

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ПОЛНЫЙ ЦИКЛ РАЗРАБОТКИ КРОСС-ПЛАТФОРМЕННОГО
МОБИЛЬНОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА
ПРОГРАММИРОВАНИЯ JAVA SCRIPT И БИБЛИОТЕКИ REACT**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные
технологии

факультета компьютерных наук и информационных технологий

Ефимова Андрея Андреевича

Научный руководитель

зав. каф., к. ф.-м. н., доцент

С. В. Миронов

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В современном мире мобильные приложения стали частью повседневной рутины каждого человека. В связи с неотъемлемостью мобильных приложений в нашей жизни рынок мобильной разработки развивается, а разработчики мобильного программного обеспечения становятся востребованнее с каждым годом. Люди уже не замечают, как легко стало узнать температуру за окном, не вставая с дивана, заказать свежие овощи в несколько жестов, совершить банковский перевод без необходимости стоять в бесконечных очередях или заказать такси, не совершая при этом абсолютно никаких звонков. И все это доступно нам благодаря нашим мобильным устройствам, которые помогают упростить ежедневные задачи и выделить драгоценное время для чего-то действительно важного.

Собравшись в команду совместно с психологическим факультетом СГУ, было принято коллективное решение создать современное мобильное приложение в тематике психологической помощи. Главная идея нашего приложения — помочь людям с тревожностью, апатией и другими психологическими проблемами, преследующими современных людей.

Целью данного проекта является разработка современного UI для мобильного приложения, онлайн и оффлайн функционала приложения, а также кросс-платформенность, выраженная в доступности приложения для двух основных мобильных операционных систем.

Чтобы достичь поставленную цель, необходимо реализовать следующие задачи:

1. Проанализировать требования заказчика (психологического факультета).
2. Разработать современный дизайн мобильного приложения.
3. Перенести получившиеся мокапы в работающий MVP-прототип.
4. Добавить весь дополнительный и необязательный функционал.
5. Протестировать мобильное приложение на различных устройствах с разными операционными системами.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 3 приложений. Общий объём работы – 50 страниц, из них 40 страниц – основное содержание, включая 47 рисунков и 1 таблицу, CD диск в качестве приложения, список использованных источников информации – 20 наименований.

Первый раздел «Анализ предметной области» посвящен описанию основных определений используемых в работе, а также теоретической информации, касающейся инструментов разработки.

Несколько лет назад рынок мобильных устройств мог похвастаться изобилием операционных систем. Но в наши дни многие из них стали не востребова­ными, начиная с 2019 года почти 100 процентов мобильных устройств находятся под управлением одной из систем: Android либо iOS.

Существует множество языков программирования, подходящих для разработки ПО для мобильных устройств, но два основных и нативных языка — это Kotlin и Swift. Для реализации проекта необходимо было выбрать язык программирования, подходящий для разработки продукт, при этом основное требование к кросс-платформенности приложения должно было выполняться.

Язык программирования Kotlin был создан 22 июля 2011 года российскими разработчиками из компании JetBrains. Является статически типизированным, объектно-ориентированным языком программирования, работающим поверх Java Virtual Machine. Также компилируется в JavaScript и в исполняемый код ряда платформ через инфраструктуру LLVM. Kotlin был создан с целью быть более современным и удобным языком, чем Java, и он включает многие функции, которых нет в Java. Некоторые из основных преимуществ Kotlin: безопасность типов, удобный синтаксис, null-безопасность, расширения функций, мультиплатформенность. Язык получил поддержку от Google — разработчиков самой операционной системы, и в мае 2017 года компания Google сообщила, что инструменты языка Kotlin, основанные на JetBrains IDE, будут по стандарту включены в официальный инструмент разработки для ОС Android.

Язык программирования Swift был создан в 2014 году компанией Apple для разработки не только мобильных, но и десктопных приложений для своих продуктов: iPhone, iPad, умных часов Apple Watch, а также компьютеров iMac и MacBook. Является открытым мультипарадигмальным компилируемым языком программирования общего назначения. Swift совместим с основной кодовой базой Apple, написанной на более раннем языке Objective-C. Swift задумывался как более лёгкий для чтения и устойчивый к ошибкам программиста язык, нежели предшествовавший ему Objective-C. Программы на Swift компилируются при помощи LLVM, входящей в интегрированную среду разработки XCode. Swift может использовать рантайм Objective-C, что делает возможным исполь-

зование обоих языков (а также языка C) в рамках одной программы. Более того, код самой операционной системы с каждым годом все больше состоит из Swift, заменяя устаревший Objective-C.

Так как параллельная разработка приложения на двух различных языках является трудозатратной и может быть невыгодна для малых проектов, было принято решение найти компромисс в виде языка программирования, который подходил бы для обеих мобильных операционных систем, но при этом обеспечивал бы необходимый функционал и инструментарий разработки. Таким языком был выбран Java Script с фреймворком React Native. Фреймворк React Native был представлен в 2015 году компанией Facebook для кроссплатформенной разработки мобильных приложений под операционные системы iOS и Android с использованием языка JavaScript и библиотеки React. React Native позволяет создавать нативные приложения с единой кодовой базой, что ускоряет процесс разработки и упрощает поддержку продукта. React Native использует архитектуру bridge — «моста» между JavaScript и нативными компонентами платформы, благодаря чему интерфейсы получаются гибкими и отзывчивыми. Приложения, созданные с помощью React Native, работают с высокой производительностью и визуально неотличимы от нативных. Разработка ведётся в любой IDE, поддерживающей JavaScript, чаще всего в Visual Studio Code. React Native также поддерживает интеграцию с нативным кодом на Java, Kotlin, Objective-C и Swift, что делает его удобным для масштабных гибридных решений. Было решено использовать именно его.

Второй раздел «Разработка мобильного приложения» посвящен подробному описанию кодовой базы инструментов разработки, проектированию и реализации приложения, его пользовательского интерфейса, компонентов и расширений.

В процессе создания интерфейса мобильного приложения было принято решение использовать следующий список технологий. Для прототипирования и создания современного дизайна, подходящего под критерии заказчика, был использован веб-сервис Figma, который позволил визуализировать интерфейс и динамически демонстрировать его заказчику для динамичного внедрения возможных правок. Так как фреймворк React Native работает на языке Java Script, то основной код продукта напоминал разработку веб-сайта: разметка страниц и экранов состояла из HTML, визуальные стили были реализованы при по-

мощи языка CSS и встроенных инструментов React Native, обеспечивающих быструю работу со стилями функциональных компонентов. Сборка проекта и управление его зависимостями осуществлялось с использованием пакетного менеджера npm внутри среды Visual Studio Code. Для удобной работы с React Native использовался фреймворк Expo, что позволило протестировать наше кросс-платформенное приложение на различных операционных системах, без необходимости публикации и получения лицензии в нативных магазинах приложений.

На этапе проектирования пользовательского интерфейса основное внимание было уделено интеграции функциональных требований заказчика с современными принципами UI/UX-дизайна. Визуальная составляющая разрабатывалась с учетом предварительно собранных пожеланий представителей психологического факультета, включая предпочтения по цветовой палитре, структуре навигации, логике размещения элементов и стилистике оформления. При этом важной задачей являлось не только формальное следование запросам заказчика, но и обеспечение интуитивной понятности интерфейса, высокой эргономичности и соответствия ожиданиям конечного пользователя. Были применены принципы и рекомендации Material Design, обеспечивающие предсказуемость поведения элементов и визуальную иерархию. Навигация была спроектирована с учетом когнитивной нагрузки: предпочтение отдано простым и повторяющимся паттернам взаимодействия, позволяющим пользователю быстро освоиться в среде приложения. В процессе проектирования активно применялись практики прототипирования и тестирования, включая разработку интерактивных макетов в Figma, что позволило на раннем этапе выявить потенциальные проблемы в логике взаимодействия и оперативно адаптировать структуру интерфейса под реальные сценарии использования. Такое сочетание индивидуального подхода и современных стандартов дизайна позволило достичь баланса между визуальной привлекательностью, функциональностью и пользовательской эффективностью интерфейса.

Разработка программного кода нашего мобильного приложения велась в соответствии с рекомендуемыми правилами создания компонентов React Native для достижения приемлемого уровня производительности продукта и возможности его масштабировать. Также в процессе написания кода использовались часто используемые паттерны фреймворка Expo для написания стандартных

типовых элементов мобильного приложения. Примером одного из таких элементов может служить навигационное меню в нижней части экрана, используемое для переключения между главными разделами приложения. Данное меню является самостоятельным компонентом и отображается лишь на тех вкладках, где оно требуется.

Страницы основных разделов мобильного приложения разработаны в стандартном стиле написания кода, аналогичном традиционному подходу к созданию веб-страниц. Используются привычные элементы интерфейса и верстка, обеспечивающая простоту восприятия и удобство навигации для пользователя. Внутри таких страниц преимущественно применялись функциональные компоненты React Native – компактные и переиспользуемые блоки кода, описывающие логику и отображение интерфейса. Такой подход способствует упрощению структуры проекта, повышает читаемость кода и облегчает сопровождение приложения. Показательным примером использования данного подхода является страница раздела с расслабляющими звуками. На данной странице повторно используется функциональный компонент для карточки отдельного звука, который включает в себя текстовый заголовок звука, его текстовое описание, путь к файлу с изображением его иконки и путь для навигации к экрану запуска данного звука внутри приложения.

Также в процессе разработки дизайна интерфейса мобильного приложения упоминалось, что на экранах процесса прослушивания расслабляющего звука и процесса воспроизведения дыхательной практики необходима анимированная фигура, состоящая из двух окружностей: внешней и внутренней. Программный код для данного анимированного элемента был разработан в стиле создания функционального компонента React. Такой подход был выбран для возможности повторного использования анимации на необходимых экранах. Для реализации запуска анимации был использован один из стандартных хуков React под названием `useEffect`. Время анимации было представлено в виде опционального параметра, стандартным значением которого было решено выбрать 4 секунды. Но при вызове данного компонента внутри требуемого экрана, данный параметр может меняться в соответствии с необходимым темпом звукового сопровождения. Сама анимация является зацикленной и повторяется бесконечное количество раз, пока пользователь не закроет экран, на котором она расположена.

Для публикации приложения в App Store издателю необходимо оформить платную подписку разработчика Apple стоимостью 99 долларов в год, а в Google Play требуется единовременная плата за аккаунт разработчика. Данные расходы необходимо учитывать компании при планировании выпуска мобильного приложения, так как они входят в состав операционных затрат, связанных с поддержкой и распространением программного продукта. Так как финансовый план заказчика всё ещё находится в стадии активной разработки, а само приложение уже технически готово, было принято решение производить сборку и тестирование готового продукта при помощи фреймворка Expo и его собственного одноимённого мобильного приложения Expo Go, которое позволяет запускать и тестировать собственный продукт без необходимости публикации его на сторонних площадках. На момент разработки продукта фреймворк Expo имел версию 53.0.9 и был установлен на компьютере с операционной системой Windows 10, мобильное приложение Expo Go имело версию 2.33.13 и было установлено на устройство Apple iPhone 14 Pro с операционной системой iOS 17.4.1.

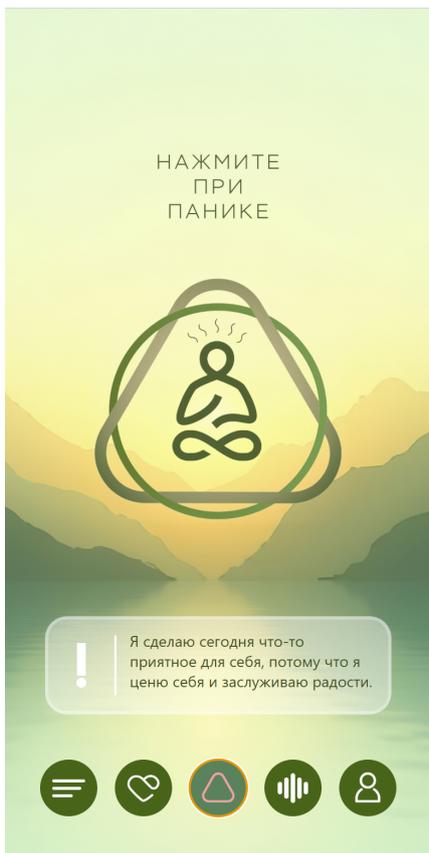


Рисунок 1 – Домашний экран



Рисунок 2 – Экран со звуками



Рисунок 3 – Экран с практиками

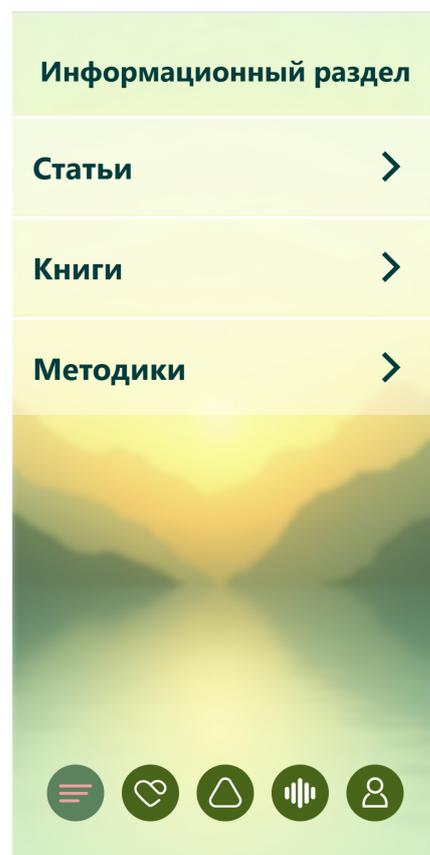


Рисунок 4 – Информационный раздел



Рисунок 5 – Экран выполнения практики

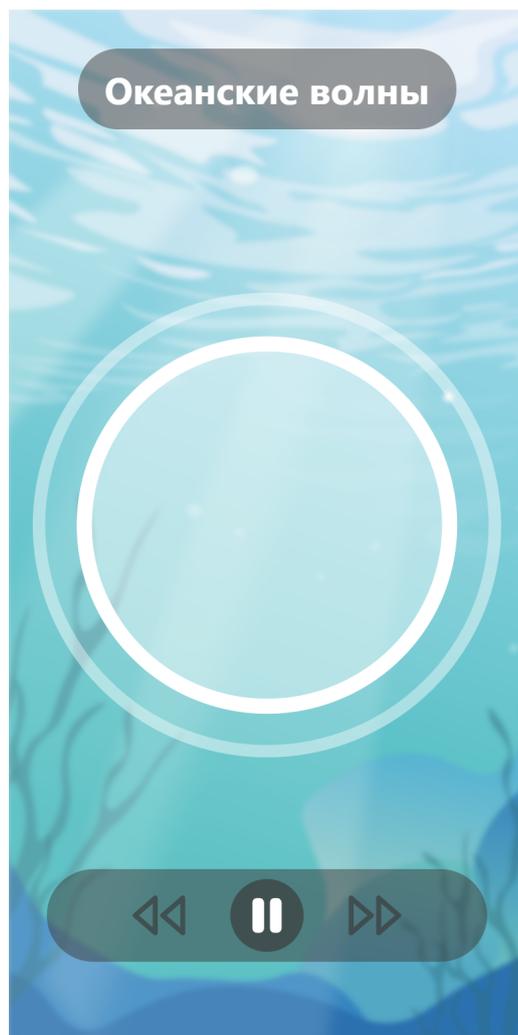


Рисунок 6 – Экран воспроизведения звука



Рисунок 7 – Экран раздела дыхательных практик

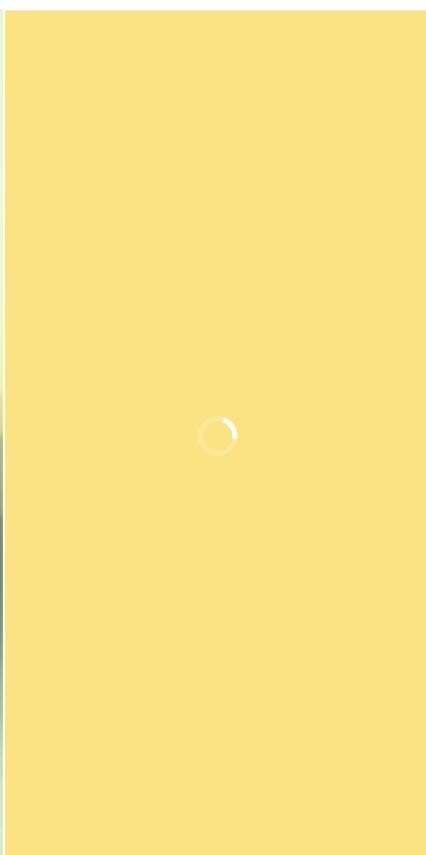


Рисунок 8 – Загрузочный экран практики

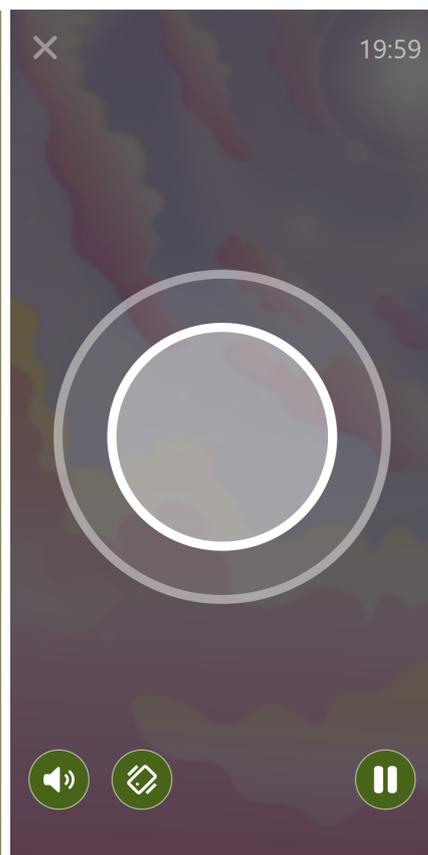


Рисунок 9 – Экран выполнения практики



Рисунок 10 – Экран информационного раздела



Рисунок 12 – Экран режима чтения статьи

Рисунок 11 – Экран списка статей

ЗАКЛЮЧЕНИЕ

Таким образом, в процессе разработки приложения были применены как теоретические знания, так и практические навыки, полученные в ходе изучения инструментов React Native разработки. Использование языка Java Script и библиотеки React Native для построения интерфейса, понимание процесса управления приложением и взаимодействия клиент-серверной части, применение React Router DOM и фреймворка Ехро для тестирования, а также гибкое взаимодействие с заказчиком и поиск компромиссов для выполнения всех изначальных требований, позволило создать удобное и функциональное мобильное кросс-платформенное приложение для психологической помощи нуждающимся пользователям.