

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Гуманитарный институт
Кафедра информационных технологий
в креативных и культурных индустриях

УТВЕРЖДАЮ

И. о. заведующего кафедрой

_____ М. А. Лаптева

« _____ » _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

Создание приложения для людей с ограниченными возможностями
по зрению с функцией Accessibility

Направление подготовки: 09.03.03 Прикладная информатика

Наименование программы: 09.03.03.30 Прикладная информатика

Руководитель доц., канд. филос. наук П. А. Солкина

Выпускник В. А. Быков

Нормоконтролер И. Р. Нигматуллин

СОДЕРЖАНИЕ

Введение.....	3
1 Теоретические аспекты ограниченных возможностей по зрению и функции доступности в мобильных приложениях.....	6
1.1 Нарушения, приводящие к ограниченным возможностям по зрению и остаточные возможности при них	6
1.2 Виды нарушения зрения, которые приводят к ОВЗ и трудности при использовании мобильных приложений, вызываемые ими	7
1.3 Формирование таблицы функциональных возможностей остаточного зрения и функций доступности в соответствии с ними	16
1.3 Анализ встроенных функций доступности в устройствах на платформе ОС Android	17
1.3 Исследование функциональности существующих приложений для людей с ОВЗ по зрению.....	23
2 Проектирование и разработка приложения для людей с ограниченными возможностями по зрению.....	30
2.1 Создание концепции приложения.....	30
2.2 Разработка требований к функциональности и дизайну	31
2.3 Определение используемых технологий и инструментов разработки.....	33
2.4 Создание макета интерфейса	38
2.5 Проектирование архитектуры приложения	41
2.6 Разработка приложения.....	45
Заключение	55
Список использованных источников	57
Приложение А	63

ВВЕДЕНИЕ

В наступающую эпоху индустрии 4.0 мобильные устройства начинают набирать все большую популярность, так согласно отчету о состоянии цифровой сферы Global Digital 2022 [1] число пользователей мобильных устройств достигает 5.31 миллиарда людей, а прирост за 2021 год, время, когда в связи с пандемией COVID-19, мобильные устройства стали еще более востребованными в повседневной жизни людей, составил более 95 миллионов пользователей. Сейчас мобильные приложения используются во многих областях жизни, начиная от развлечений и игр, и заканчивая бизнесом и научными исследованиями. Одним из важных аспектов при разработке мобильных приложений является учет особенностей и потребностей конечных пользователей и одной из таких групп пользователей, которые нуждаются в специальном подходе при разработке мобильных приложений, являются люди с ограниченными возможностями здоровья по зрению.

Так по данным Всемирной Организации Здравоохранения нарушениями зрения страдают более 2.2 миллиардов человек по всему миру [2], если же рассматривать Российскую Федерацию, то, по состоянию на 1 января 2022 года, инвалидность по зрению имеют более 450 тысяч человек, среди которых 423 тысячи взрослых и 27 тысяч детей [3]. Но даже при таком количестве людей с ограниченными возможностями по зрению около 95% приложений не являются адаптированными для данной категории пользователей [4].

Поэтому, исходя из представленной ранее информации, а именно данных о популярности мобильных устройств и количестве людей, страдающих от нарушений зрения, актуальность темы данной дипломной работы заключается во внесении вклада в создание инклюзивного общества, путем разработки мобильного приложения для людей с ограниченными возможностями по зрению, содержащего функции accessibility. Разработка приложения для людей с ограниченными возможностями по зрению имеет

решающее значение, поскольку оно улучшит жизнь людей с нарушениями зрения, предоставив им инструмент, который позволит им получать доступ к информации и легче взаимодействовать с цифровым миром.

Что же такое функции accessibility? Функции доступности (accessibility) – это функциональность, которая делает мобильное приложение доступным для пользователей с ограниченными возможностями. Это может включать в себя функции, которые облегчают чтение и навигацию для людей с нарушениями зрения, слуха, моторных навыков и других ограничений. Таким образом, разрабатываемое приложение для людей с ограниченными возможностями здоровья по зрению должно не только помогать улучшать жизнь или делать более доступным использование мобильного устройства, но и делать доступной информацию и функциональность приложения, предлагаемые в нем.

Объектом исследования являются мобильные приложения для устройств на базе ОС Android, а в качестве предмета выступают функции accessibility и функциональная составляющая для людей с ОВЗ по зрению.

Целью работы является разработка мобильного приложения для людей с ограниченными возможностями по зрению с функцией Accessibility.

Для достижения цели были поставлены следующие задачи:

1. Рассмотреть теоретические аспекты классификации людей с ограниченными возможностями по зрению и выделить остаточные возможности, опираясь на которые необходимо составить список реализуемых в приложении функций доступности.

2. Проанализировать существующие приложения в исследуемой сфере, а также встроенные функции устройств на платформе операционной системы Android, которые способствуют их использованию людьми с ограниченными возможностями по зрению.

3. Опираясь на проведенный анализ спроектировать и разработать мобильное приложение на платформе ОС Android для людей с ограниченными возможностями по зрению с функциями accessibility.

Чтобы разработать разработать мобильное приложения для людей с ограниченными возможностями по зрению с функцией accessibility необходимо изучить теоретические аспекты видов визуальных нарушений, это нужно для понимания, как люди с ними воспринимают визуальную информацию, выделить функциональные возможности остаточного зрения и на их основе составить таблицу функций доступности, которые должны быть реализованы в приложении. После этого нужно проанализировать встроенные функции доступности в мобильных телефонах под управлением ОС Android, а также рассмотреть представление приложения со схожим функционалом. Это необходимо для составления концепции приложения, после чего начнется сам процесс разработки.

1 Теоретические аспекты ограниченных возможностей по зрению и функции доступности в мобильных приложениях

1.1 Нарушения, приводящие к ограниченным возможностям по зрению и остаточные возможности при них

Зрение является одним из важнейших органов чувств человека. Способность видеть и воспринимать окружающий мир имеет решающее значение для нашей повседневной деятельности, включая работу, общение и социальные взаимодействия. Однако многие люди страдают заболеваниями глаз и нарушением зрения, что может существенно повлиять на качество их жизни.

Нарушение зрения относится к любому состоянию, которое влияет на способность человека видеть. Это может включать частичную или полную слепоту, а также другие состояния, влияющие на зрительное восприятие, такие как дальтонизм или нарушения зрительной обработки. Нарушение зрения может быть вызвано различными факторами, в том числе генетическими условиями, травмами, заболеваниями или возрастными изменениями.

Люди с нарушениями зрения могут сталкиваться с широким спектром проблем в своей повседневной жизни, в зависимости от тяжести и типа нарушения, которое у них есть. Некоторые распространенные проблемы включают трудности с чтением, навигацией в незнакомой среде, распознаванием лиц или объектов и выполнением повседневных задач, таких как приготовление пищи или уборка. Поэтому при разработке приложения важно учитывать, что каждый тип заболевания и нарушения зрения может иметь свои особенности и требования, поэтому необходимо принимать во внимание все возможные варианты, чтобы обеспечить наиболее широкую доступность для пользователей.

Принимая во внимание описанные выше факты, важно рассмотреть основные виды заболеваний и нарушений зрения, а также определить остаточные возможности людей с ограниченными возможностями здоровья

(далее ОВЗ) по зрению, чтобы использовать эту информацию для построения таблицы необходимых функций доступности (accessibility), которые будут реализованы в приложении.

1.2 Виды нарушения зрения, которые приводят к ОВЗ и трудности при использовании мобильных приложений, вызываемые ими

В мире существует множество видов нарушений зрения, которые могут ограничивать способность человека выполнять повседневные задачи. Некоторые из этих нарушений могут быть врожденными, тогда как другие могут развиваться в процессе жизни. Кроме того, некоторые нарушения зрения могут быть временными и вызваны заболеваниями или травмами. В ходе работы для обозначения нарушений зрения будет использоваться термин “визуальные нарушения”. Визуальные нарушения - это группа заболеваний и нарушений зрительной функции, которые могут привести к снижению качества зрения и в некоторых случаях даже к полной потере зрения.

Существует несколько подходов к классификации видов визуальных нарушений. Один из подходов основан на расположении аномалии или повреждения в глазу, таких как нарушения рефракции, катаракта, глаукома и другие. Другой подход основан на виде нарушения, таком как низкое зрение, слепота, дальтонизм, ночная слепота, фотофобия и другие. Кроме того, визуальные нарушения могут быть классифицированы по причине их возникновения, такими как генетические, возрастные, инфекционные, травматические и другие.

В данной работе внимание будет сосредоточено на втором подходе к классификации видов визуальных нарушений, основанном на видах заболеваний и нарушений зрительной функции. При использовании данного подхода выделяют следующие виды визуальных нарушений:

1. Низкое зрение: у человека с низким зрением есть определенный уровень зрения, но его недостаточно для выполнения повседневных задач,

таких как чтение, вождение автомобиля или распознавание лиц. Низкое зрение часто вызывается возрастной дегенерацией сетчатки, глаукомой или диабетической ретинопатией.

2. Слепота: слепота определяется как отсутствие зрения или очень ограниченное зрение. Люди с полной слепотой не могут видеть ничего вообще, тогда как те, у кого частичная слепота, могут различать свет и тень, а также некоторые формы и цвета.

3. Дальтонизм: дальтонизм - неспособность видеть определенные цвета или различать между ними. Самый распространенный тип дальтонизма - дальтонизм красного и зеленого цветов.

4. Ночная слепота: ночная слепота - неспособность видеть в условиях низкой освещенности. Она часто вызывается дефицитом витамина А.

5. Фотофобия: фотофобия — это чувствительность к свету, которая вызывает дискомфорт или боль. Она может быть вызвана различными состояниями, включая мигрень, сотрясение мозга или определенные лекарства.

6. Кортикальное нарушение зрения: кортикальное нарушение зрения (CVI) — это неврологическое состояние, которое влияет на центры обработки зрительной информации в мозгу. У людей с CVI может быть затруднение в интерпретации визуальной информации или распознавании объектов.

7. Пигментный ретинит: пигментный ретинит — это генетическое заболевание, которое влияет на клетки сетчатки. Оно может вызывать туннельное зрение, ночную слепоту и, в конечном итоге, привести к полной слепоте.

8. Катаракта: катаракта — это помутнение хрусталика в глазу, которое может вызывать размытость зрения, блики и чувствительность к свету

Далее необходимо получить больше информации о каждом из представленных видов и привести иллюстрации для лучшего понимания

того, как воспринимают визуальную информацию люди с данными нарушениями.

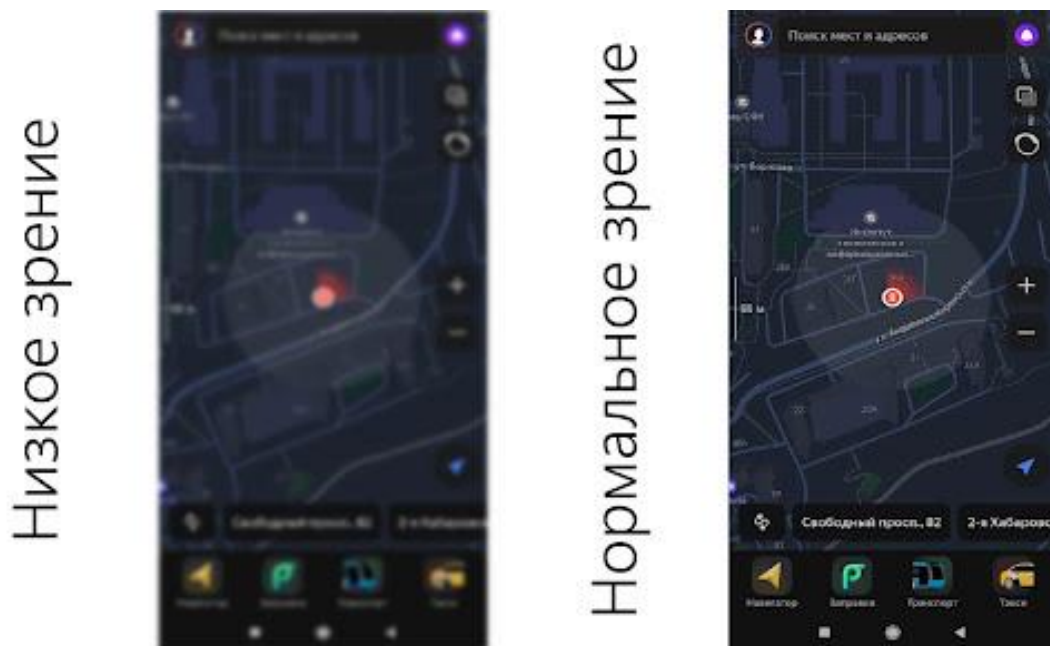


Рисунок 1 – Низкое зрение

Начать следует с низкого зрения, иллюстрация которого представлена на рисунке 1, так как данное нарушение является самым распространенным среди представленных. Низкое зрение, также известное как частичная слепота или потеря зрения, описывает состояние, при котором у человека есть значительное нарушение зрения, но все же есть некоторое оставшееся зрение. Низкое зрение может повлиять на людей любого возраста и может быть вызвано различными факторами, такими как генетические состояния, заболевания глаз или травмы.

В соответствии с определением Всемирной организации здравоохранения (ВОЗ), низкое зрение — это "острота зрения менее 6/18 (20/60) до светового восприятия или поле зрения менее 20 градусов в лучшем глазе с наилучшей возможной коррекцией". Острота зрения относится к четкости зрения и измеряется с помощью таблицы Снеллена, а поле зрения относится к области, которую человек видит, смотря прямо вперед.

Низкое зрение классифицируется на три категории на основе тяжести состояния: легкое, умеренное и тяжелое. Легкое низкое зрение

характеризуется остротой зрения от 20/40 до 20/70, умеренное низкое зрение характеризуется остротой зрения от 20/80 до 20/160, а тяжелое низкое зрение характеризуется остротой зрения от 20/200 до 20/400.

Люди с низким зрением могут столкнуться с трудностями при использовании мобильных приложений, поскольку они могут иметь ограниченный доступ к визуальной информации на экране.

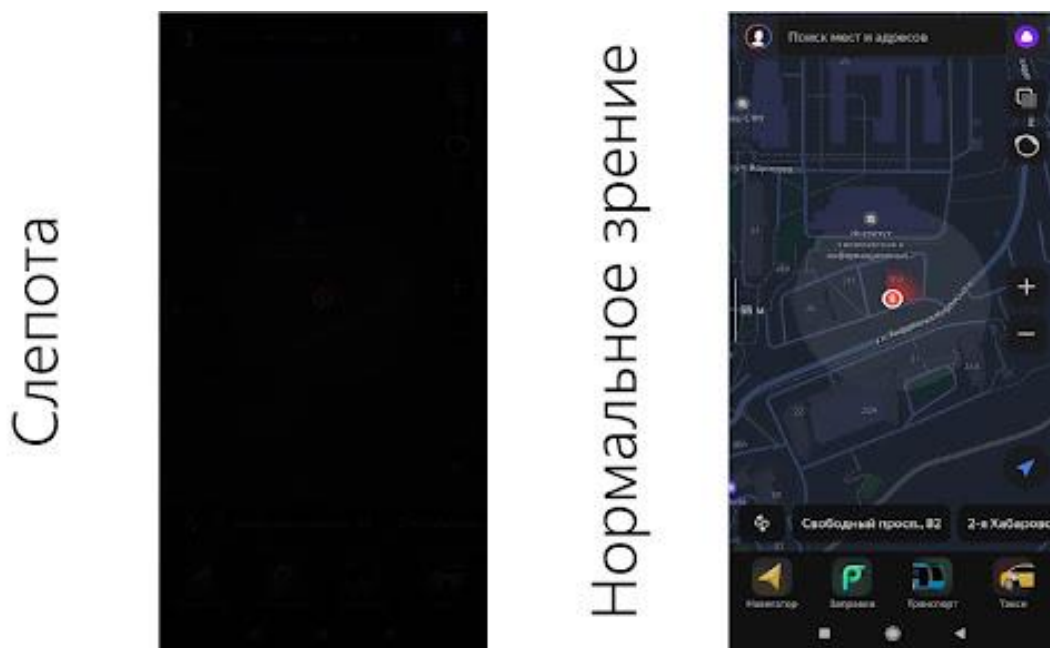


Рисунок 2 – Слепота

Вторым типом будет слепота, иллюстрация которой представлена на рисунке 2, в конкретном примере изображена слепота с сохранением светоощущения, поскольку без него человек не видит ничего. Слепота — это полная потеря зрения или невозможность использовать оставшийся зрительный орган для получения информации о внешнем мире. Кроме того, слепоту можно разделить на две основные категории: "слепота с сохранением светоощущения" и "слепота без светоощущения". Первый тип характеризуется наличием остаточного светоощущения, что позволяет человеку различать свет и темноту, но недостаточно, чтобы видеть объекты и изображения. Второй тип означает полное отсутствие светоощущения и способности видеть окружающий мир.

У людей с низким зрением могут возникнуть такие же трудности, что и у слепых людей. Однако, для слепых людей, особенно в случае полной слепоты, существуют дополнительные трудности, такие как невозможность использовать визуальные элементы, такие как иконки или изображения. Вместо этого, приложение должно предоставлять альтернативные методы взаимодействия.

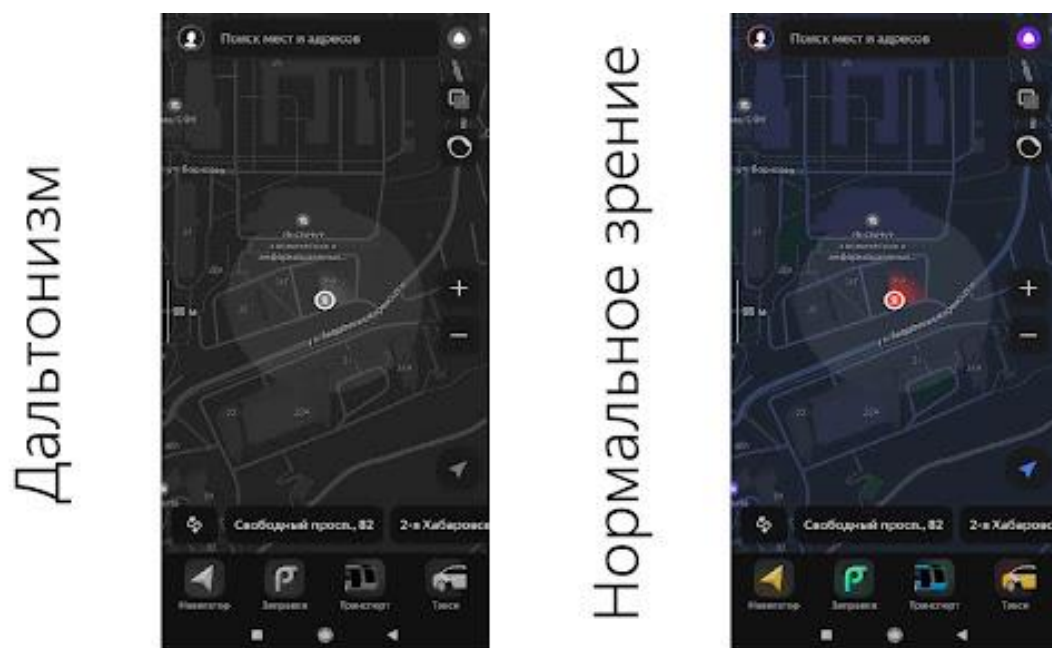


Рисунок 3 – Дальтонизм

Дальтонизм, который представлен на рисунке 3 — это нарушение цветоощущения, которое приводит к тому, что человек не может правильно различать цвета. Это состояние происходит из-за нехватки или неправильного функционирования конусных клеток глаза, которые отвечают за различие цветов. В результате, дальтоник может видеть цвета сильно искаженными или не видеть некоторые цвета вообще.

Существуют разные типы дальтонизма, основные из которых включают дихромазию и монохромазию. Дихромазия — это состояние, когда у человека отсутствует один из трех типов конусных клеток, необходимых для различения цветов, что приводит к тому, что дальтоник видит только два основных цвета. Монохромазия — это состояние, когда у человека все три

типа конусных клеток глаза не функционируют правильно, что приводит к тому, что он видит все цвета в оттенках серого.

Дальтонизм может создавать проблемы при использовании мобильных приложений, которые основаны на цветовой гамме. Например, дальтоник может не смочь правильно различить цвета кнопок или индикаторов состояния, что может привести к неправильному использованию приложения.

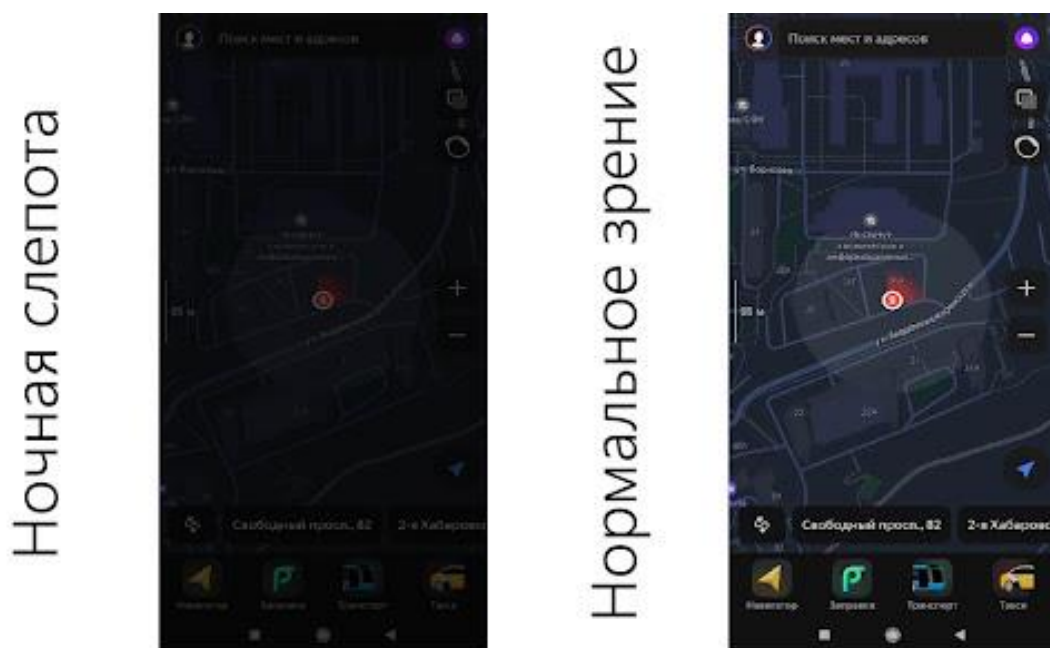


Рисунок 4 – Ночная слепота

Ночная слепота и фотофобия - две крайности. Ночная слепота, изображенная на рисунке 4 — это состояние, когда человек не может видеть при недостаточном освещении, даже если его зрение в целом не имеет проблем. Это может произойти из-за нехватки родопсина - светочувствительного белка, находящегося в глазах. Родопсин необходим для правильной работы фоторецепторов глаза, называемых палочками, которые активируются при низком освещении.

Фотофобия



Нормальное зрение



Рисунок 5 – Фотофобия

Фотофобия, представленная на рисунке 5, с другой стороны, — это чувство болезненной чувствительности к свету. Оно может проявляться как дискомфорт при ярком свете, а также головные боли и агрессивное сужение зрачков.

Кортикальное нарушение



Нормальное зрение

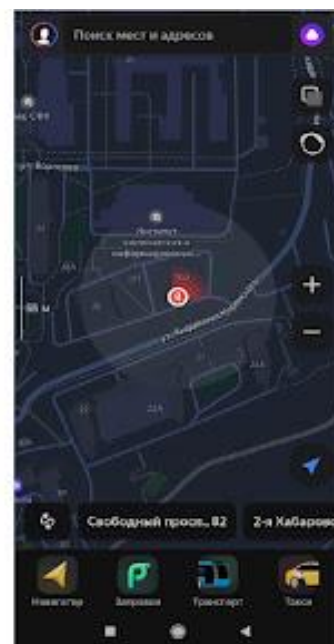


Рисунок 6 – Кортикальное нарушение зрения

Кортикальное нарушение зрения, представленное на рисунке 6, также известное как церебральная амблиопия, представляет собой нарушение

зрения, вызванное повреждением зрительной коры головного мозга. Это может произойти вследствие различных причин, включая травму, инфекцию, инсульт или опухоль.

Симптомы кортикального нарушения зрения могут варьироваться в зависимости от того, какая область зрительной коры была повреждена. В некоторых случаях человек может чувствовать, что некоторые области зрительного поля исчезают, тогда как в других случаях зрительное восприятие может быть сильно искажено. При разработке мобильных приложений для людей с кортикальным нарушением зрения важно учитывать их особенности и предоставлять специальные возможности.

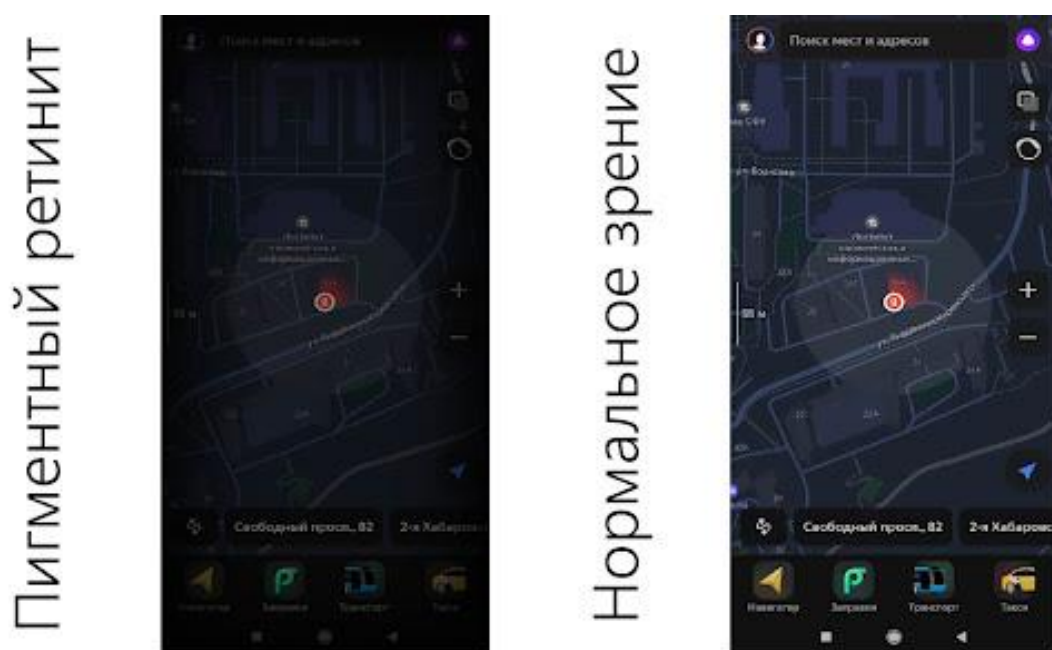


Рисунок 7 – Пигментный ретинит

Пигментный ретинит (также известный как дистрофия сетчатки), изображенный на рисунке 7 — это группа заболеваний глаз, которые влияют на сетчатку. Они приводят к потере или ухудшению зрения, особенно в периферической области поля зрения. У людей с пигментным ретинитом могут возникнуть следующие симптомы: потеря цветоощущения, снижение остроты зрения, проблемы с ночным зрением, туннельное зрение.

Пигментный ретинит может привести к снижению чувствительности к определенным цветам, особенно к красному и зеленому. Это может

затруднить использование мобильных приложений, которые зависят от цветовой индикации, например, приложений, основанных на цветовой разметке, или использования иконок различных цветов для представления различных функций. Также при данном заболевании область зрения может быть сильно ограничена, что может затруднить взаимодействие с мобильным приложением. Визуальные элементы могут быть слишком мелкими, чтобы быть видимыми, или находиться за пределами области зрения, что может привести к пропуску важной информации. У людей с пигментным ретинитом может возникать трудность с контрастом, что может затруднить различение объектов, особенно при низком освещении. Это может привести к трудностям с использованием мобильных приложений, особенно тех, которые требуют четкого различия между различными элементами.

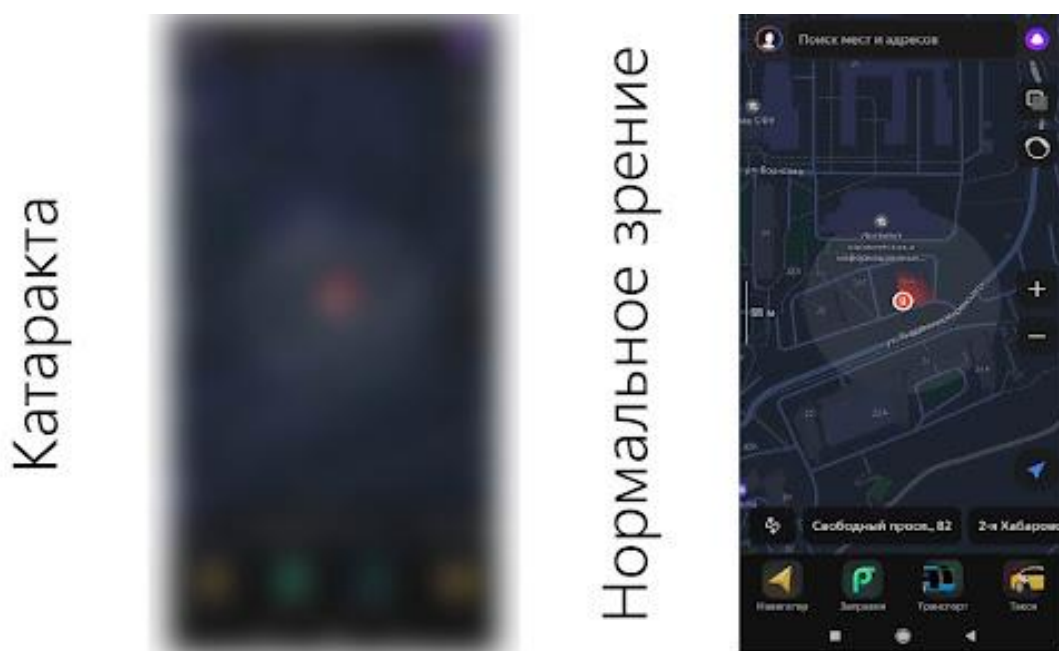


Рисунок 8 – Катаракта

Катаракта (представлена на рисунке 8) — это заболевание глаз, при котором хрусталик, расположенный за зрачком, темнеет или становится мутным, что может привести к ухудшению зрения и даже к слепоте.

У людей с катарактой могут возникать трудности при использовании мобильных приложений, например, помутнение хрусталика может привести к снижению яркости и контрастности изображений на экране. Кроме того,

люди с катарактой могут испытывать трудности с фокусировкой на мелких деталях, таких как текст или изображения.

1.3 Формирование таблицы функциональных возможностей остаточного зрения и функций доступности в соответствии с ними

Ранее, в предыдущем подпункте, были разобраны основные виды визуальных нарушений и продемонстрировано видение мира людьми, страдающими от них. Но даже при всех ограничениях, накладываемых нарушениями зрения, у человека остаются остаточные возможности. Остаточные возможности в контексте ОВЗ (ограниченных возможностей здоровья) относятся к способности человека компенсировать свои ограничения и добиваться успеха в различных сферах жизни, несмотря на них.

В контексте мобильных приложений для людей с ограниченными возможностями зрения, слуха, моторики и т.д., остаточные возможности — это те возможности, которыми обладает пользователь, несмотря на свои ограничения. Например, человек с ограниченными возможностями зрения может иметь остаточные возможности в использовании голосовых команд или увеличении размера шрифта, чтобы улучшить восприятие информации на экране. Человек с ограниченными возможностями слуха может использовать вибрацию вместо звуковых уведомлений.

Остаточные возможности в мобильных приложениях помогают людям с ограниченными возможностями получать доступ к информации и использовать приложения наравне со всеми остальными пользователями. Для разработчиков приложений это означает, что они должны учитывать эти остаточные возможности при проектировании и тестировании приложений, чтобы обеспечить максимально возможный уровень доступности и удобства использования для всех пользователей.

Поскольку темой дипломной работы является разработка мобильного приложения для людей с ограниченными возможностями по зрению с функциями accessibility (доступности) - необходимо определить эти самые

функции доступности. Для этого будет составлена таблица, в которой представлена следующая информация:

9. Виды визуальных нарушений: указываются виды визуальных нарушений, описанных ранее

10. Характер нарушения основных компонентов остаточного зрения: Описаны основные нарушения зрительной функции

11. Функциональные возможности остаточного зрения: Выведены оставшиеся возможности зрительной функции, основываясь на которых необходимо определить используемые функции доступности

12. Предлагаемые функции доступности: функции accessibility (доступности), которые с помощью функциональных возможностей остаточного зрения предоставляют информацию пользователям мобильного приложения.

Сформированная таблица представлена в приложении А.

1.3 Анализ встроенных функций доступности в устройствах на платформе ОС Android

В настоящее время мобильные устройства, такие как смартфоны и планшеты, стали неотъемлемой частью нашей жизни, и мы используем их для многих задач, включая работу, общение, развлечения и многое другое. Однако, не все пользователи могут иметь доступ к полному функционалу мобильных устройств, так как у них могут быть физические или когнитивные ограничения.

Одной из основных проблем, с которыми сталкиваются люди с ограничениями, является доступность мобильных приложений. В связи с этим, существует необходимость в разработке специальных функций доступности, которые позволят использовать мобильные устройства всем пользователям.

ОС Android является одной из самых популярных мобильных платформ в мире, так на момент 2022 года в Америке пользователей Android

насчитывалось более 74 процентов населения[7], и она предлагает встроенные функции доступности для поддержки пользователей с ограничениями. В данном разделе будет проведен анализ встроенных функций доступности в устройствах на платформе ОС Android, а также их эффективность для обеспечения доступности мобильных приложений для всех пользователей. Будут рассмотрены функции доступности для людей с различными ограничениями, такими как слепота, низкое зрение, цветовая слепота, нарушения когнитивных функций и др.

Чтобы найти меню специальных возможностей нужно перейти в настройки устройства, там найти расширенные настройки и нажать на пункт меню “специальные возможности”. Открывшийся экран настроек мы видим на рисунке 9.

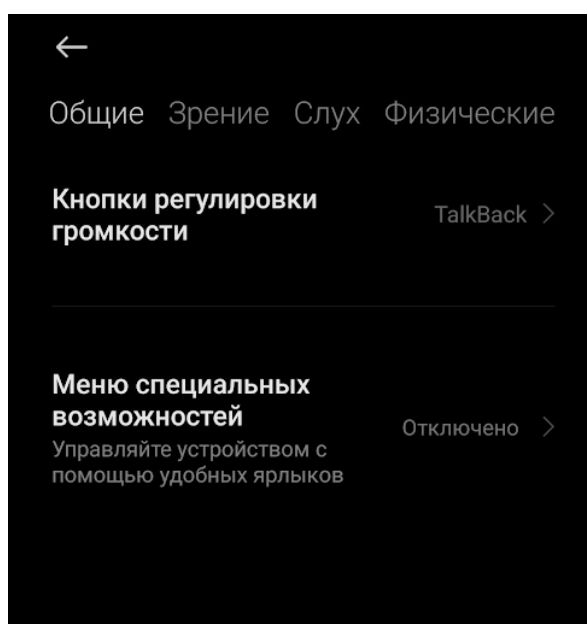


Рисунок 9 – Раздел “Общие” специальных возможностей

В данном разделе представлено 4 пункта меню: “Общие” (текущий раздел), “Зрение”, “Слух”, “Физические”. Разберем каждый по порядку.

Данный раздел предоставляет возможность включения меню специальных возможностей, а именно, на сенсорной панели управления внизу экрана появляется значок, который включает меню с ярлыками главных функций телефона, что позволяет сократить количество

необходимых действий для использования той или иной функции мобильного устройства.

На рисунке 10 представлен раскрытый пункт меню “Кнопки регулировки громкости”. Задача данного пункта заключается в назначении на кнопку регулировки громкости функционала быстрого запуска одной из представленных функций.

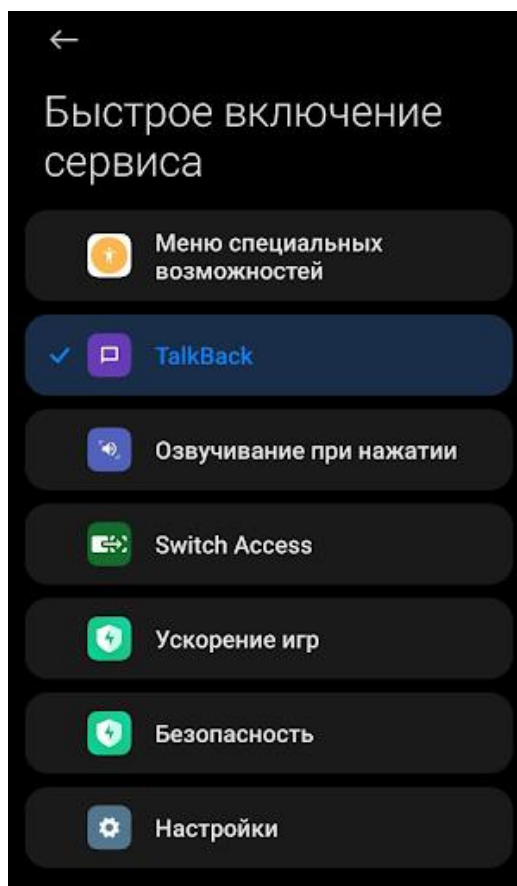


Рисунок 10 – “Кнопки регулировки громкости”

Из функций обеспечения доступности мобильного устройства тут присутствуют “TalkBack”, “Озвучивание при нажатии” и открытие меню специальных возможностей (такое же меню открывает значок из пункта, описанного выше).

TalkBack — это функция озвучивания текста на экране, что позволяет людям с ОВЗ по зрению или больным дислексией воспринимать представленную информацию. Чтобы перейти от одного объекта на экране к другому необходимо провести по экрану вправо или влево, чтобы

активировать - нажать дважды. Озвучивание при нажатии позволяет получить аудиопредставление любого объекта на экране смартфона.

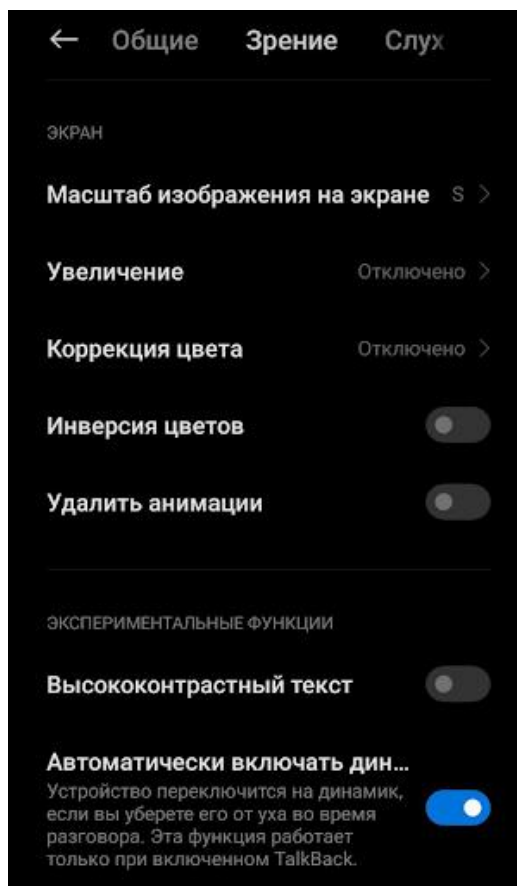


Рисунок 11 – Раздел “Зрение”

На рисунке 11 представлен раздел “Зрение”, в котором собраны функции доступности для людей с нарушениями зрения. Тут можно увеличить размер всех объектов на экране, включая любые значки, изображения и текст, что является одной из основных функций, обеспечивающей доступность использования для людей с ОВЗ по зрению. Также, если увеличение по-умолчанию мешает восприятию информации, то можно увеличивать отдельные объекты тройным касанием экрана или по нажатию на заранее определенную кнопку.

Инверсия цвета меняет местами темные и яркие цвета, чтобы способствовать лучшему восприятию информации и избеганию негативных эффектов определенному кругу лиц с ОВЗ.

Удаление анимаций и повышение контраста текста позволяет убрать отвлекающие факторы для концентрации внимания на основной информации.



Рисунок 12 – “Настройки субтитров” в разделе “Слух”

На рисунке 12 представлен единственный пункт меню раздела “Слух”, который позволяет включить субтитры к аудио и видеоматериалам. У субтитров можно выбрать язык, размер их текста и определенные стилевые настройки. Но стоит отметить, что хоть такая функция и присутствует, но работает она не со всеми медиа, а только в приложениях, которые изначально реализовали данную функцию.

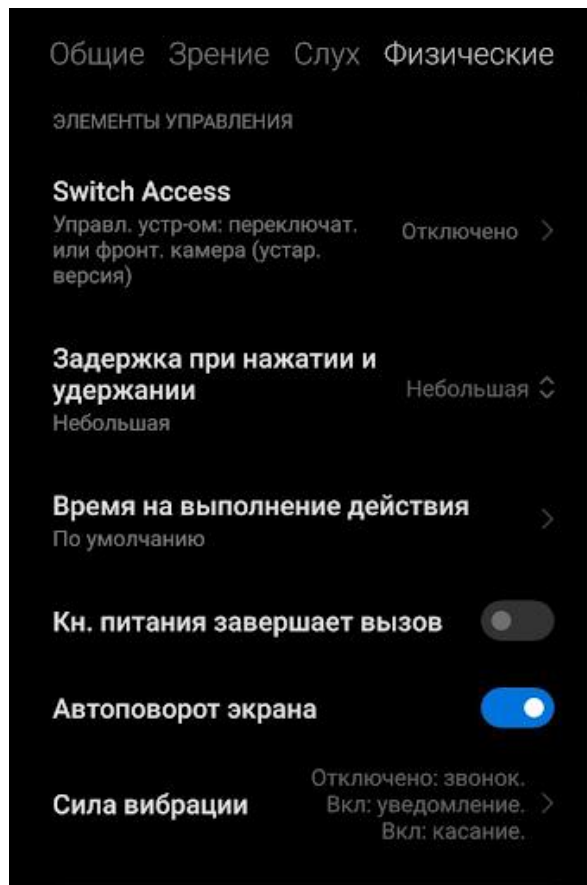


Рисунок 13 – Раздел “Физические”

Раздел физические предоставляет функции доступности для людей с ограниченными возможностями опорно-двигательного аппарата. Упор функций в данном разделе идет на управление устройством с помощью специальных переключателей, которые позволяют выбрать объекты на экране, прокручивать страницы, выводить текст и т.д. (Switch Access).

Отдельное внимание стоит уделить функциям, которые регулируют задержку нажатия и время на выполнение действий. Данные функции позволяют людям с ОВЗ пользоваться устройством без дискомфорта, а может и в общем пользоваться им.

Подводя итоги, можно сказать, что операционная система Android имеет достаточно функций доступности, чтобы обеспечить комфортное использование устройства людьми с ОВЗ. Но есть и исключения, которые заключается в том, что некоторые функции хоть и присутствуют, но не выполняют свои задачи в полной мере или вообще (на примере функции создания субтитров). Для поддержания их работоспособности необходимо

участие разработчиков приложений, которые будут реализовывать поддержку функций доступности в своих приложениях.

1.3 Исследование функциональности существующих приложений для людей с ОВЗ по зрению

Анализ существующих приложений для людей с ограниченными возможностями по зрению является важным этапом в создании более доступной и инклюзивной цифровой среды. Люди с ограниченными возможностями зрения могут столкнуться с трудностями при использовании приложений, которые не учитывают их потребности. В то же время, существует множество приложений, которые специально созданы для улучшения качества жизни людей с ограниченными возможностями зрения.

Кроме того, анализ приложений позволяет определить их функциональность и полезность для конкретной аудитории. Некоторые приложения могут содержать функции, которые особенно важны для людей с ограниченными возможностями зрения, например, функцию чтения текста или поддержку голосового управления. Однако, может быть и так, что некоторые приложения не содержат нужных функций, которые были бы полезны для людей с ограниченными возможностями зрения.

В контексте данной работы анализ сторонних приложений выполняется с целью выведения концепции функциональной составляющей создаваемого мобильного приложения для людей с ограниченными возможностями по зрению с функциями доступности.

Существующие мобильные приложения для людей с ограниченными возможностями по зрению можно разделить на следующие категории:

1. Навигаторы
2. Определители купюр
3. Распознаватели знаков и людей
4. Приложения для работы с волонтерами
5. Путешествия

6. Тифлокомментирование
7. Приложения с виртуальной реальностью
8. Онлайн банки

Что хотелось бы отметить, очень большой процент приложений более не поддерживается, какие-то перестали обновляться для использования на устройствах с более новыми версиями операционных систем, а какие полностью убраны из публичного доступа. Среди которых и находится следующее приложение.

Этим приложением является определитель купюр, который был благотворительным проектом компании “Мегафон” и имел следующее описание на PlayMarket: “Приложение позволяет слепому и слепоглохому пользователю легко и быстро понять, какая именно банкнота находится перед ним. Для определения купюр приложение использует интеллектуальные алгоритмы. Приложение непригодно для определения фальшивых банкнот или банкнот для розыгрышей! Принцип работы: нужно запустить приложение, навести камеру смартфона на купюру, коснуться экрана для начала распознавания, результат выводится в визуальной (крупные контрастные цифры), звуковой (проговаривание голосовым помощником) и тактильной (специальные вибрации для каждой купюры) формах. Приложение может использоваться как слепыми, так и слепоглохими людьми.” [7]

К сожалению, поддержка данного приложения была завершена и установить его на новые версии операционной системы Android невозможно

Смотря на раздел навигаторов выбор пал на приложение ViaOpta Nav. ViaOpta Nav - это приложение для навигации, разработанное специально для людей с ограниченными возможностями зрения. Оно обеспечивает возможность планирования маршрутов, поиска мест и ориентации в пространстве, используя голосовые команды и звуковые сигналы.

Одной из ключевых функций приложения ViaOpta Nav является функция голосовой навигации, которая сообщает пользователю, каким путем

нужно идти, сколько осталось времени до точки назначения, а также предупреждает о возможных препятствиях на пути. Кроме того, приложение имеет функцию вибрационного уведомления, которая дополнительно помогает пользователям определить направление движения.

Приложение также содержит функцию поиска мест, таких как рестораны, магазины, банки и другие объекты, которые могут быть полезны для пользователей. Функция поиска мест обеспечивает подробную информацию о каждом объекте, включая адрес, контактную информацию и отзывы.

Кроме того, ViaOpta Nav имеет функцию сохранения маршрутов, что позволяет пользователям сохранять свои любимые маршруты и быстро находить нужное место в будущем. Также в приложении предусмотрена функция поддержки языковых настроек, которая позволяет пользователю выбрать язык голосовой навигации.

В целом, функциональность приложения ViaOpta Nav предназначена для облегчения жизни людей с ограниченными возможностями зрения, обеспечивая им доступную навигацию и возможность быстрого и комфортного перемещения в пространстве. Интерфейс данного приложения представлен на рисунке 14.



Рисунок 14 – Интерфейс приложения ViaOpta Nav

Что сразу бросается в глаза при открытии приложения - в нем отсутствует русский язык, хоть и при поиске мест на карте он все же работает. Настройки приложения позволяют определить разные виды уведомлений, чтобы человеку было проще ориентироваться. Но как итог - для использования в условиях России это приложение не адаптировано: карты имеют сравнительно большой масштаб, нет русской локализации и не работает часть меток на карте.

В одной из самых полезных категорий - "Распознаватели купюр и знаков", внимание привлекло приложение Lookout - Assisted Vision. Это приложение, разработанное для помощи людям с ограниченными возможностями зрения. Оно предлагает ряд функций, которые облегчают повседневную жизнь пользователей и помогают им получить дополнительную информацию о мире вокруг.

Одной из ключевых функций приложения Lookout является функция "Explore". Она позволяет пользователям получать аудио-обратную связь о том, что находится перед ними, например, описывать предметы, тексты или людей. Это особенно полезно при чтении меток на продуктах, опознавании денежных купюр или просто для получения информации о окружающей обстановке.

Приложение также предлагает функцию "Scan", которая позволяет пользователям сканировать и распознавать текст на фотографиях или в реальном времени. Это может быть полезно при чтении документов, книг, меню или других печатных материалов. Пользователь просто направляет камеру устройства на текст, и Lookout анализирует его, затем озвучивает результаты.

Еще одна важная функция приложения Lookout — это "Currency Recognition", то есть распознавание валюты. С ее помощью пользователи могут легко определить номинал банкноты. Приложение использует камеру устройства для захвата изображения денежной купюры и затем озвучивает информацию о ней.

Моим же фаворитом является приложение Be My Eyes это бесплатное приложение для людей с ограниченными возможностями зрения, которое соединяет пользователей с добровольцами из всего мира. Оно предлагает несколько функций, которые могут помочь людям с ограниченными возможностями зрения в повседневной жизни.

Одной из ключевых функций приложения является возможность видеозвонка с добровольцем. Пользователи могут попросить помощи в распознавании предметов, нахождении конкретного места в комнате, чтении меток на продуктах, выборе одежды или просто для описания окружающей обстановки. Добровольцы, в свою очередь, могут помочь, предоставив пользователям аудио-описание с помощью камеры своего смартфона.

Помимо видеозвонков, приложение также предоставляет возможность отправки текстовых сообщений. Это может быть особенно полезно в ситуациях, когда пользователи не могут использовать звук, или если они не хотят или не могут общаться с добровольцами через видеозвонок.

Be My Eyes также предлагает функцию "Specialized Help", то есть возможность связаться с профессионалами, например, оптиками или медицинскими работниками, чтобы получить конкретную медицинскую или зрительную помощь.

В приложении также есть функция "Call History", которая позволяет пользователям просматривать и сохранять свои предыдущие запросы помощи, что может быть полезно для будущих ситуаций. Скриншоты интерфейса приложения представлены на рисунке 15.

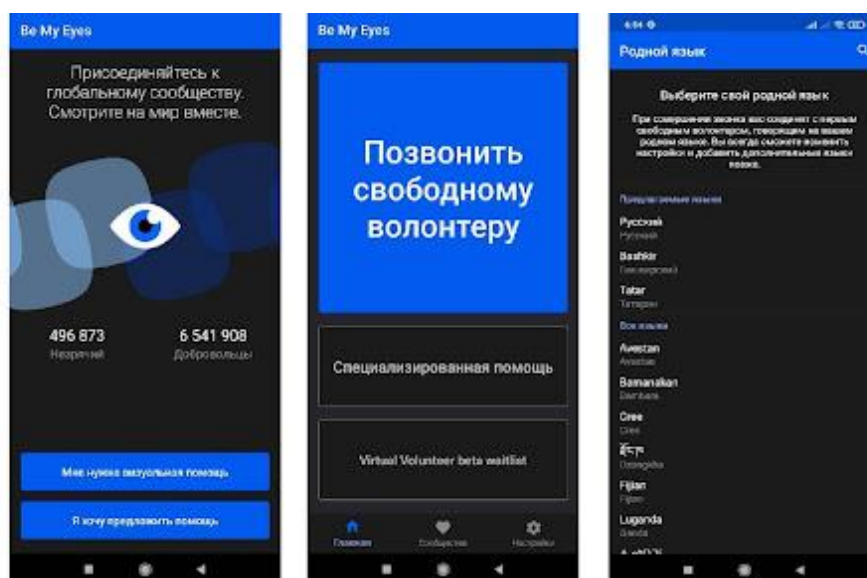


Рисунок 15 – Интерфейс приложения Be My Eyes

В целом, функциональность рассмотренных приложений нацелена на обеспечение поддержки и помощи людям с ограниченными возможностями зрения, предоставляя им узконаправленные возможности, за исключением приложения Be My Eyes, оно позволяет покрывать большинство потребностей людей с ограниченными возможностями здоровья.

Но в рамках этой работы наибольший интерес представляет приложение Voice Access — это приложение, разработанное Google, которое позволяет пользователям управлять своим устройством с помощью голосовых команд. Это особенно полезно для людей с ограниченными возможностями, таких как моторные нарушения или ограниченная моторная функция.

С помощью Voice Access пользователи могут использовать голосовые команды для запуска приложений, открытия ссылок в браузере, набора текста и других функций устройства. Это также позволяет пользователю контролировать элементы интерфейса, такие как кнопки, поля ввода и другие элементы, с помощью голосовых команд.

Приложение обучается голосовым командам пользователя и предоставляет рекомендации по их использованию. Например, пользователь

может сказать "нажми на кнопку меню" или "открой приложение Gmail", чтобы управлять своим устройством.

Voice Access также имеет функцию "Навигация по экрану", которая позволяет пользователю перемещаться по интерфейсу устройства с помощью голосовых команд. Это может быть особенно полезно для людей с ограниченной моторной функцией, которым трудно использовать экранное управление.

Одна из дополнительных функций приложения - возможность использования голосовых команд для настройки устройства. Пользователи могут настроить свои предпочтения и управлять своими устройствами, говоря: "Открой настройки" и используя голосовые команды для настройки своих приложений и функций.

В целом, функциональность приложения Voice Access нацелена на упрощение управления устройствами и обеспечение доступа к ним людям с ограниченными возможностями, используя голосовые команды. Это позволяет пользователям управлять своими устройствами, не используя экранный интерфейс, что может быть особенно полезно для людей с моторными нарушениями или ограниченной моторной функцией.

Стоит отметить, что на русском языке данное приложение работает крайне плохо, а также, во многих программах, оно не позволяет взаимодействовать с элементами интерфейса в сторонних приложениях. Аналогов данному приложению в открытом доступе нет.

Итогом работы над данным пунктом является проведенный анализ функциональности определенных приложений для людей с ограниченными возможностями по зрению, а также ситуация с ними в целом. Благодаря этому сложилась теоретическая база для работы над созданием концепции собственного разрабатываемого приложения.

2 Проектирование и разработка приложения для людей с ограниченными возможностями по зрению

2.1 Создание концепции приложения

Как и говорилось ранее, наше общество не стоит на месте в плане инклюзивности, стали все чаще и чаще проводиться мероприятия, направленные на включение в общественную жизнь людей с ограниченными возможностями здоровья, мероприятия, которые рассказывают другим, здоровым людям, что есть те, кто имеет дополнительные потребности, которые нужно учитывать. Начинают модернизироваться школы и другие образовательные учреждения, в которые дети с ограниченными возможностями здоровья могут практически на равных, а главное - в комфортных социальных условиях, получать образование и социальный опыт. А также все больше объектов инфраструктуры повышают уровень своей доступности для разных категорий граждан.

Но есть одно но, такие мероприятия, досуговые объекты и объекты инфраструктуры редко освещаются в крупных СМИ или они не становятся достаточно крупным инфоповодом, чтобы попасть на первые страницы, поэтому о них знает не столь большое количество человек.

Поэтому, опираясь на анализ видов визуальных нарушений, исследование встроенных функций доступности в устройствах на базе операционной системы Android, а также рассмотрение сторонних приложений для людей с ограниченными возможностями здоровья по зрению было принято решение делать упор в функциональной составляющей разрабатываемого приложения на информирование людей о мерах и мероприятиях, направленных на повышение уровня инклюзивности нашего общества. А также, чтобы людям с ОВЗ по зрению было проще получать данную информацию и пользоваться своим мобильным телефоном, а о их важности в нашем мире говорилось ранее, было решено включить в данное приложение функционал голосового помощника.

Основной функцией данного мобильного приложения будет являться информирование целевой аудитории об открытии новых объектов гражданской инфраструктуры с повышенной доступностью или модернизации старых, которая привела к тем же результатам, а также проводимых мероприятиях для лиц с ограниченными возможностями по зрению. Информация будет подаваться с помощью новостей, а также удобным списком с выборкой доступных мест для посещения людьми с ОВЗ по зрению. Чтобы сделать данное приложение более доступным в него будут не только включены основные функции доступности, но и интегрирован голосовой помощник, который сделает обращение с мобильным устройством более доступным для определенной группы лиц.

Поскольку данное приложение должно быть обновляемым и легко модерируемым - лучшим вариантом будет построение его на клиент-серверной архитектуре, при которой все данные будут храниться на сервере и отправляться по запросам на устройства пользователей. Это позволит постоянно пополнять базу актуальной информацией. Исходя из сказанного выше - для разработки данного приложения как нельзя подходит язык программирования JavaScript, а конкретно его библиотека React Native, о которой будет рассказано в используемых технологиях. Для серверной части будет наиболее объективным использовать одно из представленных готовых решений, наподобие Firebase.

2.2 Разработка требований к функциональности и дизайну

Разработка мобильного приложения, ориентированного на людей с ограниченными возможностями зрения, требует особого внимания к функциональности и дизайну. Целевая аудитория, состоящая из пользователей с различными видами визуальных нарушений, имеет особые потребности и требования, которые должны быть учтены на этапе разработки.

Именно поэтому был проведен анализ функциональных возможностей остаточного зрения, исходя из которого и подводя итоги сформированной таблицы можно выделить следующее.

При разработке функциональности приложения необходимо обеспечить доступность и удобство использования для людей с ограничениями зрения. Это означает, что приложение должно предоставлять возможность получения информации через альтернативные каналы, такие как голосовой помощник или аудиоинтерфейс. Кроме того, следует предусмотреть функции, позволяющие пользователю настраивать интерфейс на основе своих индивидуальных предпочтений и возможностей.

Важным аспектом разработки мобильного приложения для людей с ограниченными возможностями по зрению является дизайн. Интерфейс приложения должен быть интуитивно понятным, с учетом специфических потребностей и ограничений пользователей. Для этого следует использовать четкую и ясную навигацию, контрастные цветовые схемы и достаточно большие элементы управления, чтобы облегчить чтение и навигацию по приложению.

Также необходимо обратить внимание на доступность контента внутри приложения. Текстовая информация должна быть предоставлена в удобном для чтения формате, с возможностью увеличения шрифта или использования специальных шрифтов для людей с ограничениями зрения. Важно также предоставить возможность прослушивания текста с помощью синтеза речи или аудиозаписей.

Что же касается функциональной составляющей. Приложение должно предоставлять доступ к новостным статьям и списку мест с повышенным уровнем доступности для людей с ограниченными возможностями здоровья по зрению. А также, с помощью голосового ассистента, делать и само приложение более доступным, путем использования альтернативных каналов коммуникации и восприятия визуальной информации.

2.3 Определение используемых технологий и инструментов разработки

Говоря о технологиях разработки нельзя забывать и о технологиях, с помощью которых происходит проектирование разрабатываемого продукта. В данном кейсе для проектирования использовался продукт фирмы Adobe, а именно Adobe Illustrator.

Adobe Illustrator является мощным инструментом для проектирования интерфейса мобильного приложения. Он предоставляет широкий набор функций и возможностей, которые помогают дизайнерам создавать привлекательные и интуитивно понятные интерфейсы. Одним из главных преимуществ Adobe Illustrator при проектировании интерфейса мобильного приложения является его векторная природа. Векторная графика позволяет создавать масштабируемые элементы интерфейса, которые могут быть легко изменены без потери качества. Это особенно полезно при разработке для различных устройств с разными разрешениями экрана.

В Adobe Illustrator доступны многочисленные инструменты для создания форм, иконок, кнопок и других элементов интерфейса. Дизайнеры могут использовать инструменты для рисования, обрезки, выравнивания и создания эффектов, чтобы создать современный и стильный интерфейс мобильного приложения.

Интеграция Adobe Illustrator с другими программами Adobe, такими как Adobe XD или Photoshop, облегчает рабочий процесс при создании интерфейса мобильного приложения. Дизайнеры могут создавать и редактировать векторные элементы в Illustrator, а затем легко экспортировать их в другие программы для создания прототипов или дополнительной обработки. Adobe Illustrator также предоставляет возможность создавать анимированные элементы интерфейса мобильного приложения. С помощью инструментов анимации и временной шкалы можно добавить движение и эффекты к элементам интерфейса, что поможет создать более привлекательный и динамичный пользовательский опыт.

Далее хотелось сказать про текстовый редактор - это именно то место, в котором люди, разрабатывающие программное обеспечение, проводят наибольшее количество времени. В ходе разработки данного проекта использовался VisualStudio Code. Это интегрированная среда разработки (IDE), которая обладает широким спектром функций и расширений, упрощающих процесс создания мобильных приложений.

Одним из главных преимуществ Visual Studio Code при разработке мобильных приложений является его поддержка различных языков программирования, таких как JavaScript, TypeScript, HTML, CSS и многих других. Разработчики могут использовать синтаксическое подсвечивание, автодополнение кода и другие инструменты, которые значительно повышают эффективность и точность написания кода.

Он также обладает интегрированными средствами отладки, что позволяет разработчикам мобильных приложений легко отслеживать и исправлять ошибки в своем коде. Отладчик визуализирует выполнение программы, предоставляет информацию о переменных и позволяет устанавливать точки останова, что помогает значительно ускорить процесс отладки.

Благодаря разнообразным расширениям, доступным для данного текстового редактора, разработчики мобильных приложений могут значительно расширить возможности IDE в соответствии с своими потребностями. Эти расширения включают поддержку фреймворков, инструменты для автоматизации задач, системы контроля версий и многие другие, что делает VS Code идеальным инструментом для разработки мобильных приложений различных типов и технологий.

Важным аспектом использования Visual Studio Code при разработке мобильных приложений является его легкая настраиваемость. Разработчики могут настроить расположение панелей, цветовые схемы, шрифты и другие параметры, чтобы создать комфортное рабочее окружение, соответствующее их предпочтениям. Это помогает повысить продуктивность и удовлетворение

от работы, а также создать более эффективный рабочий процесс при разработке мобильных приложений. На рисунке 16 представлен интерфейс VS Code.

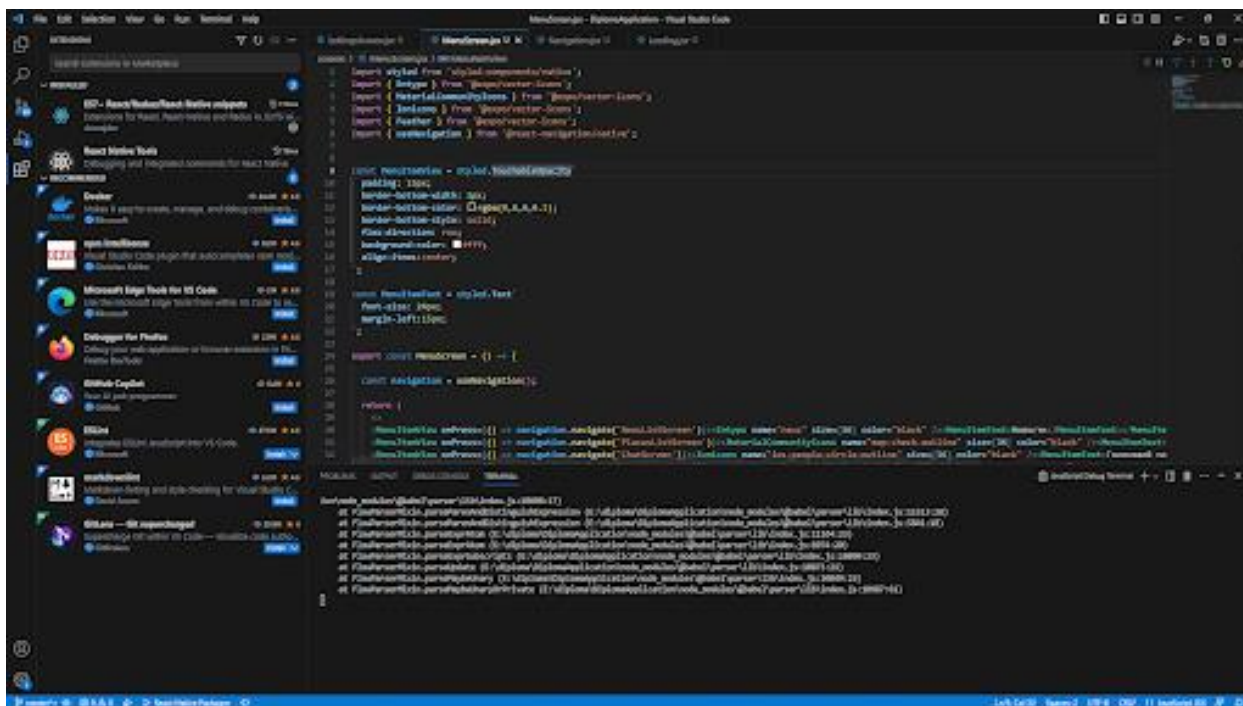


Рисунок 16 – Интерфейс VS Code

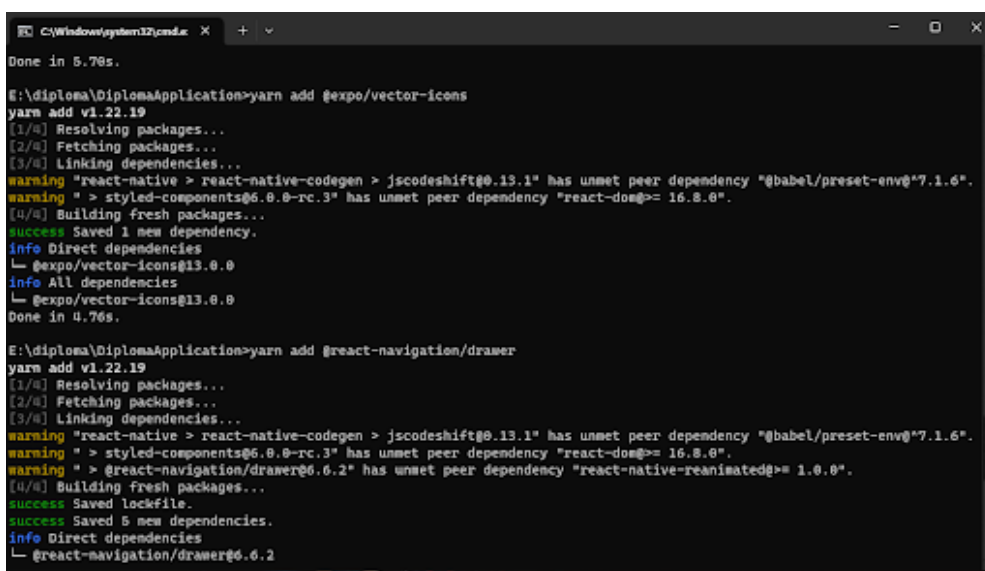
Далее переходим к пакетному менеджеру, а если точнее двум. При разработке использовались такие пакетные менеджеры как `npm` и `yarn`, при этом `npm` работает поверх движка `node.js`. В ходе разработки мобильного приложения, использование инструментов, таких как `npm`, `Yarn` и `Node.js`, является важной практикой, обеспечивающей эффективное управление зависимостями и сборкой проекта. Эти инструменты позволяют разработчикам легко управлять пакетами, внешними библиотеками и другими ресурсами, необходимыми для успешной разработки приложения.

`Node.js` представляет собой платформу, основанную на `JavaScript`, которая позволяет выполнять код на стороне сервера. Он обеспечивает среду выполнения `JavaScript` вне браузера, что является важным при разработке серверной логики для мобильных приложений. `Node.js` также интегрируется с `npm` и `Yarn`, обеспечивая возможность установки и управления пакетами и зависимостями. `npm` (`Node Package Manager`) и `Yarn` являются менеджерами

пакетов для Node.js. Они дают возможность управлять зависимостями и установкой пакетов, необходимых для проекта. Оба инструмента предлагают простой способ установки, обновления и удаления пакетов, а также позволяют управлять версиями зависимостей.

Одним из основных преимуществ использования npm, Yarn и Node.js при разработке мобильного приложения является наличие обширного экосистемы пакетов и библиотек. Среди которых можно легко найти и использовать пакеты, которые решают определенные задачи или предоставляют готовые компоненты для разработки интерфейса, обработки данных и других функций, что значительно ускоряет процесс разработки.

Кроме того, npm, Yarn и Node.js обеспечивают простоту воспроизводимости проекта. Файлы package.json или yarn.lock позволяют фиксировать версии пакетов и их зависимостей, что гарантирует, что каждый разработчик, присоединяющийся к проекту, будет работать с одними и теми же версиями пакетов. Это способствует согласованности и предотвращает возможные конфликты, связанные с версионированием пакетов. На рисунке 17 изображен вывод после установки определенных пакетов с помощью yarn



```
C:\Windows\system32\cmd.exe X + v
Done in 5.78s.

E:\diploma\diplomaApplication>yarn add @expo/vector-icons
yarn add v1.22.19
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning "react-native > react-native-codegen > jscodeshift@0.13.1" has unmet peer dependency "@babel/preset-env@^7.1.6".
warning " > styled-components@6.0.0-rc.3" has unmet peer dependency "react-dom@>= 16.8.0".
[4/4] Building fresh packages...
success Saved 1 new dependency.
info Direct dependencies
├─ @expo/vector-icons@13.0.0
info All dependencies
├─ @expo/vector-icons@13.0.0
Done in 4.76s.

E:\diploma\diplomaApplication>yarn add @react-navigation/drawer
yarn add v1.22.19
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning "react-native > react-native-codegen > jscodeshift@0.13.1" has unmet peer dependency "@babel/preset-env@^7.1.6".
warning " > styled-components@6.0.0-rc.3" has unmet peer dependency "react-dom@>= 16.8.0".
warning " > @react-navigation/drawer@6.6.2" has unmet peer dependency "react-native-reanimated@>= 1.0.0".
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 5 new dependencies.
info Direct dependencies
├─ @react-navigation/drawer@6.6.2
```

Рисунок 17 – Пакетный менеджер Yarn

Но что устанавливается и на чем, собственно, разрабатывалось данное мобильное приложение? В качестве основного языка программирования был

выбран JavaScript, а именно его библиотека под названием react-native. Рассмотрим их подробнее.

JavaScript является одним из основных языков программирования, используемых при разработке мобильных приложений. Он предоставляет множество возможностей для создания интерактивных и динамичных пользовательских интерфейсов, обработки данных и взаимодействия с внешними сервисами. React Native - это популярный фреймворк, основанный на JavaScript, который позволяет создавать мобильные приложения с использованием нативных компонентов и функций. Данный фреймворк упрощает разработку мобильных приложений, позволяя переиспользовать код и создавать приложения, которые могут работать как на iOS, так и на Android.

JavaScript и React Native вместе обеспечивают эффективное создание кросс-платформенных мобильных приложений. Разработчики могут использовать общий код на JavaScript для создания интерфейса, обработки данных и логики приложения, и затем использовать React Native для компиляции этого кода в нативный код, который может выполняться на различных платформах. Также данные инструменты предлагают богатую экосистему инструментов и библиотек для ускорения процесса разработки мобильных приложений. Существует множество готовых компонентов и библиотек, которые помогают создавать стильные пользовательские интерфейсы, обеспечивать навигацию, работать с анимациями и другими функциями.

Для передачи настроек через все приложения было решено использовать библиотеку Redux. Redux - это библиотека для управления состоянием приложения в JavaScript-приложениях. Она обеспечивает предсказуемое и однонаправленное поток данных, что упрощает разработку и поддержку больших и сложных приложений.

Центральным понятием в Redux является хранилище (store), которое содержит все данные приложения. Это единое место, где хранится состояние

приложения, и его изменения могут быть отслежены и обрабатываться в централизованном месте.

Для изменения состояния в Redux используются действия (actions), которые представляют собой объекты, описывающие какое-либо событие или изменение в приложении. Действия могут быть вызваны из компонентов приложения или из асинхронных операций.

Чтобы изменить состояние в хранилище, действия передаются в редюсеры (reducers). Редюсеры - это функции, которые принимают текущее состояние и действие, и возвращают новое состояние приложения. Редюсеры должны быть чистыми функциями без побочных эффектов, что делает изменение состояния в Redux предсказуемым.

Он также поддерживает концепцию селекторов (selectors), которые позволяют выбирать и комбинировать части состояния из хранилища. Селекторы помогают получать только необходимые данные из хранилища для компонентов приложения, что способствует оптимизации производительности.

В итоге, Redux предоставляет удобный и мощный инструментарий для управления состоянием приложения. Он позволяет создавать предсказуемые и расширяемые приложения, упрощает отладку и тестирование, а также облегчает совместную работу в команде разработчиков.

2.4 Создание макета интерфейса

Создание макетов приложений перед началом разработки является важным шагом в процессе разработки программного обеспечения. Макеты представляют собой визуальное представление будущего приложения, которое включает в себя расположение элементов интерфейса, дизайн, взаимодействие и функциональность. Несколько основных причин, по которым создание макетов является неотъемлемой частью процесса разработки приложений:

Понимание требований клиента: Макеты помогают разработчикам и заказчикам лучше понять требования и ожидания клиента по отношению к приложению. Они позволяют визуализировать концепцию и убедиться, что все заинтересованные стороны имеют единое представление о функциональности и внешнем виде приложения.

Визуальное представление: Макеты позволяют визуализировать дизайн и расположение элементов интерфейса. Они помогают определить, как информация будет представлена пользователю, и какие элементы будут взаимодействовать друг с другом. Макеты позволяют создать эффективный и интуитивно понятный пользовательский интерфейс.

Обратная связь и тестирование: Макеты дают возможность собрать обратную связь от пользователей и заинтересованных сторон ещё на ранней стадии разработки. Это позволяет выявить потенциальные проблемы и недочёты в дизайне или функциональности приложения. Путём итераций над макетами можно внести коррективы и улучшения до того, как начнется фаза активной разработки.

Экономия времени и ресурсов: Создание макетов позволяет избежать потери времени и ресурсов на разработку приложения, которое не соответствует требованиям клиента или пользовательским ожиданиям. Предварительная оценка и корректировка макетов позволяет снизить риски и повысить эффективность разработки.

Согласованность команды разработчиков: Макеты служат основой для команды разработчиков, дизайнеров и других заинтересованных сторон, чтобы работать в соответствии с одним видением и целями. Они помогают избежать недоразумений и разногласий, что способствует более гладкому и эффективному процессу разработки.

Макет интерфейса будет создаваться в программе adobe illustrator. Для интерфейса будут использоваться две темы, темная и светлая. Основным требованием к интерфейсу является его простота и лаконичность, а также крупные элементы управления и неперегруженность информацией, это все

необходимо для увеличения уровня его доступности для людей с ограниченными возможностями здоровья по зрению. На рисунке 18 представлены экраны настроек и чата.

В настройках пользователь сможет выбрать город, чтобы данные сортировались по нему. Размер шрифта также можно регулировать, как и присутствует функция доступности под названием texttospeech, которая позволяет переводить текст на экране в аудио-формат. Изменение темы поможет людям с фотофобией или ночной слепотой пользоваться приложением. В целом все эти функции повышают уровень доступности приложения для людей с ОВЗ по зрению.

На экране “чат” пользователь сможет общаться с голосовым ассистентом, при этом сообщения бота высвечиваются с левой стороны, а пользователя, наоборот, с правой. Активация микрофона происходит путем нажатия на кнопку микрофона справа снизу.

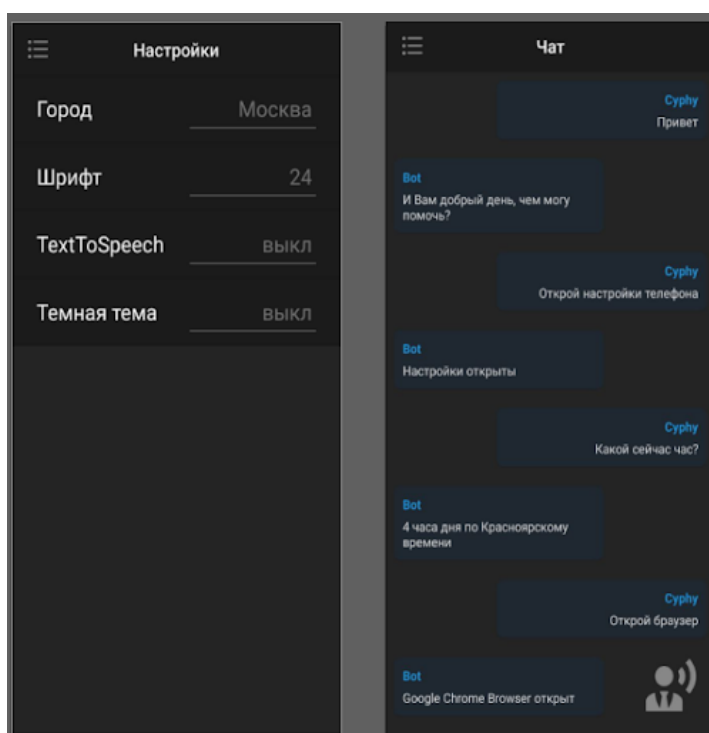


Рисунок 18 – Настройки и чат

На рисунке 19 изображены экраны списка новостей и списка мест, доступных для людей с ОВЗ по зрению. Список новостей состоит из элементов-контейнеров, которые содержат информацию про одну новость. В

эту информацию входит изображение, название новости и краткое его описание. Данные контейнеры являются кликабельными и при нажатии на них открывается полная информация про выбранную новость.

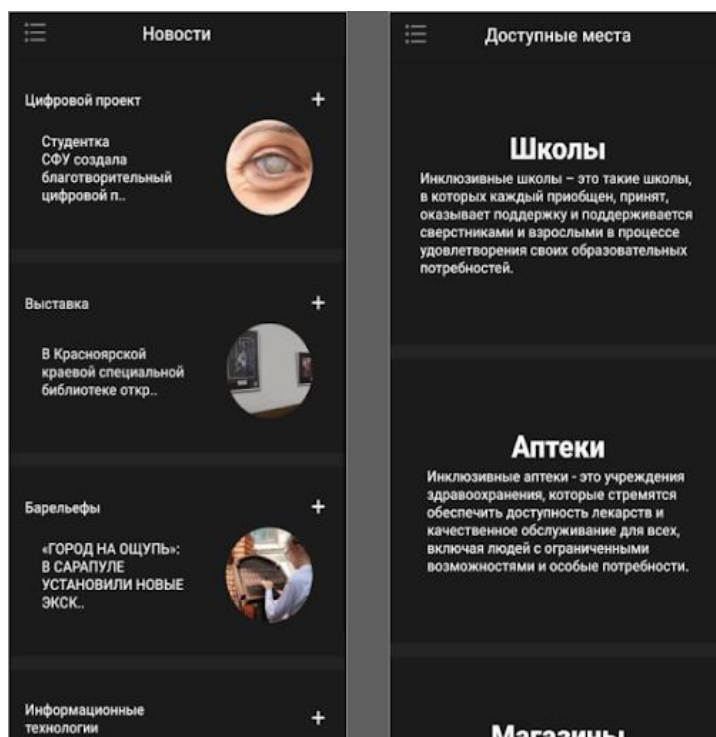


Рисунок 19 – Экран “Новости” и “Доступные места”

На экране списка доступных мест пользователя встречаются плашки с разделением на категории и при нажатии на такую плашку открывается уже список мест, доступных для людей с ОВЗ по зрению . Данные плашки содержат название категории и ее краткое описание.

2.5 Проектирование архитектуры приложения

Исходя из концепции - разрабатываемое приложение должно строиться на клиент-серверной архитектуре, при которой все данные хранятся и обновляются на сервере, а потом, через запросы, передаются на клиентское приложение, которое занимается их отрисовкой.

Также основным моментом при разработке мобильного приложения является используемая технология - React Native. Даже учитывая то, что это библиотека, а не фреймворк, она все равно заставляет проектировать приложение по определенной структуре. Если точнее, то все приложение

состоит из переиспользуемых компонентов, как небольших, наподобие кнопок или полей для ввода, так и целых экранов, отображаемых в мобильном приложении.

Проектирование архитектуры мобильного приложения с использованием React Native, React Native Router, Styled Components и Axios представляет собой важный процесс, который обеспечивает эффективное развитие и поддержку приложения. Далее будут рассмотрены основные принципы и практики, которые используются для создания гибкой и масштабируемой архитектуры мобильного приложения.

Первым шагом в проектировании архитектуры является определение основных компонентов приложения. В случае React Native это могут быть экраны, навигация, компоненты пользовательского интерфейса и служебные модули. Так что были определены какие экраны мне понадобятся, как они будут взаимодействовать друг с другом и какую функциональность они должны предоставлять.

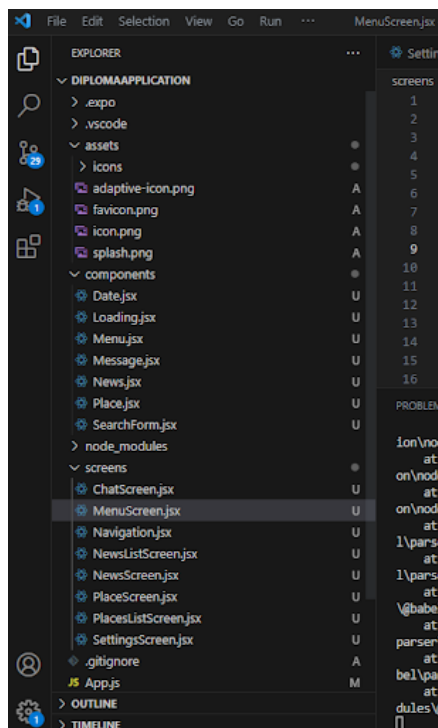


Рисунок 20 – Файловая структура проекта

Свое приложение я разделил на 8 отдельных экранов, 4 из которых являлись основными: настройки, новости, список доступных мест и чат с

голосовым помощником. Остальные же выступали в роли страниц с полной информацией или в качестве элемента меню приложения. В ходе работы, предварительно, получилась следующая файловая структура, представленная на рисунке 20.

Также данная файловая структура включает в себя используемые в ходе работы и в проекте временные файлы, иконки и изображения

React Native Router используется для управления навигацией между экранами. Он позволяет определить стеки и табы для организации экранов и предоставляет API для навигации между ними. Разделение экранов на логические блоки помогает сделать архитектуру более понятной и поддерживаемой.

Styled Components - это библиотека, которая позволяет создавать переиспользуемые и стилизованные компоненты в React Native. С ее помощью определялись стили для компонентов и, что хотелось бы отметить, они использовались повторно в разных частях приложения. Это способствовало улучшению читаемости кода и облегчению его поддержки в дальнейшем.

Для общения с сервером и получения данных использовалась библиотека Axios. Она предоставляет простой и мощный интерфейс для выполнения HTTP-запросов. С помощью нее отправлялись запросы на сервер, обрабатывались ответы и происходило управления состоянием загрузки данных. Это позволило эффективно работать с API и обеспечить плавный поток данных в вашем приложении.

В ходе проектирования приложения упор делался на соответствие принципам разделения ответственности (Separation of Concerns). Следуя которым код разделяется на модули, компоненты и служебные функции, чтобы сделать его более читаемым, понятным и переиспользуемым.

Поскольку данное приложение является клиент-серверным, самым логичным было бы спроектировать его на rest-api архитектуре. REST (Representational State Transfer) — это архитектурный стиль, используемый

для проектирования распределенных систем, включая веб-сервисы. REST API (Application Programming Interface) является интерфейсом, который позволяет различным программным приложениям взаимодействовать друг с другом по принципам REST. REST API основан на нескольких ключевых принципах:

Клиент-серверная архитектура: REST API разделяет клиентскую и серверную части. Клиентская часть отвечает за отправку запросов к серверу, а серверная часть обрабатывает эти запросы и возвращает соответствующие ответы.

Без состояния (Stateless): Каждый запрос к REST API должен содержать всю необходимую информацию для его обработки, и сервер не должен хранить никакое состояние о клиенте между запросами. Это упрощает масштабирование и обеспечивает независимость между запросами.

Кэширование: REST API поддерживает кэширование для улучшения производительности и снижения нагрузки на сервер. Клиент может кэшировать ответы сервера и повторно использовать их для повторных запросов при неизменности ресурса.

Единообразный интерфейс: REST API использует единообразный набор методов, таких как GET, POST, PUT и DELETE, для работы с ресурсами. Каждый метод имеет определенное назначение, например, GET используется для получения данных, POST - для создания новых ресурсов, PUT - для обновления существующих ресурсов, DELETE - для удаления ресурсов.

Инкапсуляция: REST API позволяет клиентам манипулировать ресурсами через предоставленные единообразные интерфейсы, скрывая внутреннюю реализацию сервера и обеспечивая отделение между клиентом и сервером.

REST API обычно использует протокол HTTP для обмена данными между клиентом и сервером. Клиент отправляет запросы на сервер, указывая

метод, URL и дополнительные параметры или данные. Сервер обрабатывает запросы и возвращает соответствующий HTTP-статус и данные в ответ.

REST API широко применяется во множестве сфер, включая разработку веб-приложений, мобильных приложений и сервисов. Он предоставляет гибкую и масштабируемую архитектуру для обмена данными между приложениями, а также обеспечивает простоту в использовании и понимании для разработчиков.

2.6 Разработка приложения

Разработка приложения начинается с установки используемых технологий. В контексте данного проекта основой является программная платформа Node js [9], она устанавливается через обычный .exe файл. Следующим шагом является установка пакетного менеджера, хоть в комплекте с node js поставляется npm, но я привык использовать пакетный менеджер yarn, он устанавливается путем выполнения исполняемой команды в npm.

```
npm install --global yarn
```

После необходимо установить платформу для удобной разработки приложений на react native - expo cli, она устанавливается с помощью пакетного менеджера npm.

```
npm install --global expo-cli
```

Новый проект создается с помощью yarn. На выбор даются несколько шаблонов с разным набором подключенных заранее инструментов, от которых меняется и создаваемая структура. Для разработки этого приложения я использовал чистый шаблон, установив его командой:

```
yarn create expo-app [Projectname]
```

Созданный шаблон включает в себя следующее: Папка assets - служит директорией для хранения используемых изображений и иконок из шаблоны. Папка node_modules - служебная директория в которой хранятся файлы используемых библиотек и фреймворков. Файл настроек системы контроля

версий .gitignore. Файлы настроек app.json, babel.config.json, package.json и yarn.lock. Входная точка для приложения - файл App.js

Содержание файла app.js и файловая структура шаблона представлена на рисунке 21.

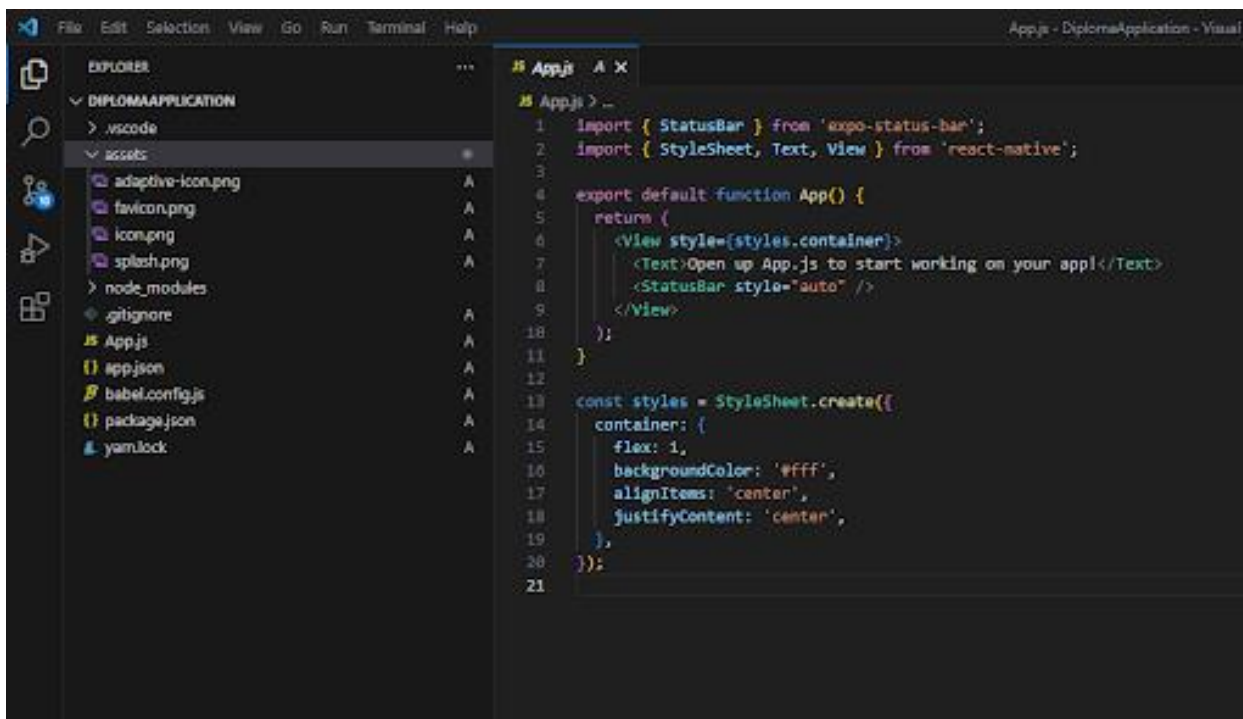


Рисунок 21 – Структура шаблона проекта

Когда проект готов - нужно подключить остальные используемые инструменты, а именно react-navigation/native, axios, styled-components, expo/vector-icons. Подключаются они с помощью пакетного менеджера yarn выполнением следующих команд:

```
yarn add axios
```

```
yarn add styled-components
```

```
yarn add @react-navigation/native @react-navigation/native-stack react-native-screens react-native-safe-area-context react-native-gesture-handler
```

```
yarn add expo/vector-icons
```

После установки всех необходимых инструментов создается первичная структура разрабатываемого приложения. Была создана директория screens, которая будет включать в себя файлы .jsx содержащие представления экранов приложения. В данной директории были созданы файлы SettingsScreen.jsx - экран для настроек, PlacesListScreen.jsx - экран для

списка мест, доступных для людей с ОВЗ, PlaceScreen.jsx - экран развернутого отображения информации про конкретное место, NewsListScreen.jsx - экран отображения списка новостей, NewsScreen.jsx - экран отображения развернутой информации про одну конкретную новость, MenuScreen.jsx - экран меню приложения, ChatScreen.jsx - экран для ведения диалога с голосовым помощником.

Следующим шагом было создание файла для навигации в приложении, которая позволит переключаться между экранами путем нажатия на кнопки. Данное действие выполняется с помощью библиотеки react-router. Код данного файла и структура проекта представлена на рисунке 22.

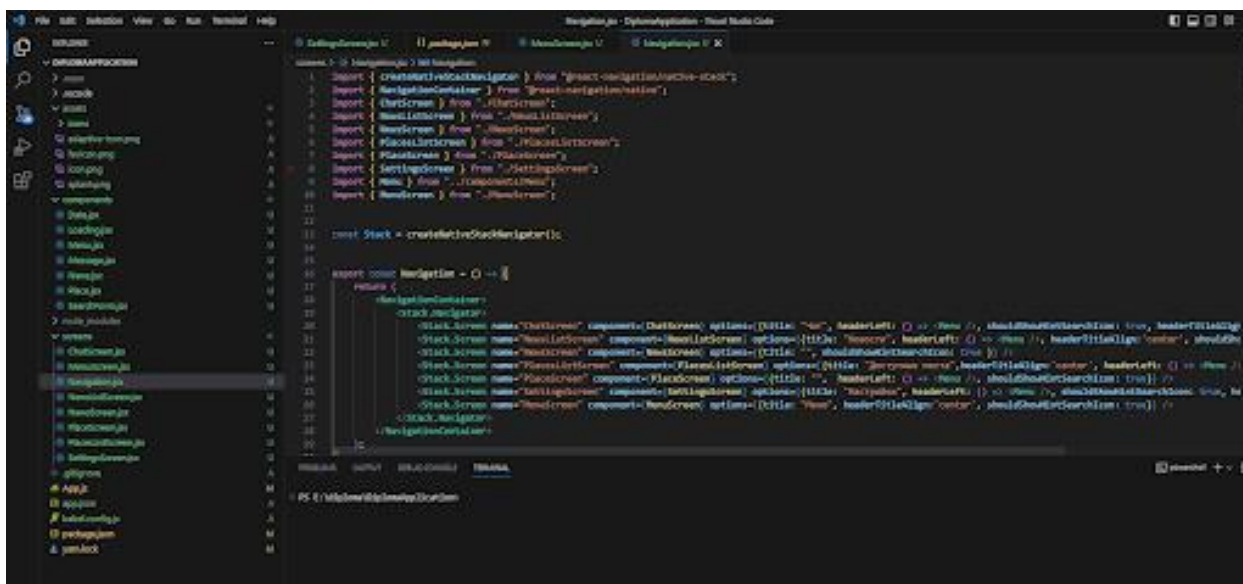


Рисунок 22 – Файл навигации

В данном файле можно увидеть как импортируются используемые компоненты с помощью инструкции import, далее создается обертка для навигации с помощью элементов <NavigationContainer> и <Stack.Navigator>, а также в нее помещаются экраны приложения с помощью элемента <Stack.Screen> и производится их настройка. Следует отметить, что тут настраивается не только сама навигация, но и верхняя панель приложения. В данных настройках можно изменить внешний вид и функциональность отображаемых на ней элементов, собственно, как и сами отображаемые

элементы. Настройка происходит с помощью передачи объекта в свойство `option` компонента. К примеру:

```
<Stack.Screen
  name = "SettingsScreen"
  component = {SettingsScreen}
  options = {{
title: "Настройки",
headerLeft: () => <Menu />,
shouldShowHintSearchIcon: true,
headerTitleAlign:'center'
  }}
/>
```

Тут приведен пример настройки элемента навигации. Сам элемент создается с помощью компонента `Stack.Screen` импортируемого из библиотеки `@react-navigation/native-stack`. В свойство `name` передается строка с именем, используя которое можно будет сказать приложению открыть именно этот экран. Компонент экрана же импортируется из его файла и передается в свойство `component`. Говоря про объект настройки, передаваемый в `options`: `title` - Название экрана, отображаемое в шапке приложения. `headerLeft` - в данное свойство передается коллбэк функция, возвращающая элемент, который будет отображаться в левой части шапки вместо стандартной кнопки назад. В данном случае передается компонент меню, именно через него была создана навигация в приложении. `shouldShowHintSearchIcon` - это свойство отвечает за отображение стандартного поля поиска в шапке навигации приложения. `headerTitleAlign` - свойство, которое отвечает за выравнивание названия экрана, может иметь значения `left` - название прижимается к левому краю и `center` - название становится по центру.

Давайте рассмотрим компонент `Menu`, который заменяет стандартную кнопку возврата на предыдущий экран. Отрывок его кода приведен ниже.

```
export const Menu = () => {
  const navigation = useNavigation();
  return (
    <View>
```



```

        <TouchableOpacity onPress={() => navigation.navigate('MenuScreen')}>
            <AntDesign name="bars" size={32} color="gray" />
        </TouchableOpacity>
    </View>
);
}

```

Одним из вариантов создания компонента в react-native является использование стрелочной функции, возвращающей элемент для рендеринга. Инструкция `export` указывает на то, что данный элемент будет экспортироваться из данного файла, при попытке его импорта. Стрелочной функцией и, соответственно компонентом, является `Main`. В начале тела функции создается объект `navigation` путем вызова импортируемой из библиотеки `@react-navigation/native` функции `useNavigation()`. Данный объект нужен для выполнения навигации и используется дальше. Возвращает же созданный компонент стандартную обертку, содержащую элемент `TouchableOpacity`, который служит в роли нажимаемой кнопки и в него включена иконка бургер-меню. В свойство `onPress`, обозначающего событие срабатывающее при нажатии на данный элемент, передана стрелочная функция, которая при срабатывании открывает экран меню приложения с помощью ссылки на псевдоним, указанный ранее в компоненте навигации.

Дальше стоит перейти к экрану меню приложения. Его код состоит из 3 частей. Первая это блок, в котором импортируются используемые компоненты и модули с помощью инструкции `import`. Во втором происходит определение стилей для элементов, созданных с помощью библиотеки `styled-components`. Пример одного такого элемента приведен ниже.

```

const MenuItemView = styled.TouchableOpacity`
    padding: 15px;
    border-bottom-width: 1px;
    border-bottom-color: rgba(0,0,0,0.1);
    border-bottom-style: solid;
    flex-direction: row;
    background-color: #fff;
    align-items:center;
`;

```

В данном отрывке кода создается элемент `MenuItemView`, который будет служить одновременно как контейнер для используемой иконки с текстом и как кнопка - элемент меню, который при нажатии будет включать определенный экран приложения.

Особенностью и огромным плюсом библиотеки `styled-components` является возможность использования почти всех CSS-свойств для управления дизайном элемента с привычной их записью. На конкретном примере задается внутренний отступ в 15 пикселей свойством `padding`, далее с помощью вариаций `border` задается сплошная рамка внизу элемента, шириной 1 пиксель и имеющая серый цвет. Использованием `background-color` устанавливается белый цвет заднего фона и `align-items` выравнивает вложенные элементы по центру блока. Возвращает же компонент `MenuScreen` набор из следующих элементов:

```
<MenuItemView onPress={() => navigation.navigate('NewsListScreen')}>
  <Entypo name="news" size={36} color="black"/>
  <MenuItemText>Новости</MenuItemText>
</MenuItemView>
```

Оговоренный ранее `MenuItemView` служит контейнером и кнопкой, которая содержит импортируемую векторную иконку из библиотеки `@expo/vector-icons` и элемент с названием этой самой кнопки.

Для сохранения настроек в приложении и передачи их всем компонентам была использована библиотека `redux`, которая позволяет хранить глобальное состояние. Чтобы реализовать голосового помощника я подключил две библиотеки `react-native-speech` и `react-native-voice`

```
yarn add react-native-speech react-native-voice
```

Далее был создан компонент со следующим кодом:

```
const SpeechRecognition = () => {
  const [isListening, setIsListening] = useState(false);

  const startSpeechRecognition = async () => {
    setIsListening(true);
    try {
      await Voice.start('en-US');
    }
  }
}
```

```

    } catch (error) {
      console.log('Error occurred during speech recognition:', error);
    }
  };

const stopSpeechRecognition = async () => {
  setIsListening(false);
  try {
    const speechResults = await Voice.stop();
    // Обработка результатов распознавания голоса
    console.log('Speech recognition results:', speechResults);
  } catch (error) {
    console.log('Error occurred during speech recognition:', error);
  }
};

const speakText = async (text) => {
  try {
    await Tts.speak({
      text,
      voice: 'en-US', // Установите нужный язык и голос
    });
  } catch (error) {
    console.log('Error occurred during text-to-speech:', error);
  }
};

return (
  <View>
    <Button
      title={isListening ? 'Stop' : 'Start'}
      onPress={isListening ? stopSpeechRecognition : startSpeechRecognition}
    />
    <Button
      title="Speak"
      onPress={() => speakText('Hello, how are you?')}
    />
  </View>
);
};

```

Что при его использовании позволило добиться поставленных целей, так были разработаны страницы Настройки и Меню. Через экран меню можно перейти на экраны новостей и к списку доступных мест, а в настройках изменить город проживания, шрифт, включить озвучивание текста и сменить тему приложения. Данные экраны представлены на рисунке 23.

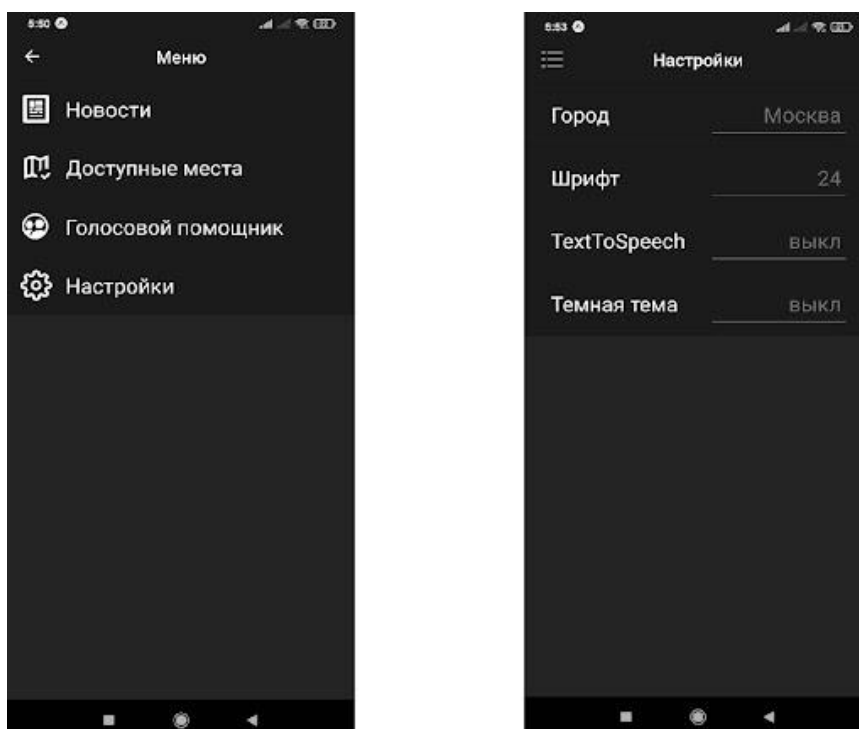


Рисунок 23 – Экран настройки и меню

Чтобы рассказать о разработке следующих страниц - нужно описать работу библиотеки для запросов axios, поскольку именно с помощью нее будут получаться данные с сервера. Ниже приведен код функции, которая делает запрос на сервер, получает список новостей, распаковывает из формата json и обрабатывает ошибки.

```
const fetchPosts = () => {
  setIsLoading(true);
  axios
    .get('https://6488be910e2469c038fe4b29.mockapi.io/news')
    .then(({ data }) => {
      setItems(data);
    });
}
```

```

    })
    .catch(err => {
      console.log(err);
      Alert.alert('Ошибка', 'Не удалось получить новости')
    })
    .finally(() => {
      setIsLoading(false);
    });
  }
}

```

Главным тут, на что стоит обратить внимание, является вызов функции `axios.get('url')`. Данный запрос отправляет запрос на сервер по указанному `url` и получает с него данные. Остальные разработанные экраны представлены на рисунке 24.

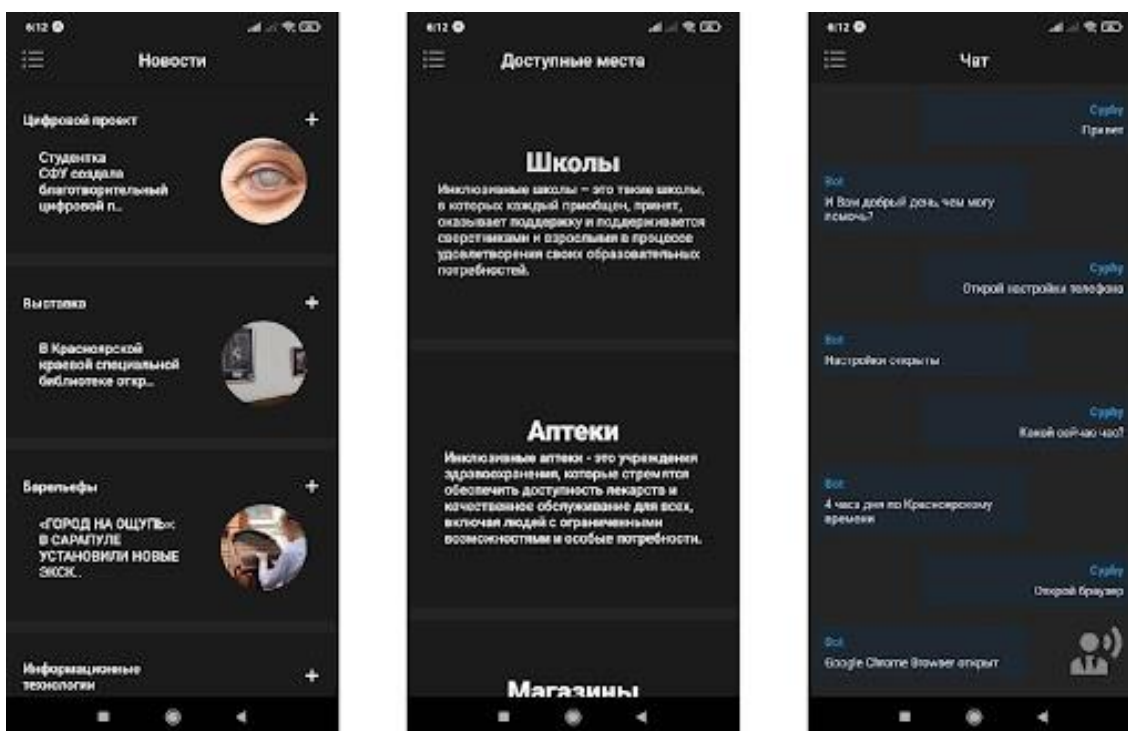


Рисунок 24 – Экраны приложения новости, доступные места и чат

На экране “Новости” пользователь сможет узнать новости, касающиеся людей с ОВЗ по зрению, также тапнув по одной из них откроется еще один экран с подробным просмотром. Экран “Доступные места” предоставит возможность пользователю узнать о местах с повышенным уровнем доступности для него, что сможет положительно повлиять на его социальную жизнь. На экране “Чат” пользователь сможет пообщаться с голосовым

ассистентом, который может помочь в управлении устройством или данным приложением в частности. Весь код данного приложения хранится на сайте по следующей ссылке: <https://github.com/noir-vladislav/Diploma>

ЗАКЛЮЧЕНИЕ

Мобильные приложения стали неотъемлемой частью нашей жизни, они применяются в разных сферах - от развлечений до бизнеса и научных исследований. Но, к сожалению, не все люди могут в полной мере потреблять подобные блага современного общества, некоторые испытывают проблемы связанные с ограниченными возможностями здоровья, в частности, со зрением. Да, несомненно, современное общество начинает уделять все больше и больше внимания инклюзивности - созданию комфортных социальных условий жизни в обществе для людей с ОВЗ, но этого пока недостаточно. Именно поэтому, чтобы внести свой вклад в развитие инклюзивного общества, было решено разработать мобильное приложение для людей с ограниченными возможностями по зрению с функцией Accessibility.

Чтобы не просто разработать мобильное приложение, а сделать его доступным для максимального количества людей с проблемой ОВЗ по зрению, то есть создать те самые функции accessibility - доступности, были изучены теоретические аспекты видов визуальных нарушений, описано и проиллюстрировано их влияние на восприятие визуальной информации на экране мобильного телефона. Данная информация помогла выделить функциональные возможности остаточного зрения для каждого вида визуального нарушения и, опираясь на них, вывести функции доступности, которые помогут людям использовать данное мобильное приложение.

Для определения концепции разрабатываемого приложения был проведен анализ как встроенных функций доступности в мобильных устройствах на базе операционной системы android, так и сторонних приложений в данной нише. Исходя из этого анализа был сделан вывод, что большинство приложений помогают решать определенные, узконаправленные, задачи и мало таких, которые могли бы доносить полезную для своей целевой аудитории - людей с ограниченными возможностями здоровья, полезную для них информацию.

В настоящее время общество проявляет все больший интерес к инклюзивности и включению людей с ограниченными возможностями здоровья в общественную жизнь. Множество мероприятий и инициатив были запущены для привлечения внимания общества к потребностям этой группы людей. Однако, несмотря на это, данные инициативы редко получают должное освещение в крупных СМИ. Они не становятся достаточно значимыми новостными событиями, чтобы привлечь внимание широкой аудитории. В результате, информация о таких инициативах и доступных объектах остается неизвестной для большинства людей.

В связи с представленной выше информацией была выбрана концепция приложения, которое направлено на информирование людей с ограниченными возможностями здоровья по зрению посредством новостей и предоставлении информации о местах, которые являются доступными для них. Доступность же самого приложения повышают функции accessibility и встроенный голосовой ассистент.

Таким образом, разработанное мобильное приложение представляет собой значимый вклад как в развитие инклюзивного общества, так и в область доступности и технологий для людей с ограниченными возможностями зрения. Оно открывает новые перспективы для пользователей, помогая им стать активными участниками современного цифрового мира. В дальнейшем, разработка подобных приложений и их внедрение может способствовать еще большему повышению качества жизни людей с ОВЗ и обеспечить им равные возможности в области информационных технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. DIGITAL 2022: GLOBAL OVERVIEW REPORT. — Текст: электронный // datareportal.com: [сайт]. — URL: <https://datareportal.com/reports/digital-2022-global-overview-report> (дата обращения: 16.06.2023).

2. ВОЗ: около миллиарда случаев слепоты и нарушений зрения можно было бы предотвратить. — Текст: электронный // news.un : [сайт]. — URL: <https://news.un.org/ru/story/2019/10/1364582> (дата обращения: 16.06.2023).

3. 95% мобильных приложений не адаптированы для людей с ограниченными возможностями здоровья. — Текст: электронный // roskachestvo.gov: [сайт]. — URL: <https://roskachestvo.gov.ru/news/95-mobilnykh-prilozheniy-ne-adaptirovany-dlya-lyudey-s-ogranichennymi-vozmozhnostyami-zdorovya/> (дата обращения: 16.06.2023).

4. Eye Conditions and Diseases. — Текст: электронный // www.nei.nih.gov: [сайт]. — URL: <https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases> (дата обращения: 16.06.2023).

5. Вагин, Д. И. Библиотека React.JS / Д. И. Вагин // Инновационные технологии и образование : Материалы международной научно-практической конференции. В 2 частях, Минск, 28 апреля 2022 года. Том Часть 2. – Минск: Белорусский национальный технический университет, 2022. – С. 41-44.

6. Волнейкина, Е. С. Анализ web-разработки на Reactjs адаптивным дизайном / Е. С. Волнейкина, Д. К. Гек // Актуальные проблемы авиации и космонавтики : Сборник материалов VII Международной научно-практической конференции, посвященной Дню космонавтики: в 3 томах, Красноярск, 12–16 апреля 2021 года / Под общей редакцией Ю. Ю. Логинова. Том 2. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский

государственный университет науки и технологий имени академика М.Ф. Решетнева", 2021. – С. 478-480.

7. Бетеев, К. Ю. Концепция Virtual Dom в библиотеке React.js / К. Ю. Бетеев, Г. В. Муратова // Инженерный вестник Дона. – 2022. – № 3(87). – С. 170-180.

8. Курбесов, А. В. Разработка библиотеки для построения сложных форм для React.js / А. В. Курбесов, А. А. Тараненко // Проблемы проектирования, применения и безопасности информационных систем в условиях цифровой экономики : Материалы XX Международной научно-практической конференции, посвященной 85-летию экономиста-математика, доктора экономических наук, профессора Виктора Алексеевича Кардаша, Ростов-на-Дону, 26–27 октября 2020 года. – Ростов-на-Дону: Ростовский государственный экономический университет "РИНХ", 2020. – С. 77-82.

9. Гарбарчук, А. С. Знакомимся с react.js и его жизненным циклом / А. С. Гарбарчук, Е. М. Гриценко // Лесной и химический комплексы - проблемы и решения : сборник статей по материалам всероссийской научно-практической конференции : в 2 т., Красноярск, 09 декабря 2016 года. Том II. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2016. – С. 249-251.

10. Винокуров, А. Ю. React.js: компонентный подход к разработке электронных курсов / А. Ю. Винокуров, А. А. Леванов // Электронное обучение в непрерывном образовании. – 2017. – № 1. – С. 60-64.

11. Пивоваров, С. А. React JS. Фреймворк для пользовательских интерфейсов / С. А. Пивоваров, Л. Л. Романов, А. А. Воробьев // ТЕХНИКА и ТЕХНОЛОГИЯ: НОВЫЕ ПЕРСПЕКТИВЫ РАЗВИТИЯ : сборник статей Международной научно-практической конференции,

Самара, 25 сентября 2017 года. – Самара: Общество с ограниченной ответственностью "Аэтерна", 2017. – С. 60-61.

12. Отладка JavaScript-скриптов с помощью браузера JavaScript // Web-программирование и базы данных : Учебный практикум. – Москва : Московский гуманитарный университет, 2014. – С. 13.

13. Жарков, А. П. Особенности работы Javascript / А. П. Жарков, Е. А. Макаров, В. А. Терехин // Молодой исследователь: вызовы и перспективы : сборник статей по материалам СХХVI международной научно-практической конференции, Москва, 15 июля 2019 года. Том 26 (126). – Москва: Общество с ограниченной ответственностью "Интернаука", 2019. – С. 140-142.

14. Горбачев, А. А. Применение web-sockets технологии с использованием платформы Node.js и языка Javascript / А. А. Горбачев // Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство : сборник научных статей по итогам одиннадцатой международной научной конференции, Казань, 30 декабря 2019 года. Том Часть 2. – Казань: ООО "КОНВЕРТ", 2019. – С. 131-133.

15. Дерябин, Е. А. Методы оптимизации разработки веб-приложений с использованием javascript-фреймворков / Е. А. Дерябин // Современное образование: традиции и инновации. – 2017. – № 1. – С. 191-193.

16. Кургасов, В. В. Javascript фреймворки / В. В. Кургасов, А. Г. Лапшова // Центральный научный вестник. – 2018. – Т. 3, № 15-16(56-57). – С. 40-41.

17. Авгуль, М. Е. Принципы UI/UX-дизайна мобильных приложений UI/UX-design principles for mobile apps / М. Е. Авгуль // Регионы. Города. Ракурсы и параллели : Сборник научных статей VIII Всероссийской научно-практической конференции, посвященной 110-летию со дня рождения В.И. Кочедамова, с конкурсом докладов молодых ученых, магистрантов и студентов, Омск, 29–30 ноября 2022 года / Редколлегия:

Л.В. Чуйко (науч. ред.) [и др.]. – Омск: Омский государственный технический университет, 2023. – С. 74-78.

18. Пеньшина, Д. С. Использование принципов UX/UI дизайна при разработке пользовательского интерфейса / Д. С. Пеньшина, А. В. Свищев // Моя профессиональная карьера. – 2023. – Т. 3, № 48. – С. 191-198.

19. Белоблоцкая, Я. С. Системный подход в UX/UI дизайне / Я. С. Белоблоцкая, А. А. Искрова, А. А. Новикова // Электронные системы и технологии : Сборник материалов 58-й научной конференции аспирантов, магистрантов и студентов БГУИР, Минск, 18–22 апреля 2022 года. – Минск: Белорусский государственный университет информатики и радиоэлектроники, 2022. – С. 698-700.

20. Кондрашова, Е. И. Контент-маркетинг, UX и UI-дизайн как три кита современной рекламы / Е. И. Кондрашова, С. Ю. Живенкова // Социально-гуманитарные проблемы образования и профессиональной самореализации (Социальный инженер-2022) : сборник материалов Всероссийская научная конференция молодых исследователей с международным участием, Москва, 12–16 декабря 2022 года. Том Часть 2. – Москва: Федеральное государственное бюджетное образовательное учреждение высшего образования "Российский государственный университет имени А.Н. Косыгина (Технологии. Дизайн. Искусство)", 2022. – С. 115-120.

21. Modular Teaching Didactic Foundations To Responsive Web Design And User Experience Design / V. D. Munister, A. L. Zolkin, E. A. Vereschagina, O. V. Kosnikova // European Proceedings of Social and Behavioural Sciences EpSBS, Krasnoyarsk, 19–21 мая 2021 года. Vol. 116. – Krasnoyarsk, Russia: ISO LONDON LIMITED - European Publisher, 2021. – P. 1070-1077. – DOI 10.15405/epsbs.2021.09.02.118.

22. Аникин, Д. А. Анализ методов авторизации и аутентификации REST API / Д. А. Аникин // Международный журнал информационных технологий и энергоэффективности. – 2023. – Т. 8, № 5-2(31). – С. 120-124.

23. Никонова, Е. З. Анализ архитектурного стиля rest API / Е. З. Никонова, Р. И. Королев // Современные вопросы устойчивого развития общества в эпоху трансформационных процессов : Сборник материалов VIII Международной научно-практической конференции, Москва, 17 апреля 2023 года. – Москва: Общество с ограниченной ответственностью "Издательство АЛЕФ", 2023. – С. 176-179.

24. Ананченко, И. В. Оптимизация HTTP-запросов с помощью перехода с REST API на graphql / И. В. Ананченко, Е. А. Чуриков // Актуальные вопросы современной науки : Сборник статей III Международной научно-практической конференции, Пенза, 25 сентября 2022 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2022. – С. 11-14.

25. Лукьянов, А. С. Особенности реализации программного интерфейса REST API для маркетплейса услуг / А. С. Лукьянов, В. А. Лукьянова, И. В. Михеев // Актуальные проблемы и пути развития энергетики, техники и технологий : Сборник трудов VIII Международной научно-практической конференции, Балаково, 20 апреля 2022 года. Том 1. – Балаково: Национальный исследовательский ядерный университет "МИФИ", 2022. – С. 288-295.

26. Воробьев, А. А. Фреймворк Vue.js как основной инструмент фронтенд разработчика / А. А. Воробьев, С. С. Лихолетов, Л. Л. Романов // ТЕХНОЛОГИЧЕСКОЕ РАЗВИТИЕ СОВРЕМЕННОЙ науки: ТЕНДЕНЦИИ, ПРОБЛЕМЫ и ПЕРСПЕКТИВЫ : сборник статей Международной научно-практической конференции, Волгоград, 11 февраля 2018 года. – Волгоград: Общество с ограниченной ответственностью "ОМЕГА САЙНС", 2018. – С. 10-11.

27. Пермяков, А. А. Создание web-проекта на Vue.js / А. А. Пермяков, Л. Ю. Забелин // Современные проблемы телекоммуникаций : Материалы Российской научно-технической конференции, Новосибирск,

26–27 апреля 2018 года. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2018. – С. 488-492.

28. Бондаренко, Ю. В. Почему стоит использовать Vue.js / Ю. В. Бондаренко // НАУКА В СОВРЕМЕННОМ ОБЩЕСТВЕ: ЗАКОНОМЕРНОСТИ И ТЕНДЕНЦИИ РАЗВИТИЯ : сборник статей Международной научно-практической конференции, Магнитогорск, 10 ноября 2018 года. – Магнитогорск: Общество с ограниченной ответственностью "Аэтерна", 2018. – С. 22-24.

ПРИЛОЖЕНИЕ А

Таблица А.1 – Функциональные возможности остаточного зрения и функции доступности для них

Виды визуальных нарушений	Характер нарушения основных компонентов остаточного зрения	Функциональные возможности остаточного зрения	Предлагаемые функции доступности
Низкое зрение	снижение четкости зрения, возможны проблемы с фокусировкой и периферическим зрением	различают контуры, цвета, но с ограничениями	увеличение шрифта и изображений, использование высококонтрастных цветов, использование звуковых уведомлений и голосовых подсказок
Слепота	отсутствие зрения или очень ограниченное зрение	могут различать свет и тень, а также некоторые формы и цвета	использование голосовых подсказок, аудиоописаний и звуковых уведомлений, использование тактильных вибраций
Дальтонизм	неспособность видеть определенные цвета или различать между ними	ограниченное восприятие красного и зеленого цветов	использование улучшенных цветовых схем, использование меток и обозначений текстом
Ночная слепота	неспособность видеть в условиях низкой освещенности	ограниченное восприятие деталей при недостаточном освещении	использование подсветки экрана и функции автоматического увеличения яркости
Фотофобия	чувствительность к свету, вызывающая дискомфорт или боль	имеют нормальное зрение, но испытывают дискомфорт при ярком освещении	использование затемнения экрана, автоматическое снижение яркости

Окончание таблицы А.1

Виды визуальных нарушений	Характер нарушения основных компонентов остаточного зрения	Функциональные возможности остаточного зрения	Предлагаемые функции доступности
Кортикальное нарушение зрения	искаженное восприятие или исчезновение областей зрительного поля	различают объекты, но могут иметь затруднения в интерпретации визуальной информации	использование упрощенных изображений и интерфейсов, использование голосовых
Пигментный ретинит	потеря цветоощущения, снижение остроты зрения, проблемы с ночным зрением, туннельное зрение	имеют ограниченное туннельное зрение с потерей резкости	использование больших шрифтов и ярких цветов, адаптивный интерфейс
Катаракта	снижение яркости и четкости зрения, помутнение и образование бликов	в зависимости от степени развития могут иметь нормальное зрение с потерей резкости, а на поздних стадиях различать цвета и контуры	использование голосовых подсказок, аудиоописаний и звуковых уведомлений, использование тактильных вибраций

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Гуманитарный институт
Кафедра информационных технологий
в креативных и культурных индустриях

УТВЕРЖДАЮ

И. о. заведующего кафедрой

 М. А. Лаптева


« 29 » июня 2023 г.

БАКАЛАВРСКАЯ РАБОТА

Создание приложения для людей с ограниченными возможностями
по зрению с функцией Accessibility

Направление подготовки: 09.03.03 Прикладная информатика

Наименование программы: 09.03.03.30 Прикладная информатика

Руководитель  доц., канд. филос. наук П. А. Солкина

Выпускник  В. А. Быков

Нормоконтролер  И. Р. Нигматуллин