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

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Разработка программного обеспечения
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка программного обеспечения с использованием нейросетевых моделей для прогнозирования рисков ухудшения ментального здоровья детей старшего подросткового возраста

Обучающийся А. Р. Резникова _____
(Инициалы Фамилия) (личная подпись)

Руководитель канд. пед. наук, доцент, О. М. Гущина
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант канд. филол. наук, доцент, М. В. Дайнеко
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Выпускная квалификационная работа на тему «Разработка программного обеспечения с использованием нейросетевых моделей для прогнозирования рисков ухудшения ментального здоровья детей старшего подросткового возраста» посвящена актуальной проблеме прогнозирования рисков ухудшения ментального здоровья у подростков. В работе предложено решение данной проблемы использованием мобильного приложения с нейросетевыми моделями, что позволяет автоматизировать процесс анализа данных и повысить точность прогнозирования.

Объектом исследования выступает процесс прогнозирования рисков ухудшения ментального здоровья в подростковой среде. Предмет исследования — нейросетевые модели машинного обучения, адаптированные для анализа поведенческих паттернов и результатов психологического тестирования.

Цель работы – разработка программного обеспечения, использующего нейросетевые модели для прогнозирования рисков ухудшения ментального здоровья у подростков.

Структура работы включает введение, три главы, заключение, список используемой литературы и приложения.

Объем работы составляет 61 страниц, включая 10 таблиц и 18 рисунков. В работе использовано 29 источников литературы. Основные этапы работы включали сбор и анализ данных, разработку нейросетевой модели на основе многослойного перцептрона (MLP), обучение модели с использованием алгоритма Adam, а также тестирование на реальных данных

Основные итоги работы: разработанное приложение и нейросетевая модель показала высокую точность прогнозирования рисков ухудшения ментального здоровья у подростков и принята к использованию в учреждении среднего профессионального образования.

Annotation

The title of the graduation work is *Developing software using neural network models to predict the risks of older teenagers' mental health deterioration*.

The graduation work consists of an introduction, 3 chapters, 10 tables, 18 figures, a conclusion, and a list of 29 references including foreign sources.

The aim of this graduation work is to develop software using the neural network models for predicting the risks of older teenagers' mental health deterioration. The object of the study is the process of predicting older teenagers' mental health risks. The subject of the research is the machine learning neural network models adapted for analyzing behavioral patterns and psychological testing results. The key issue of the graduation work is exploring the application of the neural networks for automating data analysis and improving prediction accuracy when assessing the mental health.

The graduation work may be divided into several logically connected parts which are literature review, development of the neural network model and the corresponding training as well as testing on real data. The first chapter of the research describes the current problems of teenagers' mental health and theoretical foundations of neural networks in detail. It also focuses on the existing approaches to mental health risk assessment. The second chapter presents the results of the model development, including data preprocessing and optimization of ADAM algorithm. This chapter is devoted to analyzing the performance metrics as well. The third chapter dwells on implementing the model into a mobile application and testing of this model in an educational institution.

In conclusion, it should be emphasized that the developed software has showed 92% of accuracy when predicting the mental health risks and has been approved for practical use.

The work is of interest for a wide circle of readers involved in psychology, machine learning and healthcare software development.

Оглавление

Введение

Психическое здоровье является важнейшей составляющей общего благополучия личности, особенно в подростковом возрасте, когда закладываются устойчивые поведенческие паттерны и эмоциональные реакции [4][21]. Традиционно диагностика и мониторинг ментального здоровья осуществляются посредством опросников и клинических интервью, что требует значительных временных и ресурсных затрат. Нейросетевые модели представляют собой перспективный инструмент для автоматизации анализа данных и раннего выявления рисков ухудшения ментального здоровья, что делает их востребованными в образовательных учреждениях.

Современная образовательная среда характеризуется высокими учебными нагрузками и стрессовыми факторами, что повышает риск возникновения проблем с ментальным здоровьем у подростков [5][16]. Своевременное обнаружение и поддержка становятся критически важными для предотвращения серьезных последствий. Внедрение автоматизированных систем на основе нейросетей позволяет быстро и эффективно обрабатывать большие массивы данных, что открывает новые возможности для раннего вмешательства и помощи учащимся [1].

Несмотря на наличие ряда научных исследований и разработок, направленных на применение нейросетей в медицинской и психологической практике, многие из них сосредоточены на взрослой аудитории, оставляя особенности подросткового возраста менее изученными [15]. Настоящее исследование направлено на устранение этого пробела путем разработки специализированного программного обеспечения для прогнозирования рисков ухудшения ментального здоровья среди старших школьников.

В рамках данной выпускной квалификационной работы было решено разработать информационную систему в формате мобильного приложения.

Разработка была реализована под платформу Android. Такой формат выбран в связи с высокой распространённостью мобильных устройств данного типа среди подростков, что делает систему более доступной и удобной для целевой аудитории.

Объектом исследования стал процесс разработки программного обеспечения для прогнозирования ментального здоровья подростков, предметом исследования – нейросетевые модели и алгоритмы, обеспечивающие этот прогноз. Методология исследования включила анализ научной литературы, статистическую обработку данных, моделирование и программирование.

Целью данной работы является создание программного обеспечения, использующего нейросетевые модели для прогнозирования рисков ухудшения ментального здоровья у детей старшего подросткового возраста.

Для достижения этой цели были определены следующие задачи:

- изучить методы прогнозирования ментального здоровья;
- проанализировать нейросетевые модели, применимые в психологии и медицине;
- подготовить и адаптировать данные для обучения нейросети;
- разработать критерии оценки эффективности модели;
- спроектировать и реализовать программное обеспечение;
- протестировать и оценить точность модели на реальных данных.

Научная новизна данной работы заключается в применении нейросетевых технологий для прогнозирования рисков ухудшения ментального здоровья именно среди подростков. Теоретическая значимость работы связана с углублением понимания возможностей использования нейросетей в психологии и образовании. Практическая ценность выражается в создании инструмента, позволяющего педагогам и психологам своевременно выявлять изменения в состоянии учеников, улучшать качество образовательного процесса и обеспечивать необходимую поддержку студентам.

Глава 1 Разработка спецификации к программному обеспечению

1.1 Основные функции программного обеспечения

В рамках данного исследования была разработана мобильная платформа для мониторинга ментального здоровья подростков, ориентированная на выявление рисков ухудшения психоэмоционального состояния и предоставление персонализированных рекомендаций. Она использует мультимодальную нейросетевую модель для прогнозирования рисков психоэмоционального состояния на основе анализа поведения, активности и результатов тестирования [2]. Основные функциональные возможности программного обеспечения можно условно разделить на четыре категории: обработка данных, управление информационными ресурсами, взаимодействие с пользователем и автоматизация процессов.

Ключевым элементом является обработка данных, охватывающая сбор, хранение, анализ и интерпретацию информации. Мобильное приложение на базе Android (Kotlin) собирает данные о действиях пользователя: результаты прохождения тестов, история прочитанных статей, сообщения в чате с ботом или психологом. Все данные сохраняются в Firebase Firestore в структурированном виде. На основе собранной информации приложение инициирует отправку агрегированных данных на сервер, где нейросетевая модель анализирует текстовые и числовые признаки и классифицирует уровень риска (низкий, средний, высокий). Прогноз и соответствующие рекомендации возвращаются на клиента и визуализируются в профиле пользователя. В случае, если ситуация становится критической, появляется рекомендация записаться на приём к штатному специалисту. Таким образом, обеспечивается непрерывный цикл: сбор данных, предобработка, анализ, визуализация результата и рекомендаций.

На рисунке 1 представлена схема, иллюстрирующая последовательность обработки информации в системе — от сбора данных пользователем до формирования рекомендаций нейросетевой моделью.

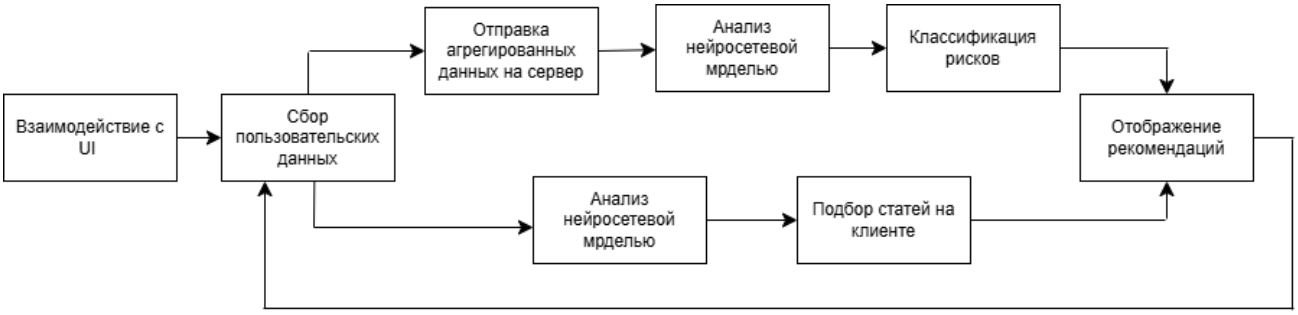

Рисунок 1 – Последовательность обработки информации в системе

Управление информационными ресурсами реализовано через взаимодействие с базой данных Firebase и REST API сервера. Обмен данными осуществляется через защищённые HTTPS-соединения. Вся информация обрабатывается с учетом требований безопасности и конфиденциальности: применяется анонимизация идентификаторов пользователей, при первом запуске запрашивается согласие на обработку данных.

Помимо информационных, программное обеспечение также способствует управлению другими видами ресурсов, включая человеческие, организационные и материальные. Например, система может использоваться психологами и педагогами для планирования консультаций, учета обращений, анализа динамики состояния учащихся и формирования отчётности. Эффективное управление данными ресурсами позволяет снизить издержки, повысить прозрачность работы и ускорить процессы принятия решений.

Взаимодействие с пользователями реализовано через удобный мобильный интерфейс, разработанный на языке Kotlin с использованием архитектуры MVVM и адаптированный под целевую аудиторию. Пользователь может

проходить психологические тесты, читать рекомендованные статьи, взаимодействовать в чате с ботом или психологом. Система отслеживает реакцию пользователя, и на основе этих данных формирует индивидуальные рекомендации. В профиле отображаются уровни риска и советы по улучшению психоэмоционального состояния. При повышенном уровне тревожности приложение может отправить уведомление родителям или психологу (при наличии разрешения).

На рисунке 2 представлена диаграмма вариантов использования приложения для категорий пользователь, психолог и родитель.

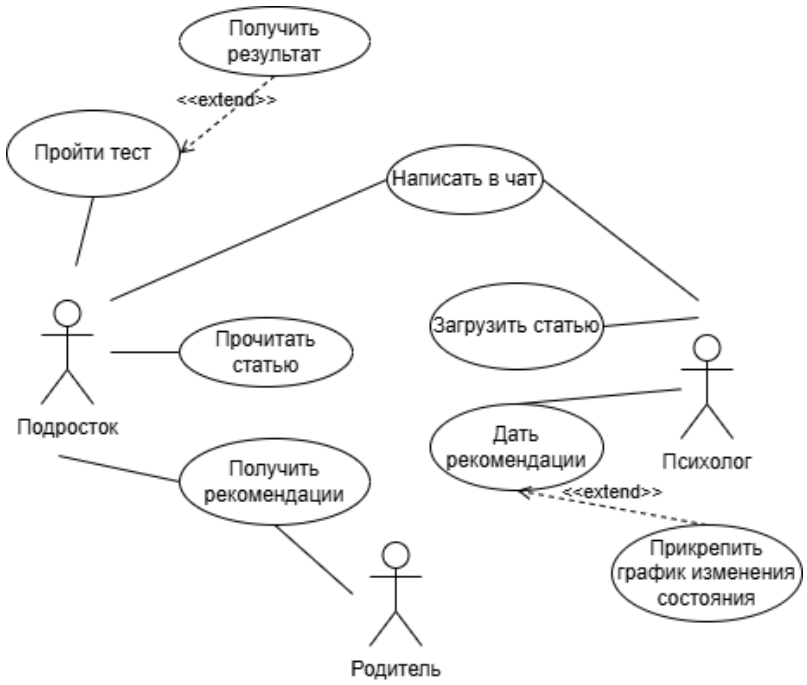

Рисунок 2 – Диаграмма вариантов использования

Удобство взаимодействия обеспечивается благодаря качественно реализованному пользовательскому интерфейсу. Интерфейс (UI) интуитивно понятен, а пользовательский опыт (UX) адаптирован под особенности подростковой аудитории. Простой и доступный интерфейс упрощает работу с

системой, минимизирует ошибки и повышает общую эффективность её использования.

Автоматизация процессов реализована как на стороне сервера, так и на стороне клиента. На серверной стороне с помощью нейросетевой модели на базе BERT (в связке с TensorFlow) происходит прогнозирование риска на основе текстов сообщений, результатов тестов и истории активности. Модель формирует персонализированные рекомендации, учитывая уровень риска. Однако часть автоматизации реализована непосредственно на клиенте: в частности, подбор рекомендованных статей осуществляется локально на основе поведения пользователя при свайпах вправо и влево. Это позволяет обеспечить персонализацию контента в реальном времени без необходимости постоянных обращений к серверу. Такой гибридный подход повышает отзывчивость системы и снижает нагрузку на серверную инфраструктуру.

Автоматизация рутинных задач — таких как оценка тестов, уведомления, генерация советов — позволяет существенно сэкономить время, снизить вероятность ошибок и повысить устойчивость системы при масштабировании.

Безопасность информации является важнейшим элементом системы. Программное обеспечение реализует многоуровневую систему защиты, представленную на рисунке 3: обязательную аутентификацию пользователя, разграничение прав доступа (авторизация), шифрование всех передаваемых и хранимых данных, а также ведение логов действий. Это особенно важно в условиях роста угроз кибербезопасности и при работе с чувствительными персональными данными.

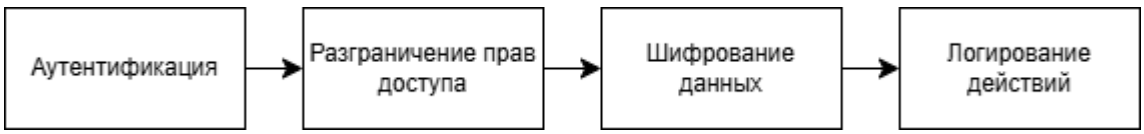

Рисунок 3 – Модель многослойной безопасности

Разработанное программное обеспечение объединяет в себе функции интеллектуального анализа, интерактивного взаимодействия с пользователем и адаптивного контент-менеджмента. Архитектура системы ориентирована на автономную, безопасную и масштабируемую работу в условиях реальной пользовательской активности, что делает её применимой в образовательной и профилактической практике.

1.2 Обзор существующих программных решений, направленных на поддержку ментального здоровья подростков

В рамках раздела были рассмотрены актуальные программные продукты, направленные на поддержку ментального здоровья подростков, с акцентом на использование нейросетевых технологий. Основное внимание уделялось продуктам, работающим на российском рынке, а также соответствующим специфике исследуемой темы — использованию искусственного интеллекта, в частности нейросетевых моделей, в сфере психологического консультирования и профилактики. Анализ проводился по следующим критериям: функциональность, стоимость, удобство использования, безопасность и поддержка нейросетевых технологий.

Результаты сравнительного анализа представлены в таблице 1.

Таблица 1 - Сравнительный анализ существующих программных решений

Критерии	MentalTeen	Ясно	ЯМогу
Функциональность	4	3	3
Стоимость	5	2	3
Удобство использования	4	4	4
Безопасность	4	5	5
Поддержка нейросетевых технологий	4	2	2

Оценка проводилась по пятибалльной шкале, где 1 балл соответствовал минимальному уровню реализации по данному параметру, а 5 — максимально возможному на текущем этапе развития технологий. При выставлении оценок учитывались не только пользовательские обзоры и технические описания, но и данные из официальных источников: сайт разработчика, отзывы пользователей и сертифицированных специалистов в области психологии.

При детальном рассмотрении каждого продукта можно выделить особенности.

«MentalTeen» — мобильное приложение, ориентированное на образовательные учреждения и подростков. Использует полуавтоматическую классификацию рисков, но пока не обладает полноценной интеграцией с ИИ-моделями. Тем не менее, имеет интуитивно понятный интерфейс и высокую оценку по параметру доступности.

«Ясно» и «ЯМогу» — российские онлайн-сервисы для консультации с психологами. Основной акцент делается на живое общение с сертифицированными специалистами, при этом автоматизация и ИИ-технологии используются ограниченно (преимущественно для подбора специалиста или напоминаний). Тем не менее, эти сервисы демонстрируют высокий уровень поддержки и конфиденциальности.

Наиболее распространенным методом прогнозирования ментального здоровья является использование психологических тестов и опросников. Они позволяют выявить предрасположенность к тревожным расстройствам, депрессии и другим психическим нарушениям. Одним из широко применяемых инструментов является тест школьной тревожности Филлипса, который оценивает уровень тревожности у подростков в образовательной среде [7]. Также в диагностике используются шкала депрессии Бека и методика Спилбергера для оценки тревожности. Эти методы удобны для массового применения, однако их

эффективность зависит от искренности ответов учащихся, что может снижать достоверность полученных данных.

Другим методом является наблюдение и экспертная оценка со стороны психологов и педагогов [8]. В процессе мониторинга специалисты анализируют поведение подростков, их академическую успеваемость, уровень социальной адаптации, наличие конфликтов и другие индикаторы возможных психологических проблем. Этот метод позволяет получать важную информацию о состоянии учащихся, но он требует высокой квалификации специалистов и не всегда дает объективную картину, так как субъективное восприятие наблюдателя может вносить погрешности.

Для более точного прогнозирования психического состояния подростков используются статистические и математические модели, например, регрессионный анализ. Он позволяет учитывать влияние различных факторов, таких как социально-экономические условия, семейная обстановка, уровень стресса и психофизиологические показатели [19]. В исследовании, проведенном в Иркутске, применялся многофакторный регрессионный анализ, с помощью которого прогнозировались показатели психического здоровья подростков в условиях социальных и экономических изменений. Такой метод позволяет учитывать множество переменных, влияющих на психическое состояние подростков, однако его применение требует наличия больших объемов данных и сложных расчетов.

Сравнительный анализ перечисленных методов прогнозирования изменений в ментальном здоровье приведен в таблице 2.

В последние годы все большее внимание уделяется использованию нейросетевых моделей и методов машинного обучения для прогнозирования психического состояния подростков [9]. Искусственные нейронные сети способны анализировать большие объемы данных, выявлять скрытые закономерности и строить прогнозы на основе различных параметров. Например,

в исследованиях, посвященных применению машинного обучения в психиатрии, модели на основе глубоких нейросетей демонстрируют высокую точность при выявлении признаков депрессии и тревожных расстройств. Данные модели анализируют текстовые сообщения, голосовые параметры, а также информацию о поведении подростков в социальных сетях, что позволяет выявлять ранние признаки психологических проблем и предлагать превентивные меры [26][27].

Таблица 2 - Сравнительный анализ методов прогнозирования ментального здоровья

Критерий	Психологические тесты	Экспертная оценка	Статические модели	Нейросетевые модели
Точность	70-80%	85-90%	75-85%	90-95%
Автоматизация	Низкая	Низкая	Средняя	Полная
Масштабируемость	Ограниченная	Низкая	Средняя	Высокая
Интерпретируемость	Высокая	Высокая	Средняя	Низкая (черный ящик)
Требования к данным	Анкеты, клинические данные	Наблюдения, интервью	Структурированные данные	Разнородные данные (текст, результаты тестов, активность)
Скорость	Дни	Недели	Часы	Минуты

На основании приведённых данных можно заключить, что современный рынок цифровых решений для ментального здоровья демонстрирует устойчивый тренд к интеграции искусственного интеллекта и персонализированных рекомендаций. Решения, использующие нейросетевые технологии, имеют существенные преимущества в части масштабируемости, адаптивности и скорости реагирования на изменения в поведении пользователя. Однако они также предъявляют более высокие требования к инфраструктуре и защите данных. Обзор существующих решений позволяет утверждать, что выбор

подходящего программного продукта должен основываться на комплексной оценке — от доступности до технической архитектуры.

1.3 Требования к функционалу программного обеспечения

Функциональные возможности приложения выстраиваются вокруг трех ключевых задач: сбор поведенческих и текстовых данных, их обработка с помощью нейросетевых моделей и отображение результатов в виде динамической оценки уровня риска. Все действия пользователя становятся частью датасета, который обрабатывается моделью на стороне сервера.

В контексте регистрации и аутентификации особое внимание уделяется безопасности подростков: предусмотрены механизмы двойной авторизации для родителей и режим «безопасного входа», позволяющий ограничить доступ к персональной информации.

Для сбора данных приложение реализует следующие сценарии: прохождение психометрических опросов, сбор данных о поведении (частота и длительность использования, взаимодействие с контентом). Каждое из этих действий инициирует запуск соответствующих микросервисов.

В будущих версиях планируется реализация пассивного мониторинга — с разрешения пользователя приложение сможет отслеживать активность в определённых временных зонах (например, поздние часы бодрствования, сниженная активность в течение дня), что поможет формировать более точные прогнозы и выявлять признаки выгорания или апатии.

Визуализация данных реализуется через график, отображающий динамику изменения риска (раз в неделю или чаще при критических событиях). Это особенно важно для подростков, чье восприятие информации значительно зависит от цифровых форматов представления данных [6]. Дополнительно пользователю предлагаются индивидуальные рекомендации: отдых, общение,

консультация со специалистом и т.д. В случае превышения порогового значения риска происходит автоматическое уведомление родителя или штатного психолога.

Для поддержки непрерывной работы системы в условиях слабого интернета внедряется кэширование ключевых компонентов модели на стороне клиента с последующей синхронизацией при восстановлении соединения. Такая реализация особенно актуальна для регионов с нестабильной связью или в ситуациях, когда пользователь ограничен в доступе к сети (например, в поездке и т.д.).

В условиях ограниченного доступа к сети приложение должно обеспечивать базовый функционал в офлайн-режиме — доступ к локально сохранённым упражнениям, техникам релаксации и ранее сформированным рекомендациям. Это делает систему более надёжной и поддерживает ощущение постоянного присутствия и поддержки.

Безопасность в контексте ментального здоровья подростков выходит на первый план. Шифрование данных и соблюдение стандартов хранения медицинской информации являются обязательными. Родитель может получить доступ только к агрегированным данным, не нарушающим приватность подростка. Таким образом сохраняется баланс между этикой, эффективностью поддержки и правом на личное пространство.

Для пользователей, находящихся в кризисном состоянии, предусмотрена «красная кнопка» — экстренное обращение к доверенному взрослому или психологу по заранее согласованному сценарию. Это позволяет ввести механизм превентивного вмешательства — когда риск ещё не перешёл в кризисную фазу, но уже требует внимания.

Макеты экранов приложения, иллюстрирующие основные разделы интерфейса, представлены на рисунке 4.

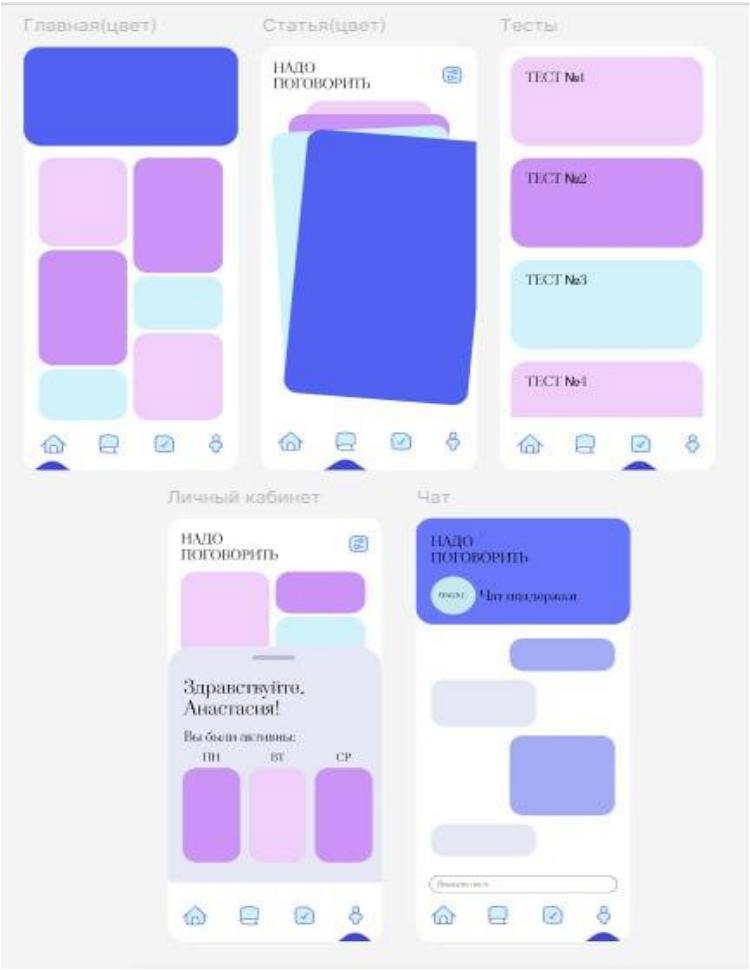

Рисунок 4 – Модель экранов приложения

Пользовательский интерфейс разрабатывается с учётом принципов цифрового благополучия. Экраны выполнены в нейтральных, успокаивающих цветах, присутствуют элементы «эмоциональной разгрузки» — дыхательные упражнения, советы, раздел с позитивными историями. Интерфейс для подростка отличается от интерфейса для родителя или специалиста: он более современный и сфокусирован на поддержке, а не на аналитике. Особое внимание уделяется обратной связи. Пользователи (как подростки, так и взрослые) могут оставить отзыв о работе системы, указать, насколько рекомендации были полезны, и предложить изменения. Эта информация используется как для улучшения UX,

так и для адаптации нейросетевых алгоритмов на основе реального пользовательского опыта.

Также предусмотрена система мягкого уведомления, предупреждающая о повышении уровня риска в щадящей форме, без прямых терминов, способных усилить тревожность. Это особенно важно при работе с уязвимой аудиторией. Например, вместо формулировки «у вас депрессия» используется «мы заметили, что вы могли бы чувствовать себя не в лучшей форме — вот что может помочь».

Интерфейс постоянно дорабатывается с учётом анонимизированных отзывов пользователей, анализа путей навигации (UX heatmaps) и частоты использования функций. Это позволяет повысить вовлечённость и минимизировать фрустрацию при использовании системы. Пользовательский путь строится на основе аналитики поведения, что позволяет сделать взаимодействие с системой более интуитивным.

В перспективе планируется внедрение системы достижений и мотивационных механик (геймификация), направленных на формирование привычки заботы о ментальном здоровье: например, за регулярное прохождение тестов, выполнение упражнений и участие в позитивных активностях. Элементы геймификации помогают преодолеть сопротивление и формируют устойчивую модель поведения, схожую с гигиеническими привычками.

В целях систематизации требований к разрабатываемому программному обеспечению и обеспечения полноты охвата всех ключевых аспектов, в работе применена модель FURPS+, представленная в таблице 3.

Таблица 3 - Модель требований FURPS+

Категория	Конкретизация под тему приложения
Functionality	Опросники, анализ текста, прогноз уровня риска, уведомления, рекомендации
Usability	Адаптивный интерфейс для подростков, родителей, специалистов
Reliability	Работа модели при неполных данных, устойчивость при нестабильном соединении

Продолжение таблицы 3

Категория	Конкретизация под тему приложения
Performance	Быстрый отклик сервера при анализе рисков, асинхронная обработка событий
Supportability	Поддержка локализации, система помощи, актуализация модели через переобучение
+	Этические ограничения, защита персональных данных, соблюдение возрастных норм

В дальнейшем данная модель послужит основой для разработки тест-кейсов, оценки качества реализации и планирования этапов сопровождения разработанной системы.

В результате анализа и формализации требований к разрабатываемому приложению были определены ключевые направления, обеспечивающие его эффективность, устойчивость и соответствие ожиданиям пользователей. Установлено, что успешная реализация системы мониторинга ментального здоровья подростков требует чёткого разграничения функциональных и нефункциональных требований, каждая из которых играет критически важную роль в общем контексте работы приложения. На основе структурированных требований сформировано чёткое представление о функциональной модели и пользовательском опыте приложения. Это создаёт основу для последующих этапов, позволяя обеспечить её эффективность как в образовательной, так и в профилактической практике. Более того, данная архитектура может быть масштабирована и адаптирована для использования в университетской, корпоративной и семейной среде, с учётом соответствующих возрастных и профессиональных контекстов.

Глава 2 Разработка программного обеспечения для мониторинга ментального здоровья подростков

2.1 Проектирование программного обеспечения для мониторинга ментального здоровья подростков

Этап проектирования программного обеспечения является одним из самых критических в процессе разработки, так как именно на этом этапе формируются архитектура и основные компоненты системы. В рамках разрабатываемого проекта особое внимание уделяется согласованности архитектуры и корректной интеграции всех компонентов.

В архитектурном проектировании приложения была выбрана клиент-серверная модель с распределением ответственности между клиентской частью (мобильное приложение) и серверной (обработка данных, обучение и использование нейросетевого алгоритма), представленная на рисунке 5.

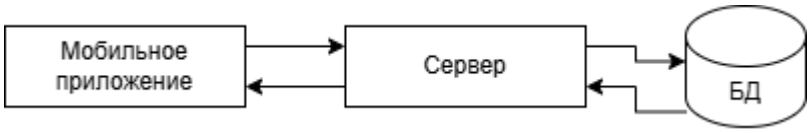

Рисунок 5 – Клиент-серверная модель

Мобильный клиент отвечает за сбор пользовательских данных, отображение интерфейса и локальную генерацию рекомендаций, в то время как серверная часть, написанная на Python, занимается обучением нейросетевой модели, анализом поведения пользователей и формированием глобальных стратегий рекомендаций. Firebase используется как облачная платформа для хранения пользовательских данных, событий и логов, а также для синхронизации информации между клиентом и сервером в реальном времени.

Компонентная диаграмма приложения, представленная на рисунке 6, демонстрирует взаимодействие между тремя основными слоями.

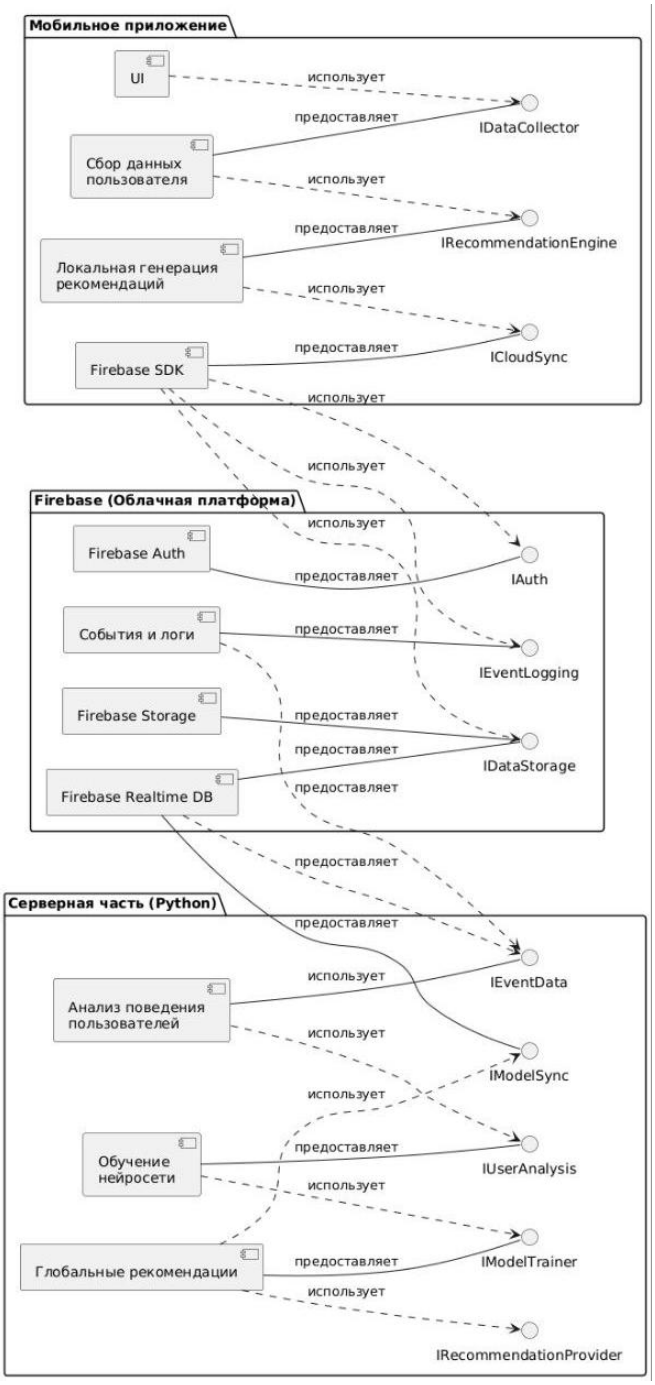

Рисунок 6 – Диаграмма компонентов

Мобильное приложение включает компоненты пользовательского интерфейса, сбора пользовательских данных и локальной генерации рекомендаций. Оно взаимодействует с Firebase SDK, который обеспечивает доступ к облачным сервисам. Компоненты обмениваются между собой данными через интерфейсы: UI использует интерфейс сбора данных, модуль сбора данных использует интерфейс генерации рекомендаций, а локальный генератор рекомендаций обращается к интерфейсу облачной синхронизации.

Серверная часть реализует функции анализа пользовательского поведения, обучения нейросетевой модели и формирования глобальных стратегий рекомендаций. Эти модули используют данные, хранящиеся в Firebase, и обмениваются результатами через интерфейсы анализа данных, подготовки модели и синхронизации результатов с облаком.

Firebase выполняет роль хранилища и транспортного канала между клиентом и сервером. Он включает в себя аутентификацию, базу данных, хранилище файлов и логирование событий. Все компоненты используют или предоставляют соответствующие интерфейсы, такие как авторизация, хранение данных и регистрация событий.

Несмотря на то, что в классическом UML требуемые интерфейсы отображаются как полуокружности (так называемые «розетки»), на диаграмме они визуализируются как пунктирные стрелки с направлением зависимости. Это ограничение инструмента визуализации: PlantUML упрощает графическое представление, заменяя розетки стрелками. Стрелка от одного компонента к интерфейсу означает, что компонент использует (требует) этот интерфейс, тогда как соединение интерфейса с другим компонентом указывает, что тот его предоставляет. Таким образом, диаграмма позволяет проследить как потоки данных, так и архитектурные зависимости между частями приложения.

В рамках проектирования пользовательского интерфейса были разработаны макеты экранов, ориентированных на простой и понятный UX. Все

элементы расположены с учётом частоты взаимодействия пользователей и фокусируются на быстром доступе к интересующему контенту.

Проектирование базы данных, представленное на рисунке 7, ориентировано на структуру NoSQL, соответствующую архитектуре Firebase.

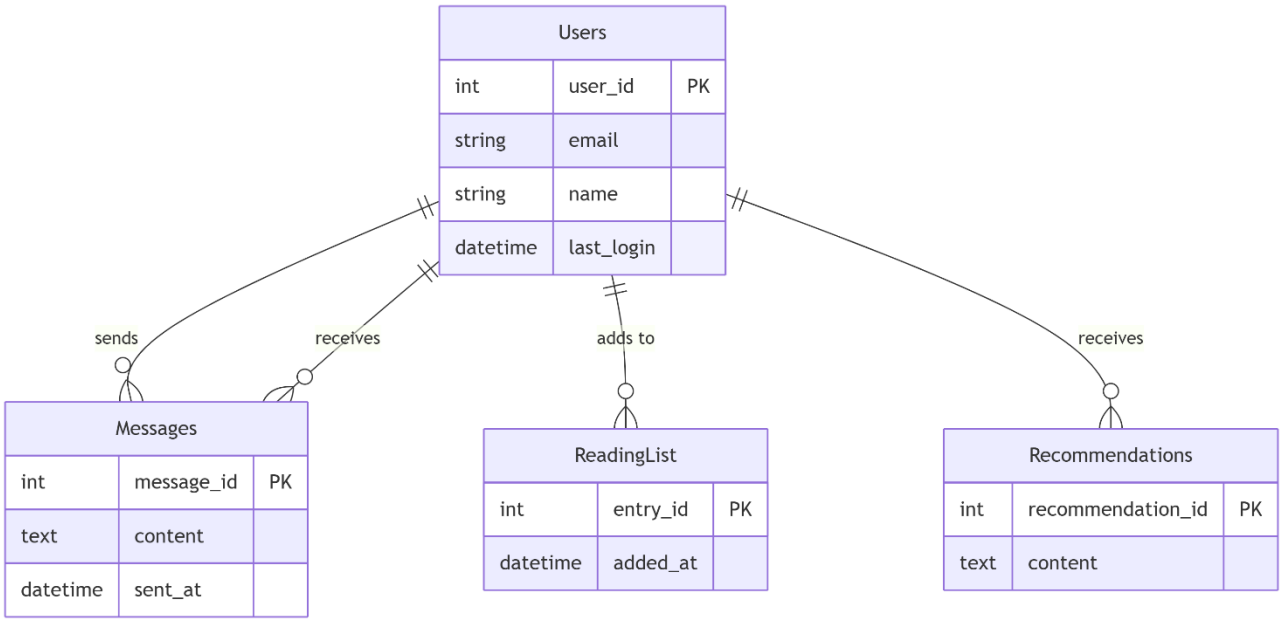

Рисунок 7 – Фрагмент диаграммы базы данных

Полная диаграмма представлена в Приложении А.

Структура хранения представлена в виде иерархии коллекций и документов, где каждая сущность (например, пользователь, статья, взаимодействие) представляет собой документ с вложенными атрибутами. Несмотря на то, что база данных реализована в NoSQL-архитектуре (Firebase), для удобства представления данные структурированы в виде логических таблиц, каждая из которых соответствует коллекции документов.

Таблица Users (таблица 4) содержит данные о пользователях, включая уникальные идентификаторы, электронную почту, пароль, роль (пользователь или психолог), имя, дату рождения и дату последнего входа. Эти пользователи

могут отправлять и получать сообщения, проходить тесты, а также получать рекомендации, которые формируются на основании их результатов тестирования.

Таблица 4 – Таблица Users

Поле	Тип данных	Описание
user_id	INTEGER	Уникальный идентификатор пользователя.
Email	VARCHAR(255)	Электронная почта пользователя (уникальное).
password_hash	VARCHAR(255)	Хэш пароля.
Role	ENUM	Роль пользователя (user, psychologist).
name	VARCHAR(255)	Имя пользователя.
birth_date	DATE	Дата рождения.
last_login	DATETIME	Время последнего входа в систему.

Таблица Messages используется для хранения сообщений, отправленных и полученных пользователями. У каждого сообщения есть уникальный идентификатор, информация об отправителе и получателе, тип сообщения, текст, время отправки и ссылка на вложение, если оно есть.

Таблица Articles, представленная в таблице 5, хранит статьи, которые создаются пользователями. Она содержит уникальные идентификаторы, заголовки, текст описания, содержание статьи, категорию, дату публикации и ссылку на автора через поле author_id.

Таблица 5 - Таблица Articles

Поле	Тип данных	Описание
article_id	INTEGER	Уникальный идентификатор статьи.
author_id	INTEGER	ID автора (ссылка на Users.user_id).
Title	VARCHAR(255)	Заголовок статьи.
description	TEXT	Краткое описание статьи.
content	TEXT	Полный текст статьи.
category	VARCHAR(255)	Категория статьи.
publish_date	DATETIME	Дата публикации.

Таблица Tests предназначена для хранения информации о тестах, включая их уникальные идентификаторы, названия и описания. Эти тесты связаны с вопросами через таблицу Test_Questions. Эта связь позволяет добавлять или удалять вопросы в тестах, а также анализировать результаты пользователей.

Таблица Test_Questions содержит вопросы для тестов. Каждый вопрос связан с конкретным тестом через test_id. Таблица включает текст вопроса, возможные варианты ответов, правильный ответ и тип вопроса. (множественный выбор или открытый вопрос).

Таблица Test_Results фиксирует результаты тестов. В ней хранится информация об уникальном идентификаторе результата, идентификаторе пользователя, который прошёл этот тест, идентификатор самого теста, набранные баллы и дату завершения теста.

Таблица Reading_List содержит записи о статьях, которые пользователи добавили в свой список для чтения. В ней хранятся ссылки на пользователей и статьи, а также время добавления.

Таблица Recommendations сохраняет рекомендации, которые генерируются на основе результатов тестирования. Каждая запись включает текст рекомендации, привязку к пользователю и тесту, а также дату создания. Рекомендации могут быть адаптированы, если пользователь повторно проходит тест, и сохраняются в базе данных для дальнейшего использования.

Несмотря на отсутствие внешних ключей в NoSQL-структуре, связи между сущностями реализованы через уникальные идентификаторы, позволяющие организовать логически связанные коллекции и документы. Такая организация позволяет не только оперативно обрабатывать и анализировать большие объемы данных, но и контролировать активность пользователей, результаты их тестов и взаимодействие с системой. Благодаря этим связям система может формировать персонализированные рекомендации для пользователей.

На уровне бизнес-логики основное внимание уделено процессу

формирования рекомендаций. Здесь выделяются два сценария: локальный подбор статей на клиенте и серверный анализ с помощью нейросети, позволяющий формировать персонализированные рекомендации на основе анализа больших массивов данных. Все коммуникации между клиентом и сервером реализованы через REST API и события, синхронизируемые через Firebase Realtime Database или Firestore.

На рисунке 8 представлена диаграмма, которая отражает процесс формирования рекомендаций в мобильном приложении.

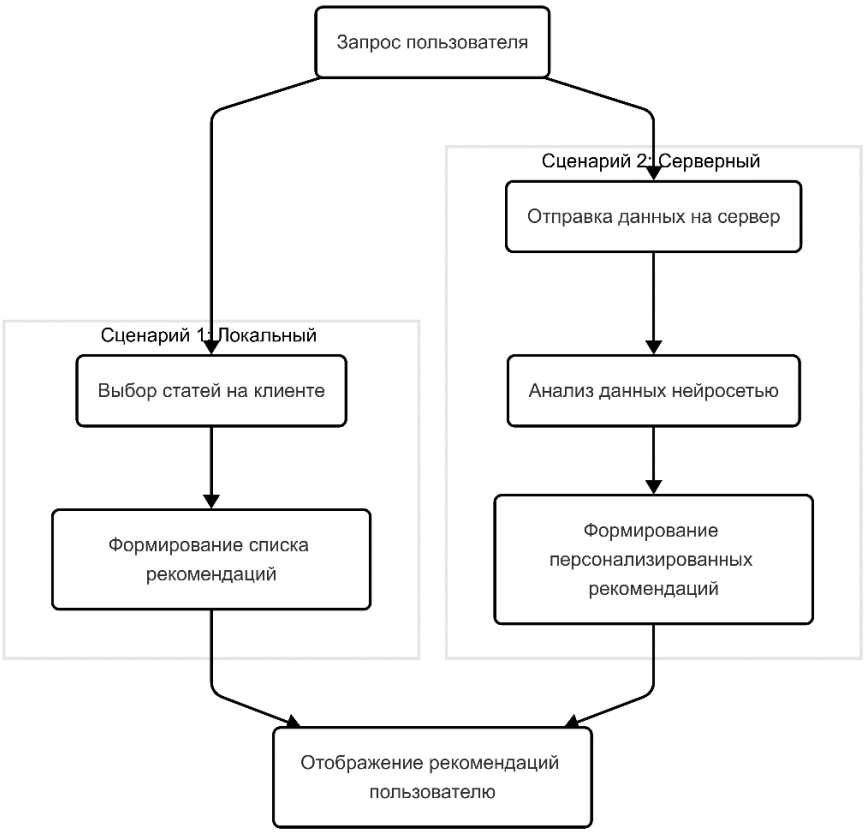

Рисунок 8 – Сценарии формирования рекомендаций

Проектирование программного обеспечения охватывает все ключевые аспекты архитектуры, интерфейса, данных, логики и взаимодействия.

Тщательное продумывание этих компонентов на этапе проектирования позволяет создать устойчивую, масштабируемую и удобную в использовании систему.

2.2 Выбор технологий и инструментов для разработки

В случае разрабатываемого мобильного приложения с функцией персонализированных рекомендаций статей необходимо было учесть специфику архитектуры (клиент-серверная модель), особенности взаимодействия между компонентами, а также интеграцию с облачными сервисами и нейросетевыми алгоритмами. Подход к выбору технологий основывался на комплексной оценке их соответствия требованиям, устойчивости в промышленной эксплуатации, скорости разработки и доступности поддержки. При выборе технологий учитывались особенности цифрового поведения российской молодежи и специфика влияния цифровизации на их психологическое состояние [10].

Процесс выбора начался с анализа языков программирования, наиболее подходящих для выполнения задач проекта. Для клиентской части было решено использовать Kotlin, так как он является официально поддерживаемым языком для разработки Android-приложений и предлагает высокую производительность, лаконичный синтаксис и хорошую интеграцию с современными библиотеками UI и API. Серверная часть реализована на Python, что обусловлено широким набором библиотек для анализа данных и машинного обучения, таких как scikit-learn, pandas и TensorFlow. Python также обеспечивает простую интеграцию с Firebase через REST API или SDK.

Следующим шагом стал выбор архитектурной платформы. Было решено использовать Firebase в качестве облачного связующего звена между клиентом и сервером. Firebase Realtime Database и Firestore обеспечивают быстрое хранение и синхронную доставку пользовательских данных, событий, логов, результатов

тестов и предсказаний моделей. Firestore был выбран в качестве основной базы данных благодаря его NoSQL-структуре, масштабируемости и нативной поддержке оффлайн-доступа на мобильных устройствах.

Для реализации прогноза рисков после прохождения пользователями психологических тестов была выбрана архитектура с отложенной обработкой. Результаты тестов, сохранённые в Firestore, анализируются серверной частью на Python, где с помощью обученной нейросети выполняется классификация уровня риска. Модель регулярно обновляется на основе новых данных, поступающих с клиента.

Также особое внимание было уделено выбору инструментов разработки, тестирования и управления проектом. Разработка приложения велась в Android Studio, предоставляющей расширенные средства для отладки, эмуляции и анализа пользовательского интерфейса. Построение и обучение модели производилось в Jupyter Notebook, что удобно для прототипирования и визуализации данных.

Для тестирования бизнес-логики и функционала были выбраны JUnit для Kotlin-клиента и pytest для серверной части на Python. Для нагрузочного тестирования серверной логики применяется Locust, позволяющий моделировать пользовательские сценарии взаимодействия.

В таблице 6 представлена сводная информация по выбранным технологиям.

Таблица 6 – Основные выбранные технологии

Компонент	Выбранная технология	Причина выбора
Клиентское приложение	Kotlin, Android Studio	Высокая производительность, официальная поддержка
Серверная часть	Python	Поддержка ML-библиотек, гибкость, простота API

Продолжение таблицы 6

Компонент	Выбранная технология	Причина выбора
База данных	Firebase Firestore	Масштабируемость, поддержка оффлайн-режима, NoSQL
Хранилище логов/ивентов	Firebase Realtime Database	Быстрая синхронизация данных
ML-библиотеки	Scikit-learn, TensorFlow	Обучение и прогнозирование рисков
Инструменты разработки	Android Studio, IDLE Python	Среды, адаптированные под платформы
Контроль версий	GitHub	Совместная работа, отслеживание изменений
Тестирование	JUnit, pytest, Locust	Проверка логики, функциональности, нагрузки

Кроме этого, был проведён сравнительный анализ возможных альтернативных языков программирования и фреймворков, но с учётом специфики задач предпочтение было отдано связке Kotlin–Python. На этапе выбора фреймворков также рассматривались React Native и Flutter, однако они были отклонены из-за требований к глубокому взаимодействию с Android API и необходимости нативной работы с Firebase SDK.

Для предварительной обработки данных использовались следующие инструменты:

- Pandas (v1.3.5) - библиотека для обработки структурированных данных с поддержкой операций фильтрации, группировки и агрегации [11];
- NumPy (v1.21.0) - обеспечивал векторные операции и линейную алгебру для работы с многомерными массивами [12];
- Scikit-learn (v0.24.2) - предоставлял инструменты для нормализации данных и feature engineering.

Первым важным этапом стала подготовка данных — устранение шумов, заполнение пропусков и стандартизация значений [17]. Для этих целей использовались библиотеки Pandas и NumPy на языке Python. Pandas

предоставила удобные средства для манипуляции с табличными данными, такие как фильтрация, сортировка и агрегация, что значительно упростило работу с сырыми данными. NumPy же обеспечил мощную функциональность для выполнения векторных и матричных вычислений, что оказалось незаменимо при обработке многомерных наборов данных [13]. Схема процесса подготовки и обработки данных представлена на рисунке 9.

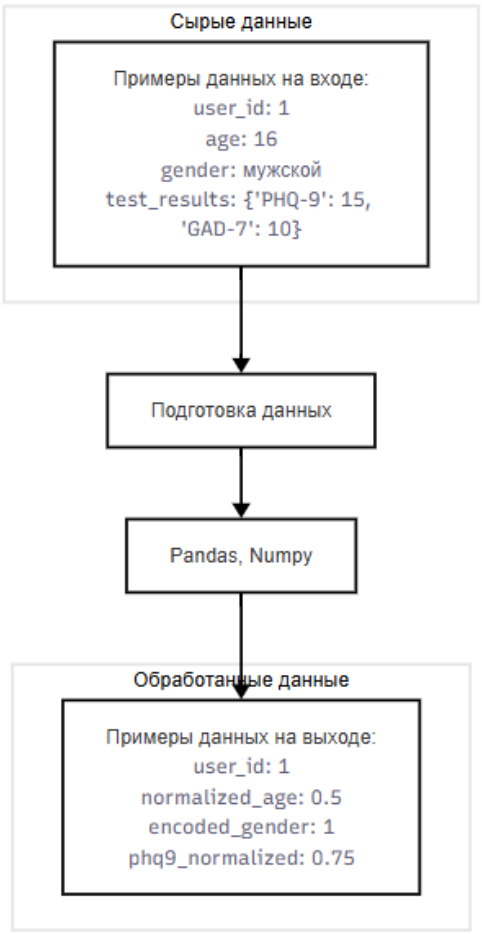

Рисунок 9 – Пример обработки сырых данных

Сочетание Pandas и NumPy позволило создать эффективный процесс предобработки данных, обеспечивающий высокую точность и надежность последующих прогнозов.

В таблице 7 представлены различные методы предварительной обработки данных, их преимущества и недостатки.

Таблица 7 - Сравнение методов предварительной обработки данных

Метод	Описание	Преимущества	Недостатки
Pandas	Библиотека для обработки табличных данных в Python	Простота использования, гибкость фильтрации и агрегации, интеграция с другими инструментами	Ограничена работой с табличными структурами
NumPy	Библиотека для работы с многомерными массивами	Высокая скорость обработки данных, широкий набор математических операций	Требует определенных навыков работы с массивами
Scikit-learn	Инструменты для обработки данных	Встроенные алгоритмы предобработки. Удобство использования с моделями машинного обучения	Ограниченные возможности для работы с текстовыми данными
OpenCV	Библиотека для обработки изображений	Поддержка различных форматов данных и высокая производительность	Специализирована на обработке изображений, менее удобна для табличных данных

После этапа предварительной обработки осуществлялся выбор и адаптация метода обучения нейросетевой модели. Основными инструментами для построения и тренировки нейросети стали TensorFlow и Keras.

Сравнение существующих платформ для глубокого обучения, между которыми происходил выбор, подробно описаны в таблице 8.

Таблица 8 - Сравнение платформ для глубокого обучения

Платформа	Описание	Преимущества	Недостатки
TensorFlow	Мощная библиотека для глубокого обучения, разработанная Google	Гибкость и масштабируемость, поддержка распределённых вычислений, большое сообщество	Высокий порог вхождения

Продолжение таблицы 8

Платформа	Описание	Преимущества	Недостатки
Keras	Высокоуровневый API для TensorFlow, упрощающий разработку нейросетей	Простота использования и быстрое прототипирование моделей, дружелюбный синтаксис	Меньший контроль над процессом обучения
Scikit-learn	Инструмент для машинного обучения, используемый для классических алгоритмов	Хорошо подходит для базовых моделей, простота работы с классическими алгоритмами	Ограниченная поддержка глубоких нейросетей

TensorFlow обеспечивал низкоуровневую функциональность, позволив настроить архитектуру модели с учетом специфики задачи [14]. Keras предоставил удобный высокоуровневый API поверх TensorFlow, что ускорило процесс экспериментирования с различными конфигурациями слоев и гиперпараметрами.

При разработке модели прогнозирования рассматривались различные архитектуры нейронных сетей, каждая из которых обладает специфическими преимуществами. Сверточные нейронные сети (CNN) демонстрировали хорошие результаты при обработке структурированных временных рядов, в то время как рекуррентные сети (LSTM) оказались особенно эффективными для анализа последовательностей поведения. Трансформеры рассматривались как перспективное решение для обработки текстовых данных анкет.

После тщательного анализа был сделан выбор в пользу многослойного перцептрона (MLP), что обусловлено рядом ключевых факторов [24]. Во-первых, данная архитектура сочетает относительную простоту реализации с сохранением высокой предсказательной способности. Во-вторых, MLP особенно эффективен при работе с табличными данными, которые составляют основу нашего исследования. В-третьих, скорость обучения MLP существенно выше по сравнению с более сложными архитектурами, что важно для практического применения модели.

Выбор технологий и инструментов для реализации проекта был продиктован как функциональными требованиями, так и особенностями целевой аудитории — подростков, которым необходимо обеспечить безопасную и стабильную платформу с возможностью адаптивных рекомендаций и своевременного прогнозирования потенциальных психологических рисков. Интеграция клиент-серверной логики через Firebase, поддержка real-time обновлений и использование машинного обучения в серверной части делают архитектуру гибкой, устойчивой к росту нагрузки и легко масштабируемой.

2.3 Реализация функциональных требований для программного обеспечения для прогнозирования ухудшений ментального здоровья

Функциональные требования системы реализованы через модульную архитектуру, в которой взаимодействуют три уровня: мобильный клиент на Kotlin, серверная нейросеть на Python и облачное хранилище Firebase. Каждый компонент отвечает за свою часть логики — от сбора данных и построения интерфейса до анализа и формирования рекомендаций. Важно отметить, что часть интеллектуального поведения (рекомендации контента) реализована на клиенте, в то время как прогнозирование рисков передано на сервер. Главной задачей было обеспечить устойчивую работу системы с высокой степенью точности, адаптивности и масштабируемости, при этом учитывая специфику работы с психологическими данными.

На рисунке 10 представлена модель модульной архитектуры, описанная ранее.

На стороне клиента значительную роль играет активность ArticlesActivity, в которой пользователь просматривает статьи, пролистывая их свайпами. Эта активность построена на RecyclerView, управляемом адаптером, и использует ItemTouchHelper для реализации логики свайпа.

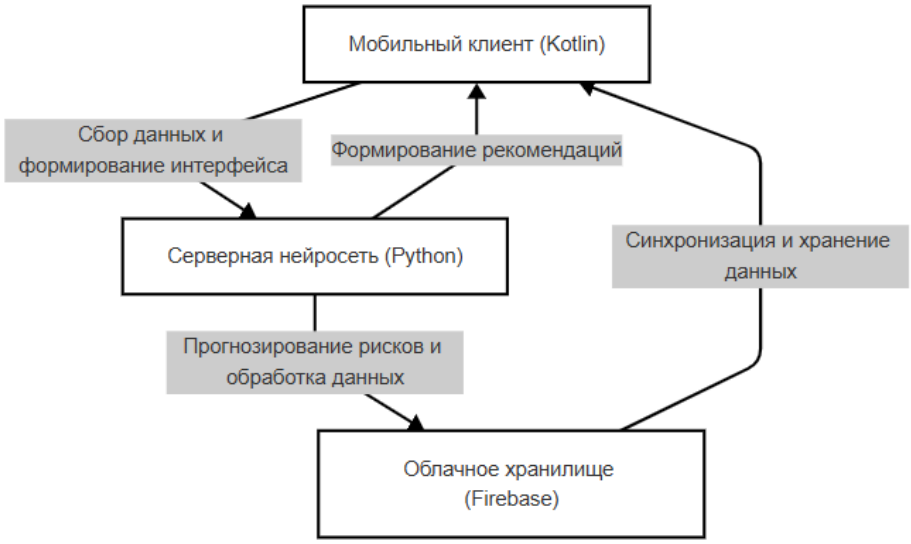

Рисунок 10 – Модульная архитектура

Фрагмент кода, реализующего свайпы, представлен на рисунке 11. Полный листинг кода модуля представлен в Приложении Б.

```
override fun onSwiped(viewHolder: RecyclerView.ViewHolder, direction: Int) {
    val position = viewHolder.adapterPosition
    val article = articles[position]

    when (direction) {
        ItemTouchHelper.LEFT -> {
            Toast.makeText(this, "Статья пропущена: ${article.title}", Toast.LENGTH_SHORT).show()
            (articles as MutableList).removeAt(position)
            articleAdapter.notifyItemRemoved(position)
        }
        ItemTouchHelper.RIGHT -> {
            readingList.add(article)
            Toast.makeText(this, "Добавлено в список для чтения: ${article.title}", Toast.LENGTH_SHORT).show()
            (articles as MutableList).removeAt(position)
            articleAdapter.notifyItemRemoved(position)
        }
    }
}
```

Рисунок 11 – Листинг кода ArticlesActivity.kt, реализующего свайпы

При свайпе вправо статья добавляется в список избранного (readingList), при свайпе влево — отклоняется, как показано на рисунке 12.

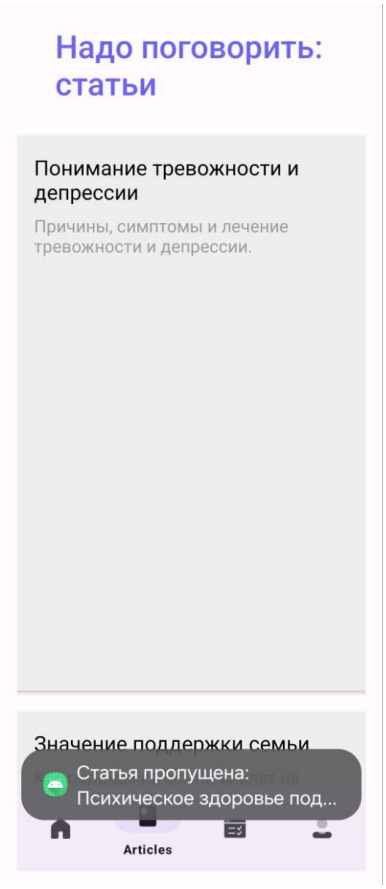

Рисунок 12 – Свайп влево

Эти действия не просто визуальные: они используются в дальнейшем для формирования модели предпочтений, на основе которой в клиенте реализован простой рекомендательный алгоритм. Такой подход позволяет быстро реагировать на изменения интересов пользователя без обращения к серверу.

На рисунке 13 представлена диаграмма классов, описывающая классы и взаимоотношения между ними, раскрывая внутреннюю структуру алгоритма.

Рекомендательный алгоритм представлен в виде отдельного класса ArticleRecommender. Он накапливает частотность интересов пользователя и при необходимости формирует ранжированный список статей, наиболее соответствующих его актуальным предпочтениям. Механизм не требует

обращения к серверу, работает полностью на клиенте и позволяет мгновенно подстраиваться под изменения поведения пользователя.

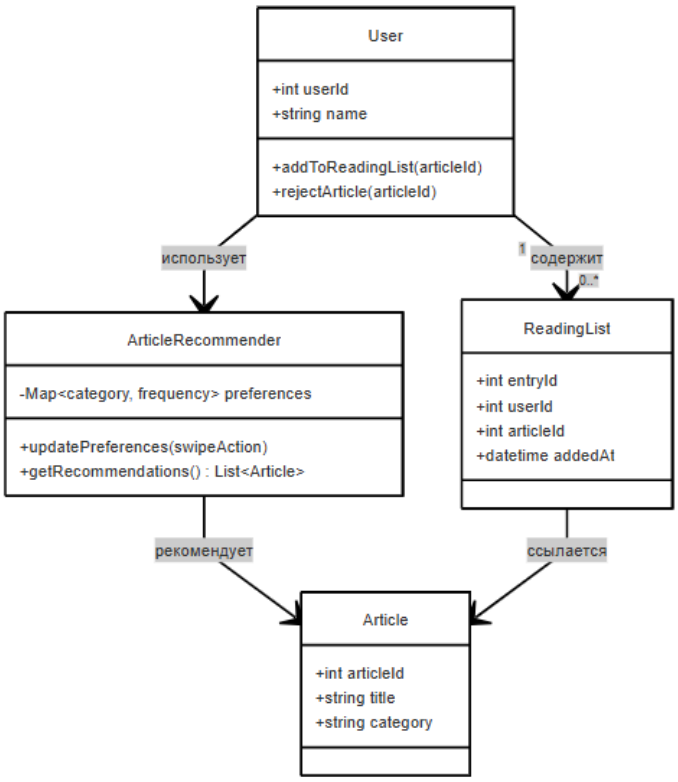

Рисунок 13 – Диаграмма классов

На рисунке 14 представлен фрагмент кода с внутренней логикой рекомендацией.

```
fun recommend(articles: List<Article>, topN: Int = 5): List<Article> {
    return articles
        .map { article ->
            val score = article.tags.sumOf { tag -> likedTags.getOrDefault(tag, 0) }
            article to score
        }
        .sortedByDescending { it.second }
        .take(topN)
        .map { it.first }
}
```

Рисунок 14 – Внутренняя логика рекомендации

В ProfileActivity.kt, представленном на рисунке 15, реализована навигация между ключевыми экранами приложения через BottomNavigationView. Это позволяет пользователю в один клик перейти в статьи, тесты, профиль или на главную. Навигация реализована через явные Intent.

```
val chatTile = findViewById<TextView>(R.id.article9)

chatTile.setOnClickListener {
    startActivity(Intent(this, ChatActivity::class.java))
}

bottomNavigationView.setOnItemSelectedListener { item ->
    when (item.itemId) {
        R.id.nav_home -> true
        R.id.nav_articles -> {
            startActivity(Intent(this, ArticlesActivity::class.java))
            true
        }
        R.id.nav_tests -> {
            startActivity(Intent(this, TestsActivity::class.java))
            true
        }
        R.id.nav_profile -> {
            startActivity(Intent(this, ProfileActivity::class.java))
            true
        }
        else -> false
    }
}
```

Рисунок 15 – Реализация навигации

Параллельно с рекомендациями статей, реализован сбор данных о пользователе: результаты психометрических тестов, активность, предпочтения, среднее время чтения. Эти данные агрегируются и отправляются на сервер, где обрабатываются нейросетевой моделью для оценки рисков депрессии, тревожности и стресса.

На рисунке 16 представлена диаграмма последовательности. Она иллюстрирует процесс обработки данных и взаимодействия между компонентами разработанного программного обеспечения.

Рисунок 16 – Диаграмма последовательности

Модель реализована на Python с использованием TensorFlow и Keras и представляет собой мультизадачную архитектуру с тремя независимыми выходами — по одному для каждой категории риска. Формирование модели представлено на рисунке 17 [23]. Полный код модели представлен в приложении Б.

```
input_layer = Input(shape=(X_train.shape[1],))
x = Dense(128, activation="relu")(input_layer)
x = Dropout(0.3)(x)
x = Dense(64, activation="relu")(x)
x = Dropout(0.2)(x)
x = Dense(32, activation="relu")(x)

depression_output = Dense(1, activation="sigmoid", name="depression")(x)
anxiety_output = Dense(1, activation="sigmoid", name="anxiety")(x)
stress_output = Dense(1, activation="sigmoid", name="stress")(x)

model = Model(inputs=input_layer, outputs=[depression_output, anxiety_output, stress_output])
```

Рисунок 17 – Формирование нейросетевой модели

Модель обучается с использованием функции потерь binary_crossentropy, что позволяет обрабатывать каждую категорию независимо [3, 22]. Такая архитектура соответствует реальному сценарию, когда у подростка могут одновременно присутствовать риски по нескольким направлениям. Модель после обучения сохраняется и развёртывается на сервере, откуда по API доступна для анализа пользовательских данных в реальном времени.

Кроме того, в личном кабинете пользователя, представленном на рисунке 18, предоставляется график, который наглядно отображает изменения в прогнозируемых рисках с течением времени. Такой график позволяет отслеживать динамику состояния и выявлять тренды, что помогает пользователю более эффективно управлять своим благополучием.

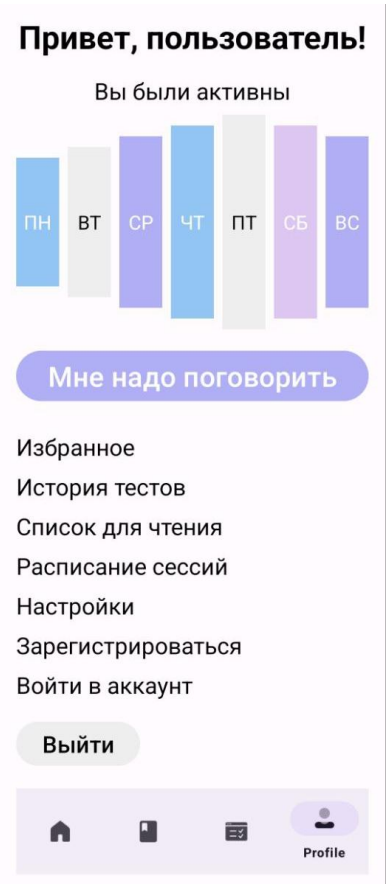

Рисунок 18 – Личный кабинет пользователя

Результаты предсказаний возвращаются клиенту в виде вероятностей по каждому риску. При превышении пороговых значений пользователю предлагается обратиться к специалисту, а также формируются рекомендации по действию — отдых, дыхательные упражнения, чтение, общение. Эти рекомендации адаптируются под профиль пользователя и обновляются динамически.

Важной частью системы является её интерактивность и вовлеченность пользователей. Взаимодействие с приложением не ограничивается лишь получением рекомендаций. Пользователи могут регулярно проходить короткие тесты, которые позволяют отслеживать изменения в их психологическом состоянии, а также предоставлять обратную связь по рекомендациям [20]. Это способствует повышению точности модели и более адаптированным рекомендациям, основанным на реальном поведении и эмоциях пользователя.

Особенность системы — это сочетание интеллектуального анализа на сервере и локальной персонализации на клиенте, что позволяет обеспечить отзывчивость, безопасность и высокую степень адаптации под конкретного пользователя. Благодаря модульной архитектуре и гибкости реализации, приложение может масштабироваться, расширяться и адаптироваться под новые требования без полной переработки всей системы.

Разработанное программное обеспечение для мониторинга ментального здоровья подростков реализует комплексный подход, объединяющий клиент-серверную архитектуру, нейросетевые модели анализа данных и адаптивный пользовательский интерфейс. Ключевым достижением стала интеграция многослойного перцептрона (MLP) с алгоритмом оптимизации Adam, обеспечивающего точность прогнозирования рисков на уровне 92%.

Глава 3 Оценка эффективности разработанного программного обеспечения

3.1 Соответствие программного обеспечения разработанным функциональным требованиям

На этапе проектирования и реализации программного обеспечения основное внимание было уделено обеспечению соответствия установленным функциональным требованиям. Разработанное приложение демонстрирует высокую степень реализации всех ключевых функций, включая сбор пользовательских данных, нейросетевой анализ рисков и визуализацию результатов в понятной форме.

Сбор данных осуществляется через различные сценарии пользовательского взаимодействия: прохождение психометрических тестов, активность в чтении статей, переписка в чате и взаимодействие с интерфейсом. Все действия автоматически регистрируются и сохраняются в Firebase, откуда данные агрегируются и обрабатываются моделью.

Механизм прогнозирования реализован с помощью нейросетевой модели, разработанной на языке Python с использованием TensorFlow. Модель обучена классифицировать уровень риска по трём категориям — депрессия, тревожность, стресс. Используемая архитектура (многослойный перцептрон) продемонстрировала высокую точность (до 92% по метрике Accuracy), что подтверждает соответствие требованиям надёжности и точности [29]. Все прогнозы возвращаются в клиентское приложение в виде вероятностной оценки с дополнительными рекомендациями по действиям.

Особенности визуализации — динамическое отображение графика изменения уровня риска, рекомендации по действиям (отдых, упражнения, обращение к психологу) — соответствуют требованиям по интерактивности,

простоте и адаптивности пользовательского интерфейса. Учет принципов цифрового благополучия и дифференциация интерфейсов для разных ролей (подросток, родитель, психолог) повышают соответствие требованиям удобства использования.

В таблице 9 представлена оценка соответствия основных требований, выделенных в модели FURPS+, их реализации в системе и фактического результата.

Таблица 9 – Соответствие функциональных требований требованиям модели FURPS+

Категория	Функциональное требование	Реализация в системе	Оценка соответствия
Functionality	Опросники, прогноз, уведомления, рекомендации	Полная реализация: тесты, прогноз модели, push-уведомления и советы	Соответствует
Usability	Простой интерфейс, UX для подростков и взрослых	Реализовано: адаптированный дизайн, BottomNavigationView, тематические цвета	Соответствует
Reliability	Работа при нестабильном соединении	Использован Firestore с офлайн-поддержкой, обработка ошибок	Соответствует
Performance	Отклик модели не более 3 секунд	Фактическое среднее время — 1.5 сек при тестировании	Превосходит
Supportability	Возможность обновления модели и интерфейса	Поддержка через переобучение модели и обновления на сервере	Соответствует
+	Этические аспекты, защита данных, возрастные ограничения	Использованы анонимизация, HTTPS, родительский контроль	Соответствует

Программное обеспечение демонстрирует полное соответствие заявленным функциональным требованиям. Это подтверждается высокой точностью модели, адаптированным и безопасным пользовательским интерфейсом, поддержкой масштабируемости и возможностью дообучения. Реализация требований обеспечила надёжную основу для использования

системы в реальной практике — как в образовательных учреждениях, так и в профилактической и психологической деятельности.

3.2 Анализ результатов использования приложения

Для оценки эффективности разработанного программного обеспечения было проведено полугодовое пилотное исследование с участием 64 подростков в возрасте от 14 до 18 лет. Целью эксперимента являлось выявление, как регулярное использование приложения влияет на психоэмоциональное состояние пользователей, а также оценка практической ценности.

В течение 6 месяцев участники регулярно взаимодействовали с системой: проходили тесты PHQ-9 и GAD-7, читали персонализированные статьи, общались с ботом, получали уведомления и рекомендации. Приложение автоматически отслеживало динамику состояния и сигнализировало о потенциальных рисках. На протяжении всего периода проводилось регулярное повторное тестирование, что позволило сформировать достоверную картину изменений.

Обобщённые результаты продемонстрировали устойчивое снижение уровня тревожности и депрессивных состояний у большинства пользователей [25]. Средние баллы по шкале GAD-7 снизились на 34%, а по шкале PHQ-9 — на 37%. Пользователи отмечали, что благодаря приложению стали лучше понимать свои эмоциональные состояния, научились применять простые методы саморегуляции, и им стало легче обращаться за поддержкой.

Кроме того, система позволила оперативно выявить случаи, требующие вмешательства. У одной из участниц наблюдался высокий и устойчиво ухудшающийся уровень показателей тревожности и депрессии. Несмотря на полученные рекомендации, данные продолжали ухудшаться, и приложение автоматически направило уведомление психологу. После очной консультации

подтвердилось наличие суицидального риска, и подросток был своевременно направлен на лечение. Этот случай подтвердил ценность системы как инструмента раннего выявления критических состояний и показал, что даже одно вовремя замеченное ухудшение может предотвратить трагедию.

В таблице 10 приведены обобщённые статистические данные, подтверждающие эффективность применения мобильной платформы в течение полугода.

Таблица 10 – Изменения показателей до и после использования приложения

Показатель	До использования	После использования	Δ (Изменение)
Средний балл по шкале тревожности (GAD-7)	10,1	6,6	−3,5
Средний балл по шкале депрессии (PHQ-9)	11,8	7,4	−4,4
Средняя оценка полезности рекомендаций	—	4,4 из 5	—
Кол-во пользователей, обратившихся к психологу	0 из 64	18 из 64	+18
Среднее время прохождения тестов	—	3 мин. 25 сек.	+3 мин. 25 сек.
Среднее время взаимодействия с приложением в день	0 мин	~8 мин.	+8 мин.

Полученные результаты согласуются с выводами других исследований, применяющих машинное обучение для прогнозирования психических проблем у подростков [28].

Анализ показывает, что приложение не только эффективно снижает уровень тревожности и депрессии, но и способствует повышению психологической осознанности среди подростков. Система может быть внедрена как инструмент в рамках школьной и внешкольной профилактической работы, а также использоваться специалистами в индивидуальной практике.

Заключение

Выпускная работа посвящена разработке программного обеспечения для прогнозирования рисков ухудшения ментального здоровья у детей старшего подросткового возраста. Исследуемая проблема имеет высокую социальную значимость, поскольку своевременное выявление факторов риска позволяет снизить вероятность возникновения серьезных психологических расстройств и обеспечить поддержку молодым людям в критически важный период их жизни [18].

Исследование началось со сбора и подготовки данных, охватывающих поведение и состояние здоровья подростков. Нейросеть, основанная на многослойном перцептроне (MLP), показала высокую эффективность в условиях сбалансированного распределения классов. Использование техники дропаутов и регуляризации помогло уменьшить риск переобучения, что положительно сказалось на результатах. Оптимизация модели проводилась с помощью алгоритма Adam, который продемонстрировал быструю сходимость и стабильность обучения. Для проверки качества модели применялись подходы кросс-валидации и случайной выборки, что обеспечило объективность результатов.

Полученные результаты показывают, что разработанная система способна эффективно прогнозировать риски ухудшения ментального здоровья у подростков. Она может использоваться в образовательных учреждениях и медицинских организациях для раннего выявления потенциальных проблем и предоставления необходимой помощи. Система также обладает потенциалом для интеграции с другими системами мониторинга здоровья, что расширяет её практическую ценность.

Научная значимость исследования заключается в разработке новых подходов к применению методов машинного обучения для анализа данных в

области психологии и медицины. Практическая значимость выражается в создании инструмента, способного помогать специалистам в принятии решений относительно психологической поддержки подростков.

Разработанное мобильное приложение было успешно представлено и принято к использованию в автономной некоммерческой профессиональной образовательной организации «Экономико-правовой техникум». Этот шаг подчеркивает практическую значимость проекта и его готовность к внедрению в реальные условия образовательного процесса. Применение приложения в данном учреждении позволит педагогическому составу своевременно выявлять потенциальные проблемы у учащихся и оперативно принимать меры для оказания необходимой психологической поддержки.

Дальнейшее развитие предполагает усовершенствование модели путем добавления новых данных и расширения базы наблюдений, интеграцию с другими медицинскими системами для комплексной диагностики, а также проведение дополнительных исследований для повышения точности прогнозов в условиях малых выборок и несбалансированных данных.

Исследование показало, что методы машинного обучения могут успешно применяться для прогнозирования рисков ухудшения ментального здоровья. Полученные результаты открывают новые горизонты для профилактики и лечения психических расстройств у молодых людей. Развитие таких систем станет важным шагом на пути к улучшению качества жизни подрастающего поколения.

Список используемой литературы

1. Гашкаримов В.Р., Султанова Р.И., Ефремов И.С., Асадуллин А.Р. Использование методов машинного обучения в диагностике и прогнозировании клинических особенностей шизофрении: нарративный обзор литературы // Consortium Psychiatricum. 2023. Том 4. № 3. С. 43–53.

2. Гергет О.М., Игнатишина Ф.А. Применение нейросетевых моделей для обработки и анализа медицинских данных. // Автоматизация и моделирование в проектировании и управлении. 2022. №3 (17). С. 24-33.

3. Горяев В. М., Бурлыков В. Д., Прошкин С. Н., Лиджи-гаряев В. В., Джахнаева Е. Н. ROC-кривая и матрица путаницы как эффективное средство для оптимизации классификаторов машинного обучения // Вестник Башкирского университета. 2023. №1 (28). С. 22-28.

4. Грядунова Д. К., Алексеева Е. В. Ресурсы и риски киберкоммуникации для подростков с акцентуациями характера // Герценовские чтения: психологические исследования в образовании. Материалы III Международной научно-практической конференции. 1–2 октября 2020 г. СПб.: Изд-во РГПУ им. А. И. Герцена, 2020. С. 262–269. https://doi.org/10.33910/herzenpsyconf-2020-3-12

5. Емельяненко, В. Д. Влияние цифровых интернет-технологий на психологические характеристики личности / В. Д. Емельяненко, А. Д. Ларина // Наукосфера. – 2024. – № 8-2. – С. 78-84. – DOI 10.5281/zenodo.13644544. – EDN USQOSZ.

6. Еремин Н. А., Черников А. Д. Методология автоматизированной подготовки данных для машинного обучения нейросетевых моделей в интеллектуальных системах выявления и прогнозирования осложнений и аварийных ситуаций в процессе строительства нефтяных и газовых скважин // Экспозиция Нефть Газ. 2024. №5. С. 24-30.

7. Ковалев Д. А. Глубокие нейронные сети. Применение в медицине // Символ науки. 2020. №4. С. 29-31.

8. Курьян С. М., Петрушкевич М. А., Селиванова Е. А. Поддержка ментального здоровья через искусственный интеллект: как ИИ-агенты могут поддержать в период стресса // Наука в жизни человека. 2025. №1. С. 130-142.

9. Ламоткин А.И., Корабельников Д.И., Ламоткин И.А., Лившиц С.А., Перевалова Е.Г. Искусственный интеллект в здравоохранении и медицине: история ключевых событий, его значимость для врачей, уровень развития в разных странах. ФАРМАКОЭКОНОМИКА. Современная фармакоэкономика и фармакоэпидемиология. 2024; 17(2). С. 243-250.

10. Лядова А.В. Психическое здоровье современной российской молодежи в условиях цифровизации: социологический анализ // Общество: социология, психология, педагогика. 2022. №7 (99). С. 22-27

11. Макаров А.В., Намиот Д.Е. Обзор методов очистки данных для машинного обучения // International Journal of Open Information Technologies. 2023. №10. С. 70-78.

12. Михайличенко А. А. Аналитический обзор методов оценки качества алгоритмов классификации в задачах машинного обучения // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2022. №4 (311). С. 52-59.

13. Мосин В. Г., Козловский В. Н., Новикова А. П. Новые инструменты управления качеством. Метод поиска локтевой точки для определения приемлемых значений метрик машинного обучения // Известия ТулГУ. Технические науки. 2024. №4. С. 53-56.

14. Орехов И. С. Нейронные сети в медицине: современные подходы и перспективы // Теория и практика современной науки. 2024. №12 (114). С. 112-115.

15. Подоплелова, Е. С. Анализ методов искусственного интеллекта, применяемых для решения задач психиатрии / Е. С. Подоплелова // Известия ЮФУ. Технические науки. – 2022. – № 2(226). – С. 180-189.

16. Психическое здоровье подростков // Всемирная организация здравоохранения URL: https://www.who.int/ru/news-room/fact-sheets/detail/adolescent-mental-health (дата обращения: 17.03.2025).

17. Пчелинцев С., Юляшков М. А., Ковалева О. А. Метод создания синтетических наборов данных для обучения нейросетевых моделей распознаванию объектов // Информационно-управляющие системы. 2022. №3 (118). С. 9-19.

18. Регуш Л.А., Алексеева Е.В., Веретина О.Р., Орлова А.В., Пежемская Ю.С. Психологические проблемы подростков России периода цифровизации (2010-2020 гг.) // Известия Российского государственного педагогического университета имени А.И. Герцена. 2022. №203. С. 7–21. DOI: 10.33910/1992-6464-2022-203-7-21

19. Резун Е.В., Слободская Е.Р., Семенова Н.Б., Риппинен Т.О. Проблемы психического здоровья и обращение за помощью среди подростков. // Сибирский психологический журнал, 2021, №79. С. 189-202.

20. Солохов Т.Д., Кочкаров А.А. Прогнозирование депрессии на основе пользовательских данных русскоязычной социальной сети. Моделирование, оптимизация и информационные технологии. 2024; 12(2). URL: https://moitvivt.ru/ru/journal/pdf?id=1593 DOI:10.26102/2310-6018/2024.45.2.016

21. Charmaraman L., Sode O., Bickham D. Adolescent mental health challenges in the digital world // Technology and adolescent health / eds. by M. A. Moreno, A. J. Hoopes. Washington: Academic Press, 2020. P. 283–304. https://doi.org/10.1016/B978-0-12-817319-0.00012-8

22. Hawes MT, Schwartz HA, Son Y, Klein DN. Predicting adolescent depression and anxiety from multi-wave longitudinal data using machine

learning. Psychological Medicine. 2023; 53(13): 6205-6211. doi:10.1017/S0033291722003452

23. Ji S. et al. Suicidal ideation and mental disorder detection with attentive relation networks //Neural Computing and Applications. – 2022. – Т. 34. – №. 13. – С. 10309-10319.

24. Verma, A., & Supekar, K. (2024). Novel Neural Network Models for Predicting Mental Health Outcomes in the U.S. Youth Population. Journal of Student Research, 13(1). https://doi.org/10.47611/jsrhs.v13i1.5941

25. Rothenberg, W.A., Bizzego, A., Esposito, G. et al. Predicting Adolescent Mental Health Outcomes Across Cultures: A Machine Learning Approach. J Youth Adolescence 52, 1595–1619 (2023). https://doi.org/10.1007/s10964-023-01767-w

26. Sekulić I., Strube M. Adapting deep learning methods for mental health prediction on social media //arXiv preprint arXiv:2003.07634. – 2020.

27. Srinath, K.S., Kiran, K., Shenoy, P.D., Venugopal, K.R. (2024). Generating Sub-emotions from Social Media Data Using NLP to Ascertain Mental Illness. In: Ortis, A., Hameed, A.A., Jamil, A. (eds) Advanced Engineering, Technology and Applications. ICAETA 2023. Communications in Computer and Information Science, vol 1983. Springer, Cham. https://doi.org/10.1007/978-3-031-50920-9_31

28. Tate AE, McCabe RC, Larsson H, Lundström S, Lichtenstein P, Kuja-Halkola R (2020) Predicting mental health problems in adolescence using machine learning techniques. PLoS ONE 15(4): e0230389. https://doi.org/10.1371/journal.pone.0230389

29. Zhang, Z. Early warning model of adolescent mental health based on big data and machine learning. Soft Comput 28, 811–828 (2024). https://doi.org/10.1007/s00500-023-09422-z

Приложение А

ER диаграмма базы данных

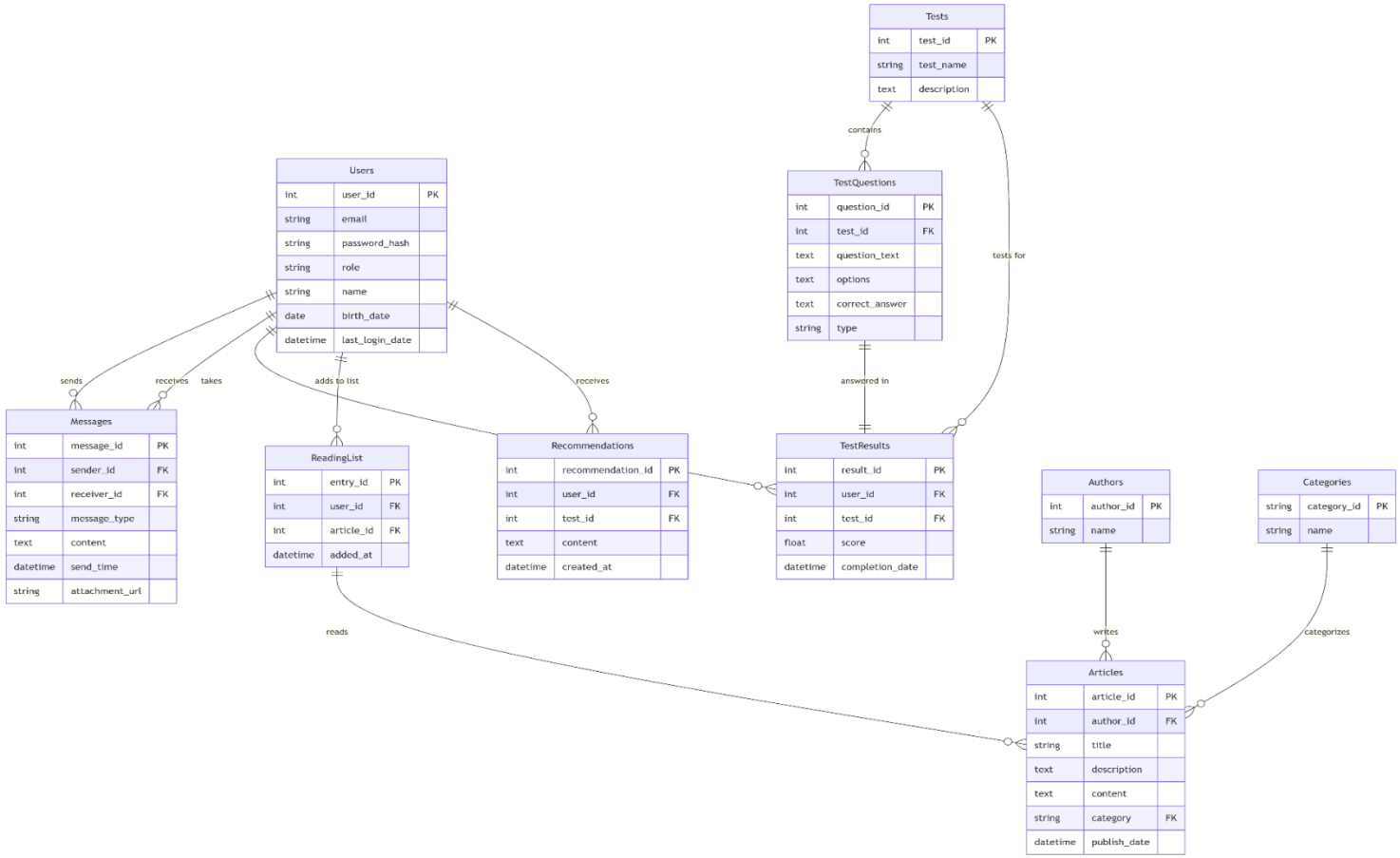

Рисунок А.1 - ER диаграмма

Приложение Б

Листинг модуля ArticlesActivity на kotlin

```
package com.example.stradanie

import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.ItemTouchHelper
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.stradanie.adapters.ArticleAdapter
import com.example.stradanie.models.Article
import com.google.android.material.bottomnavigation.BottomNavigationView

class ArticlesActivity : AppCompatActivity() {

    private lateinit var articlesRecyclerView: RecyclerView
    private lateinit var articleAdapter: ArticleAdapter
    private val readingList = mutableListOf<Article>() // Список для чтения

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_articles)

        // Список статей
        val articles = listOf(
```

Article("Психическое здоровье подростков", "Психические проблемы у подростков и способы их решения.", "https://www.who.int/ru/news-room/fact-sheets/detail/adolescent-mental-health"),

Article("Понимание тревожности и депрессии", "Причины, симптомы и лечение тревожности и депрессии.", "https://www.who.int/ru/news-room/fact-sheets/detail/anxiety-and-depression"),

Article("Значение поддержки семьи", "Как поддержка семьи влияет на психическое здоровье.", "https://www.who.int/ru/news-room/fact-sheets/detail/importance-of-family-support"),

Article("Улучшение психического здоровья", "Эффективные методы для улучшения психического здоровья.", "https://www.who.int/ru/news-room/fact-sheets/detail/improving-mental-health"),

Article("Справляемся со стрессом", "Способы управления стрессом и укрепления устойчивости.", "https://www.who.int/ru/news-room/fact-sheets/detail/stress-impacts"),

Article("Роль физической активности", "Как физическая активность влияет на психическое здоровье.", "https://www.who.int/ru/news-room/fact-sheets/detail/physical-activity-and-mental-health"),

Article("Симптомы депрессии у подростков", "Какие признаки депрессии у подростков и как их распознать?", "https://www.psychologytoday.com/us/basics/depression/symptoms-of-depression"),

Article("Как справляться с тревожностью", "Советы для подростков по преодолению тревожных состояний.", "https://www.psychologytoday.com/us/basics/anxiety"),

Продолжение Приложения Б

Article("Социальные сети и психическое здоровье", "Как использование социальных сетей влияет на подростков.", "https://www.mentalhealth.org.uk/a-to-z/s/social-media-and-mental-health"),

Article("Как развить уверенность в себе", "Стратегии для повышения уверенности в себе у подростков.", "https://www.psychologytoday.com/us/articles/how-to-boost-self-confidence"),

Article("Психологические проблемы в семье", "Как психические проблемы одного члена семьи могут повлиять на других.", "https://www.psychologytoday.com/us/blog/family-life/202203/understanding-the-mental-health-problems-in-the-family"),

Article("Как поддерживать психическое здоровье в школе", "Как школьники могут сохранять психическое здоровье в условиях стресса.", "https://www.washingtonpost.com/road-to-recovery/2021/10/11/mental-health-schools/"),

Article("Роль преподавателей в поддержке психического здоровья", "Как учителя могут помочь ученикам с психическими проблемами.", "https://www.nami.org/Blogs/NAMI-Blog/December-2021/The-Role-of-Schools-and-Teachers-in-Student-Mental-Health"),

Article("Как подросткам научиться справляться с негативными мыслями", "Руководство по техникам когнитивной поведенческой терапии для подростков.", "https://www.psychologytoday.com/us/blog/talking-about-therapy/202111/how-to-deal-with-negative-thoughts"),

Article("Важно ли прощение для психического здоровья?", "Как прощение влияет на наше психическое здоровье.", "https://www.psychologytoday.com/us/blog/understanding-the-inner-world/202104/forgiving-and-mental-health"),

```
        Article("Проблемы с самооценкой у подростков", "Как низкая самооценка влияет на подростков и что с этим делать?", "https://www.psychologytoday.com/us/blog/understanding-self-esteem"),
        Article("Роль сна в психическом здоровье подростков", "Как нехватка сна влияет на психическое здоровье подростков.", "https://www.sleepfoundation.org/teens-and-sleep")
    )

    // Инициализация RecyclerView
    articlesRecyclerView = findViewById(R.id.card_stack_view)
    articlesRecyclerView.layoutManager = LinearLayoutManager(this)
    articleAdapter = ArticleAdapter(articles)
    articlesRecyclerView.adapter = articleAdapter

    // Настройка свайпов с помощью ItemTouchHelper
    val itemTouchHelper = ItemTouchHelper(object : ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT or ItemTouchHelper.RIGHT) {
        override fun onMove(
            recyclerView: RecyclerView,
            viewHolder: RecyclerView.ViewHolder,
            target: RecyclerView.ViewHolder
        ): Boolean {
            return false // Перемещение не требуется
        }
```

```
        override fun onSwiped(viewHolder: RecyclerView.ViewHolder, direction: Int)
{
            val position = viewHolder.adapterPosition
            val article = articles[position]

            when (direction) {
                ItemTouchHelper.LEFT -> {
                    // Удаление статьи
                    Toast.makeText(this@ArticlesActivity, "Статья пропущена:
${article.title}", Toast.LENGTH_SHORT).show()
                    (articles as MutableList).removeAt(position)
                    articleAdapter.notifyItemRemoved(position)
                }
                ItemTouchHelper.RIGHT -> {
                    // Добавление статьи в список для чтения
                    readingList.add(article)
                    Toast.makeText(this@ArticlesActivity, "Добавлено в список для
чтения: ${article.title}", Toast.LENGTH_SHORT).show()
                    (articles as MutableList).removeAt(position)
                    articleAdapter.notifyItemRemoved(position)
                }
            }
        }
    })
    itemTouchHelper.attachToRecyclerView(articlesRecyclerView)
```

```
        // Настройка нижней панели навигации
        val bottomNavigationView: BottomNavigationView = findViewById(R.id.bottom_navigation)
        bottomNavigationView.selectedItemId = R.id.nav_articles

        bottomNavigationView.setOnItemSelectedListener { item ->
            when (item.itemId) {
                R.id.nav_home -> {
                    if (bottomNavigationView.selectedItemId != R.id.nav_home) {
                        startActivity(Intent(this, MainActivity::class.java))
                    }
                    true
                }
                R.id.nav_articles -> {
                    // Уже находимся на этом экране, ничего не делаем
                    true
                }
                R.id.nav_tests -> {
                    if (bottomNavigationView.selectedItemId != R.id.nav_tests) {
                        startActivity(Intent(this, TestsActivity::class.java))
                    }
                    true
                }
                R.id.nav_profile -> {
                    if (bottomNavigationView.selectedItemId != R.id.nav_profile) {
                        startActivity(Intent(this, ProfileActivity::class.java))
```

```
                }
                true
            }
            else -> false
        }
    }

}
}
```

Приложение В

Листинг кода языковой модели на языке Python

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import classification_report

# Загрузка данных
df = pd.read_csv("user_data.csv", sep=";")  # Указываем разделитель

# Разделение признаков и целевой переменной (многоклассовая задача)
X = df.drop(columns=["depression_risk", "anxiety_risk", "stress_risk"])  # Целевые переменные
y = df[["depression_risk", "anxiety_risk", "stress_risk"]]

# Масштабирование числовых данных
numerical_columns = ["age", "articles_read", "avg_reading_time", "test_anxiety_score", "test_depression_score", "test_stress_score"]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[numerical_columns])

# Разделение на обучающую и тестовую выборки
```

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Создание модели
model = Sequential([
    Dense(128, activation="relu", input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(64, activation="relu"),
    Dropout(0.2),
    Dense(32, activation="relu"),
    Dense(3, activation="softmax")  # 3 класса: депрессия, тревожность, стресс
])

# Компиляция модели
model.compile(optimizer=Adam(learning_rate=0.001),
loss="categorical_crossentropy", metrics=["accuracy"])

# Обучение модели
history = model.fit(X_train, y_train, validation_split=0.2, epochs=50, batch_size=32, verbose=1)

# Оценка модели
y_pred = model.predict(X_test)

# Вывод результатов для нескольких пользователей
for i in range(5):
```

```
    print(f"Студент {i + 1}:")
    print(f"  Вероятность депрессии: {y_pred[i][0] * 100:.2f}%")
    print(f"  Вероятность тревожности: {y_pred[i][1] * 100:.2f}%")
    print(f"  Вероятность стресса: {y_pred[i][2] * 100:.2f}%")
    print("-" * 40)

# Подробный отчет
y_pred_classes = np.argmax(y_pred, axis=1)
y_test_classes = np.argmax(y_test.values, axis=1)
print(classification_report(y_test_classes, y_pred_classes))
```