**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«Национальный исследовательский университет**

**«Высшая школа экономики»**

**Факультет математики, информатики и компьютерных наук**

***«Приложение для слабовидящих покупателей»***

***Курсовая работа***

по направлению подготовки 09.03.04 «Программная инженерия»

Студент группы 18-ПИ2

Данилов С. Д.

Научный руководитель

Старший преподаватель,

НН Базовая кафедра группы компаний "MERA"

Лейкин М.В.

Нижний Новгород, 2021

**Оглавление**

**Введение**

Рынок современных мобильных приложений стремительно растет. С каждым годом создается все больше библиотек, инновационных решений в сфере дизайна, интерфейса, функциональности. Смартфон стал незаменимой частью нашей жизни во многих сферах жизни - это и доставка еды, и вызов такси, покупка билетов. Все это явно улучшает качество жизни, делает обыденные вещи проще.

При всем многообразии решений в сфере мобильных технологий, не многие продукты адаптированы к использованию людьми с ограниченными возможностями, такими как нарушения зрения, слуха, ориентирования и так далее. Пользователю с перечисленными нарушениями важны такие вспомогательные возможности, как крупный, удобочитаемый шрифт, озвучивание текста, простой и понятный интерфейс и дополнительные улучшения, упрощающие работу с приложением Многие компании разработчики программного обеспечения просто игнорируют этот факт, что потенциальным пользователем может быть пользователь с отклонениями по здоровью, так как адаптация – это дорогой процесс, затратный по времени и требует дополнительных знаний. К счастью, многие исследовательские компании начали понимать необходимость поддержки решений для людей с ограниченными возможностями, и такой параметр, как accessibility (англ. доступность) стал неотъемлемым пунктом в оценке качества готового приложения (не только мобильного, но и веб приложения) Только если приложения, которые упрощают рутину во всех сферах жизни, станут более доступны и удобны людям с ограничениями по здоровью, жизнь таких людей станет намного приятнее, полноценнее, они смогут познавать и осуществлять деятельность наравне с обычными пользователями. Разве не в этом смысл создания мобильных приложений?

**Актуальность выбранной темы**

Проблемами доступности приложений на данный момент озабочены в большей степени IT “гиганты” – компании Google, Apple, Facebook, . На данный момент существует большое количество приложений в секции "Распознавание текста" или распознавание предметов на фото. Но не так много приложений удовлетворяют высокому уровню удобства для использования им людям со специальными возможностями. Я уверен, что устранение этого недочетаочень упростит жизнь таким пользователям. Поэтому изучение рекомендаций по дизайну и технологиям для слабовидящих и применение полученных знаний при проектировании приложений является остро актуальной темой.

**Цели и задачи**

Цель данной работы - изучить технологии создания приложений для людей с нарушениями зрения и на основе этих знаний создать Android приложение для слабовидящих покупателей. Данная цель подразумевает выполнение таких задач, как:

* Изучение пособий и материалов по проектированию интерфейсов мобильных приложений для людей с ограниченными возможностями, в частности - для пользователей с нарушениями зрения
* Приобретение знаний и навыков разработки мобильных приложений для платформы Android
* Изучение решений в области обработки изображений, распознавания и обработки текста, облачного хранения данных пользователей
* Проектирование приложения для слабовидящих пользователей на основе полученных знаний и его реализация

**Особенности приложений для слабовидящих**

При проектировании приложения для слабовидящих пользователей необходимо на самом раннем этапе выстроить особую стратегию построения дизайна и схемы взаимодействия между пользователем и программой. Главная мысль, которой нужно придерживаться при разработке стратегии – минимализм, фокус на звук, а не изображение и использование специальных возможностей.

* Минимализм

Что подразумевается под минимализмом если мы говорим о мобильных приложениях? Порой при использовании приложения мы можем столкнуться с большим количеством компонент интерфейса на одном экране или диалоге, такими как всевозможные кнопки, поля ввода, меню и так далее. Такое многообразие элементов усложняет процесс использования продукта, особенно если пользователь имеет проблемы со зрением. Именно в этом случае очень помогает соблюдение разумного минимализма, соблюдая который, мы оставляем только самое необходимое из интерфейса. К этому понятию можно также отнести отсутствие анимации и сложных элементов дизайна, так как их наличие может рассеять внимание пользователя и усложнить восприятие интерфейса, особенно если используется система озвучивания интерфейса TalkBack.

* Быстрое достижение цели пользователем

Также важным показателем качества приложения для слабовидящих является то, как быстро сможет пользователь достичь своей цели использования приложения. Порой поиск нужного элемента интерфейса, его распознавание и в целом ориентирование по приложению может занимать больше времени и сил для лиц с ограниченными возможностями. Если приложение не соответствует этому критерию качества, пользователь не сможет комфортно пользоваться продуктом, поэтому важно проработать сценарии использования и посторить граф активити таким образом, чтобы путь к критически важным функциям приложения был как можно меньше.

* Четкий текст и дизайн элементов

Эти два критерия являются необходимыми к соблюдению для всех приложений, не только к продуктам для людей с ограниченными возможностями, так как качественный дизайн делает процесс использования программного продукта намного приятнее. Но что касается типа приложений, описанных в данной работе, эти два критерия обретают больший смысл. В случае, если не используется система озвучивания интерфейса, пользователь будет ориентироваться на текст, символы, кнопки и т.д. , составляющие интерфейс, и важно, чтобы эти элемента интерфейса не были расплывчатыми и сложными к восприятию у людей с ограниченными возможностями. Поможет поддержать это требование соблюдение данных конкретных рекомендаций к дизайну:

* Контраст цветов (между текстом и фоном например)
* Крупный размер кегля и удобочитаемый шрифт
* Строгий дизайн элементов (минимальный угол, монотонная заливка, минимум анимации)
* Текстовое обозначение важных элементов интерфейса
* Голосовое сопровождение, интеграция с TalkBack

Незаменимым помощником для слабовидящих пользователей мобильных устройств на базе ОС Android является система озвучки интерфейса TalkBack (аналог для операционной системы iOS – VoiceOver) Он позволяет озвучивать информацию о происходящем на экране в целом или о выделенном элементе интерфейса. Так пользователь с ограниченными возможностями по зрению сможет найти нужный ему элемент интерфейса или получить результат работы приложения.

* Использование специальных возможностей

Зачастую

**Применение технологий компьютерного зрения   
в приложениях для слабовидящих**

Зачастую приложения со специальными возможностями становятся «глазами» человека, иными словами, они выполняют функции обработки изображения методами искусственного интеллекта, позволяя определять и различать предметы на изображении, распознавать текст и лица и многое другое, что потенциальный пользователь не может выполнять совсем, или это очень затруднительно из за проблем со зрением.

Описать **Google Firebase ML Vision**

**Обзор аналогов**

* TapTapSee: расскажет, что происходит вокруг

Приложение позволяет распознавать объекты, окружающие пользователя, позволяя ему легко ориентироваться в пространстве.

**Алгоритм приложения**

Основной функционал приложения включает в себя распознавание текста с этикетки продукта, его обработка и озвучивание.

Блок схема

Также приложение может распознавать информацию с ценников, находить подходящий к распознанному ранее продукту ценник и выводить информацию о цене продукта

И наконец, приложение может сохранять результаты сканирования в облаке для последующего их использования пользователем.

1. Создадим проект на Google Firebase

Нашему приложению понадобится доступ к облачным сервисам компании Google. Я выбрал именно эту компанию в качестве провайдера облачных технологий и решений в системе искусственного интеллекта по нескольким причинам – во-первых, корпорация предлагает очень низкую цену предоставления услуг и предоставляет полностью прозрачную систему начисления платежей. Также важным фактором является большая популярность применения облачных ИИ технологий и баз данных в разработке IT продуктов по всему миру, таких как Telegram, VK, . Конкретно нашему приложению понадобятся – распознавание текста с изображения, перевод со стороны искусственного интеллекта и доступ к базе данных для хранения информации пользователей.

1. Опишем view модель главного экрана

Руководствуясь правилами дизайна, описанными в теоретической части, создадим макет главного экрана. Интерфейс приложения будет состоять из таких элементов, как:

* Кнопка создания фото
* Кнопка перехода к истории сканирований
* Поле с изображением сканированной этикетки
* Поле с текстом

Кнопку с созданием нового фото я специально поместил в нижней части экрана, чтобы этот элемент можно было использовать без поиска по экрану, машинально. Кнопку с переходом к истории сканирований поместим повыше, так как она будет менее часто использоваться пользователем. Далее, поместим в центральной части приложения поле с текстом. Там будут размещаться либо распознанный текст, либо подсказки пользователю. Наконец, в верхней части экрана будет высвечиваться фото, созданное пользователем, которое больше будет иметь смысл для пользователей со слабыми нарушениями со зрением и не будет особо отвлекать внимание для людей с серьезными заболеваниями органов зрения.

Данный макет был спроектирован в приложении Figma, где я разместил вышеперечисленные элементы пользовательского интерфейса, настроил внешний вид элементов и описал получившийся дизайн в XML формате.

1. Опишем переход к диалогу камеры

Создадим обработчик нажатия кнопки сканирования этикетки. Подключим к кнопке обработчик событий, который будет инициировать переход к новому экрану с диалогом камеры при нажатии. У меня было два пути, как реализовать диалог камеры – либо спроектировать собственный интерфейс, либо воспользоваться системным интерфейсом Android. Первый подход намного сложнее в реализации, но зато он позволил бы адаптировать интерфейс под особенности приложений со специальными возможностями. Второй же подход легче и лаконичнее в реализации, плюс если пользователь уже работал с камерой, то ему будет легче разобраться в интерфейсе и совершить необходимое действе, но в таком случае невозможно переопределить элементы диалога, а это грозит тем, что важные элементы интерфейса будут малые по размеру или нечеткие для пользователя с нарушениями со зрением. Я выбрал второй вариант, но дополнительно вставил голосовые подсказки, которые помогут сориентироваться пользователю в случае, если будут затруднения.

1. Опишем класс обработки изображения

Для наиболее точного распознавания следует дополнительно обработать изображение. Для этого мы применим несколько алгоритмов – черно-белый фильтр, повышение контраста, … .

Изображения представлены в проекте как объекты класса *Bitmap*, содержащие матрицу пикселей. Фильтры, применяемые к изображению, являются небольшими матрицами, последовательно применяемые к участкам исходной матрицы изображения. Мы будем использовать черно-белый фильтр, чтобы убрать не имеющую особой важности при распознавании текста информацию о цвете пикселя, важен только оттенок черного или белого. Далее мы инвертируем изображение, меняя цвет или яркость на противоположный. Так мы сможем добиться высокого контраста изображения, более четкого выделения границ и, соответственно, качества распознавания символов. И наконец применим метод, нейтрализующий шумы с фотографии. Так мы уберем те выбросы в данных, которые могут сильно повлиять в худшую сторону при распознавании символов. Применив данные фильтры, мы отдаем обработанное изображение этикетки далее модулю распознавания текста.

1. Опишем класс распознавания текста

Для распознавания текста с этикетки мы будем использовать облачный сервис Google Firebase ML, алгоритм распознавания описан в теоретической части работы. Для обращения к сервису будем использовать Firebase API, на основе которого создадим метод передачи изображения на облачный сервер и последующее ожидание результата. После успешного подключения к сервисам Firebase наше приложение пересылает изображение на облачный сервер, где методами ИИ извлекаются объекты FirebaseTextBlock, которые содержат координаты области с текстом и сам распознанный текст. В случае ошибки обработки (нет стабильного соединения с интернетом, не возможно подключиться к сервисам Firebase) выводится оповещение на экран.

1. Оптимизация решения

К сожалению, решение с использованием объекта FirebaseVisionImage, построенного напрямую с объекта Bitmap, не дает должного качества распознавания, так как при таком подходе изображение сжимается перед отправкой, следовательно теряется качество исходного изображения. Вместо этого можно сохранять обработанное изображение в файле JPG в памяти устройства и передавать файл на сервер целиком. Такой подход заметно увеличивает качество распознавания, но вместе с этим увеличивает продолжительность передачи файла и в целом ожидания пользователя. Я выбрал решение с передачей целостного файла, так как качество распознавание текста намного важнее, чем скорость обработки. Если приложение отдаст пользователю в итоге неразборчивый, бессмысленный текст, но за короткий промежуток времени, тогда приложение не будет выполнять поставленной цели.

1. Опишем класс обработки и перевода текста

Так как точность распознавания текста не может быть 100%, в конечном результате анализа могут находиться лишние символы, слова либо с ошибками, либо не несущие никакого смысла выражения. Нужно попытаться максимально избавиться от этих элементов текста, чтобы на выходе получить максимально полезную для пользователя информацию о сканируемом приложением продукте. Для одиночных служебных символов, таких как «\», «(», можно применить регулярное выражение, используя которое обработчик будет проходить по тексту и удалять символы, удовлетворяющие заданному правилу. Для проверки орфографии наиболее подходящим будет использование сервиса \*\*\*. После отправки интересующего нас слова в API запросе, сервис предоставляет результат в виде более подходящих по написанию слов, либо ничего не возвращает если не возможно подобрать исправленный вариант.   
Наконец, для большего удобства подключим перевод распознанного текста на русский язык. На некоторых этикетках информация о продукте может быть только на иностранном языке, или в описании продукта могут встречаться термины на английском, непонятные пользователю. Для перевода используем сервис Google Cloud Translate, которому мы через API будем передавать распознанный текст с этикетки и запрос на перевод на русский язык.   
Применяя вышеперечисленные фильтры к тексту, в итоге приложение предоставляет информативный текст, который уже можно предоставить пользователю. Этот текст мы отображаем в текстовом поле на главном экране вместе с фотографией этикетки.

1. Добавим класс озвучки текста

Одного вывода текста на экран не достаточно для приложения в сегменте продуктов для слабовидящих. Необходимо также озвучить распознанный и обработанный приложением текст, чтобы информация с этикетки была понятна и удобна для восприятия пользователю с нарушениями со зрением. В реализации озвучивания самым распространённым инструментом является библиотека Text To Speech, входящая в стандартный Android API версии 4 и выше. На этапе инициализации приложения создается объект класса TextToSpeech, который в последствии будет озвучивать женским голосом на русском языке передаваемый ему текст. Как только текст будет распознан и к нему применены фильтры и перевод, он сразу же передается на озвучку. Средство TTS идеально справляется с поставленной задачей озвучивания текста, предоставляя плавное, слитное озвучивание предложений, что делает процесс использования приложения слабовидящим пользователям намного удобнее.

1. Добавим авторизацию через отпечаток пальца
2. Добавим сохранение результата сканирования
3. Добавим экран истории сканирований

**Тестирование получившегося решения**

Протестируем наше приложение в условиях приближенных к реальным. Первая часть – ручное UX тестирование. Максимальное

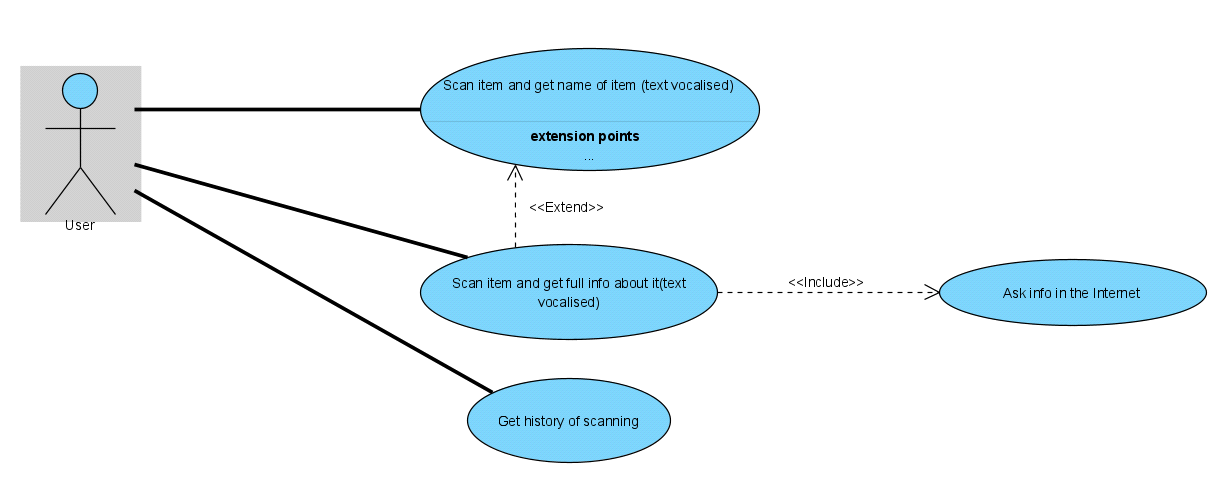
**Вывод**

Мной была поставлена цель разработать приложение для слабовидящих покупателей. Эта тема очень актуальна на текущий момент, так как создание или адаптация готовых приложений под нужды слабовидящих пользователей не часто применяется при разработке, а иногда. В ходе реализации заданной цели мною были выполнены изучение материалов по разработке интерфейсов для людей с ограниченными возможностями и разработке приложений для операционной системы Android. Руководствуясь полученными знаниями, я реализовал функциональную часть приложения и спроектировал дизайн. На каждом этапе создания продукта я строго следовал советам и рекомендациям по проектированию приложений для слабовидящих, чтобы в итоге получить качественный продукт. Полученное решение я протестировал, используя метод «слепого» тестирования, чтобы максимально приблизиться к ситуации реального использования продукта. Разработанное приложение имеет свои недостатки на текущем этапе и еще не может называться полноценным помощником для слабовидящих покупателей. Но функционал продукта может пополняться такими полезными возможностями, как предоставление советов по подбору продукта, сравнение цен в магазинах и так далее. Я уверен, что если разработчики мобильных приложений будут проявлять активность в сфере поддержки пользователей с нарушениями зрения, они не только не потеряют в прибыли, но приобретут станут известными как компании, заботящиеся о своих пользователях и стремящихся сделать жизнь более полноценной и удобной. «Доступность – хорошая метрика компании».

***Приложение***

**Техническое задание**

Техническое задание по созданию приложения для слабовидящих покупателей

* **Введение**
* **Название – «**Мобильное приложение для слабовидящих покупателей» (BuyVision)
* **Назначение и область применения**

Приложение позволяет сканировать этикетку\упаковку товаров и озвучивать наименование продукта, информацию с этикетки (состав, срок годности и т.д.) Мобильное приложение предназначено для людей с нарушениями зрения, которым тяжело разглядеть информацию на упаковке.

Функционал приложения включает в себя:

* Сканирование этикетки с помощью камеры
* Извлечение из изображения названия товара, описание, дополнительной информации
* Озвучивание полученной информации
* Получение дополнительной информации в сети Интернет
* Сохранение отсканированных товаров в облаке
* **Требования к приложению**
* Приложение должно быть адаптированно для людей с ограниченными возможностями: иметь минималистичный интерфейс, в дизайне долны преобладать крупные значки, шрифты и контрастные цвета, большая часть текстовой информации должна озвучиваться.
* Приложение должно делать снимки товара и передавать их на облачный сервис Firebase ML для распознавания текста или обрабатывать на устройстве с помощью библиотеки ML Kit. В случии успешного распознавания, полученный текст должен анализироваться специальным алгоритмом для выявления названия товара, описания, срока годности, состава, дополнительной информации.
* Приложение должно озвучивать наименование товара и, при желании пользователя, остальную информацию с помощью класса стандартной библиотеки Android «TextToSpeech» (возможно использование облачного сервиса Yandex SpeechKit).
* История сканирований и информации о продукте сохраняется на устройстве. Также она может быть сохранена и в базе данных Firebase Firestore, но для этого потребуется авторизация. Пользователь может создать аккаунт с помощью почты и пароля, или воспользоваться аккаунтом Google для авторизации. После успешной авторизации история сканирования синхронизируется с облаком на всех устройствах пользователя.
* Системные требования: ОС Android 6 и выше, камера 10 Мегапикселей и выше, подключение к Интернету.

**Стадии разработки**

* Разработка проектной документации
* Проектирование дизайна и интерфейса приложения
* Проектирование архитектуры приложения
* Проектирование функционала распознавания текста
* Создание алгоритма анализа полученной информации
* Проектирование системы озвучивания
* Разработка экранов приложения и интерфейса
* Добавление авторизации
* Добавление функционала работы с облачной базой данных

d. Тестирование готового продукта, испраление ошибок

e. Загрузка готового приложения на репозиторий Google Play

Начало разработки проекта – 1 января 2021 г.

Проект должен быть разработан – 1 июня 2021 г.

**Требования к дизайну**

Главный упор в дизайне должен быть на удобство использование слабовидящими пользователями. Следуя указаниям нормативов качества мобильных приложений (***источник***) – главные особенности такого приложения:

* Минималистичекий дизайн (меньше элементов управления)
* Контраст цветов ( палитра желтый\черный - #000000 #212121 #424242 #FFEA00 #c7fea0
* Крупные значки, шрифт (лучше динамический)
* Надписи к значкам – для озвучивания TalkBack
* Поддержка TalkBack, если отключен – озвучивание интерфейса
* Возможно управление с помощью кнопок громкости (возможно)
* Озвучивание текста (информация о продукте)