

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра информатики и программирования

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ СО
СПЕЦИАЛЬНЫМИ ВОЗМОЖНОСТЯМИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Кулакова Максима Сергеевича

Научный руководитель:

к.э.н., доцент

Л.В. Кабанова

Зав. Кафедрой ИиП:

к.ф.-м.н., доцент

М.В. Огнева

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. Интернет разрушает препятствия для общения и взаимодействия даже между людьми с ограничениями здоровья, за счёт которых его развитие меняется коренным образом. В то же время, если веб-сайты, онлайн-приложения и прочие технологии спроектированы неудачным образом, Интернет может быть труднодоступен людям, создавая преграды на пути к использованию.

Спроектированные должным образом веб-сайты и онлайн-приложения могут использоваться людьми, которые имеют ограниченные возможности своего здоровья. В настоящее время наблюдаются сайты и приложения с ограниченной для данной группы людей доступностью, в связи с чем их использование является затруднительным или даже невозможным.

Доступность имеет принципиальное значения для организаций, стремящихся создавать качественные сервисы без утраты возможности комфортно пользоваться их продуктами и услугами. Поэтому обеспечение доступности Интернет-пространства выгодно для отдельных граждан, коммерческих предприятий и общества в целом. Международные стандарты определяют необходимые условия для реализации веб-доступности. Таким образом, задача разработки мобильного приложения, предоставляющего различные возможности взаимодействия с устройством при помощи азбуки Морзе, является актуальной задачей.

Цель бакалаврской работы – разработка мобильного приложения, предоставляющего различные возможности взаимодействия с устройством при помощи азбуки Морзе.

Поставленная цель определила **следующие задачи:**

1. Изучение основных понятий веб-доступности;
2. Исследование существующих методов для разработки доступных веб-технологий и средств их реализации;
3. Обзор существующих приложений;
4. Разработка мобильного приложения, позволяющего взаимодействовать

- с пользователем при помощи вибраций;
- 5. Реализация авторизации пользователей;
- 6. Реализация back-end части приложения;
- 7. Реализация ввода информации через азбуку Морзе.

Методологические основы разработки мобильного Android-приложения, предоставляющего различные возможности взаимодействия с устройством при помощи азбуки Морзе, на React Native с использованием Google FireBase представлены в работе А. Л. Мархакшинова, А. А. Тонхоновой, Е. Р. Урмакшиновой и так далее [6] [7].

Теоретическая и практическая значимость бакалаврской работы. Представленная бакалаврская работа имеет огромную значимость для современной IT сферы. Теоретическая значимость заключается в объяснении всех нюансов и аспектов для разработки специального приложения, которое способно облегчить использование смартфона многим лицам. Практической значимостью является идея разработки нового способа взаимодействия с устройством, который не был реализован до настоящего времени.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объём работы – 69 страниц, из них 50 страниц – основное содержание, включая 7 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 30 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические аспекты создания доступных приложений» посвящен основам веб-доступности, изучению факторов для разработки доступного приложения, средств для их разработки, реализации авторизации, а также изучение существующих способов ввода информации.

Веб-доступность в первую очередь представляет собой проектирование и разработку веб-сайтов, онлайн-приложений и прочих технологических

решений с учетом возможности их использования людьми с различными нарушениями здоровья для того, чтобы пользователи могли самостоятельно:

1. Пользоваться информацией, доступной в сети Интернет;
2. Коммуницировать посредством Интернета;
3. Принимать участие в работе Всемирной сети.

Интернет помогает предоставить большому количеству людей с нарушениями здоровья неограниченный доступ к информации и возможностям взаимодействия пользователей с веб-технологиями: устраняют препятствия на пути к комфортной печати, прослушиванию звукового контента и просмотру материалов. Однако для преодоления таких информационных препятствий нет единых методов, ведь каждая группа пользователей использует привычные для него инструменты [1].

При создании доступного веб-контента для лиц, имеющих ограничения по здоровью, рекомендуется придерживаться WCAG - руководством по доступности веб-контента. Также он является стандартом ISO 40500 и принятым европейским стандартом EN 301 549.

WCAG основывается на четырёх основных принципах:

1. Восприятие. Пользователи могут видеть или слышать контент, с которым они взаимодействуют;
2. Управляемость. Пользователи должны беспрепятственно управлять компьютером или мобильным устройством при помощи клавиатуры или голоса;
3. Понятность. Веб-контент должен быть реализован на простом и понятном для пользователя языке;
4. Надежность. Пользователи могут использовать различные ассистивные технологии [2].

Для создания мобильных приложений существует множество языков программирования, библиотек и сервисов, позволяющих разработчику адаптировать свой продукт под мобильные устройства и находящейся в них операционных системах. Каждые из них имеют как преимущественные

достоинства, так и недостатки. Самыми популярными языками программирования и сервисов разработки являются Swift, Java, Flutter, Xamarin, React Native и Avocode [3].

Для полного комфорта взаимодействия пользователя с разработанным приложением необходимо учесть способы получения информации и его ввода. Существует множество альтернативных источников ввода и вывода информации, которое может облегчить пользователю работу с компьютером, смартфоном и другими электронными устройствами.

Для надёжного хранения данных пользователя и возможности его восстановления необходимо продумать, каким образом можно сохранить эти самые данные. Самым очевидным вариантом является облачная авторизация. Средства аутентификации и идентификации значительно снижают риск утечек информации, которые могут быть связаны с человеческим фактором (ошибки персонала) или с преднамеренными действиями лиц, а также обеспечивают надёжную аутентификацию пользователей при доступе к сетевым ресурсам.

Аутентификация является обязательным условием любой защищённой системы, так как механизмы защиты информации рассчитаны на работу с поименованными субъектами и объектами информационных систем. Они представляют пользователю возможность авторизироваться на сервере и использовать все возможности сайта или приложения [7].

Стоит вернуться к теме веб-доступности в интернете. Большинство людей со слепотой пользуются интернет-ресурсами при помощи предустановленных или загруженных позже специальных озвучивающих программ (англ. "Screen Reader", программы экранного доступа). Принцип действия такой программы заключается в следующем: отображаемый на экране устройства текст передается в форму, которую может обработать пользователь с ослабленным зрением, чаще всего слуховую.

В зависимости от операционной системы на устройствах предустановлены программы:

- Android: TalkBack;
- MacOS / iOS: VoiceOver;
- Linux: Orca, BRLTTY, Emacspeak;
- Windows: NVDA, Serotek, «Специальные возможности»;
- Браузеры: WebAnywhere, SpokenWeb, ChromeVox, ChromeVis [4].

На основе изучения перечисленных выше приложений можно сказать, что основной акцент идёт на озвучивание объектов на экране и средств для увеличения контрастности отображаемой информации. Однако не все из них используют такие средства, как вибрация, что служит основополагающим фактором для написания собственного приложения.

В качестве основного средства взаимодействия с объектами и текстом на смартфоне выступают вибро-отклики, которые являются результатом кодирования и декодирования названий и содержания интерфейса в азбуку Морзе. Он является оптимальным вариантом для передачи необходимой информации пользователю.

Авторизация в приложении и наличие самодостаточного чата для возможности общения с другими людьми также является полезной и универсальной составляющей разработанного приложения, так как благодаря данным функциям пользователю не нужно переключаться на другие приложения.

Второй раздел «Практическое применение React Native для создания приложений» посвящен реализации мобильного приложения, включающего в себя новый способ взаимодействия пользователей с устройством через азбуку Морзе – знаковое кодирование символов в последовательность сигналов: тире и точек, которые будут преобразованы в вибрацию.

Для создания прототипа используем приложение Figma, так как оно является бесплатным и полностью подходит для реализации данного этапа разработки. Были изображены экраны домашней страницы, сообщений,

контактов, звонка, календаря, поиска, карты и настроек, для которых была выстроена навигация между ними.

Для работы приложения необходимы вспомогательные модули для построения собственных блоков, такие как: геолокация, информация об устройстве, календарь и браузер. Их подключение происходит с помощью пакетного менеджера npm.

Хранения всех параметров реализовано через асинхронное хранилище – незашифрованную постоянную систему хранения ключей и значений, являющуюся глобальной для приложения. Асинхронность необходима для обхода ограничения отсутствующей многопоточности в JavaScript. В отличие от синхронных операций асинхронные выполняются параллельно, не ожидая выполнения предыдущей команды.

Реализация перевода азбуки Морзе в вибрацию начинается с транслитерации латинских символов в кириллические. По заданному словарю происходит последовательное сопоставление символов изначальной строки с ключом, после чего значение добавляется в новую промежуточную строку. Полученный кириллический текст переведём в символы Морзе. Функция возвращает преобразованную исходную строку, где берётся каждый символ, приводится в нижний регистр и заменяется на код из словаря-алфавита. После этого реализуем функцию, принимающую код Морзе и скорость, и возвращающую вибрацию в необходимом для работы приложения виде – массиве времени пауз и вибраций. За минимальную продолжительность принимается точка, обозначим её как одну секунду. В зависимости от переданной скорости, данное значение делится на параметр и добавляется в массив. При дальнейшем масштабировании приложения потребуется реализация декодирования символов азбуки Морзе в текст, что реализуется аналогично кодированию.

Приложение в React Native строится на компонентах – блоках, из которых можно организовать общую структуру для отображения и работы функций.

Компонент для возврата назад принимает текст, короткое нажатие, удержание кнопки и иконку, возвращая визуальный блок. Тег `TouchableOpacity` отвечает за взаимодействие с элементом, его свойства `onPress` за нажатие, `onLongPress` за долгое нажатие с вызовом вибрации для реализации взаимодействия с помощью кода Морзе, в которое передаётся текст кнопки. Теги `View` со свойствами `style` необходимы для выстраивания стилизации структуры блока, аналогично тегам `div` в веб-разработке. Стили кнопки описываются в константе `styles` и отвечают за позиционирование и оформление. Аналогично кнопке возврата назад реализуются компоненты для получения статуса батареи, навигации, текущего времени, элементов чатов, сообщений и контактов, блок настроек и поиска в браузере.

После написания компонентов становится возможно выстраивание экранов приложения на основе блочной структуры. Данная структура позволяет связать название экрана с его расположением, что позволит выстроить переходы между экранами.

Из основных компонентов реализуется главная страница, являющаяся логической связующей в навигации (данный экран отображается первым при старте приложения и с него возможен переход по всем остальным структурным элементам), блоки отображения заряда батареи и погоды, экран контактов (с возможностью написания сообщения и набора номера для звонка), экран настроек (стандартные константные значения, которые могут быть изменены пользователем и сохранены благодаря асинхронному хранилищу, после чего будут применены в приложении), экраны списка чатов, поиска, календаря и локации.

Для разработки клавиатуры необходимо добавить компонент в файл `Keyboard.js`, состоящий из отдельной кликабельной области-кнопки и поля введённой информации. Функционал самой клавиатуры описан в файле `readMorzeKey.js`. Добавляются слушатели области клавиатуры и сохранение времени в миллисекундах, когда произошло действие. `onKeyDownListener` работает таким образом, что при зажатой клавише происходит постоянный

вызов обработчика, что не позволит нам сохранить время, поэтому добавляем временную переменную, суммирующую время всех вызовов функции, что требует вспомогательные функции, возвращающие среднее и минимальное значения массива. Они необходимы для динамического вычисления среднего значения зажатия кнопки и подстройки под текущего пользователя.

В дальнейшем описываем компонент отображения кнопки логина: `GoogleButton.js` и добавляем экран авторизации `./screens/AuthorizationScreen.js`, который содержит два окна для ввода данных и кнопку авторизации через Google или по почте-паролю.

Создаём проект на <https://firebase.google.com> с включённой аналитикой. Подключаем аутентификацию для проекта по почте, а также через логин и пароль от Google-аккаунта. Подключаем возможность использования мобильного приложения для данного проекта Firebase.

Для аутентификации через Google в приложении будем генерировать `webClientId` и вызывать экран авторизации. При успешном выполнении процесса приложение получит всю необходимую для работы информацию: уникальный токен, имя, фотография.

Создаём директорию серверной части, добавляем точку входа `index.js` и инициализируем проект. Устанавливаем необходимые для работы пакеты в проект.

Описываем работу приложения с файловой системой, для реализации функционала чтения и обновления информации в процессе работы, после чего описываем переменные окружения в файле `.env` и перейдём к написанию основного функционала в файле `./server/index.js`.

Создаём сам сервер и подключаем промежуточные обработчики окружения к нему для того, чтобы `express` мог отдавать статичные файлы относительно своей точки входа, обрабатывать сами запросы и сохранять информацию о них. Дополнительно добавляем способность возврата списка пользователей с помощью написанного помощника `fs.js`. Указываем порт, через который будем принимать запросы и подключаем к нему `socket.io`.

После этого описываем логику работы на клиентской части приложения. Подключаем необходимые библиотеки и создаём подключение по сокету к серверу.

ЗАКЛЮЧЕНИЕ

В данной работе были изучены основные понятия веб-доступности, проведено исследование существующих методов для разработки доступных веб-технологий, средств их реализации.

Веб-доступность является важной составляющей современного Интернета. Помимо обычных пользователей веб-контентом пользуются также люди с ограничениями здоровья, для которых иногда веб-сайты и приложения являются единственным способом для коммуникации и взаимодействию с окружающими. Поэтому современная всемирная паутина должна быть максимально доступной для группы лиц с ограничениями здоровья и должна позволять им пользоваться своими ресурсами так же, как и здоровым людям: понятным, доступным и удобным способом.

В практической части было разработано мобильное приложение, которое включает в себя новый способ взаимодействия пользователей, имеющими ограничения здоровья, со своим устройством, используя кодирование и декодирование названий и содержания деталей интерфейса в азбуку Морзе и передачу вибраций, соответствующей закодированной фразе. Для удобства пользования приложением были реализованы функции добавления персональной клавиатуры, безопасную авторизацию посредством сервиса Google и добавлена реализация серверной части приложения.

Основные источники информации:

1. Введение в веб-доступность [Электронный ресурс]:/ URL: <https://www.w3.org/WAI/fundamentals/accessibility-intro/ru#context> (Дата обращения: 12.05.2022)

2. Веб-доступность и стандарты W3C [Электронный ресурс]:/ URL: <https://www.w3.org/WAI/videos/standards-and-benefits/ru> (Дата обращения: 12.05.2022)
3. React Native [Электронный ресурс]:/ URL: <https://reactnative.dev> (Дата обращения: 16.05.2022)
4. Input and output devices for physically challenged users [Электронный ресурс] :/ URL: <https://www.siyavula.com/read/cat/grade-11-cat/social-implications-of-hardware/06-social-implications-of-hardware> (Дата обращения: 26.05.2022)
5. Ashish Nandan Singh «React and Firebase are all you need to host your web apps» [Статья]:/ URL: <https://www.freecodecamp.org/news/react-and-firebase-are-all-you-need-to-host-your-web-apps-f7ab55919f53> (Дата обращения: 26.05.2021)
6. Самарин А. Новые сенсорные технологии для компьютерных устройств ввода [Статья] // А. Самарин, «Компоненты и технологии», 2007 г. :/ URL: <https://cyberleninka.ru/article/n/novye-sensornye-tehnologii-dlya-kompyuternyh-ustroystv-vvoda> (Дата обращения: 26.05.2022)
7. А. Л. Мархакшинов, А. А. Тонхоноева, Е. Р. Урмакшинова. Разработка бессерверных мобильных приложений [Статья] // Вестник Новосибирского государственного университета. Серия: Информационные технологии, 2019 г. :/ URL: <https://cyberleninka.ru/article/n/razrabotka-besservernyh-mobilnyh-prilozheniy> (Дата обращения: 26.05.2022)