# 2 Выбор средств и методов для создания приложения

Выбор средств и методов является важнейшим аспектом процесса разработки, поскольку он напрямую влияет на эффективность, масштабируемость и общий успех проекта. Для этой дипломной работы на степень бакалавра мы тщательно оценили и выбрали набор мощных и универсальных инструментов, чтобы обеспечить бесперебойную и надежную разработку.

Выбранные инструменты для этого проекта включают React Native, Firebase Cloud Firestore, Expo и Visual Studio Code (VS Code) в качестве редактора кода. Каждый инструмент обладает своими уникальными возможностями и преимуществами, позволяя разработчикам создавать высококачественное мобильное приложение, соответствующее требованиям и целям проекта.

# 2.1 React Native

React Native — это JavaScript-фреймворк для написания реальных мобильных приложений с собственным рендерингом для iOS и Android [11]. Он основан на React, JavaScript-библиотеке Facebook для создания пользовательских интерфейсов, но ориентирован не на браузер, а на мобильные платформы. Другими словами: веб-разработчики теперь могут писать мобильные приложения, которые выглядят и ощущаются по-настоящему “родными”, и все это с комфортом, используя библиотеку JavaScript, которую мы уже знаем и любим. Кроме того, поскольку большая часть кода, который вы пишете, может быть разделена между платформами, React Native упрощает одновременную разработку как для Android, так и для iOS.

Подобно React for the Web, приложения React Native написаны с использованием смеси JavaScript и разметки в стиле XML, известной как JSX. Затем, под капотом, React Native “bridge” вызывает собственные API рендеринга в Objective-C (для iOS) или Java (для Android). Таким образом, ваше приложение будет отображаться с использованием реальных компонентов мобильного пользовательского интерфейса, а не веб-просмотров, и будет выглядеть и чувствовать себя как любое другое мобильное приложение. React Native также предоставляет интерфейсы JavaScript для API-интерфейсов платформы, поэтому ваши приложения React Native могут получать доступ к функциям платформы, таким как камера телефона или местоположение пользователя.

React Native в настоящее время поддерживает как iOS, так и Android и имеет потенциал для расширения и на будущие платформы. В этой книге мы рассмотрим как iOS, так и Android. Подавляющее большинство кода, который мы напишем, будет кроссплатформенным. И да: вы действительно можете использовать React Native для создания готовых к работе мобильных приложений! Немного анекдота: Facebook, Palantir и TaskRabbit уже используют его в производстве для пользовательских приложений.

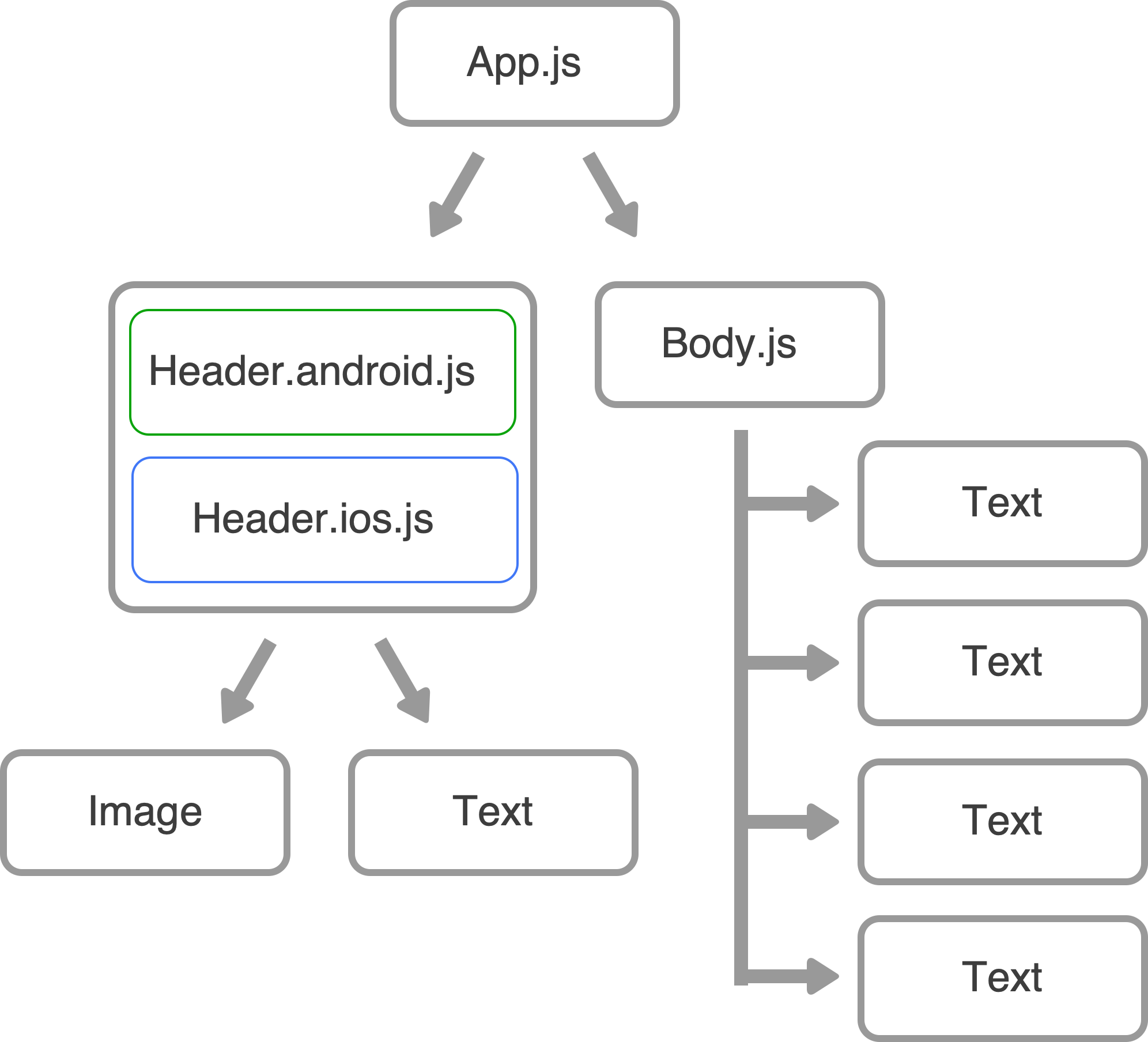


Рисунок 2.2.1 – Кроссплатформенная разработка с использованием React Native

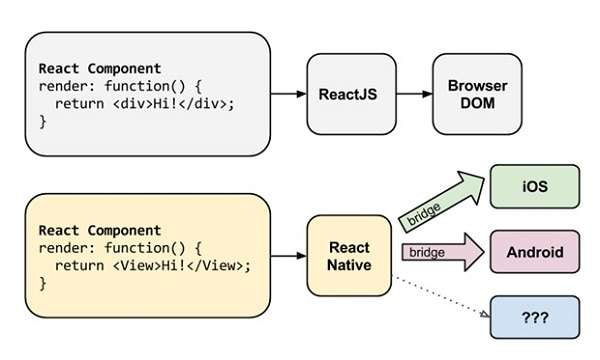


Рисунок 2.2.2 – Рендеринг компонентов в React Native

Решение использовать React Native для этого проекта было основано на нескольких ключевых факторах:

1. Кроссплатформенная совместимость: Одним из существенных преимуществ React Native является его способность создавать приложения, которые без проблем работают как на платформах iOS, так и на Android. Благодаря общей кодовой базе разработчики могут писать один раз и развертывать на нескольких платформах, экономя время и усилия. Такая кроссплатформенная совместимость обеспечивает более быстрые циклы разработки и более широкий охват аудитории.

2. Производительность, подобная нативной: React Native устраняет разрыв между нативной и веб-разработкой, используя нативные компоненты и рендеря их с помощью JavaScript. Такой подход позволяет приложению обеспечивать производительность, близкую к нативной, обеспечивая плавный и отзывчивый пользовательский интерфейс. React Native достигает этого за счет оптимизации для конкретной платформы и рендеринга компонентов пользовательского интерфейса с использованием базовых собственных API.

3. Возможность повторного использования кода: Одним из ключевых преимуществ React Native является его способность повторно использовать код на разных платформах. Большинство бизнес-логики и компонентов пользовательского интерфейса могут быть разделены между iOS и Android, что сокращает дублирование и улучшает удобство обслуживания. Эта возможность совместного использования кода не только ускоряет разработку, но и упрощает процесс обслуживания, поскольку обновления и исправления ошибок могут применяться единообразно к обеим платформам.

4. Богатая экосистема и поддержка сообщества: React Native извлекает выгоду из динамичного и активного сообщества разработчиков и надежной экосистемы библиотек, инструментов и ресурсов. Эта обширная сеть поддержки обеспечивает доступ к широкому спектру готовых компонентов, сторонних интеграций и удобных для разработчиков утилит, которые могут значительно ускорить разработку и расширить функциональность приложения.

5. Быстрая итерация и скорость разработки: Функция горячей перезагрузки React Native позволяет разработчикам видеть изменения, внесенные в код, практически мгновенно, не требуя полной перестройки приложения. Этот быстрый цикл итераций значительно повышает скорость разработки, позволяя разработчикам быстро выполнять итерации, тестировать и совершенствовать функции, макеты и функциональность приложения.

6. Эффективность разработчика: React Native предлагает декларативную модель программирования, использующую возможности архитектуры React, основанной на компонентах. Такой подход упрощает разработку пользовательского интерфейса, поскольку разработчики могут создавать повторно используемые компоненты пользовательского интерфейса и эффективно управлять их состоянием. Знакомый и интуитивно понятный характер модели программирования React в сочетании с обширными возможностями инструментария и отладки React Native повышает производительность разработчиков и сокращает время обучения.

7. Широкое внедрение в отрасли: React Native завоевал значительную популярность в отрасли, и многие известные компании и организации используют его для своих нужд в разработке мобильных приложений. Такое широкое внедрение обеспечивает зрелую и стабильную платформу с постоянной поддержкой и обновлениями, что делает ее надежным выбором для создания готовых к работе приложений.

# 2.2 Firebase Cloud Firestore

Cloud Firestore — это гибкая, масштабируемая база данных для разработки мобильных устройств, веб-приложений и серверов от Firebase и Google Cloud. Как и Firebase Realtime Database, она синхронизирует ваши данные между клиентскими приложениями с помощью прослушивателей реального времени и предлагает автономную поддержку для мобильных устройств и Интернета, так что вы можете создавать адаптивные приложения, которые работают независимо от задержки в сети или подключения к Интернету. Cloud Firestore также предлагает бесшовную интеграцию с другими продуктами Firebase и Google Cloud, включая облачные функции.

Cloud Firestore — это размещенная в облаке база данных NoSQL, доступ к которой ваши приложения Apple, Android и веб-приложения могут получить напрямую через собственные SDK. Облачный Firestore также доступен на родном языке Node.js, Java, Python, Unity, C++ и Go SDK, в дополнение к REST и RPC API.

В соответствии с моделью данных NoSQL Cloud Firestore данные хранятся в документах, содержащих поля, соответствующие значениям. Эти документы хранятся в коллекциях, которые являются контейнерами для документов, которые можно использовать для организации данных и построения запросов. Документы поддерживают множество различных типов данных, от простых строк и чисел до сложных вложенных объектов. Можно также создавать вложенные коллекции внутри документов и создавать иерархические структуры данных, которые будут изменяться по мере роста базы данных. Модель данных Cloud Firestore поддерживает любую структуру данных, которая лучше всего подходит для вашего приложения.

Кроме того, запросы в Cloud Firestore являются выразительными, эффективными и гибкими. Создавайте неглубокие запросы для извлечения данных на уровне документа без необходимости извлекать всю коллекцию или какие-либо вложенные полколлекции. Добавьте сортировку, фильтрацию и ограничения к вашим запросам или курсорам для разбивки результатов на страницы. Чтобы сохранить данные в актуальном состоянии вашего приложения, не извлекая всю базу данных при каждом обновлении, добавьте прослушиватели в реальном времени. Добавление прослушивателей в реальном времени в ваше приложение уведомляет вас моментальным снимком данных всякий раз, когда данные, которые прослушивают ваши клиентские приложения, изменяются, извлекая только новые изменения.

Защитите доступ к своим данным в Cloud Firestore с помощью аутентификации Firebase и правил безопасности Cloud Firestore для платформ Android, Apple и JavaScript или управления идентификацией и доступом (IAM) для серверных языков.

Решение о включении Firebase Cloud Firestore в проект было основано на его многочисленных ключевых функциях и преимуществах, которые способствуют общей эффективности и функциональности приложения.

1. Гибкость - Cloud Firestore предоставляет гибкую модель данных, которая поддерживает иерархические структуры данных. Вы можете хранить свои данные в документах, организованных в коллекции. Документы могут содержать сложные вложенные объекты в дополнение к вложенным коллекциям. Такая гибкость позволяет вам организовать ваши данные таким образом, который наилучшим образом соответствует требованиям вашего приложения. Независимо от того, есть ли у вас простые пары ключ-значение или сложные вложенные структуры данных, Cloud Firestore может адаптировать вашу модель данных.

2. Выразительный запрос - Облачный Firestore позволяет использовать запросы для извлечения определенных документов или для извлечения всех документов в коллекции, соответствующих определенным критериям. Запросы могут включать в себя несколько связанных фильтров, сочетающих операции фильтрации и сортировки. По умолчанию Cloud Firestore индексирует запросы, обеспечивая эффективную производительность, пропорциональную размеру результирующего набора, а не всего набора данных. Это делает запрос и извлечение данных из базы данных очень выразительными, эффективными и масштабируемыми.

3. Обновления в реальном времени - Cloud Firestore, подобно базе данных реального времени, использует синхронизацию данных для предоставления обновлений в режиме реального времени на любом подключенном устройстве. Он эффективно выполняет одноразовые запросы для извлечения данных и обеспечивает синхронизацию данных между всеми подключенными клиентами. Всякий раз, когда в данных, которые прослушивают ваши клиентские приложения, происходят изменения, Cloud Firestore немедленно уведомляет их, гарантируя, что ваши пользователи всегда будут иметь актуальную информацию. Эта возможность синхронизации в режиме реального времени позволяет разрабатывать динамичные и отзывчивые приложения.

4. Автономная поддержка - Cloud Firestore предлагает встроенную автономную поддержку, позволяющую вашему приложению бесперебойно функционировать, даже если устройство отключено от сети или имеет плохое сетевое подключение. Он кэширует активно используемые данные на устройстве, позволяя пользователям читать, записывать, прослушивать и запрашивать данные локально. Как только устройство повторно подключается к сети, Cloud Firestore автоматически синхронизирует локальные изменения с удаленной базой данных, обеспечивая согласованность данных. Эта автономная поддержка улучшает пользовательский опыт, обеспечивая бесперебойное использование приложения независимо от доступности сети.

5. Cloud Firestore предназначен для работы с приложениями любого масштаба. Он использует мощную инфраструктуру Google Cloud, обеспечивая автоматическую репликацию данных в нескольких регионах, надежные гарантии согласованности, атомарные пакетные операции и поддержку реальных транзакций. Независимо от того, имеет ли ваше приложение небольшую базу пользователей или сталкивается с большим трафиком и объемами данных, Cloud Firestore справится с требованиями сложности и масштабируемости. Он был разработан в соответствии с требованиями крупнейших глобальных приложений, гарантируя, что ваше приложение сможет беспрепятственно развиваться по мере расширения вашей пользовательской базы.

# 2.3 Expo CLI

Expo — это фреймворк и платформа для создания мобильных приложений с использованием React Native. Он предоставляет набор инструментов и сервисов, которые упрощают процесс разработки и позволяют разработчикам быстро создавать, тестировать и развертывать мобильные приложения на различных платформах, таких как iOS и Android.

Expo строится поверх React Native и расширяет его возможности, предлагая дополнительные функции и инструменты. Некоторые ключевые особенности выставки включают в себя:

1. Единая среда разработки: Expo предлагает единую среду разработки, которая позволяет разработчикам создавать приложения как для платформ iOS, так и для Android, используя единую кодовую базу. Этот подход, известный как кроссплатформенная разработка, значительно сокращает время и усилия разработки за счет устранения необходимости поддерживать отдельные базы кода для каждой платформы. С помощью Expo разработчики могут написать логику своего приложения один раз и без особых усилий развернуть ее на нескольких платформах, обеспечивая согласованность работы пользователей на разных устройствах.

2. Простая настройка: Expo упрощает первоначальную настройку среды разработки. Он предоставляет инструмент интерфейса командной строки (CLI), который автоматизирует процесс создания нового проекта, управления зависимостями и настройки необходимых параметров сборки. Интуитивно понятный интерфейс CLI помогает разработчикам в процессе настройки, делая его доступным даже для тех, кто новичок в разработке мобильных приложений. Такая простота настройки позволяет разработчикам быстро приступить к разработке, не увязая в сложных задачах настройки.

3. Обширные встроенные API: Expo предлагает богатый набор готовых встроенных API, которые обеспечивают доступ к различным функциям устройств и датчикам. Эти API охватывают широкий спектр возможностей, включая доступ к камере, геолокацию, акселерометр, push-уведомления, доступ к файловой системе и многое другое. Используя собственные API-интерфейсы Expo, разработчики могут легко внедрять функции, зависящие от устройства, в свои приложения без необходимости писать код, зависящий от платформы. Это значительно упрощает процесс разработки и позволяет разработчикам сосредоточиться на создании основной функциональности своих приложений, а не тратить время на детали реализации, зависящие от конкретной платформы.

4. Обновления в режиме реального времени (OTA): Одной из выдающихся особенностей Expo является ее способность предоставлять обновления в режиме реального времени (OTA) для развернутых приложений. Это означает, что разработчики могут вносить обновления и исправления ошибок в свои приложения, не требуя от пользователей скачивать и устанавливать новую версию из app Store. OTA-обновления Expo позволяют разработчикам беспрепятственно загружать обновления на устройства своих пользователей, гарантируя, что у них всегда будет последняя версия приложения. Эта функция особенно полезна для быстрого выполнения итераций и развертывания обновлений для пользовательской базы приложения, поскольку избавляет пользователей от необходимости проходить процесс обновления вручную.

5. Инструменты тестирования и отладки: Expo предоставляет набор инструментов тестирования и отладки, которые упрощают процесс выявления и устранения проблем в приложении. Среда разработки Expo включает встроенный симулятор и функцию перезагрузки в режиме реального времени, которая позволяет разработчикам мгновенно видеть изменения, которые они вносят в свой код, не перестраивая все приложение целиком. Это облегчает быструю итерацию и тестирование, повышая производительность разработчика. Кроме того, Expo предлагает интеграцию с популярными инструментами отладки, такими как React Native Debugger и Flipper, что позволяет разработчикам отлаживать свои приложения непосредственно из среды разработки.

6. Сообщество и экосистема: у Expo есть динамичное и активное сообщество разработчиков, которые вносят свой вклад в ее экосистему. Сообщество Expo предоставляет множество ресурсов, включая учебные пособия, документацию и примеры кода, которые могут помочь разработчикам ускорить процесс обучения и разработки. Сообщество активно поддерживает Expo и предлагает помощь разработчикам, сталкивающимся с проблемами или нуждающимся в руководстве. Кроме того, Expo располагает обширной коллекцией созданных сообществом библиотек и компонентов, которые расширяют его функциональность и предлагают готовые решения для общих задач разработки. Эта обширная экосистема гарантирует разработчикам доступ к постоянной поддержке, возможность использовать коллективные знания и опыт других пользователей Expo и ускорить разработку за счет повторного использования существующих компонентов и решений.

# 2.4 Visual Studio Code

Visual Studio Code, также обычно называемый VS Code, представляет собой редактор исходного кода, созданный Microsoft на платформе Electron Framework для Windows, Linux и macOS. Функции включают поддержку отладки, подсветку синтаксиса, интеллектуальное завершение кода, фрагменты, рефакторинг кода и встроенный Git.

VS Code разработан как легкий, но многофункциональный редактор кода, который предлагает широкий спектр функциональных возможностей для разработчиков на различных языках программирования и платформах. Он обеспечивает интуитивно понятный и настраиваемый пользовательский интерфейс, что делает его легко адаптируемым к индивидуальным предпочтениям и рабочим процессам.

Решение использовать VS Code для этого проекта было основано на нескольких ключевых факторах:

1. Легкий и быстрый: Одним из заметных преимуществ VS Code является его легкий характер. Он разработан для обеспечения плавного и отзывчивого кодирования даже при работе с большими базами кода. В отличие от более тяжелых интегрированных сред разработки (IDE), VS Code предлагает минималистичную, но мощную среду, которая фокусируется на основных инструментах, необходимых для эффективного кодирования. Его эффективная архитектура обеспечивает быстрое время запуска, минимальное потребление ресурсов и плавную производительность, позволяя разработчикам сосредоточиться на написании кода без каких-либо отвлекающих факторов или задержек.

2. Кроссплатформенная совместимость: VS Code создан для работы на нескольких платформах, включая Windows, macOS и Linux. Такая кроссплатформенная совместимость позволяет разработчикам работать в предпочитаемой ими среде операционной системы без ущерба для функциональности или производительности. Разработчики могут легко переключаться между различными платформами, сохраняя при этом один и тот же набор функций и рабочих процессов, обеспечивая согласованность процесса разработки на разных компьютерах. Такая гибкость позволяет разработчикам беспрепятственно сотрудничать с членами команды, которые могут использовать разные операционные системы.

3. Богатая экосистема расширений: VS Code может похвастаться обширной и постоянно растущей экосистемой расширений, которые представляют собой плагины, созданные сообществом и расширяющие его функциональность. Эти расширения охватывают широкий спектр областей, включая языковую поддержку, форматирование кода, отладку, интеграцию с системой контроля версий, управление проектами и многое другое. Разработчики могут легко настроить свой редактор, установив расширения, которые наилучшим образом соответствуют их потребностям, что позволяет им адаптировать среду программирования к своим предпочтениям и работать более эффективно. VS Code Marketplace предоставляет централизованную платформу, где разработчики могут находить и устанавливать расширения всего несколькими щелчками мыши, что еще больше повышает их производительность и расширяет возможности редактора.

4. Надежные инструменты редактирования и рефакторинга: VS Code предлагает полный набор инструментов редактирования и рефакторинга, которые упрощают модификацию кода и обслуживание. Эти инструменты включают расширенные возможности поиска и замены, функции навигации по коду, такие как Перейти к определению и найти все ссылки, и мощные операции рефакторинга, такие как переименование переменных и извлечение методов. Имея в своем распоряжении эти инструменты, разработчики могут эффективно управлять своей кодовой базой, проводить рефакторинг кода для улучшения удобства сопровождения и с легкостью ориентироваться в сложных проектах. Интуитивно понятный и удобный интерфейс VS Code в сочетании с функциями редактирования и рефакторинга позволяет разработчикам писать чистый, хорошо структурированный код и поддерживать его качество на протяжении всего процесса разработки.

5. Интегрированный контроль версий: VS Code легко интегрируется с популярными системами контроля версий, такими как Git, предоставляя разработчикам плавный и интегрированный контроль версий. Встроенная интеграция с Git позволяет разработчикам просматривать различия в файлах, поэтапные изменения, фиксировать код и разрешать конфликты слияния непосредственно в редакторе. Такая тесная интеграция упрощает рабочий процесс разработки, упрощая отслеживание изменений, совместную работу с членами команды и обеспечивая целостность кода на протяжении всего процесса разработки. Разработчики могут получать доступ к общим операциям управления версиями, не переключаясь на внешние инструменты или терминалы, что позволяет им поддерживать бесперебойный рабочий процесс в редакторе.

6. Возможности отладки: VS Code предлагает надежные возможности отладки, которые помогают разработчикам эффективно выявлять и устранять проблемы в их коде. Он предоставляет интуитивно понятный интерфейс для установки точек останова, проверки переменных, пошагового выполнения кода и анализа стеков вызовов. VS Code поддерживает отладку для широкого спектра языков программирования и фреймворков, что делает его универсальным инструментом для отладки как серверного, так и интерфейсного кода. Возможность отладки кода непосредственно в редакторе значительно сокращает время и усилия, необходимые для диагностики и исправления ошибок, улучшая общий опыт разработки и ускоряя цикл разработки.

7. Встроенный терминал: VS Code включает в себя встроенный терминал, который позволяет разработчикам запускать инструменты командной строки и скрипты, не выходя из редактора. Эта функция устраняет необходимость переключаться между редактором и отдельным окном терминала, обеспечивая оптимизированную и унифицированную среду разработки. Встроенный терминал поддерживает различные оболочки и интерфейсы командной строки, позволяя разработчикам выполнять процессы сборки, запускать тесты или выполнять другие операции командной строки непосредственно в редакторе. Такая бесшовная интеграция повышает производительность разработчика и устраняет необходимость во внешних терминалах или дополнительных инструментах для выполнения рутинных задач командной строки.

8. Широкие возможности настройки: VS Code предлагает высокую степень настройки, позволяя разработчикам адаптировать редактор к своим конкретным предпочтениям и рабочим процессам. Разработчики могут настроить внешний вид редактора, выбирая из широкого спектра тем, цветовых схем и наборов значков. Они также могут настраивать сочетания клавиш, правила форматирования кода и поведение редактора в соответствии со своим стилем кодирования. Настройки VS Code легко настраиваются и могут быть персонализированы как на уровне пользователя, так и на уровне рабочей области, обеспечивая гибкость и адаптируемость к индивидуальным потребностям и потребностям команды.

9. Активное сообщество и поддержка: VS Code извлекает выгоду из динамичного сообщества разработчиков, которые вносят свой вклад в его постоянное совершенствование и делятся своими знаниями и расширениями. Ориентированный на сообщество характер VS Code гарантирует разработчикам доступ к множеству ресурсов, руководств и обсуждений для устранения неполадок, открытия новых функций и изучения передового опыта. Кроме того, корпорация Майкрософт предоставляет регулярные обновления и поддержку, гарантируя, что VS Code остается надежным и современным редактором.

# 2.5 Выводы по главе

В данной главе подробно рассматриваются основные технологии и инструменты, используемые при разработке мобильных приложений. В частности, в нем рассматривается React Native, широко распространенный фреймворк, позволяющий разработчикам создавать кроссплатформенные мобильные приложения с использованием JavaScript и React. Также объясняется обоснование использования Firebase Cloud Firestore, подчеркиваются его достоинства как гибкого и масштабируемого решения для баз данных NoSQL, которое облегчает синхронизацию данных в режиме реального времени и надежные возможности выполнения запросов. Кроме того, в главе рассматривается Expo - комплексный набор инструментов и сервисов, которые упрощают разработку и развертывание приложений React Native. Наконец, в нем рассказывается об использовании Visual Studio Code (VS Code), универсального и настраиваемого редактора исходного кода, известного тем, что он предлагает легкую, но мощную среду, способствующую эффективному кодированию.