Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ТЕЛЕФОНОМ

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.201800.010.000ПЗ

Разработал:

студент группы ПР-21.101

Портнов М.А.

**Содержание**

[Введение 4](#_Toc166426129)

[1 Исследовательский раздел 5](#_Toc166426130)

[Описание предметной области 5](#_Toc166426131)

[Образ клиента 8](#_Toc166426132)

[Сценарии 8](#_Toc166426133)

[Сбор и анализ прототипов 9](#_Toc166426134)

[2 Проектирование приложения 12](#_Toc166426135)

[UI/UX дизайн проекта 12](#_Toc166426136)

[Выбор технологии, языка и среды программирования 14](#_Toc166426137)

[3 Разработка мобильного приложения 16](#_Toc166426138)

[3.1 Разработка мультимедийного контента 16](#_Toc166426139)

[3.2 Описание используемых плагинов 16](#_Toc166426140)

[3.3 Описание разработанных процедур и функций 17](#_Toc166426141)

[4 Тестирование 21](#_Toc166426142)

[Протокол тестирования дизайна приложения 21](#_Toc166426143)

[Протокол тестирования функционала приложения 21](#_Toc166426144)

[Заключение 24](#_Toc166426145)

[Библиография 2](#_Toc166426146)5

[Приложение А (обязательное) Техническое задание 2](#_Toc166426147)6

Введение

Мобильные устройства становятся все более важными инструментами для работы, общения и развлечений. С развитием мобильных технологий и расширением функциональности смартфонов возникает необходимость в создании инновационных способов управления устройствами. Одним из перспективных направлений в этой области является разработка мобильных приложений для дистанционного управления телефоном с использованием жестов. Это позволяет пользователям эффективно управлять устройством, не прибегая к физическому контакту с экраном, что особенно актуально в ситуациях, когда доступ к устройству ограничен или неудобен.

Целью курсового проекта является создание мобильного приложения дистанционного управления телефоном.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления функций приложения;
* разработка и реализация дизайна приложения;
* написание кода приложения;
* тестирование полученного продукта.

# Исследовательский раздел

## Описание предметной области

Одним из глобальных явлений современного времени стало распространение мобильных телефонов до такой степени, что они стали неотъемлемой частью повседневной жизни для миллиардов людей по всему миру. Подобно человеческому инстинкту к социализации, мобильные телефоны стали не просто устройствами связи, а центральным элементом взаимодействия с окружающим миром.

Каждый день миллионы людей используют мобильные телефоны для общения, работы, развлечений, поиска информации и многих других целей. Мобильные приложения и сервисы позволяют нам оставаться в контакте с друзьями и семьей, управлять своими делами, получать новости из мира и развлекаться в любое время и в любом месте.

Мобильные телефоны − великолепный инструмент, который стал неотъемлемой частью нашей жизни, однако, несмотря на все их преимущества, возникают ситуации, когда мы оказываемся в неудобном положении из-за невозможности использовать телефон.

Чувствительные экраны мобильных устройств требуют чистых рук для эффективного взаимодействия. Когда мы находимся в ситуации, когда руки загрязнены, например, при готовке, ремонте или работе в саду, нам приходится идти мыть руки, прежде чем воспользоваться телефоном. Это может занять дополнительное время и создать неудобства, особенно если у пользователя есть срочные вопросы или задачи, требующие его внимания.

В таких случаях становится очевидной необходимость в эффективном способе управления телефоном без использования чистых рук. Быстрые и интуитивно понятные жесты могут стать решением этой проблемы, позволяя нам взаимодействовать с устройством без необходимости прикасаться к экрану. Это не только экономит время, но и позволяет нам оставаться связанными с миром в любых условиях, сохраняя нашу продуктивность и комфорт.

Можно представить такую ситуацию: пользователь занят готовкой любимого блюда, и ему нужно переключить музыку. Вместо того чтобы вытирать руки, подходить к телефону и нажимать на экран, пользователь может просто поставить телефон вертикально на стол напротив себя, запустить приложение, оно будет работать в фоновом режиме. И в момент, когда пользователь показывает жест рукой, например, проводит с право налево − этот жест уже предварительно настроили на переключение музыки − телефон мгновенно распознает этот жест и включает следующий трек.

Это не только экономит время и усилия, но и делает процесс гораздо удобнее. Пользователь остается сосредоточенным на приготовлении блюда, не отвлекаясь на сложности взаимодействия с телефоном.

Такой подход к управлению телефоном открывает новые возможности для использования мобильных устройств в повседневной жизни, делая их более удобными, интуитивными и функциональными. Он позволяет нам оставаться продуктивными и связанными с миром, не отвлекаясь от текущих дел.

Помимо этого, сейчас существует множество приложений с помощью использования голосового помощника, который также помогает взаимодействовать с телефоном без помощи рук. Управление жестами, особенно в сценарии использования на кухне или в других ситуациях, когда руки заняты, может быть более предпочтительным, чем управление голосом, по следующим причинам:

− взаимодействие с устройством с помощью жестов может быть более естественным и интуитивно понятным для пользователей. Не всегда управление голосом может быть столь же прямолинейным и надежным, особенно в шумных или общественных местах, где распознавание голоса может быть затруднено;

− в некоторых случаях распознавание жестов может быть более точным и надежным, особенно если это требует точного тайминга или позиционирования. В то время как управление голосом иногда может вызвать недопонимание или ошибки распознавания, жесты могут быть выполнены точно и без задержек;

− управление жестами может быть более подходящим в условиях сильного шума или в среде, где использование голосовых команд нежелательно или невозможно, например, в библиотеке, на улице или в общественном транспорте.

Таким образом, использование компьютерного зрения для управления устройством при помощи жестов может представлять собой более удобный, эффективный и приватный способ взаимодействия с телефоном, особенно в ситуациях, когда руки заняты или условия окружающей среды делают управление голосом менее практичным.

## Образ клиента

Пользователями являются владельцы сенсорных мобильных телефонов. Приложение предназначено для людей различных возрастов, но основным контингентом будет люди с возрастом от 16 до 45 лет. Приложение удобно как для людей, которые часто готовят, так и для тех, кто занимается спортом, а также просто любит запачкать руки.

## Сценарии

Сценарии – это правдоподобные ситуации, происходящие с каждым образом пользователя.

Сценарий 1: переключение музыки.

Пользователь готовит пирог, телефон стоит вертикально, и, чтобы не мыть руки, не вытирать их сухим полотенцем, он может сделать жест рукой, чтобы переключить музыку.

Сценарий 2: снимок.

Пользователь работает в саду в перчатках. В моменте его попросили прислать фотографию, здесь он может просто показать в камеру жест рукой и телефон сделает снимок.

Сценарий 3: постановка музыки на паузу.

Пользователь занят уборкой дома, ему начинает говорить что-то другой, чтобы его услышать, пользователю необходимо убрать все принадлежности, снять перчатки и выключить музыку. Вместо этого он может показать в камеру жест и музыка тут же остановится.

Сбор и анализ прототипов

Сбор и анализ прототипов мобильного приложения для дистанционного управления телефоном включает в себя анализ конкурентов и определение сильных и слабых сторон их приложений. Необходимо рассмотреть дизайн, функционал и ошибки, чтобы определить, какие функции необходимо добавить в собственное приложение, а какие - исключить. Также важно учесть потребности целевой аудитории и создать продукт, который будет отвечать их запросам.

Мобильное приложение «Air Gesture Control» представляет собой инструмент для управления мобильным устройством с помощью жестов в воздухе. Оно позволяет пользователю выполнять различные действия, такие как прокрутка экрана, управление медиафайлами, открытие приложений и другие, используя жесты рук перед датчиками устройства. Приложение обычно предоставляет пользователю возможность настраивать различные жесты в соответствии с его потребностями и предпочтениями. В основе работы приложения лежит использование встроенных сенсоров устройства, таких как датчики движения и камера, для распознавания и интерпретации жестов пользователя.

Преимущества и возможности приложения «Air Gesture Control»:

* простой интерфейс;
* позволяет настраивать различные жесты для выполнения различных действий, что обеспечивает гибкость в использовании;
* наличие дополнительных функций, таких как управление презентациями и включение фонарика.

Недостатки приложения «Air Gesture Control»:

* возможности приложения могут быть ограничены по сравнению с более сложными приложениями или устройствами для управления жестами;
* некоторые функции приложения могут быть недоступны на определенных устройствах или версиях операционной системы Android, что может создавать неудобства для пользователей.
* использование управления жестами может быть подвержено ошибкам и задержкам, особенно в условиях низкого освещения или наличия помех в окружающей среде.

В целом, приложение "Air Gesture Control" предоставляет удобный и интуитивно понятный способ управления устройством с помощью жестов, однако оно может не подходить для всех пользователей из-за своих ограниченных возможностей и возможных недочетов. Приложение доступно бесплатно для установки на устройства Android (см. Рисунок 1).

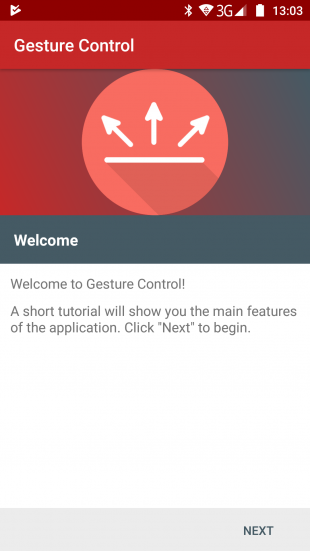


Рисунок 1 − мобильное приложение «Air Gesture Control»

Таким образом во всех таких предложениях требуются такие функции как:

− интеграция с камерой;

− пользовательские жесты.

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта разработан в программе Figma, так как рабочее пространство Figma максимально удобно и интуитивно понятно. Если человек, далекий от дизайнерских программ, откроет Photoshop или иллюстратор, то наверняка впадет в ступор от обилия кнопок, панелей и инструментов. В Figmа все гораздо проще, поэтому освоить ее на базовом уровне не составило никакого труда.

Приложение имеет один экран: главный, предназначенный для основной функциональности т.е. включение и выключения курсора.

Главный экран приложения содержит элементы управления, позволяющие пользователю выполнять различные действия с помощью жестов. Основные элементы управления на главном экране представлены интуитивно понятными жестами, которые пользователь может легко освоить.

Общий дизайн приложения основан на едином стиле, включающем в себя удобную цветовую схему и читаемый текст.

Светлая тема экрана и обеспечивает хорошую читаемость контента на экране, при этом не отвлекая внимание от основного функционала.

Целью данной работы является создание удобного и понятного пользовательского интерфейса для приложения управления телефоном с помощью жестов.

Ниже на рисунке 2 представлен логотип приложения, на котором изображена тематическая приложению иллюстрация.



Рисунок 2 – Логотип приложения

На рисунке 3 ниже представлена страница главного экрана.

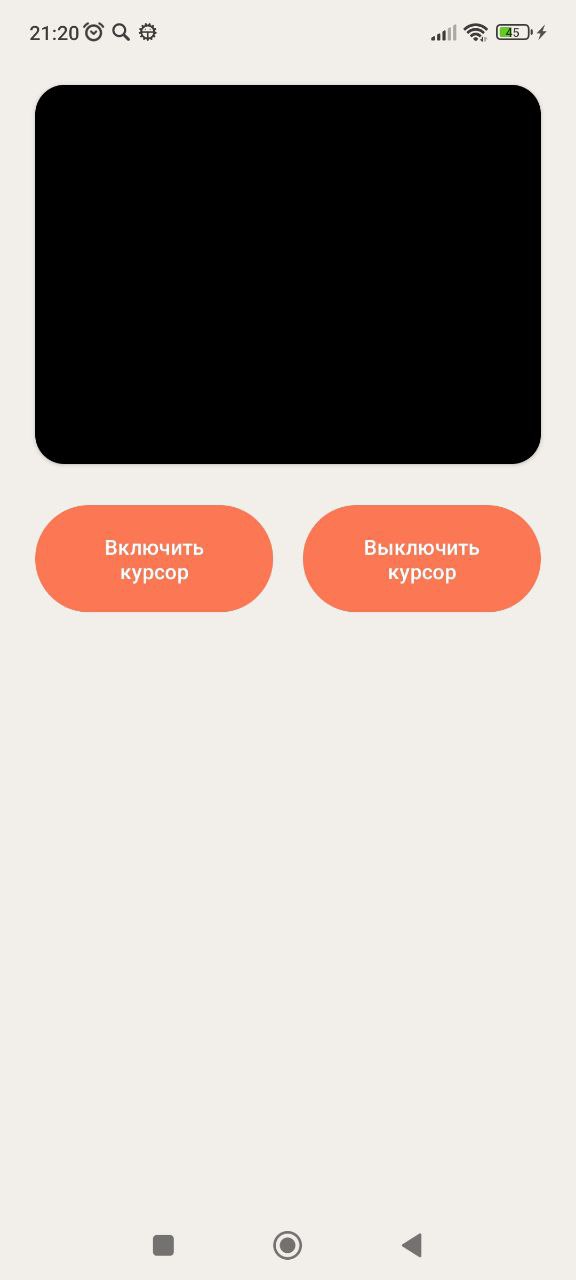


Рисунок 3 – Страница «Главный экран»

На данной странице мобильного приложения предоставлена возможность включить или выключить курсор, который будет следить за движениями руки.

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio версии: 2022.3.1 Android Studio – универсальная среда разработки, так как позволяет оптимизировать работу будущих приложения для работы не только на смартфонах, но и на планшетах, портативных ПК, которые работают на основе рассматриваемой операционной системы. В программу встроен эмулятор, позволяющий проверить корректную работу приложения на устройствах с разными экранами: с различными соотношениями сторон.

Android Studio предоставляет разнообразные инструменты для сборки и организации кода, что значительно облегчает процесс разработки и поддержки приложений. Благодаря богатому функционалу интегрированной среды разработки (IDE), разработчики получают доступ к удобным средствам для создания пользовательского интерфейса, включая интуитивно понятный Drag-n-Drop функционал. Он позволяет легко перемещать и располагать компоненты интерфейса в рабочей области IDE. Таким образом, разработчики могут более эффективно взаимодействовать с элементами пользовательского интерфейса и ускорить процесс создания интерактивных приложений.

В качестве языка программирования был выбран Кotlin. Благодаря строгой типизации и хорошо разработанным механизмам безопасности, приложения на Кotlin часто считаются надежными и стабильными. Одним из ключевых преимуществ Кotlin является его обширная документация, а также большое количество библиотек и фреймворков, которые облегчают разработку приложений. Помимо этого, Kotlin имеет простоту и ясность кода. Он предлагает более краткий и понятный синтаксис по сравнению с другими языками программирования, что упрощает написание и понимание кода.

# Разработка мобильного приложения

## 3.1 Разработка мультимедийного контента

Весь мультимедийный контент, который используется в приложении, взят из открытых источников, которые распространяют мультимедийный контент под лицензией CreativeCommons − NoDerivs.

## 3.2 Описание используемых плагинов

Ниже на рисунке 5 изображено название эмулятора для запуска приложения в Android Studio.

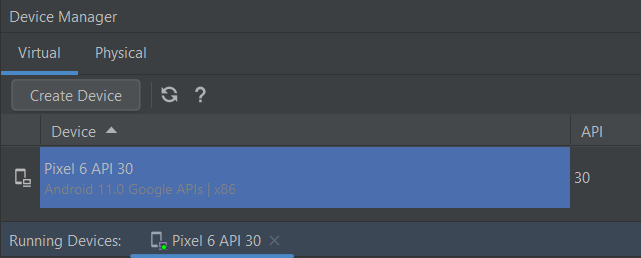


Рисунок 5 – Эмулятор

Для разработки использованы встраиваемые плагины, которые создаются вместе с проектом, а также дополнительные плагины, которые нужны для увеличения функциональности приложения в коде представлены

implementation("androidx.core:core-ktx:1.12.0")

implementation("androidx.appcompat:appcompat:1.6.1")

implementation("com.google.android.material:material:1.11.0")

implementation("androidx.constraintlayout:constraintlayout:2.1.4")

implementation("androidx.activity:activity-ktx:1.4.0")

// cameras

implementation("androidx.camera:camera-core:1.3.2")

implementation("androidx.camera:camera-camera2:1.3.2")

implementation("androidx.camera:camera-lifecycle:1.3.2")

implementation("androidx.camera:camera-view:1.3.2")

// viewModel for activity

implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.7.0")

implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0")

// MediaPipe Library

implementation("com.google.mediapipe:tasks-vision:0.20230731")

implementation("com.google.firebase:firebase-firestore-ktx:24.11.0")

implementation("androidx.camera:camera-lifecycle:1.3.3")

testImplementation("junit:junit:4.13.2")

androidTestImplementation("androidx.test.ext:junit:1.1.5")

androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

implementation ("androidx.lifecycle:lifecycle-service:2.4.1")

implementation ("androidx.lifecycle:lifecycle-runtime-ktx:2.4.1")

В проекте осуществляется работа со встроенными и сторонними библиотеками. Список всех библиотек и их описание представлен в таблице 1.

Таблица 1 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
| CameraX | Дополнение к Jetpack, которое упрощает добавление возможностей камеры в приложение. Предоставляет API для работы с камерой |
| LiveData | Компонент LiveData — предназначен для хранения объекта и разрешает подписаться на его изменения |
| ViewModel | Объект ViewModel содержит бизнес-логику обработки данных для взаимодействия с моделью |
| MediaPipe | Библиотека для выполнения задач компьютерного зрения, таких как обнаружение объектов или распознавание жестов |
| Firestore | Предоставляет клиентские библиотеки для работы с Firebase Firestore в приложениях Android, упрощая доступ к базе данных в реальном времени |
| Lifecycle | Предоставляет классы и методы для создания служб с учетом жизненного цикла приложения, обеспечивая их корректную работу в контексте жизненного цикла приложения |

Предоставленные библиотеки имеют очень важные зависимости, без которых приложение не будет функционировать в полной мере.

## 3.3 Описание разработанных процедур и функций

В мобильных приложениях процедуры и функции используются для создания модульного кода, который облегчает поддержку, тестирование и расширение приложения. В приложении реализованы методы и классы, для выполнения сразу нескольких методов.

В приложении разработаны следующие функции:

* включение курсора;
* выключение курсора.

На таблице 2 описаны классы, содержащие в себе различные методы:

Таблица 2 – Классы приложения

|  |  |
| --- | --- |
| Класс | Описание |
| HandLandmarkerHelper | класс, который обеспечивает интеграцию с API библиотеки MediaPipe для обнаружения и отслеживания рук на изображениях и видео в приложении Android |
| OverlayView | Представляет собой пользовательский вид для отображения результатов обнаружения и отслеживания рук на изображении или видео в приложении Android с использованием библиотеки MediaPipe |
| CursorService | Служба доступности Android, которая обеспечивает отображение и управление курсором на экране с использованием сервиса доступности и взаимодействия с API для обнаружения рук |
| GestureInvoker | Обрабатывает жесты рук, определяет действия клика и удерживания для управления курсором |
| HandDetectionService | Предоставляет сервис для обнаружения рук с использованием камеры и обработки изображений для анализа |
| TransferChannel | Предоставляет канал для передачи данных о обнаруженных руках между различными компонентами приложения, такими как сервисы и слушатели, позволяя им обмениваться результатами анализа положения рук |
| SmartControlApplication | Создает канал уведомлений для запускающихся сервисов |

Код, который выполняет одни и те же действия в разных частях приложения, можно упаковать в процедуры и функции, чтобы избежать его повторения.

В приложении реализованы методы, описанные на таблице 3:

Таблица 3 – Функции приложения

|  |  |
| --- | --- |
| Функции | Описание |
| startHandCursor() | Проверяет, есть ли у приложения необходимые разрешения для работы с курсором, и при их наличии запускает сервис, который управляет этим курсором |
| stopHandCursor() | Используется для остановки сервиса, который управляет курсором |

Продолжение таблицы 3

|  |  |
| --- | --- |
| checkPermissions() | Проверяет наличие необходимых разрешений. Если служба доступности не включена, то будет открыт экран настроек доступности. Затем будут запрошены разрешения на доступ к уведомлениям, камере и медиапроекции переднего плана |
| isAccessibilityServiceEnabled() | Извлекает значение настройки доступности и проверяет, соответствует ли указанная служба включенным службам доступности |
| setupHandLandmarker() | Настраивает параметры для определения положения рук, обрабатывая возможные ошибки |
| detectAsync() | Выполняет обнаружение положения рук асинхронно на изображениях, и возвращает результаты обнаружения или null в случае ошибки |
| returnLivestreamResult() | Формирует и передает результаты обнаружения положения рук в реальном времени через слушателя |
| returnLivestreamError() | Передает сообщение об ошибке через слушателя |
| onCreate() | Инициализирует службу курсора, настраивает параметры отображения и выводит всплывающее сообщение о инициализации службы |
| onDestroy() | Выполняет необходимые действия перед завершением службы, такие как удаление курсора и вывод всплывающего сообщения о завершении службы |
| onStartCommand() | Определяет действия, выполняемые в зависимости от полученного намерения, создавая или удаляя курсор |
| setupDisplayMetrics() | Настраивает параметры дисплея для корректного отображения курсора, учитывая версии Android |
| removeCursor() | удаляет курсор и останавливает поток, отвечающий за его движение, а также останавливает сервис обнаружения рук |
| createCursor() | Создает курсорную иконку, запускает службу обнаружения рук в первом плане, устанавливает параметры окна для курсора, добавляет курсорное представление на экран и инициирует поток для движения курсора, а также устанавливает слушатель для получения результатов обнаружения рук |
| move() | Перемещает курсор в соответствии с координатами обнаруженной руки, обеспечивая его плавное движение по экрану |

Продолжение таблицы 3

|  |  |
| --- | --- |
| click() | Симулирует нажатие на экран в текущих координатах курсора, создавая описание жеста и отправляя его для исполнения системой, что приводит к выполнению действия, соответствующего клику |
| stopPress() | Останавливает симуляцию нажатия на экране, создавая описание жеста от текущих координат курсора до предыдущих координат, что приводит к отпусканию "нажатия" |
| handleLandMark() | Обрабатывает обнаруженные координаты ключевых точек руки, масштабирует их относительно размеров экрана и вызывает соответствующее действие для обработки жеста |
| handleClickAction() | Определяет, следует ли выполнить клик или удержание, основываясь на расстоянии между указательным и большим пальцами, и времени удержания |
| startCamera() | Инициализирует камеру, настраивая предварительный просмотр и анализ изображения для обнаружения рук в реальном времени с помощью фронтальной камеры, используя формат изображения RGBA\_8888 и стратегию обработки задержек |
| onResults() | Получает результаты обнаружения положения рук, извлекает координаты ключевых точек руки и передает их через канал данных |

Можно подметить, что каждая из функций структурирует код и позволяет реализовать паттерны такие как MVC. Важный фактор – соответствие функции своему классу.

# Тестирование

## Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение было выбрано устройство Xiaomi Redmi Note 10S разрешение экрана 1080×2400.

Проверка была проведена на:

* оптимальный размер кнопок;
* читабельный размер шрифта;
* приятное глазу цветовое сочетание;
* корректное расположение элементов.

В результате тестирования дизайна все пункты были пройдены.

## Протокол тестирования функционала приложения

Для тестирования функционала были разработаны тест-кейсы, каждый из которых описывает небольшую последовательность действий с конкретным результатом. Благодаря тест-кейсам всегда известно, как и что протестировать оптимальным количеством проверок, и не забыть о нюансах, так как записан каждый шаг. Разработанные тест-кейсы представлены в таблицах от 4.1 до 4.9.

Таблица 4.1 – Test-case 1

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Дымовое тестирование | | |
| Функция: | Запуск приложения | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
|  | |  |  |
| Шаги теста: | |  |  |
| Нажатие на ярлык приложения | | Корректное открытие приложения | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.2 – Test-case 2

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Работа навигации на главном экране | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку | | Открытие окна настроек | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку меню «Настройки» | | Перемещение на страницу настроек | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.3 – Test-case 3

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Включение курсора | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Работоспособность курсора | | Включение курсора | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку «включить курсор» | | Включение курсора и вывод сообщение об успешном начале работы | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.5 – Test-case 5

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Выключение курсора | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |

Продолжение таблицы 4.5

|  |  |  |
| --- | --- | --- |
| Предусловие: |  |  |
| Запуск приложения | Корректная работа приложения | пройден |
| Работоспособность курсора | Выключение курсора | пройден |
| Шаги теста: |  |  |
| Нажатие на кнопку «выключить курсор» | Выключение курсора и вывод сообщение об успешном завершении работы | пройден |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 4.6 – Test-case 6

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Распознавание жеста | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Работоспособность курсора | | Включение курсора | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку «Включить курсор» на странице главного экрана | | Включение курсора | пройден |
| Выполнение жеста рукой | | Распознавание жеста с помощью камеры и вызов определенного действия | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Заключение

Курсовой проект был успешно завершен с целью разработки мобильного приложения для дистанционного управления телефоном. В ходе выполнения проекта были осуществлены исключительно важные задачи, которые в итоге позволили создать интуитивно понятное и работающее приложение для управления телефоном с помощью жестов.

Первоначально была проведена тщательная исследовательская работа по изучению предметной области, что позволило углубиться в основы управления телефоном с помощью компьютерного зрения и изучить наиболее актуальные разработки на рынке. Благодаря этому полученное понимание потребностей пользователей позволило определить основные функциональные возможности, которые должны были быть включены в приложение.

Следующим этапом был тщательный анализ приложения с точки зрения конечных пользователей. Были выявлены ключевые функции, которые пользователи ожидают от данного приложения, такие как доступ к камере, возможность скрывать вкладки, переключаться между ними, перелистывание музыкальных треков, их остановка, а также поддержка настроек.

На основе собранных данных был разработан пользовательский интерфейс, который отличается своей простотой. Необходимый функционал был реализован с высокой степенью надежности, что было достигнуто путем тщательного тестирования приложения. Тесты были проведены с целью обеспечить стабильность работы, соответствие заявленным требованиям и общую удовлетворенность пользователей. Выявленные ошибки и недочеты были исправлены, чтобы обеспечить безупречный пользовательский опыт.

В итоге, результатом данного курсового проекта стало создание приложения для дистанционного управления телефоном, которое содержит в себе мощные функциональные возможности, способное обнаруживать и интерпретировать жесты пользователя, что позволяет эффективно управлять устройством без прямого физического контакта с экраном, не прерывая текущие действия.

Библиография

Нормативно-правовые акты:

1 ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва : Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1 Android Developers [электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/media/camera/camerax (дата обращения: 15.01.2024)

2 CameraX [Электронный ресурс]. – Управление камерой. – URL: https://learningprogramming.net/mobile/android/line-chart-in-android/ (дата обращения: 21.02.2024)

3 Lifecycle [электронный ресурс]. – Жизненный цикл активности – URL: https://developer.android.com/guide/components/activities/activity-lifecycle (дата обращения: 25.02.2024)

4 MediaPipe [электронный ресурс]. – MediaPipe Framework на Android. – URL: https://developers.google.com/mediapipe/framework/getting\_started/ (дата обращения: 18.03.2024)

5 Figma [электронный ресурс]. – Прототипирование проектов. – URL: https://www.figma.com/ (дата обращения: 14.02.2024

**Приложение А**

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

Техническое задание

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ТЕЛЕФОНОМ

НАТКиГ.201800.010.000ПЗ

Выполнил:

студент группы ПР – 21.101

Портнов М. А.

2024

**Содержание**

[Введение 36](#_Toc120314003)

[1Назначение разработки 36](#_Toc120314004)

[2 Требования к мобильному приложению 37](#_Toc120314005)

[2.1 Требования к функциональным характеристикам 37](#_Toc120314006)

[2.2 Требования к надёжности 37](#_Toc120314007)

[2.3 Условия эксплуатации 37](#_Toc120314008)

[2.4 Требования к составу и параметрам технических средств 37](#_Toc120314009)

[2.5 Требования к информационной и программной совместимости 38](#_Toc120314010)

[2.6 Требования к защите информации 38](#_Toc120314011)

[2.7 Требования к маркировке и упаковке 38](#_Toc120314012)

[3 Требования к программной документации 38](#_Toc120314013)

[4 Технико – экономические показатели 38](#_Toc120314014)

[5 Стадии и этапы разработки 38](#_Toc120314015)

[6 Порядок контроля и приёмки 39](#_Toc120314016)

**Введение**

Настоящее техническое задание распространяется на разработку информационной системы «разработка приложения для дистанционного управления телефоном».

Наименование приложения: «Hand control».

Краткая характеристика области применения: приложение для пользователей, заинтересованных в дистанционном управлении телефоном при помощи жестов.

Основанием для проведения разработки является Протокол № Уч-34/1 от 22 февраля 2024 года.

Наименование темы разработки – «разработка приложения для дистанционного управления телефоном».

Условное обозначение темы разработки – «Hand control»

1. **Назначение разработки**

Основное назначение мобильного приложения заключается в:

* дистанционное управление телефоном с помощью жестов;

Лица, которые могут работать с данной системой:

* любой пользователь.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* включение курсора;
* распознавание жестов;
* выключение курсора;

Выходные данные должны быть организованы в виде выводимого, в специальную форму, результата, соответствующего определённому шаблону.

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация бесперебойного питания оборудования пользователя;
* использование лицензионного программного обеспечения;
* организация стабильного интернет – соединения.

Приложение должно контролировать выходную информацию:

* соблюдение типов данных при выводе результата;
* операции сброса результатов, при длительном нажатии на экран.

**2.3 Условия эксплуатации**

Обслуживание информационной системы включает в себя:

* информационное обслуживание – вывод информации;
* решение проблем при отладке.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения на устройстве требуется: объем свободной памяти не менее 200 МБ, версия устройства Android 7 и выше.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Kotlin.

**2.6 Требования к защите информации**

Доступ к информации предоставляется только пользователю приложения.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* проектную документацию.

**4 Техника – экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

# Стадии и этапы разработки

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название этапа | Срок, даты | Отчётность |
| 1 | 2 | 3 | 4 |
| 1 | Исследование предметной области | 20.02.2024 – 21.02.2024 | Пояснительная записка |
| 2 | Выбор моделей, описывающих предметную область | 22.02.2024 – 23.02.2024 | Пояснительная записка |
| 3 | Разработка технического задания | 24.02.2024 – 27.02.2024 | Техническое задание |
| 4 | Изучение отдельных спецификаций Kotlin | 01.03.2024–  06.03.2024 | Пояснительная записка |
| 5 | Анализ требований и уточнение спецификаций | 08.03.2024 | Спецификации программного обеспечения |
| 6 | Проектирование структуры мобильного приложения, проектирование компонентов (технический проект) | 10.03.2024 – 14.03.2024 | Схема структурная системы и спецификации компонентов |
| 8 | Кодирование клиентской части | 16.03.2023 – 16.04.2023 | Программный  Продукт |
| 9 | Тестирование компонентов.  Сборка и комплексное тестирование | 18.04.2024 | Тексты программных компонентов |
| 10 | Разработка программной  документации | 20.04.2024 – 10.05.2024 | Программная  документация |
| 12 | Защита | 15.05.2024 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;
* презентация.