователем данные и выдавать персонализированные рекомендации по поддержанию здоровья и предотвращению развития заболеваний.
6. Внедрить функционал для создания и проведения анкет среди пользователей.
7. Провести тестирование всех компонентов системы, собрать отзывы от пользователей и внести необходимые доработки перед финальным запуском.

Проект представляет собой важную инициативу, направленную на улучшение качества медицинских услуг и облегчение управления здоровьем пациентов. Реализация этого проекта внесёт значительный вклад в развитие современной медицины и повысит эффективность работы медицинских специалистов

# Анализ

## Важность своевременного отслеживания показателей здоровья

Отслеживание показателей здоровья играет важнейшую роль в поддержании общего благополучия и предотвращения развития хронических заболеваний. Регулярное измерение и анализ таких параметров, как артериальное давление, уровень сахара в крови, частота сердечных сокращений, вес и другие показатели, позволяют вовремя выявить отклонения от нормы и принять соответствующие меры.

## Преимущества отслеживания показателей здоровья в цифровом виде

Приложение можно будет интегрировать с другими медицинскими сервисами, позволяя врачам быстро получать доступ к актуальной информации о пациенте для диагностики и лечения. Также обработка данных с помощью приложения позволяет анализировать большие объемы информации за короткий срок, выявляя закономерности и тренды, которые могут быть незаметны при ручном анализе. Ещё приложение даст возможность удаленного мониторинга, что особенно важно для пациентов с хроническими заболеваниями или тех, кто проживает вдали от медицинских учреждений.

## Ключевые функции для медицинского помощника

Для успешной реализации цифрового медицинского помощника необходимо учитывать ключевые потребности его пользователей - пациентов и медицинских специалистов. С точки зрения пациентов, помощник должен обеспечивать удобный и интуитивно понятный интерфейс для ввода данных о своем здоровье, возможность отслеживать изменения в показателях и получать персонализированные рекомендации. Важно, чтобы пользователи могли легко фиксировать и анализировать свои показатели здоровья, что создаст мотивацию для регулярного использования приложения. С точки зрения медицинских специалистов, помощник должен предоставлять удобные инструменты для анализа данных пациентов, составления отчетов и получения статистики, что позволит оперативно оценивать состояние здоровья пациентов и корректировать лечение.

# Обзор существующих решений

Существуют различные программные решения, схожие с проектом «Цифровой медицинский помощник»:

## Apple Health

Apple Health – встроенное приложение для iOS, которое собирает данные о здоровье через подключаемые устройства (например, Apple Watch), предоставляет аналитику и советы по улучшению образа жизни.

Преимущества:

- Интеграция с экосистемой Apple: Легкая синхронизация с устройствами Apple, такими как Apple Watch, iPhone и iPad.

- Широкий спектр функций: Отслеживание шагов, калорий, уровня кислорода в крови, ЭКГ и других важных показателей здоровья.

- Безопасность данных: Высокий уровень защиты персональных данных благодаря встроенным механизмам безопасности Apple.

Недостатки:

- Ограниченная совместимость: Работает только на устройствах Apple, что ограничивает аудиторию пользователей.

- Недостаток кастомизации: Некоторые пользователи отмечают недостаток гибкости в настройках и интерфейсе.

## Google Fit

Google Fit – аналогичное решение от Google, которое помогает пользователям отслеживать физическую активность, следить за показателями здоровья и получать персональные рекомендации.

Преимущества:

- Кросс-платформенность: Доступен на Android и iOS, что делает его доступным для широкой аудитории.

- Простота использования: Интерфейс интуитивно понятен и удобен даже для новичков.

- Совместимость с различными устройствами: Поддержка множества фитнес-трекеров и умных часов.

Недостатки:

- Менее точные данные: По сравнению с некоторыми специализированными приложениями, точность измерений может быть ниже.

- Концентрация на физической активности: Основное внимание уделяется тренировкам и шагам, тогда как другие аспекты здоровья освещены менее подробно.

## Вывод

Для наглядности результаты сравнения перенесены в таблицу 1.

Таблица 2.1 - Преимущества и недостатки существующих решений

|  |  |  |
| --- | --- | --- |
| **Программное средство** | **Преимущества** | **Недостатки** |
| Apple Health | Интеграция с экосистемой Apple: Легкая синхронизация с устройствами Apple, такими как Apple Watch, iPhone и iPad. Широкий спектр функций: Отслеживание шагов, калорий, уровня кислорода в крови, ЭКГ и других важных показателей | Ограниченная совместимость: Работает только на устройствах Apple, что ограничивает аудиторию пользователей.  Недостаток кастомизации: Некоторые пользователи  отмечают недостаток гибкости в настройках и интерфейсе. |

Продолжение таблица 2.1

|  |  |  |
| --- | --- | --- |
| **Программное средство** | **Преимущества** | **Недостатки** |
| Apple Health | здоровья. Безопасность данных: Высокий уровень защиты персональных данных благодаря встроенным механизмам безопасности Apple. |  |
| Google Fit | Кросс-платформенность: Доступен на Android и iOS, что делает его доступным для широкой аудитории. Простота использования: Интерфейс интуитивно понятен и удобен даже для новичков. Совместимость с различными устройствами: Поддержка множества фитнес-трекеров и умных часов. | Менее точные данные: По сравнению с некоторыми специализированными приложениями, точность измерений может быть ниже.  Концентрация на физической активности: Основное внимание уделяется тренировкам и шагам, тогда как другие аспекты здоровья освещены менее подробно. |

Apple Health ограничен устройствами Apple, что сокращает целевую аудиторию, и у него есть недостатки в плане настройки и удобства использования. Google Fit, хоть и доступен на разных платформах, акцентирует внимание главным образом на физической активности, упуская из виду другие важные аспекты здоровья. Разрабатываемое ПО будет лишено этих недостатков.

# Процесс AS IS vs TO BE

Процесс разработки программного обеспечения и управления проектами можно представить в двух состояниях:

- AS IS - текущее состояние, как процессы выполняются в настоящее время.

- TO BE - целевое состояние, которое будет достигнуто после внедрения новой программы.

## Процесс AS IS

Описание текущего состояния:

- Записи хранятся в бумажном виде, что создает риск потери или повреждения данных.

- Анализ данных осуществляется вручную, путем просмотра записей и сравнения значений.

Недостатки:

- Трудоемкость процесса высокая, так как требуется много времени на обработку данных.

- Сложно выявить тенденции и закономерности, поскольку данные разбросаны и трудно сопоставимы.

- Доступ к данным ограничен, необходимо физическое присутствие для работы с ними.

## Процесс TO BE

Описание целевого состояния:

- Данные автоматически передаются в приложение и сохраняются в цифровой форме.

- Уменьшение вероятности ошибок при вводе данных и повышение точности измерений.

- Автоматическая агрегация и анализ данных внутри приложения.

- Возможность построения графиков и отчетов для визуализации тенденций и динамики изменений.

- Персонализированные рекомендации на основе анализа исторических данных и текущих показаний.

- Возможность отправки уведомлений и предупреждений при достижении критических значений.

- Интерактивные подсказки и советы по улучшению здоровья.

- Хранение данных в облачных хранилищах с возможностью резервного копирования.

- Легкий доступ к данным через интернет с любого устройства.

# Описание вариантов использования

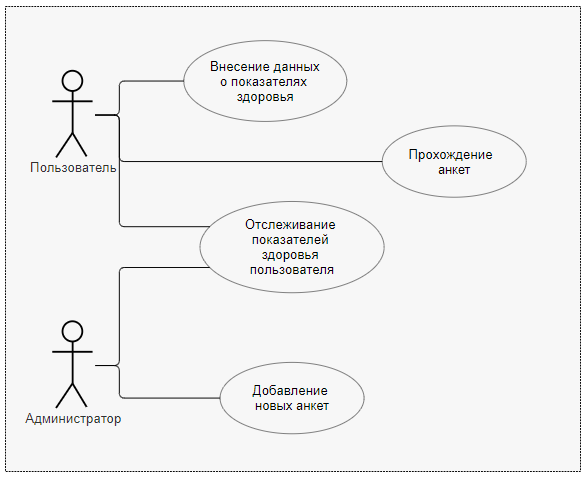


Рисунок 4.1 - Диаграмма вариантов использования

Пользователем приложения сможет стать любой человек. Пользователь с помощью программы сможет проходить анкеты, вносить свои показатели здоровья и отслеживать их. Также будет назначен специалист (медицинский работник), отвечающий за серверную часть. Он будет добавлять новые анкеты и просматривать данные, внесённые пользователями.

Диаграмма вариантов использования представлена на рисунке 1.

# Выработка требований и постановка задачи

Создать платформу, которая упростит отслеживание показателей здоровья как для пользователей, так и для медицинских работников.

Для этого необходимо выполнить следующие задачи:

- Разработать базу данных для хранения анкет и показателей здоровья.

- Разработать API.

- Разработать мобильное приложения.

- Реализовать возможности внесения данных пользователями.

- Разработать алгоритмы для анализа данных.

- Внедрить функционал для создания и проведения анкет.

- Провести тестирование всех компонентов системы.

# Выбор и обоснование средств проектирования и реализации

## Средства проектирования

- Draw.io – веб-приложение, с помощью которого можно строить различные диаграммы. Имеет множество инструментов, необходимых для выполнения разнообразных задач.

- StarUML – приложение для визуального моделирования. Он поддерживает стандартизованный язык графического описания UML для моделирования систем и программного обеспечения.

## Средства реализации

- PostgreSQL – это современная свободная объектно-реляционная СУБД, используемая для хранения и организации данных. Информация о проектах будет храниться, используя таблицы в базе данных данной СУБД.

- pgAdmin4 – это программа для управления СУБД PostgreSQL. Она используется для написания SQL-запросов, разработки процедур и функций, а также для администрирования PostgreSQL. С помощью этой программы будут созданы таблицы, процедуры и функции внутри базы данных PostgreSQL, а также будет проведено тестирование функциональности СУБД отдельно от программы, чтобы удостовериться в работоспособности СУБД.

- Python - это высокоуровневый язык программирования, известный своим простым и читаемым синтаксисом. Он поддерживает разные парадигмы программирования и подходит для широкого спектра задач, от веб-разработки до науки о данных. Python работает на многих операционных системах и имеет богатую стандартную библиотеку и большое сообщество. В данном проекте он использован для создания точек доступа к базе данных.

- FastAPI - это современный фреймворк для создания API на Python, основанный на стандартах ASGI. FastAPI известен своей скоростью, простотой использования и поддержкой асинхронного программирования. Он предоставляет удобные инструменты для документирования API, автоматической проверки типов данных и обработки запросов.

- Kotlin - это статически типизированный язык программирования, разработанный компанией JetBrains. Он предназначен для разработки на платформе Java Virtual Machine (JVM), а также для Android, браузеров и нативных приложений. Kotlin совместим с Java и отличается лаконичностью, безопасностью и мощными функциями, такими как null-безопасность и корутины.

# Проектирование архитектуры приложения

Для проекта "Цифровой медицинский помощник" была выбрана клиент-серверная архитектура, так как она соответствует основным целям проекта - предоставить пользователям простое и удобное решение для управления своими медицинскими данными. Такая архитектура позволит: иметь доступ к данным удаленно, изменять информацию на сервере без необходимости обновлять приложение пользователя, создать систему аутентификации и авторизации пользователей для проверки прав доступа к данным. Схема взаимодействия между клиентом и сервером представлена на рисунке 7.1.



Рисунок 7.1 – Архитектура приложения

# Проектирование хранилища данных

Рассмотрим основные таблицы базы данных.

## Таблицы для хранения анкет

Таблица Interview – хранит записи об Анкетах.

Таблица 8.1 – Атрибуты таблицы Interview

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id анкет |
| name | text | Хранит название анкеты |
| description | text | Хранит описание для анкеты |

Таблица Question – хранит записи об вопросах.

Таблица 8.2 – Атрибуты таблицы Question

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id вопросов |
| name | text | Хранит название вопроса |
| description | text | Хранит описание для вопроса |
| type | int4 | Хранит тип вопроса |

Таблица Options – хранит записи об опциях-ответах на вопрос.

Таблица 8.3 – Атрибуты таблицы Options

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id опций для ответов |
| name | text | Хранит название опции |
| type | int4 | Хранит тип опции |
| constraint | json | Хранит ограничение для опции |

Таблица Question\_of\_interview – соединяет таблицы Interview и Question. Добаляет больше информации о вопросе в определённой анкете.

Таблица 8.4 – Атрибуты таблицы Question\_of\_interview

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id вопросов для определённой анкеты |
| question\_id | int8 (FK) | Хранит id базового вопроса, связывает с таблицей Question |
| interview\_id | int8 (FK) | Хранит id анкеты, связывает с таблицей Interview |
| priority | int4 | Хранит степень важности вопроса в анкете |
| transition\_type | int4 | Хранит тип перехода к этому вопросу |

Таблица Options\_of\_question – соединяет таблицы Options и Question. Добаляет больше информации об опции в определённой вопросе.

Таблица 8.5 – Атрибуты таблицы Options\_of\_question

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id опций для определённого вопроса |
| question\_id | int8 (FK) | Хранит id базового вопроса, связывает с таблицей Question |
| option\_id | int8 (FK) | Хранит id опции, связывает с таблицей Options |
| priority | int4 | Хранит степень важности вопроса в анкете |
| transition\_type | int4 | Хранит тип перехода к этому вопросу |

## Таблицы для хранения условий к вопросам

Таблица Сonditions – хранит записи об условиях.

Таблица 8.6 – Атрибуты таблицы Сonditions

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id условий |
| condition\_question\_id | int8 (FK) | Хранит id вопроса анкеты, от которого идёт условие, связывает с таблицей Question\_of\_interview |
| question\_of\_interview \_id | int8 (FK) | Хранит id вопроса анкеты, к которому направлено условие, связывает с таблицей Question\_of\_interview |
| compare\_type\_id | int8 (FK) | Хранит id типа сравнения, к которому направлено условие, связывает с таблицей Compare\_type |
| description | text | Хранит описание для условия |

Таблица Compare\_type – хранит записи о типах сравнения.

Таблица 8.7 – Атрибуты таблицы Compare\_type

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id типов сравнения |
| name | text | Хранит тип сравнения |

Таблица Option\_condition – хранит записи об опциях условия.

Таблица 8.8 – Атрибуты таблицы Option\_condition

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id для опции сравнения |
| condition\_operand\_id | int8 (FK) | Хранит id условия, связывает с таблицей Сonditions |
| option\_of\_question\_id | int8 (FK) | Хранит id опции, связывает с таблицей Options\_of\_question |
| value | text | Хранит значение, с которым сравнивают |

## Таблицы для хранения ответов анкеты

Таблица Answers – хранит записи об ответах.

Таблица 8.9 – Атрибуты таблицы Answers

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id ответов |
| user\_id | int8 (FK) | Хранит id пользователя, связывает с таблицей Users |

Продолжение таблицы 8.9

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| question\_of\_interview\_id | int8 (FK) | Хранит id вопроса анкеты, связывает с таблицей Question\_of\_interview |
| option\_of\_question\_id | int8 (FK) | Хранит id опции, связывает с таблицей Options\_of\_question |
| answer\_text | text | Хранит ответ на вопрос |
| type | int8 (FK) | Хранит тип ответа на вопрос |
| response\_time | timestamp | Хранит время ответа на вопрос |

Таблица Type\_of\_answer – хранит записи о типах ответа.

Таблица 8.10 – Атрибуты таблицы Type\_of\_answer

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id типов ответа |
| name | text | Хранит название типа ответа |

Таблица Users – хранит записи о пользователях.

Таблица 8.10 – Атрибуты таблицы Users

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id пользователей |
| name | text | Хранит имя пользователя |
| username | text | Хранит ник пользователя |
| email | text | Хранит адрес электронной почту пользователя пользователя |
| password | text | Хранит пароль пользователя |

## Таблицы для хранения показателей здоровья

Таблица Parameter\_value\_for\_user – хранит записи о показателях.

Таблица 8.11 – Атрибуты таблицы Parameter\_value\_for\_user

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id записи определённого показателя здоровья |
| user\_id | int8 (FK) | Хранит id пользователя, связывает с таблицей Users |
| parameter\_id | int8 (FK) | Хранит id показателя, связывает с таблицей parameters |
| value1 | float4 | Хранит данные по показателю |
| value1 | float4 | Хранит дополнительные данные по показателю |

Продолжение таблицы 8.11

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| time | timestamp | Хранит время предоставления проверки показателя |

Таблица Parameters – хранит записи о показателях.

Таблица 8.11 – Атрибуты таблицы Parameters

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | bigserial (PK) | Хранит уникальные id показателей здоровья |
| name | text | Хранит название показателя |
| description | text | Хранит описание показателя |

# Проектирование точек доступа для работы с базой данных и интеграции в приложения

## Точки доступа для работы с анкетой

Для получения результатов интервью и их сохранения были созданы две точки доступа к базе данных interview\_answers\_access\_point и interview\_save\_access\_point.

Таблица 9.1 - Точки доступа для работы с анкетой

|  |  |  |
| --- | --- | --- |
| Название функции | Входные параметры | Результат |
| interview\_save\_access\_point | Данные формата json, хранящие результаты прохождения анкеты | Сохраняет полученные данные в базу данных |
| interview\_answers\_access\_point | Номер анкеты и id пользователя | Выгружает результаты анкеты для определённого пользователя из базы данных |

## Точки доступа для работы с показателями здоровья

Для получения показателей здоровья пользователя и их сохранения были созданы две точки доступа к базе данных parameter\_access\_point и parameter\_save\_access\_point.

Таблица 9.2 - Точки доступа для работы с показателями здоровья

|  |  |  |
| --- | --- | --- |
| Название функции | Входные параметры | Результат |
| parameter\_save\_access\_point | Данные формата json, хранящие информацию о показателе | Сохраняет полученные данные в базу данных |

Продолжение таблицы 9.2

|  |  |  |
| --- | --- | --- |
| Название функции | Входные параметры | Результат |
| parameter\_access\_point | id пользователя | Выгружает записи о показателях здоровья для определённого пользователя из базы данных |

# Json-форматы для отправки данных в базу данных

В ходе работы были придуманны два json-формата: для отправки результатов анкеты и для отправки показателей здоровья.

## Json-формат для отправки показателей здоровья

Для отправки показателей здоровья используется следующий формат:

*parameter = {*

*"user\_id": 2,*

*"parameter\_id": 1,*

*"value1": 36,*

*"value2": None,*

*"time": "2024-12-13 18:28:30.051437"*

*}*

Данный формат содержит ID пользователя, ID параметра, основное значение показателя, дополнительное значение показателя, время предоставления показателя.

## Json-формат для отправки результатов анкеты

Для отправки результатов анкеты используется следующий формат:

*questions\_answers = {*

*"interview\_result": {*

*"interview\_id": 1,*

*"user\_id": 1,*

*"answers": [*

*{*

*"question\_of\_interview\_id": 15,*

*"question\_id": 54,*

*"option\_id": [48],*

*"answer\_text": None,*

*],*

*},*

*{*

*"question\_of\_interview\_id": 18,*

*"question\_id": 56,*

*"option\_id": [4],*

*"answer\_text": None,*

*}*

*]*

*}*

*}*

Данный формат содержит ID анкеты, ID пользователя, ответы на вопросы. Каждый ответ представляет собой запись, состоящую из ID вопроса анкеты, ID основного вопроса, ID опции ответа на вопрос и текста ответа на вопрос.

# Разработка мобильного приложения

Мобильное приложение всё ещё находится в стадии разработки, но уже обладает некоторыми функциями. Работа мобильного приложения продемонстрирована на рисунках 11.1-11.6.

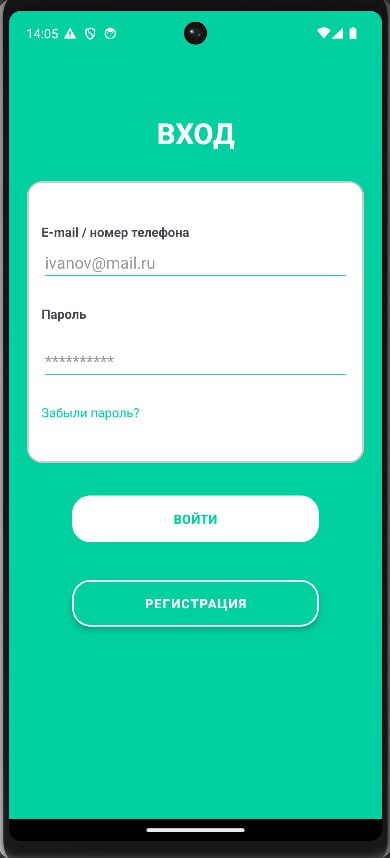


Рисунок 11.1 – Страница авторизации

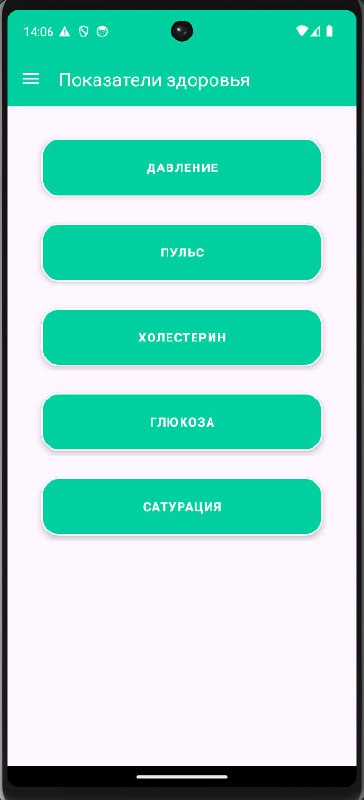


Рисунок 11.2 – Интерфейс меню

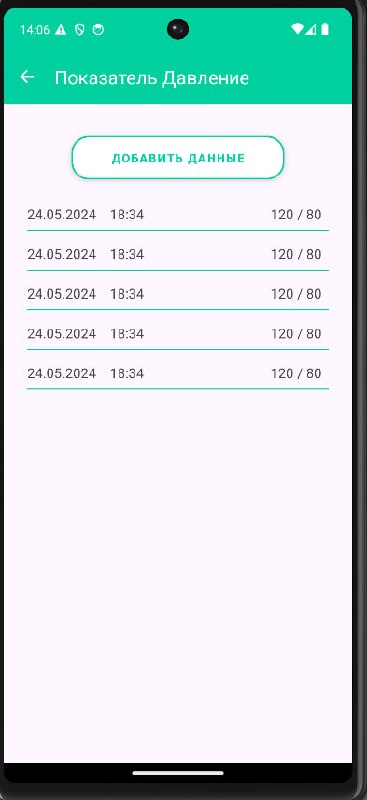


Рисунок 11.3 – Таблица показателя

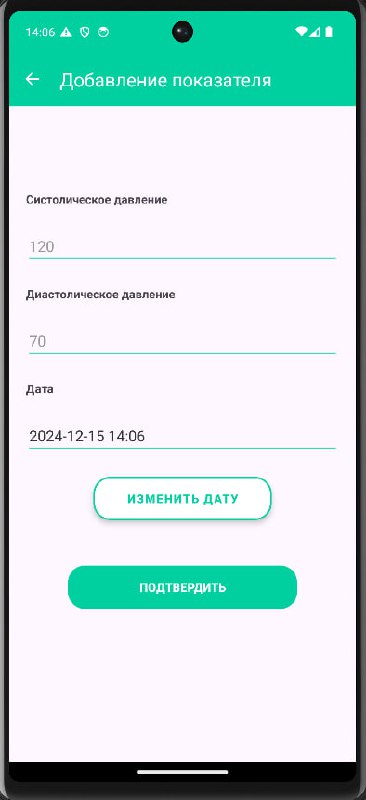


Рисунок 11.4 – Изменение показателя

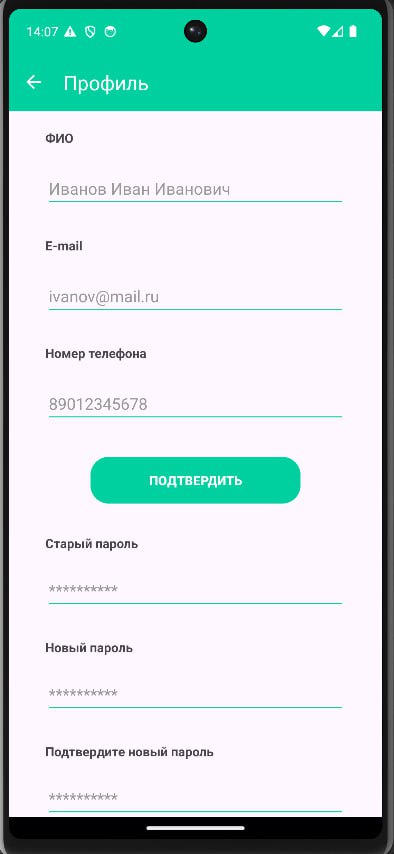


Рисунок 11.5 – Профиль

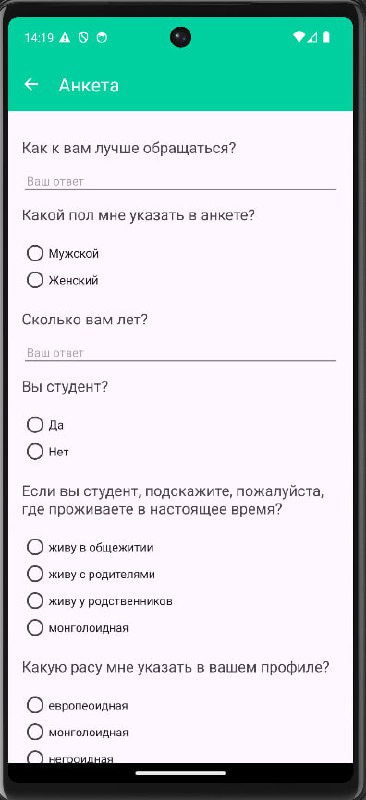


Рисунок 11.6 – Анкетирование

На данный момент в приложении не реализована работа с точками доступа.

# Заключение

Проект "Цифровой медицинский помощник" представляет собой инновационное решение для мониторинга и анализа медицинских данных, что позволяет пользователям эффективно управлять своим здоровьем. Архитектура проекта основана на клиент-серверной модели, что обеспечивает высокую производительность, стабильность и удобство использования. Создание базы данных, API точек доступа к ней и мобильного приложения позволит реализовать функциональность, необходимую для работы с медицинскими данными. Пользователи могут легко смогут вносить данные о своем здоровье, отслеживать динамику изменений и получать персонализованные рекомендации. Включение модуля для проведения анкет расширит функциональные возможности приложения, делая его полезным инструментом для профилактического наблюдения и контроля за состоянием здоровья. Проект не ориентирован на образовательный сегмент, что подчеркивает его направленность на практическое применение в повседневной жизни для повышения качества жизни.

# Список использованных источников

1. «Python Documentation» [Электронный ресурс]. – URL: <https://docs.python.org/3/> (дата обращения: 15.11.2024).
2. «FastAPI Documentation» [Электронный ресурс]. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 15.11.2024).
3. «PostgreSQL Documentation» [Электронный ресурс]. – URL: <https://www.postgresql.org/docs/> (дата обращения: 15.11.2024).
4. Лекции по разработке виртуальног ассистента [Электронный ресурс]. – URL: <https://el.istu.edu/mod/book/view.php?id=360675/> (дата обращения: 15.11.2024).
5. Гутгарц Р.Д. Проектирование автоматизированных систем обработки информации и управления: учебное пособие для академического бакалавриата. – М.: Издательство Юрайт, 2018.