

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН**

**Казахский национальный исследовательский технический университет
имени К.И. Сатпаева**

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

Эйсмонт Вячеслав Владимирович

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

Образовательная программа: 6B06102 Computer Science

Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И. Сатпаева

Институт автоматики и информационных технологий

Кафедра «Программная инженерия»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ПИ
канд. тех. наук, ассон. профессор
Ф.Н. Абдолдина
"28" 05 2025 г.



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Разработка мобильного приложения для альтернативных способов
управления Android-устройствами»

Образовательная программа: 6B06102 Computer Science

Выполнил

Эйсмонт В.В.

Рецензент

Научный руководитель
магистр технических наук
стар. преподаватель

PhD, ассистент-профессор,
МУИТ

А.Т. Бисаринова
"19" мая 2025 г.

А.М. Амреева
"27" мая 2025 г.



Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И. Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»



УТВЕРЖДАЮ

Заведующий кафедрой ПИ
канд. тех. наук, ассоц. профессор
Ф.Н. Абдолдина
2025 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся: *Эйсмонт В.В.*

Тема: Разработка мобильного приложения для альтернативных способов управления Android-устройствами

Утверждена приказом проректора по академической работе:

№ 1804-до

	От «24» ноября 2024 г.
Срок сдачи законченного проекта	« 08 » 05 2025 г.

Исходные данные к дипломному проекту:

- А) Анализ предметной области, анализ аналогичных проектов;
- Б) Разработка технического задания;
- В) Разработка архитектуры ПО;
- Г) Проектирование и разработка понятного и интерактивного дизайна;
- Д) Разработка и тестирование ПО;

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): представлены 30 слайда презентации.

Рекомендуемая основная литература: из 25 наименований.

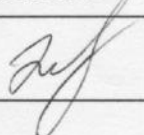
ГРАФИК
подготовки дипломного проекта

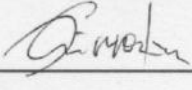
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, разработка технического задания	25.10.24	Выполнено
2. Выбор технологий для разработки	01.11.24	Выполнено
3. Разработка логики взаимодействия	10.11.24	Выполнено
4. Разработка функционала системы	01.12.24	Выполнено
5. Тестирование и оптимизация	04.01.25	Выполнено
6. Написание пояснительной записки к дипломному проекту	01.05.25	Выполнено

Подписи

консультантов и нормоконтролера на законченный дипломный проект
с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись	Оценка
Программное обеспечение	Шаханов Э.Р. Преподаватель	27.05.2025		100
Нормоконтролер	Ердалиева З. У. Преподаватель	27.05.2025		

Научный руководитель  Амреева А.М.

Задание принял к исполнению обучающийся  Эйсмонт В.В.

Дата « 08 » 05 2025 г.

АНДАТПА

Осы дипломдық жұмыста Android операциялық жүйесіне негізделген құрылғылармен баламалы өзара әрекеттесуді қамтамасыз ететін AIS-A (Alternative Interaction System for Android) мобильді қосымшасы әзірленді. Қосымша қимыл-қозғалысы шектеулі пайдаланушыларға (жарақаттар, инсульт, сал ауруы және т.б. салдарлары) арналған, сондай-ақ классикалық сенсорлық енгізу қиындаған жағдайларда (экранның зақымдануы, проекторды пайдалану, hands-free қажеттілігі және т.б.) қолдануға болады.

Төрт басқару режимі ұсынылған:

T-PAD (экран тачпад рөлінде),

HEAD (бас бұрылуы арқылы курсормен басқару),

EYES (бас бағыты мен көздің шамамен орналасуы арқылы басқару),

G-sens (құрылғыны еңкейту арқылы басқару — акселерометр/гироскоп негізінде).

Әрбір режимде виртуалды курсор, негізгі әрекеттерге арналған белсенді аймақтар («Басты бет», «Артқа», «Солға/оңға сырғыту») және бапталатын «тап» механизмі (курсордың кідірісі, дыбыс, бет қимылдары және т.б.) қарастырылған. Қосымша Kotlin тілінде, Android AccessibilityService және камера/аудио/сенсорларды өңдеуге арналған қосымша кітапханаларды пайдалану арқылы әзірленді.пайдалы.

АННОТАЦИЯ

В настоящей дипломной работе разработано мобильное приложение AIS-A (Alternative Interaction System for Android), обеспечивающее альтернативные способы взаимодействия с устройствами на базе ОС Android. Приложение предназначено для пользователей, имеющих ограничения в двигательной активности (последствия травм, инсультов, параличей и пр.), а также для ситуаций, когда классический сенсорный ввод затруднён (повреждение экрана, использование проектора, необходимость hands-free и т.д.).

Предлагается четыре режима управления:

T-PAD (экран в роли тачпада),

HEAD (управление курсором при помощи поворота головы),

EYES (управление направлением головы и ориентировочным положением глаз),

G-sens (управление путём наклона устройства — акселерометр/гироскоп).

Каждый режим предусматривает виртуальный курсор, набор активных зон для основных действий («Домой», «Назад», «Свайп влево/вправо») и настраиваемый механизм «тапа» (задержка курсора, звук, жесты лица и т.д.). Приложение разработано на языке Kotlin с использованием Android AccessibilityService и дополнительных библиотек для обработки камеры/аудио/сенсора.

ABSTRACT

In this thesis, a mobile application called AIS-A (Alternative Interaction System for Android) has been developed to provide alternative methods of interaction with devices running the Android operating system. The application is designed for users with limited motor abilities (due to injuries, strokes, paralysis, etc.) as well as for scenarios where traditional touch input is difficult (e.g., damaged screen, projector usage, hands-free operation, etc.).

Four control modes are offered:

T-PAD (using the screen as a touchpad),

HEAD (cursor control through head movements),

EYES (control via head direction and approximate eye position),

G-sens (control by tilting the device — using accelerometer/gyroscope).

Each mode includes a virtual cursor, a set of active zones for basic actions (“Home,” “Back,” “Swipe left/right”), and a customizable “tap” mechanism (cursor delay, sound input, facial gestures, etc.). The application is developed in Kotlin using Android AccessibilityService and additional libraries for camera, audio, and sensor processing.

СОДЕРЖАНИЕ

	ВВЕДЕНИЕ	10
1	Анализ предметной области и постановка задачи	11
1.1	Анализ предметной области и обоснование актуальности	11
1.2	Анализ существующих решений и постановка задачи	13
1.2.1	Функциональные	15
1.2.2	Нефункциональные требования	15
1.3	Обоснование выбора технологий и инструментов разработки	15
1.3.1	Язык программирования Kotlin	16
1.3.2	Среда разработки Android Studio	16
2	ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ (ПРИЛОЖЕНИЯ)	21
2.1	Архитектура приложения и организационная структура проекта	21
2.2	Проектирование сервисов для каждого режима (T-PAD, HEAD, EYES, G-sens)	22
2.2.1	TouchpadAccessibilityService (T-PAD)	23
2.2.2	HeadAccessibilityService (HEAD)	23
2.2.3	EyeAccessibilityService (EYES)	24
2.2.4	GSensorAccessibilityService (G-sens)	25
2.3	Описание программных модулей управления	25
2.3.1	Сервис Head (HeadAccessibilityService)	26
2.3.2	Сервис EYES (EyesAccessibilityService)	28
2.3.3	Сервис T-PAD (TouchpadAccessibilityService)	30
2.3.4	Сервис G-SENS (GSensorAccessibilityService)	32
2.4	Структура данных и хранение настроек	34
2.4.1	Хранение пользовательских параметров через SharedPreferences	35
2.4.2	Определение интерфейсов через XML-файлы	36
2.5	Взаимодействие между настройками и интерфейсом	38
2.6	Разработка и реализация приложения AIS-A	39
2.6.1	Проблемы, возникшие в ходе разработки, и пути их решения	41
2.6.2	Общий процесс разработки и интеграция модулей	41
3	ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ (методика, результаты)	44
3.1	Средства тестирования	45
3.2	Результаты тестирования	47
3.3	Тестирование пользователями с двигательными ограничениями	50
3.4	Тестирование с пользователем, имеющим последствия тяжёлой черепно-мозговой травмы	50
3.4.1	Переход к управлению через акселерометр (режим G-sens)	52

3.4.2	Попытка использования режимов EYES и HEAD	57
3.4.3	Вывод по наблюдению	57
3.5	Тестирование с пользователем, имеющим последствия спинальной травмы	58
3.5.1	Использование режима EYES	58
3.5.2	Переход к режиму G-sens	59
3.5.3	Попытка использования режима T-PAD	61
3.5.4	Переход к управлению через движения головы (HEAD)	62
3.5.5	Вывод по наблюдению	63
3.5.6	Обобщённый вывод	64
3.6	Другие реальные сценарии практического использования	65
3.7	Развёртывание и распространение	66
3.7.1	Текущий статус	67
3.8	Экономическое обоснование проекта	68
3.8.1	Трудозатраты и оценка стоимости разработки	68
3.8.2	Потенциальный экономический эффект	68
3.8.3	Перспективы масштабирования и внедрения	69
	ЗАКЛЮЧЕНИЕ	70
	Список использованной литературы	73
	Приложение А. Техническое задание	76
	Приложение Б. Листинг кода	78
	Приложение В. Акт о внедрении результатов дипломного проектирования	89

ВВЕДЕНИЕ

В настоящее время мобильные устройства на базе Android широко применяются во многих сферах жизни. Однако пользователи с ограниченными возможностями здоровья (повреждения опорно-двигательного аппарата, последствия инсультов, травм позвоночника и многих других редких заболеваний и состояний) часто испытывают трудности при взаимодействии с сенсорными экранами. Существующие решения (TalkBack, Voice Access, Switch Access и др.) не всегда удовлетворяют потребности, особенно когда необходима тонкая настройка, комбинированные способы ввода или возможность попробовать разные варианты управления для людей с ОВЗ(ограниченными возможностями здоровья).

Цель данного дипломного проекта — разработка мобильного инструмента для альтернативного взаимодействия с Android-смартфонами и планшетами. Приложение AIS-A предлагает четыре режима управления, ориентированные как на пользователей с двигательными нарушениями, так и на здоровых людей, чьи обстоятельства временно не позволяют работать с обычным сенсорным вводом.

Основные задачи работы: Исследовать существующие ассистивные решения для Android и выявить их ограничения. Разработать структуру и техническое задание приложения AIS-A, включающего четыре режима взаимодействия. Реализовать механизмы универсального виртуального курсора и активных зон, а также конфигурацию «тапа» (звуковой сигнал, задержка на месте, открывание рта и т.д.). Провести тестирование на группе пользователей, включая людей с ограниченными возможностями, и оценить результаты с точки зрения удобства, точности, производительности. Объект исследования – процесс альтернативного управления Android-устройством средствами ассистивных технологий. Предмет исследования – программные сервисы (AccessibilityService) и алгоритмы обработки сенсорных данных (камера, аудио, акселерометр) для реализации виртуального курсора.

Актуальность темы объясняется всё более широким использованием смартфонов, необходимостью их доступности и универсальностью для всех категорий пользователей.

1 Анализ предметной области и постановка задачи

1.1 Анализ предметной области и обоснование актуальности

Современные мобильные устройства стали неотъемлемой частью повседневной жизни, обеспечивая доступ к коммуникации, информации, социальным сетям и цифровым сервисам. Однако для миллионов людей с ограниченными возможностями здоровья, особенно с нарушениями опорно-двигательного аппарата (ОДА), использование таких устройств остаётся серьёзной проблемой. При этом взаимодействие с цифровой средой — не роскошь, а необходимое условие включённости в общественную и личную жизнь.

Согласно данным Всемирной организации здравоохранения (ВОЗ), более 1 миллиарда человек в мире живут с различными формами инвалидности. Из них около 190 миллионов сталкиваются с тяжёлыми двигательными нарушениями [1]. В Казахстане, по данным Департамента инклюзивности, проживает свыше 725 000 человек с ОВЗ, значительная часть которых нуждается в цифровых средствах абилитации и реабилитации.

Среди этой категории есть люди с крайне ограниченными возможностями — вплоть до состояния, приближенного к синдрому «запертого человека», когда сохранено сознание, но почти полностью утрачена способность управлять телом [2][4]. Даже при наличии минимальных движений головы, глаз или пальцев они часто бывают хаотичными и неуправляемыми, а повторные попытки не дают ожидаемого результата. Это может приводить к потере мотивации и отказу от дальнейшего взаимодействия с внешним миром.

Однако именно такие попытки имеют решающее значение — они активируют остаточные нейронные связи, способствуют нейропластичности и формируют новые пути между мозгом и телом [3]. Большинство методов реабилитации предполагает активное участие пользователя, и его вовлечённость значительно повышает эффективность восстановительных процессов.

В этой связи всё большую роль играют вспомогательные технологии: айтрекеры, адаптивные интерфейсы, сенсорные переключатели (свитчи), голосовое управление и др. Они становятся не просто средствами управления, но и инструментами включённости, тренировочными платформами и каналами мотивации. Такие решения:

- возвращают человеку ощущение контроля;
- формируют среду для восстановления утраченных навыков;
- способствуют социальной и когнитивной реабилитации;
- могут применяться в домашних условиях без необходимости медицинского сопровождения.

Тем не менее, существующие решения на рынке нередко имеют узкую специализацию, требуют дорогостоящего оборудования или не учитывают разнообразие физических состояний пользователей. Многие системы сложны в настройке, не позволяют гибко переключаться между способами управления, а использование, например, только голосового ввода оказывается невозможным для людей с нарушениями речи или дыхания.

Идея и цели проекта AIS-A

Именно на фоне описанных проблем была сформулирована идея создания универсального мобильного приложения AIS-A, способного предоставить пользователю набор hands-free режимов, работающих на базе стандартных модулей смартфона — без необходимости подключения дополнительного оборудования.

Приложение AIS-A поддерживает четыре независимых режима управления:

T-PAD — управление с помощью прикосновений к экрану, используется ограниченная зона экрана, функционирующая по принципу тачпада;

HEAD — управление движением головы (через фронтальную камеру);

EYES — управление направлением взгляда и мимикой (через ML Kit);

G-sens — управление с помощью наклона устройства (акселерометр).

Такая конфигурация обеспечивает широкий охват остаточных двигательных возможностей пользователей. Благодаря единой концепции виртуального курсора пользователь может динамически переключаться между режимами, ориентируясь на текущее самочувствие, утомляемость или прогресс в реабилитации. Например, если в течение дня устают руки — можно перейти на управление глазами или головой, а при нарушении фокусировки взгляда — использовать наклон устройства.

Актуальность проекта

AIS-A разрабатывается с учётом: реальных ограничений пользователей; необходимости минимального порога вхождения (установка без специальной подготовки); работы в домашних условиях, где нет профессионального ассистента или специалиста; открытого доступа и масштабируемости.

Кроме того, приложение фактически выполняет функции доступной БОС-платформы (биологической обратной связи). Каждый жест, движение или взгляд пользователя немедленно визуализируется на экране в виде движения курсора или срабатывания действия, что даёт моментальную обратную связь, стимулирует активность и укрепляет связь между намерением и результатом.

Таким образом, разработка AIS-A отвечает не только технической, но и социальной, психологической и реабилитационной задачам, создавая условия для цифровой инклюзии и вовлечённости людей с ограниченными возможностями в полноценную цифровую среду.

1.2 Анализ существующих решений и постановка задачи

Современные системы hands-free управления устройствами разрабатываются как для общего использования, так и специально для людей с ограниченными возможностями. Однако при ближайшем рассмотрении оказывается, что большинство доступных решений обладают значительными ограничениями по универсальности, адаптивности, стоимости или применимости в бытовых условиях.

Использование айтрекеров, адаптивных интерфейсов и вспомогательных технологий активно исследуется в рамках ACM ASSETS и HCI-исследований [5]. Наиболее известные и используемые решения:

Voice Access (Google) Встроенная в Android система голосового управления, позволяющая выполнять команды и навигацию без прикосновений.

Недостатки: требует отчётливой дикции, тишины в помещении, не работает при нарушениях речи, усталости или шуме.

Switch Access Поддержка физических и виртуальных переключателей. Пользователь может подключить кнопки, сенсоры или другие внешние устройства и использовать их для навигации. Недостатки: требует дополнительного оборудования, часто неудобна в мобильных условиях, низкая скорость взаимодействия.

Tobii PCEye, Irisbond, EyeTech Профессиональные трекары взгляда, используемые в медицинской и образовательной среде. Недостатки: высокая стоимость, ограниченная доступность, отсутствие поддержки Android, необходимость использования ПК или ноутбуков.

OpenTrack, HeadMouse, eViacam Программные head-tracking-решения, распознающие движения головы по камере. Недостатки: чаще всего рассчитаны на ПК, требуют внешней камеры, чувствительны к освещённости и положению головы, редко поддерживают мобильные платформы.

Gyro-based Remote Mouse / Phone Mouse Программы, использующие акселерометр телефона как мышь или джойстик для ПК. Недостатки: не ориентированы на людей с ОВЗ, не имеют адаптивных интерфейсов, нет поддержки триггеров, автотапа и активных зон.

Hands-free mouse (Tecla, AMAneo) Коммерческие решения, работающие на базе sip-and-puff, сенсоров EMG или Bluetooth-устройств. Недостатки: требуют дорогостоящих аппаратных модулей, часто несовместимы с Android, предназначены для стационарного применения.

Браузерные и экспериментальные интерфейсы на базе WebGazer, TensorFlow.js

Используются для слежения за взглядом через веб-камеру в браузере. Недостатки: работают только в рамках браузера, не дают системного доступа, не интегрированы с интерфейсом Android.

Ключевые проблемы существующих решений: Зависимость от дополнительного оборудования (треки, переключатели, камеры); Отсутствие универсальности — каждое решение обслуживает один способ управления; Недостаточная адаптация к нестабильным или ослабленным движениям; Отсутствие поддержки режимов работы без голоса (Voice Access недоступен при дизартрии, усталости); Сложность настройки, отсутствие интеграции с Android Accessibility API; Отсутствие поддержки мгновенного переключения между режимами, что критично при переменной симптоматике (например, усталости руки или головы).

Особенности и отличия AIS-A. В ответ на перечисленные ограничения была поставлена задача разработки универсального, недорогого и доступного решения, способного обеспечить гибкое hands-free управление Android-устройством, без необходимости приобретения специализированных устройств.

AIS-A реализует следующие ключевые преимущества:

Четыре режима управления в одном приложении: T-PAD — управление экраном как тачпадом; HEAD — управление головой (через фронтальную камеру); EYES — управление глазами и мимикой (через ML Kit); G-sens — управление акселерометром, наклонами устройства.

Единый виртуальный курсор, активные зоны на экране (BACK, HOME, SCROLL), индикаторы и всплывающие элементы для наглядного взаимодействия.

Гибкая система триггеров для клика: автотап при неподвижности, двойной тап, открытие рта (face landmark), звук (хлопок, щелчок — через FFT анализ).

Мгновенное переключение между режимами — пользователь может изменять способ управления без перезапуска приложения.

Полная интеграция с Android Accessibility API — приложение не требует root-доступа, работает с любыми смартфонами с Android 8+.

Настройка всех параметров (чувствительность, скорость, порог срабатывания) через интерфейс пользователя, без подключения к ПК.

Поддержка работы в домашних условиях — приложение рассчитано на самостоятельную установку и использование без медперсонала.

Таким образом, основная цель проекта — создание универсального Android-приложения, предоставляющего несколько способов hands-free управления в рамках единой, доступной и гибко настраиваемой системы, пригодной для реабилитации, абилитации и повседневного использования людьми с ограничениями подвижности.

AIS-A не просто объединяет лучшие принципы существующих решений, но и расширяет их применимость, делая акцент на: адаптивность к реальному состоянию пользователя; минимальные требования к оборудованию; инклюзивный дизайн и терапевтический эффект за счёт биологической обратной связи (БОС).

Проект направлен на повышение цифровой независимости и улучшение качества жизни людей с тяжёлыми двигательными нарушениями.

1.2.1 Функциональные требования

Поддержка четырёх режимов управления (T-PAD, HEAD, EYES, G-sens); Унифицированный интерфейс с виртуальным курсором и активными зонами; Настройка триггеров нажатия (удержание, открытие рта, звук, двойной тап); Возможность переключения режимов и регулировки чувствительности; Совместимость с API Android AccessibilityService [5]; Сохранение настроек для каждого режима (размер и скорость курсора, отображение зон, и т.д.).

1.2.2 Нефункциональные требования:

Оптимизированное потребление ресурсов (CPU, RAM, батарея); Простой и интуитивный интерфейс настроек, пригодный для использования людьми с ОВЗ; Совместимость с устройствами под управлением Android 8.0 и выше; Отказоустойчивость, отсутствие зависаний или сбоев при переключении режимов; Возможность использования разных сценариях.

1.3 Обоснование выбора технологий и инструментов разработки

В рамках подготовки к реализации проекта был проведён анализ наиболее популярных языков программирования и сред разработки, применяемых в мобильной среде (смотрите таб. - 1). По результатам анализа были выбраны инструменты, наиболее соответствующие требованиям проекта с точки зрения функциональности, надёжности и поддержки Android-платформы.

В рамках реализации проекта AIS-A — мобильного приложения для альтернативного hands-free управления Android-устройством — решено было использовать язык программирования Kotlin и интегрированную среду разработки Android Studio. Выбор этих технологий обусловлен как техническими требованиями проекта, так и рекомендациями официальной документации Google.

1.3.1 Язык программирования Kotlin

Kotlin является официальным языком Android-разработки [12], одобренным и активно продвигаемым компанией Google. Он обладает полной совместимостью с Java, при этом предоставляет более современный и выразительный синтаксис, что позволяет создавать надёжный, компактный и легко сопровождаемый код.

Ключевые преимущества Kotlin [12] в контексте данного проекта:

Безопасность при работе с null-значениями (Null Safety), что критически важно при взаимодействии с системными API (камерой, сенсорами, аудиовходами); Поддержка функционального программирования, включая лямбда-выражения и расширяемые функции, что упрощает реализацию логики обработки жестов, сенсорных данных и событий; Сокращение шаблонного кода (boilerplate), особенно при разработке многочисленных пользовательских настроек и экранов; Простота внедрения в проекты с использованием Java-библиотек, включая TarsosDSP, ML Kit и другие компоненты, применённые в AIS-A.

Эти особенности позволяют сфокусироваться на бизнес-логике приложения, минимизируя вероятность ошибок, связанных с управлением состоянием и пользовательским вводом, особенно в условиях нестандартного взаимодействия (голос, взгляд, наклон устройства).

1.3.2 Среда разработки Android Studio

Android Studio является официальной и наиболее функциональной IDE для создания Android-приложений, разрабатываемой и поддерживаемой Google. Она предоставляет:

Встроенные средства визуального проектирования UI, включая редактор макетов, превью-компоненты и поддержку Material Design;

Инструменты тестирования и профилирования — от логгирования и мониторинга производительности до анализа утечек памяти и откликов пользовательского интерфейса;

Полноценную интеграцию с Gradle, позволяющую управлять зависимостями, сборкой, подписыванием и публикацией приложения;

Интеграцию с Android SDK, AVD Manager и эмуляторами, что критично для тестирования режимов, зависящих от камеры, микрофона, акселерометра и служб специальных возможностей (Accessibility Services) [6].

Особое значение имеет поддержка библиотек Computer Vision (ML Kit), обработки звука (TarsosDSP), а также полноценного доступа к SensorManager,

CameraX и Gesture API, на которых основаны различные режимы управления в AIS-A.

Выбор связки Kotlin + Android Studio обеспечил: эффективную реализацию модульной архитектуры приложения; высокую скорость разработки и отладки; адаптацию под требования Android Accessibility API [5]; возможность масштабирования и расширения функциональности в будущем.

Таким образом, выбранный стек технологий оказался наиболее подходящим для задач проекта, связанных с разработкой надёжного, кросс-модульного и адаптивного интерфейса альтернативного управления на платформе Android.

Таблица 1.1– Сравнение подходящих языков программирования

Критерий	Kotlin (Android Studio)	Java (Android Studio)	Flutter (Dart)	React Native (JavaScrip
Официальная поддержка Google	Да	Да	Да	Нет (только сторонние сообщества)
Совместимость с Android SDK	Да, полная	Да, полная	Частичная (через платформенные каналы)	Частичная (требуется нативных модулей)
Поддержка AccessibilityService	Да, полная	Да, полная	Ограниченная	Ограниченная
Доступ к сенсорам (камера, гироскоп и др.)	Да, полный доступ	Да, пол. доступ	Возможен, но требует нативной интеграции	Нативной интеграции Зависит от сторонних библиотек
Язык разработки	Совр-й, безопас.	Устарев. Многослов.	Кроссплатформенный, менее распространён	Гибкий, но не нативный
Краткость и читаемость кода	Высокая	Низкая	Средняя	Средняя
Производительность	Высокая (нативная)	Высокая (нативная)	Ниже нативной	Ниже нативной
Поддержка реального времени (сенсоры, видео)	Отличная	Отличная	Огр.	Огр.

Ключевая технология — AccessibilityService, предоставляющая доступ к событиям взаимодействия с системой и возможность рисовать собственный интерфейс поверх всех окон (overlay) [6]. Это позволяет реализовать курсор и активные зоны без необходимости в системных правах.

Для отслеживания движений головы и глаз используется: CameraX [8] для доступа к фронтальной камере; ML Kit Face Detection — библиотека от Google для распознавания лица, глаз, рта (с высокой скоростью и стабильностью на Android) ее преимущества показаны в таблице 1.2; Для звуковых триггеров и FFT-анализа применяется TarsosDSP [11] — лёгкая аудиобиблиотека с широким набором функций обработки аудиосигнала; SensorManager используется для получения данных с акселерометра и гироскопа [7].

Система логирования позволяет отслеживать параметры производительности и частоту срабатываний различных жестов, что может быть полезно как для разработчика, так и для анализа эффективности процессов абилитации или реабилитации.

Таблица 1.2- Сравнение ML Kit с другими CV-библиотеками

Биб-ка		Описание	Вывод
MediaPipe (Google)	+	Высокая точность распознавания лица, глаз, положения головы и жестов. Предоставляет продвинутое модели (Face Mesh, Iris Tracking, Hand)	MediaPipe отклонён из-за избыточности, сложности сборки и несоответствия требованиям к размеру APK.
	-	Требуется сборки с использованием NDK и CMake, усложняет процесс интеграции. Существенно увеличивает размер APK (от 20 до 40 МБ на компонент). Плохо совместим с CameraX и архитектурой Android Jetpack. Сложно адаптировать поведение моделей под кастомные сценарии.	

OpenCV	+	Богатый инструментарий для обработки изображений и видео. Возможность полной локальной обработки данных.	Используется для низкоуровневых операций и экспериментов, но менее удобен в реализации задач hands-free управления.
	-	Требуется подключения и настройки нативных библиотек под все архитектуры. Не предоставляет встроенных решений для трекинга глаз и головы. Вводит дополнительные сложности при интеграции с Kotlin/AndroidX.	
DDlib	+	Open-source и кроссплатформенный. Поддерживает распознавание лиц и ключевых точек.	Подходит для десктопов и серверов, но не рекомендован для Android-устройств с ограниченными ресурсами.
	-	Требуется обёрток и C++-интеграции. Высокие требования к ресурсам.	
TensorFlow Lite + Custom Models	+	Гибкость: возможность обучать собственные модели. Поддержка оптимизации под мобильные устройства.	Перспективен в будущем, но требует больше ресурсов и времени на разработку.
	-	Требуется наличия собственных датасетов и экспертизы в ML. Более высокая нагрузка на CPU/GPU при сложных задачах.	

ML Kit	+	Простая интеграция с Android Studio и CameraX. Поддержка готовых решений для face detection и head pose estimation. Работа offline, оптимизация под ARM и низкое энергопотребление. Поддержка AndroidX и встроенные средства отладки.	Несмотря на некоторые ограничения, ML Kit обеспечивает оптимальное соотношение точности, производительности и простоты интеграции, соответствуя целям и ограничениям проекта AIS-A.
	-	Ограниченный контроль над моделью: невозможность внести глубокие изменения в алгоритмы распознавания. Невысокая точность отслеживания взгляда по сравнению с кастомными моделями или MediaPipe Face Mesh. Зависимость от Google Play Services: некоторые функции могут не работать на устройствах без Google-сервисов (например, китайские смартфоны). Отсутствие нативной поддержки трекинга глаз с высокой точностью (например, pupil tracking).	

Вывод по таблице 1.2: Несмотря на более высокую точность некоторых решений (MediaPipe, TensorFlow Lite), выбор пал на ML Kit благодаря его простоте, лёгкости интеграции и стабильной поддержке от Google. Это делает ML Kit наиболее подходящим вариантом для проектов, ориентированных на устойчивую работу на Android-устройствах без необходимости глубокого машинного обучения.

2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ (ПРИЛОЖЕНИЯ)

2.1 Архитектура приложения и организационная структура проекта

Разработка проекта AIS-A (Alternative Interaction System for Android) осуществлялась с учётом принципов модульности и структурной раздельности, что способствует удобству сопровождения и расширения функциональности системы. Особое внимание уделено логическому разграничению пользовательского интерфейса, фоновых сервисов, параметров конфигурации и ресурсных компонентов. Ниже на рисунке 2.1 представлена структура проекта:

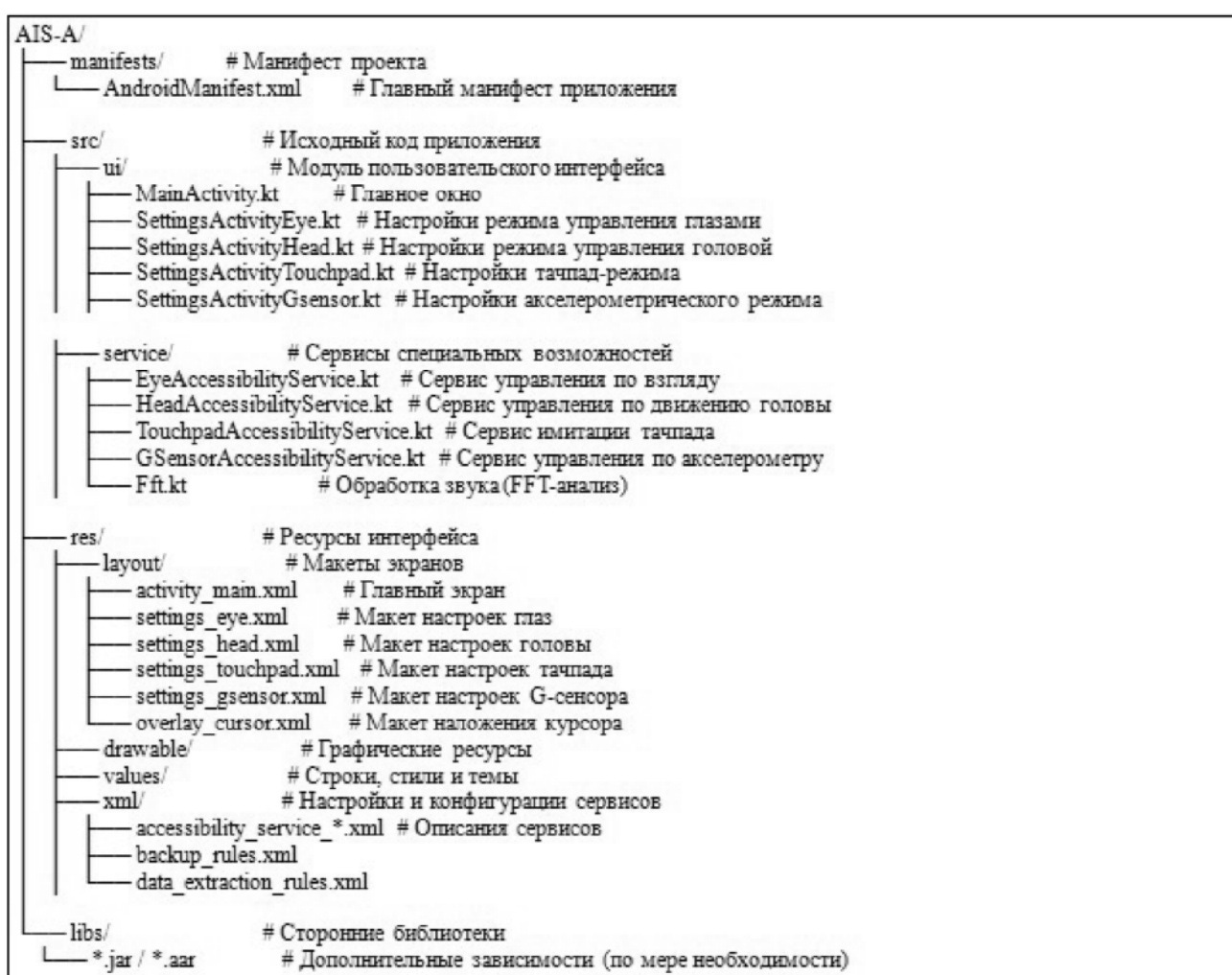


Рисунок 2.1 – Структура проекта AIS-A

Ui/ Содержит основные Activity:

- MainActivity.kt — главный экран выбора режима управления.

- SettingsActivityEye.kt, SettingsActivityHead.kt, SettingsActivityTouchpad.kt, SettingsActivityGsensor.kt — экраны настройки параметров соответствующего режима. Позволяют настраивать скорость курсора, размер, чувствительность, видимость зон и способы «тапа».

service/ Содержит реализации AccessibilityService:

- EyeAccessibilityService.kt — управление с помощью глаз, определение взгляда и событий «тапа» (удержание, открытие рта, звук).

- HeadAccessibilityService.kt — отслеживание головы и реализация курсора, опирающаяся на поворот головы.

- TouchpadAccessibilityService.kt — обработка касаний по экрану как по тачпаду.

- GSensorAccessibilityService.kt — управление с помощью акселерометра.

- Fft.kt — реализация звукового триггера «тапа» с использованием быстрого преобразования Фурье (FFT).

layout/ XML-макеты для UI:

- overlay_cursor.xml — разметка курсора и активных зон.

- settings_*.xml — формы настроек для каждого режима управления.

- activity_main.xml — главный экран запуска.

xml/

Файлы конфигурации сервисов и бэкапов:

- accessibility_service_*.xml — настройки для регистрации AccessibilityService в манифесте.

- backup_rules.xml, data_extraction_rules.xml — правила бэкапа и восстановления настроек.

2.2 Проектирование сервисов для каждого режима (T-PAD, HEAD, EYES, G-sens)

В данном разделе описывается архитектура и логика реализации сервисов, обеспечивающих альтернативное hands-free управление в приложении AIS-A. Каждый режим (T-PAD, HEAD, EYES, G-sens) реализован в виде отдельного сервиса, расширяющего функциональность Android AccessibilityService и использующего дополнительные модули, такие как CameraX, ML Kit, SensorManager и др. Полные листинги кода приведены в Приложении Б (см. листинг Приложение Б).

2.2.1 TouchpadAccessibilityService (T-PAD)

Режим T-PAD реализуется с помощью сервиса TouchpadAccessibilityService, в котором экран устройства выступает в роли тачпада. Он предназначен для пользователей, не имеющих возможности управлять устройством с помощью стандартных средств ввода (мыши, клавиатуры).

Основные функции реализуемого сервиса: управление виртуальным курсором по координатам касания; распознавание одиночного и двойного тапа (см. листинг 1.1); режим "залипания" (Sticky Selection); масштабирование с помощью жеста pinch-to-zoom; работа с триггер-зонами (Back, Home, Scroll); динамическая настройка параметров через BroadcastReceiver (см. листинг 1.2); обработка физических клавиш и ориентации экрана. Структура класса TouchpadAccessibilityService см. рисунок 2.2.



Рисунок 2.2 – UML-диаграмма классов TouchpadAccessibilityService

2.2.2 HeadAccessibilityService (HEAD)

Управление движением головы. Режим HEAD реализуется с использованием фронтальной камеры устройства и библиотеки ML Kit Face Detection, что позволяет определять координаты центра лица и направление взгляда пользователя. Это обеспечивает возможность hands-free управления курсором за счёт поворота головы. Ключевые особенности реализации: захват и калибровка координат центра лица; расчёт смещения и масштабирования координат движения (см. листинг 1.3); экспоненциальное сглаживание движения для повышения стабильности курсора; автотап по неподвижности (автоматическое подтверждение при остановке); клик по звуку путём сравнения спектров аудиосигнала (см. листинг 1.4); выполнение жестов через вызовы dispatchGesture() для имитации касаний. Структура класса HeadAccessibilityService см. рисунок 2.3

```

HeadAccessibilityService
├─ trackHeadMovement()
├─ calibrateCenter()
├─ performClickAtPosition()
├─ startIdleCheck()
└─ startSoundListener()

```

Рисунок 2.3 – UML-диаграмма классов HeadAccessibilityService

2.2.3 EyeAccessibilityService (EYES)

Управление направлением головы и ориентировочным положением глаз и мимикой

Режим EYES обеспечивает полностью бесконтактное управление устройством на основе анализа движений глаз и рта пользователя. Реализация основана на технологиях CameraX и ML Kit Face Detection, позволяющих в реальном времени обрабатывать видеопоток с фронтальной камеры. Методы управления, реализованные в данном сервисе: перемещение виртуального курсора в зависимости от координат глаз; клик по экрану при открытии рта (распознавание мимики); автотап при фиксации взгляда (неподвижности); поддержка динамической калибровки параметров отслеживания (см. листинг 1.5). Структура класса EyeAccessibilityService см. рисунок 2.4:

```

EyeAccessibilityService
├─ onFrameCaptured()
├─ calculateGazeCoordinates()
├─ applySmoothing()
├─ checkMouthOpen()
└─ performClickGesture()

```

Рисунок 2.4 – UML-диаграмма классов EyeAccessibilityService

2.2.4 GSensorAccessibilityService (G-sens)

Управление через акселерометр - Режим G-sens предоставляет возможность управления курсором с помощью наклонов устройства, используя данные встроенного акселерометра. Это решение ориентировано на пользователей с ограниченными возможностями и обеспечивает полностью hands-free взаимодействие. Основные возможности сервиса: обработка данных сенсора в методе `handleAccelerometer()`; прилипание курсора к кликабельным элементам интерфейса; автотап при неподвижности курсора (см. листинг 1.6); распознавание звука и клик по звуковому триггеру; поддержка двойного тапа как жеста подтверждения; активация действий через триггер-зоны; динамическая настройка параметров с использованием `BroadcastReceiver`. Структура класса `GSensorAccessibilityService` см. рисунок 2.5:

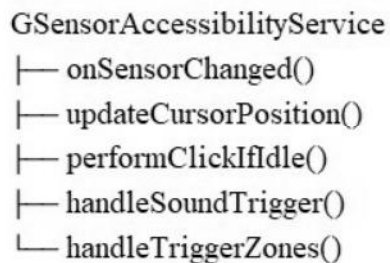


Рисунок 2.5 – UML-диаграмма классов `GSensorAccessibilityService`

Примечания: Все сервисы используют общую архитектуру виртуального курсора, реализованную через `Overlay Window` (см. листинг 1.1). Настройки конфигурации считываются и обновляются с помощью `SharedPreferences` (см. раздел 2.3 и листинг 1.7). Обработка внешних команд осуществляется через `BroadcastReceiver`, обеспечивая адаптацию поведения приложения в рантайме.

2.3 Описание программных модулей управления

Раздел посвящён детальному рассмотрению архитектуры и принципов работы программных компонентов, обеспечивающих альтернативные способы управления. Представлены особенности реализации, алгоритмы функционирования и ключевые технические решения для каждого модуля.

2.3.1 Сервис Head (HeadAccessibilityService).

Архитектура модуля HEAD построена на основе Android Accessibility API, с интеграцией следующих компонентов: CameraX + ML Kit Face Detection — для анализа положения головы, детекции лица и оценки вероятности открытия рта; Overlay-интерфейс — визуальный слой с отображением виртуального курсора, индикаторов активности и визуальных элементов управления; Система калибровки — автоматическая или ручная инициализация нейтрального положения головы, от которого рассчитываются смещения; Сглаживание координат — применяется экспоненциальное сглаживание и псевдо-Калманов фильтр для устранения мелких дрожаний;

Триггеры выполнения кликов: Задержка на месте — фиксированное время без движения вызывает зону предтапа и если пользователь выводит курсор и возвращает обратно в нее совершается тап подробнее об этом будет рассказано ниже; Открытие рта — естественный жест, позволяющий имитировать касание; Звук — активируется при совпадении текущего спектра с эталонным звуком; Active Zone — специальная область под курсором, в которой допускается выполнение действий; Trigger Zones — вспомогательные зоны у краёв экрана для имитации жестов (Back, Home, свайпы и др.).

Алгоритм работы проходит по следующим этапам: После запуска сервиса пользователь проходит процедуру калибровки — определяется центральное положение головы, которое считается нулевой точкой отсчёта. При дальнейшем использовании каждое новое положение головы сравнивается с калиброванной позицией, и курсор перемещается в противоположном направлении. При отсутствии движения курсора в течение заданного промежутка времени может сработать автотап. Если активирован метод «открытие рта» — срабатывание клика происходит при превышении заданного порога вероятности. При активации звукового метода — входящий аудиосигнал обрабатывается через FFT-анализ и сравнивается с предварительно записанным эталонным звуком. Дополнительно реализованы триггер-зоны по краям экрана, позволяющие имитировать системные жесты.

Для повышения точности реализована динамическая фильтрация движения, зависящая от скорости смещения головы: при резких движениях фильтрация ослабляется, при плавных — усиливается. Это обеспечивает баланс между отзывчивостью и стабильностью управления. Реализована система залипания (snapping) к кликабельным элементам интерфейса, которая позволяет "прилипнуть" к элементу и точно выбрать его.

Защита от ложного тапа реализуется через специальную зону предтапа — виртуальную область вокруг интерактивного элемента. Тап срабатывает только в том случае, если курсор сначала вошёл в зону, затем вышел из неё и повторно вернулся. Это позволяет исключить случайные срабатывания при

непроизвольных колебаниях головы, значительно повышая точность взаимодействия и предотвращая ошибочные нажатия см. рисунок 2.6.

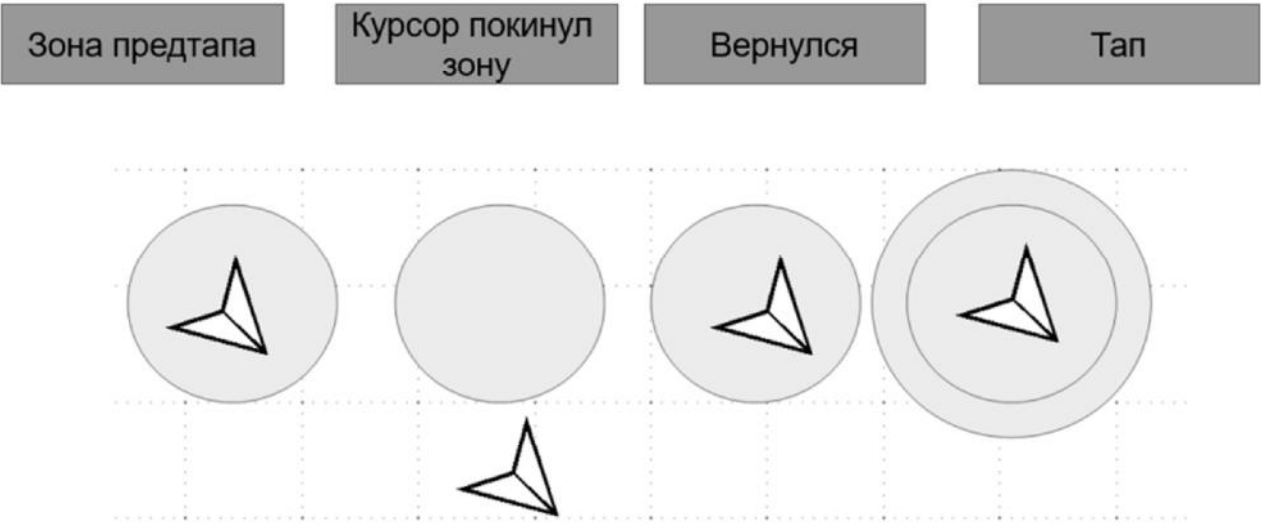


Рисунок 2.6 – Защита от ложного срабатывания тапа при режиме тапа по ожиданию

Все параметры управления — чувствительность, порог срабатывания, размер курсора, длительность задержки, включение/отключение конкретных методов клика — доступны для гибкой настройки и сохраняются с использованием SharedPreferences, что обеспечивает сохранение пользовательских предпочтений между сессиями.

Таблица 2.1- Технические детали HeadAccessibilityService

Компонент	Технология
Распознавание лица	ML Kit Face Detection
Обработка аудио	TarsosDSP + FFT
Работа с камерой	CameraX (фронтальная камера)
Интерфейс	WindowManager + Overlay Views
Управление курсором	AccessibilityService + Gesture API
Сохранение настроек	SharedPreferences

Полевые испытания сервиса проводились на группе пользователей с временными ограничениями в движениях (например, с зафиксированной рукой). Управление курсором было признано интуитивно понятным, с высокой стабильностью отклика. Особенно эффективным показал себя метод «задержки»

— благодаря простоте и надёжности. Метод «клик по звуку» показал хорошие результаты при использовании в условиях низкого уровня шума и чётко выраженного звукового шаблона. При этом ложные срабатывания успешно отсеивались за счёт зоны предтапа и спектрального анализа сигнала.

2.3.2 Сервис EYES (EyesAccessibilityService)

Управление курсором по направлению взгляда и мимике

Сервис EyesAccessibilityService является частью программного комплекса AIS-A (Alternative Interaction System for Android) и представляет собой специализированный модуль hands-free управления курсором Android-устройства с помощью направления взгляда и мимических жестов. Основной целью разработки является обеспечение доступа к мобильным технологиям для пользователей с тяжёлыми нарушениями двигательной активности, исключая возможность управления через стандартные сенсорные интерфейсы.

Архитектура и функции модуля построены на базе AccessibilityService и интегрирует библиотеки ML Kit и CameraX, обеспечивая надёжную обработку лицевых признаков и видеопотока с фронтальной камеры. Обработка координат взгляда, реакция на открытие рта, распознавание звука и визуальный отклик — все эти компоненты объединены в единую систему альтернативного управления.

Ключевые функциональные блоки EyesAccessibilityService: Инициализация жизненного цикла; Модуль реализует интерфейс LifecycleOwner, что позволяет безопасно управлять жизненным циклом камеры, обрабатывая активации и деактивации сервиса без утечек памяти. Инициализация графического интерфейса (Overlay View) работает следующим образом: Поверх всех окон системы накладывается специальный overlay-интерфейс, содержащий: окно предпросмотра камеры (PreviewView); курсор (ImageView); активные зоны; прогресс-индикаторы калибровки, автотапа и триггеров.

Калибровка направления взгляда, проходит следующим образом: При первом запуске пользователь проходит пошаговую калибровку: центр, вверх, вниз, влево и вправо. При использовании метода клика по рту также добавляются состояния `mouth_closed` и `mouth_open`. Полученные значения сохраняются и используются для нормализации координат взгляда.

Обнаружение лица и отслеживание глаз работает на основе ML Kit FaceDetector для извлечения координат глаз, после чего они нормализуются и трансформируются в экранные координаты.

Для компенсации колебаний и микродвижений головы применяется экспоненциальное сглаживание и адаптивная фильтрация на основе псевдо-Калмановского фильтра.

Автоматический тап по неподвижности. При условии, что скорость движения взгляда минимальна и курсор остаётся в пределах одной области более 2 секунд, выполняется автоматический вызов зоны предтапа. Пользователю отображается визуальный индикатор, для того что бы совершить тап пользователю нужно вывести курсор из зоны предтапа и вернуть его обратно, подобное решение максимально минимизирует ложные срабатывания.

Модуль поддерживает три метода выбора: 0 — автотап по неподвижности; 1 — распознавание открытого рта (на основе вероятности открытия); 2 — аудиосигнал, совпадающий с заранее записанным эталонным звуком (обрабатывается с использованием FFT и библиотеки TarsosDSP).

Механизм действия прилипания к элементам интерфейса (snapping) работает по следующему принципу: При приближении курсора к кликабельным элементам включается механизм залипания: визуально подсвечивается объект и запускается таймер, по истечении которого вызывается зона предтапа.

В краях экрана заданы активные области: при попадании в одну из них и удержании курсора в течение одной секунды выполняется запрограммированное действие — возврат (Back), переход на главный экран (Home), вертикальная прокрутка и пр.

Сервис отслеживает активные окна и обновляет список кликабельных элементов. Это необходимо для корректной работы механизма прилипания и исключения взаимодействия с неактивными объектами. Внедрён механизм динамического сглаживания — коэффициенты сглаживания адаптируются в зависимости от скорости движения взгляда, что позволяет повысить как стабильность, так и отзывчивость интерфейса.

Поддерживаются кастомные изображения курсора, а также возможность настройки его размеров, скорости и чувствительности.

Визуальная обратная связь сопровождает все ключевые действия: отображение калибровочной точки, прогресс-индикаторы автотапа, вспышка при распознавании звука и др. Поддерживается режим предварительной записи звука, который затем используется как эталон для точного распознавания голосового клика. Основные технические моменты представлены в Таблице 2.2.

Таблица 2.2- Технические детали EyesAccessibilityService

Компонент	Технология
AccessibilityService	Базовый компонент управления курсором
CameraX3	захват и обработка видеопотока с камеры
ML Kit	Распознавание лица, глаз и рта
AudioRecord + FFT	Анализ аудиосигнала и сравнение с эталоном
WindowManager	Управление Overlay-интерфейсом
GestureDescription	Эмуляция касаний и жестов
SharedPreferences	Хранение и восстановление пользовательских настроек

2.3.3 Сервис T-PAD (TouchpadAccessibilityService)

Сервис TouchpadAccessibilityService представляет собой модуль альтернативного hands-free управления интерфейсом Android-устройства, реализованный через имитацию трекпада. Он позволяет пользователю управлять виртуальным курсором с помощью ограниченного касания — одного пальца — по выделенной области экрана. Модуль входит в состав программного комплекса AIS-A (Alternative Interaction System for Android) и ориентирован на пользователей с частично сохранённой моторикой.

Модуль T-PAD предназначен для людей, имеющих возможность использовать хотя бы часть сенсорного экрана, но испытывающих затруднения при работе с полной клавиатурой, мышью или стандартными жестами. Это решение особенно полезно в условиях временной или частичной утраты подвижности: при переломах, после операций, при ДЦП и других формах ограниченной моторики. Интерфейс позволяет пользователю выполнять все основные действия в системе Android, используя только один палец в ограниченном безопасном пространстве экрана.

Архитектура сервиса построена на основе AccessibilityService и включает в себя несколько ключевых компонентов: Обработка касаний и жестов

Реализована через интерфейс View.OnTouchListener, позволяющий отслеживать координаты касания, распознавать одиночный и двойной тап, различать длительные нажатия, а также обрабатывать жесты масштабирования.

Поверх всех окон отображается полупрозрачная зона — виртуальный тачпад, в пределах которого пользователь может управлять курсором. Дополнительно отображаются визуальные элементы: сам курсор, границы активной зоны, индикаторы режимов (залипание, масштабирование, прогресс-бар).

С помощью GestureDescription API эмулируются действия системы: одиночный и двойной тап, прокрутка, свайпы и масштабирование. Это позволяет интегрироваться с существующими приложениями без модификации их кода.

По краям зоны тачпада определены специальные области, активирующие действия системы: возврат (BACK), переход на главный экран (HOME), прокрутка вверх/вниз (SCROLL_UP/SCROLL_DOWN), прокрутка влево/вправо (SCROLL_LEFT/SCROLL_RIGHT).

Поддерживается выбор изображений курсора, их масштабирование, настройка скорости перемещения и чувствительности, а также возможность загрузки собственных иконок из памяти устройства.

Пользователь активирует режим T-PAD в главном меню приложения AIS-A; На экране появляется полупрозрачная зона — виртуальный тачпад; При перемещении пальца по этой зоне курсор на экране перемещается соответственно направлениям касания; Касание фиксируется как одиночный или двойной тап в зависимости от длительности или количества касаний;

При использовании двух пальцев активируется режим блокировки курсора, сенсорный экран воспринимает касания в привычном режиме работы устройства, что бы активировать курсор нужно тапнуть по его изображению; При входе в Trigger Zone и удержании курсора выполняется соответствующее действие;

Все настройки поведения сохраняются через SharedPreferences и автоматически применяются при следующем запуске.

Особенности реализации Поддержка множества жестов: одиночный тап, двойной тап, прокрутка, масштабирование; Гибкая настройка зоны касания, размера и скорости курсора, чувствительности; Визуальные элементы оформлены в стиле Glassmorphism и Material You, что делает интерфейс современным и визуально доступным; Реализована возможность использования пользовательских иконок курсора с помощью BitmapFactory и масштабирования PNG-изображений; Учитывается ориентация экрана и автоматическая адаптация зоны управления к его текущей конфигурации; Возможность временной блокировки управления (например, двойным тапом двумя пальцами) с последующей активацией. Основные технические моменты показаны в таблице 2.3

Таблица 2.3- Технические детали TouchpadAccessibilityService

Компонент	Технология
Обработка касаний	View.OnTouchListener
Эмуляция жестов	GestureDescription
API Интерфейс	WindowManager + Overlay Views
Управление курсором	AccessibilityService + MotionEvent
Хранение настроек	SharedPreferences
Кастомизация курсора	BitmapFactory + PNG

Полевые испытания показали, что режим T-PAD обеспечивает высокую отзывчивость интерфейса и точное позиционирование курсора. Пользователи отметили удобство использования при минимальном движении пальца, особенно в случаях, когда полная моторика отсутствует, но доступен один палец или стилус. Trigger Zones успешно заменяли системные жесты, позволяя работать в полноэкранных приложениях без кнопок. Система показала стабильную. Визуальное оформление и возможность выбора курсора сделали интерфейс индивидуально настраиваемым и доступным.

2.3.5 Сервис G-SENS (GsensorAccessibilityService)

Сервис GSensorAccessibilityService предоставляет пользователю возможность hands-free управления Android-устройством с помощью встроенного акселерометра. Движение виртуального курсора осуществляется за счёт наклонов устройства в различных направлениях. Решение ориентировано на пользователей, неспособных полноценно использовать сенсорный экран, но обладающих возможностью поворачивать или слегка наклонять устройство в руках или с помощью подставок.

Модуль G-SENS предназначен для обеспечения доступности базового управления интерфейсом людям с нарушениями мелкой моторики, параличами, временной иммобилизацией или иными ограничениями верхних конечностей. За счёт анализа наклона устройства в пространстве, реализуется управление движением курсора по экрану. Взаимодействие осуществляется без касаний, посредством определения направления движения и устойчивости положения устройства.

Модуль реализован на основе сервиса специальных возможностей AccessibilityService и использует данные акселерометра, поступающие через API

SensorManager. Взаимодействие с курсором построено на основе анализа изменения ускорений по осям X и Y, с применением фильтрации и сглаживания движения.

Ключевые компоненты архитектуры: SensorListener — постоянное отслеживание данных акселерометра по осям, фиксация изменений и интерпретация наклона; Расчёт направления движения — определение ориентации и силы отклонения устройства; Управление курсором — обновление координат с учётом фильтрации и параметров чувствительности; Overlay-интерфейс — визуальное наложение элементов: курсора, активной зоны, индикаторов прогресса автотапа; Автотап по неподвижности — автоматическое выполнение клика при длительной стабилизации курсора в пределах одной зоны; Trigger Zones — зоны по краям экрана, при попадании в которые срабатывают системные действия (назад, домой, прокрутка и т.д.); Двойной тап/наклон — интерпретация двойного наклона или фиксации положения как команды; Звуковое управление — (опционально) выполнение клика по распознанному звуку (например, хлопок) с помощью анализа аудиосигнала.

Алгоритм работы происходит следующим образом: Пользователь активирует режим G-SENS в приложении; Сервис запускает обработку входящих данных акселерометра и рассчитывает вектор наклона; На основе направления и величины ускорения вычисляются новые координаты виртуального курсора; При снижении скорости и стабилизации положения курсора запускается таймер зоны предтапа. Если курсор покидает и возвращается в зону — происходит клик; При попадании курсора в триггер-зону (Trigger Zone) и удержании в ней более 1 секунды — активируется соответствующее действие (например, ВАСК или НОМЕ и др.); Если активирован режим клика по звуку, аудиосигналы обрабатываются в реальном времени, и при совпадении с эталоном выполняется вызов зоны предтапа при повторном звуке тапа.

Особенности реализации необходимо рассказать о триггер-зонах по краям экрана, которые позволяют имитировать системные жесты без дополнительных движений или голосовых команд; Все ключевые действия сопровождаются визуальной обратной связью: индикатором зоны активации, вспышкой при клике по звуку; Опциональное голосовое управление реализуется через запись и сравнение аудиоспектра в режиме реального времени, обеспечивая альтернативу для пользователей с минимальной подвижностью. Основные технические детали модуля представлены в таблице 2.4.

Таблица 2.4- Технические детали GsensorAccessibilityService

Компонент	Технология
Обработка сенсоров	SensorManager + Sensor.TYPE_ACCELEROMETER
Интерпретация наклона	Математическая фильтрация, нормализация данных
Интерфейс	WindowManager + Overlay Views
Эмуляция жестов	GestureDescription API
Сохранение настроек	SharedPreferences
Аудио (опционально)	AudioRecord + FFT (через TarsosDSP)

Тестирование сервиса проводилось на пользователях с ограниченной подвижностью пальцев. Управление курсором путём наклонов устройства показало высокую точность и устойчивость, особенно при настройке подходящего коэффициента чувствительности. Механизм автотапа работал стабильно даже при наличии слабых колебаний устройства. Особо была отмечена надёжность защиты от ложных срабатываний благодаря комбинированной системе сглаживания и триггеров. Визуальная обратная связь позволила пользователям быстро адаптироваться к управлению, а звуковое управление подтвердило свою актуальность в случаях, когда другие методы были затруднены.

2.4 Структура данных и хранение настроек

Для обеспечения гибкости поведения интерфейсных компонентов и адаптации приложения под индивидуальные потребности пользователя в системе AIS-A (Alternative Interaction System for Android) реализована система хранения и обработки пользовательских настроек. Эта подсистема играет важнейшую роль в обеспечении персонализации, доступности и удобства эксплуатации. Она отвечает за сохранение параметров управления, визуальных и поведенческих характеристик курсора, а также режимов взаимодействия, используемых в различных сервисах.

В архитектуре AIS-A используются два основных механизма для хранения данных: SharedPreferences и BroadcastReceiver, позволяющие реализовать как долговременное сохранение пользовательских параметров, так и их моментальное применение без необходимости перезапуска сервисов.

2.4.1 Хранение пользовательских параметров через SharedPreferences

Механизм SharedPreferences применяется в проекте AIS-A для локального хранения настроек пользователя, которые непосредственно влияют на поведение виртуального курсора и сервисов управления. С его помощью сохраняются ключевые параметры, такие как: размер, скорость и плавность перемещения курсора; способ выполнения клика (по неподвижности, по звуку, при открытии рта, двойной тап и др.); выбор изображения курсора (в том числе кастомные иконки); включение или отключение видимости Trigger Zones (триггерных зон на краях экрана); активация или деактивация режимов «залипания» (Sticky Mode) и двойного тапа и др.

Таким образом, SharedPreferences выполняет функцию централизованного механизма конфигурации, сохраняющего индивидуальные предпочтения пользователя и обеспечивающего единое поведение интерфейса.

Механизм SharedPreferences активно применяется во всех сервисах hands-free управления: TouchpadAccessibilityService - Хранит параметры чувствительности тачпада, включения триггер-зон, скорости и размера курсора, а также выбранного режима отображения. GSensorAccessibilityService - Сохраняет настройки чувствительности акселерометра, пороги автотапа, параметры залипания и поведение при попадании в Trigger Zones. Также поддерживает хранение звукового эталона для клика по звуку. EyeAccessibilityService - Использует SharedPreferences для хранения метода клика (автотап, рот, звук), индивидуальной калибровки, скорости движения курсора, выбора изображения курсора и параметров визуального интерфейса.

Каждому сервису соответствует свой набор ключей и параметров, позволяющий гибко адаптировать поведение приложения под разные сценарии использования.

Все пользовательские параметры сохраняются в виде XML-файла `cursor_prefs.xml`, расположенного в директории `shared_prefs/` на устройстве.

Формат хранения — пары ключ–значение, что обеспечивает компактность, высокую производительность при чтении и записи, а также простоту доступа из любого компонента приложения. При изменении настроек в пользовательском интерфейсе данные передаются сервисам в реальном времени с помощью механизма `BroadcastReceiver`, что исключает необходимость перезапуска сервисов.

Поддерживается отдельная настройка каждого режима управления (HEAD, EYES, T-PAD, G-SENS), включая хранение специфических параметров (например, эталонного звука, нейтрального положения головы, направления взгляда и др.). Все настройки автоматически подгружаются при запуске соответствующего сервиса и применяются без задержек.

Использование SharedPreferences в качестве механизма хранения настроек

предоставляет следующие ключевые преимущества: Простота реализации и масштабируемость — легко расширяется при добавлении новых параметров или режимов; Высокая производительность — обеспечивает мгновенное чтение/запись данных даже на устройствах с ограниченными ресурсами; Сохранность пользовательских данных — настройки автоматически сохраняются между сессиями и не сбрасываются при перезапуске приложения; Гибкость и адаптивность — поддержка индивидуальной конфигурации для каждого пользователя и каждого режима управления; Моментальное применение изменений — благодаря связке с BroadcastReceiver возможно обновление параметров без остановки и перезапуска сервисов.

2.4.2 Определение интерфейсов через XML-файлы

Эффективное проектирование пользовательского интерфейса играет важнейшую роль в обеспечении доступности и удобства эксплуатации приложения. В рамках системы AIS-A визуальные компоненты всех режимов управления определяются с использованием XML-разметки, что соответствует стандартам разработки Android-приложений [13] и обеспечивает гибкость, масштабируемость и простоту поддержки кода.

XML-файлы разметки описывают визуальную структуру экранов, элементы взаимодействия, подсказки, а также компоненты, необходимые для отображения состояния пользователя, выполнения калибровки и запуска сервисов управления курсором.

XML-файлы служат основой для построения адаптивных, визуально выразительных и функционально насыщенных интерфейсов. Благодаря разделению логики и представления, они позволяют: описывать структуру экранов без жёсткой привязки к программной реализации; централизованно управлять компонентами, применимыми в разных сервисах; легко адаптировать интерфейс под различные разрешения экранов и ориентации устройства; реализовывать элементы анимации, визуальных эффектов и динамической обратной связи; упрощать локализацию, замену ресурсов и кастомизацию внешнего вида.

Каждый режим управления в AIS-A использует свой XML-файл, предназначенный для отображения элементов интерфейса, характерных для конкретного способа взаимодействия: EyeAccessibilityService Макет accessibility_service_eye.xml включает: PreviewView[13] для отображения видеопотока с фронтальной камеры; точки калибровки направления взгляда; кнопки управления калибровкой; индикаторы состояния, прогресса автотапа и активности режима. GsensorAccessibilityService Макет accessibility_service_gsensor.xml содержит: кнопки активации режима;

визуальные элементы отображения чувствительности; индикаторы движения и автотапа; элементы обратной связи по отклонению устройства. TouchpadAccessibilityService Макет accessibility_service_touchpad.xml отображает: интерактивную зону тачпада; ползунки настройки скорости и чувствительности курсора; кнопки активации залипания и отображения Trigger Zones; визуальные обводки и курсор. Дополнительно используются общие макеты (activity_main.xml, overlay_cursor.xml) и вспомогательные drawable/values-файлы, применяемые во всех режимах.

Все макеты построены на базе ConstraintLayout, что обеспечивает максимальную адаптивность под различные разрешения и ориентации экрана, включая планшеты и устройства с нестандартными дисплеями. Компоненты связываются с логикой сервисов через методы findViewById() и обрабатываются в соответствующих классах сервисов (onServiceConnected() или onCreateOverlay()). Визуальное оформление выполнено с использованием концепций Glassmorphism и Material You, обеспечивая современный, интуитивный и минималистичный стиль.

Поддерживается динамическая настройка визуальных параметров: прозрачности, размеров, цвета, а также встроенные анимации и визуальные подсказки (например, вспышка при клике, прогресс-индикатор автотапа). Интерфейсы легко расширяются за счёт повторного использования общих компонентов и кастомных drawable-ресурсов.

Вся визуальная и структурная информация приложения организована в соответствии с рекомендациями Android SDK и располагается в каталоге res/ и shared_prefs/: см.рисунок 2.7

```
├─ res/
│   ├─ layout/
│   │   ├─ accessibility_service_eye.xml
│   │   ├─ accessibility_service_gsensor.xml
│   │   ├─ accessibility_service_touchpad.xml
│   │   └─ activity_main.xml
│   ├─ drawable/
│   │   └─ calibration_drawable.xml
│   └─ values/
│       └─ strings.xml, styles.xml
├─ shared_prefs/
│   └─ cursor_prefs.xml
```

Рисунок 2.7 – Структура хранения ресурсов и настроек проекта AIS-A

Использование XML-разметки для описания интерфейсов обеспечивает: Разделение логики и представления, упрощающее отладку и поддержку;

Визуальную гибкость — возможность изменять внешний вид без перекомпиляции логики;

Повторное использование компонентов — стандартные элементы легко включаются в разные макеты;

Адаптивность и масштабируемость — поддержка различных устройств и конфигураций;

Совместимость с дизайн-системами — полная интеграция с Material You и современными подходами к UI/UX.

2.5 Взаимодействие между настройками и интерфейсом

Одна из ключевых задач при проектировании адаптивных интерфейсов — обеспечение непрерывного и отзывчивого взаимодействия между пользовательскими настройками и визуальным интерфейсом. В системе AIS-A эта задача решается за счёт интеграции механизма хранения настроек (SharedPreferences) с динамическим обновлением компонентов интерфейса через внутренние вызовы и широковебательные сообщения (BroadcastReceiver).

Подобная архитектура позволяет пользователю в реальном времени изменять параметры поведения системы, моментально наблюдая результат на экране, без необходимости перезапуска приложения или перезагрузки сервисов. Это особенно критично для пользователей с ограниченными возможностями, для которых важна стабильность, предсказуемость и мгновенная реакция интерфейса.

Принцип работы

Все параметры, задаваемые через элементы интерфейса (ползунки, переключатели, чекбоксы), немедленно передаются в соответствующий сервис управления. Это реализуется следующим образом: При изменении значения (например, размера или скорости курсора), происходит сохранение нового параметра в SharedPreferences. Сразу после этого в соответствующий сервис отправляется Broadcast-интент, содержащий обновлённое значение. Сервис перехватывает сообщение, обрабатывает его и применяет изменение на лету — например, обновляет внешний вид курсора, режим управления или параметры фильтрации.

Все действия сопровождаются визуальной обратной связью, повышающей прозрачность работы системы для пользователя.

Пример взаимодействия: Изменение размера курсора

При перемещении ползунка в настройках курсора новое значение моментально сохраняется в SharedPreferences, после чего сервис получает сигнал и пересоздаёт изображение курсора на экране с обновлёнными параметрами. Пользователь сразу видит результат и может откорректировать значение при

необходимости. Или например: Выбор метода клика (ртом, звуком, неподвижностью). Переключение соответствующего переключателя (RadioButton или Switch) вызывает обновление конфигурации, активирует/деактивирует прослушивание микрофона или анализ движения рта. Интерфейс мгновенно адаптируется: отображаются новые кнопки, активируются прогресс-индикаторы, и т.п.

Изменение чувствительности вызова зоны предтапа или автотапа или скорости движения. Сервисы получают новые значения и корректируют алгоритмы движения, задержку или фильтрацию без прерывания сеанса. Курсор продолжает движение с уже применёнными параметрами, без потери позиции или конфликта настроек. Преимущества такого подхода: Непрерывность взаимодействия — изменения вступают в силу мгновенно, без перезапуска приложений и сервисов; Гибкость конфигурации — пользователь может поэкспериментировать с параметрами и добиться оптимального поведения под свои особенности; Надёжность и предсказуемость — система избегает зависаний, дублирующих состояний и конфликтов между модулями; Обратная связь — визуальные и аудио-индикаторы подтверждают, что изменения применены и активны; Масштабируемость — подход легко расширяется новыми параметрами и сервисами.

Таким образом, взаимодействие между настройками и пользовательским интерфейсом в AIS-A построено на основе современных принципов реактивной архитектуры: любое изменение состояния отражается в поведении системы в реальном времени, что обеспечивает комфорт и адаптацию даже в условиях ограниченной физической активности пользователя.

2.6 Разработка и реализация приложения AIS-A

Процесс разработки мобильного приложения AIS-A (Alternative Interaction System for Android) охватывал полный цикл проектирования, реализации, отладки и интеграционного тестирования системы hands-free управления Android-устройством. Главной целью являлось создание адаптивного решения, ориентированного на пользователей с ограниченными возможностями двигательной активности, не способных использовать стандартные элементы управления (сенсорный экран, мышь, клавиатура).

Разработка велась в среде Android Studio с использованием языка Kotlin, а также API Android SDK 31, что обеспечило актуальность и совместимость с современными устройствами.

Архитектура и модульная структура

Основу архитектуры приложения составляет модульный подход, при котором каждый режим управления реализован как отдельный сервис на базе

AccessibilityService. В системе предусмотрены четыре независимых модуля: T-PAD – управление через тачпад-зону экрана; HEAD – управление курсором по движению головы; EYES – наведение и выбор с помощью направления взгляда и мимики; G-sens – взаимодействие на основе данных акселерометра.

Каждый модуль обрабатывает данные из собственного источника ввода (сенсор, камера, микрофон) и взаимодействует с интерфейсом пользователя через overlay-компоненты, созданные средствами WindowManager. Объединение модулей осуществляется через главное меню, оформленное в виде панели запуска режимов. В этом меню пользователь может выбрать нужный режим управления, а также получить доступ к его настройкам.

Коммуникация между компонентами реализована посредством:

SharedPreferences — для хранения индивидуальных пользовательских настроек; BroadcastReceiver — для передачи изменений параметров и команд между интерфейсами и сервисами в реальном времени.

Это обеспечивает высокую степень адаптивности: при изменении параметров (например, чувствительности курсора или метода клика) сервисы моментально получают обновлённые значения и применяют их без перезапуска, что особенно важно для удобства пользователей.

Пользовательский интерфейс выполнен в соответствии с рекомендациями Material You и стилистикой Glassmorphism, что делает его современным, прозрачным, легко воспринимаемым. Каждый режим управления сопровождается собственным окном настроек, реализованным в виде XML-макета, включая: accessibility_service_eye.xml, accessibility_service_head.xml, accessibility_service_touchpad.xml, accessibility_service_gsensor.xml.

Все настройки сгруппированы логически и сопровождаются подсказками и визуальной обратной связью (ползунки, индикаторы, кнопки). Изменения значений передаются в соответствующий сервис, где немедленно обрабатываются.

При активации любого режима система проверяет наличие необходимых разрешений: доступ к камере, микрофону или разрешение на наложение поверх окон. В случае их отсутствия, пользователь автоматически перенаправляется в соответствующее системное окно настроек для предоставления прав. Это гарантирует корректную работу модулей и защиту приватности.

Жизненный цикл каждого сервиса управляется в соответствии с официальной документацией Android. Методы onStart, onStop, onDestroy реализованы с учётом рекомендаций по безопасности и освобождению ресурсов. При остановке сервиса освобождаются потоки, отключается камера или микрофон, закрываются overlay-элементы.

2.6.1 Проблемы, возникшие в ходе разработки, и пути их решения

В процессе реализации проекта были выявлены несколько технических и функциональных сложностей: В режиме HEAD наблюдалась инверсия координат и нестабильность курсора при быстрых поворотах головы; В EYES курсор дрожал из-за высокой чувствительности к микродвижениям глаз и шумам от FaceDetector; При реализации метода клика по звуку возникали ложные срабатывания в условиях фонового шума. Для устранения этих проблем были внедрены следующие решения: Экспоненциальное сглаживание координат — для подавления дрожаний при нестабильных данных; Псевдо-Калманов фильтр — для дополнительной стабилизации траектории движения курсора; Пороговые значения и предобработка звука с помощью FFT — для фильтрации шумов и исключения случайных срабатываний.

В результате проделанной работы приложение AIS-A было реализовано как полноценная, адаптивная и масштабируемая система hands-free управления, обладающая гибкой архитектурой, настраиваемым поведением и доступным интерфейсом. Все модули были протестированы как по отдельности, так и в составе общей системы. Интеграционные испытания подтвердили корректность взаимодействия сервисов, устойчивость приложения к нагрузке, а также эффективность реализованных методов управления.

Полный анализ и результаты тестирования представлены в разделе — 3.

2.6.2 Общий процесс разработки и интеграция модулей

Процесс разработки программного комплекса AIS-A (Alternative Interaction System for Android) строился на основе модульной и поэтапной модели, что обеспечило высокую управляемость проектом, возможность параллельной работы над отдельными компонентами и постепенное наращивание функциональности без нарушения целостности системы.[14]

Ключевым принципом разработки стало разделение функциональных блоков (сервисов управления) и их последующая интеграция в единое приложение. Такой подход обеспечил гибкость, масштабируемость и простоту отладки.

Разработка приложения велась по классическому каскадно-инкрементальному подходу, в рамках которого были последовательно реализованы следующие этапы: Построение технического задания и сбор требований. Были сформированы функциональные и нефункциональные требования к каждому из режимов управления: точность, отзывчивость, устойчивость к шумам и дрожаниям, визуальная адаптивность и простота

настройки.

В рамках подготовки архитектуры была организована файловая структура проекта с чётким разделением по задачам см. рисунок 2.8

└─ ui/	– экранные компоненты (MainActivity, Settings)
└─ service/	– сервисы управления курсором (HEAD, EYES, G-sens, T-PAD)
└─ layout/	– XML-макеты экранов и интерфейсов
└─ drawable/	– визуальные ресурсы (иконки, индикаторы)
└─ values/	– стили, цвета, строки интерфейса
└─ xml/	– конфигурации служб специальных возможностей
└─ shared_prefs/	– настройки пользователя (cursor_prefs.xml)
└─ libs/	– внешние библиотеки (.jar/.aar)

Рисунок 2.8 – Файловая структура проекта AIS-A

Каждый режим управления (T-PAD, HEAD, EYES, G-sens) разрабатывался как самостоятельный модуль, отвечающий за получение входных данных (акселерометр, камера, касания, звук) и управление курсором. Это позволило тестировать их изолированно, снижая риски ошибок на ранних этапах.

Был создан универсальный интерфейс — главное меню, через которое пользователь запускает нужный режим, а также экраны конфигурации, соответствующие каждому модулю. Интерфейсы оформлены в стиле Material You + Glassmorphism.

Все модули были связаны через универсальный механизм обмена данными — BroadcastReceiver. Это позволило динамически передавать настройки и команды, не создавая жёсткой связности между компонентами.

В каждый сервис встроены проверки наличия разрешений на доступ к камере, микрофону, overlay и акселерометру. Реализована корректная и безопасная остановка сервисов (onStop, onDestroy), освобождение ресурсов и предотвращение конфликтов между режимами.

Каждый модуль проходил модульное тестирование в изоляции, после чего проводилось интеграционное тестирование всей системы. Особое внимание было уделено конфликтам между службами, корректности остановки и запусков, обработке исключений, визуальному поведению.

Были устранены мелкие баги, оптимизированы алгоритмы фильтрации движения, адаптированы таймеры автотапа и снижена нагрузка на ресурсы устройства. Добавлены инструкции для пользователя и обеспечена совместимость с Android 12+.

Интеграция компонентов AIS-A осуществлялась на нескольких уровнях:

Интерфейсный уровень (MainActivity). Все режимы управления доступны через общее главное меню, откуда происходит запуск соответствующих сервисов и переход к настройкам.

Общая система хранения пользовательских параметров позволяет разделять и передавать настройки между модулями без дублирования кода и конфигураций. Универсальные приёмники команд обеспечивают реагирование каждого сервиса на изменения параметров и событий без необходимости прямой связи между компонентами.

Все сервисы используют общую логику отображения курсора, активных зон, индикаторов и визуальных эффектов. Переиспользуемые оверлеи позволяют сохранять единый визуальный стиль и снижают сложность поддержки.

Благодаря соблюдению принципов модульности, слабой связности и повторного использования, интеграция компонентов прошла без конфликтов. Все режимы управления работают стабильно как по отдельности, так и в составе общей системы.

Также была достигнута визуальная и функциональная унификация интерфейса, что облегчает обучение и взаимодействие пользователей. Поддержка общей системы настроек позволяет сохранять пользовательские предпочтения при смене режимов, а универсальные Broadcast-команды обеспечивают централизованное управление логикой поведения.

Таким образом, разработка и интеграция приложения AIS-A завершились формированием масштабируемой, модульной архитектуры, готовой к дальнейшему расширению, улучшению и адаптации под новые сенсоры и методы взаимодействия. Все сервисы успешно прошли интеграционное тестирование и подготовлены к реальному применению.

3.ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ (методика, результаты)

Тестирование мобильного приложения AIS-A (Alternative Interaction System for Android) являлось завершающим этапом разработки и было направлено на всестороннюю проверку корректности функционирования всех компонентов системы. Основной целью данного этапа было подтверждение стабильной работы каждого из режимов управления, надёжности архитектуры, а также совместимости пользовательского интерфейса с различными условиями эксплуатации.

В рамках тестирования применялись как формальные методы (модульное и интеграционное тестирование), так и ручные пользовательские проверки, максимально приближённые к реальным условиям использования устройства людьми с ограниченными возможностями.

Тестирование проводилось по многоуровневой схеме, включающей пять взаимосвязанных этапов: Модульное тестирование

На данном этапе каждый сервис (TouchpadAccessibilityService, HeadAccessibilityService, EyeAccessibilityService, GSensorAccessibilityService) проверялся изолированно. Цель — убедиться в правильной работе базовой логики и отсутствии сбоев в ключевых функциях: запуск и остановка сервиса; отклик на пользовательские команды; корректность обработки настроек; отображение визуальных элементов (курсора, индикаторов, зон); выполнение жестов и кликов.

Следующим этапом шло интеграционное тестирование, выполнялась поэтапная интеграция сервисов в рамках общего фреймворка. Проверялись: переключение между режимами управления без перезапуска приложения; корректная остановка предыдущего сервиса при запуске нового; передача настроек между интерфейсом и сервисом через BroadcastReceiver; отсутствие конфликтов между overlay-компонентами; единый стиль интерфейса и поведение при возврате в главное меню.

Во время функционального тестирования интерфейса, была проведена комплексная проверка всех компонентов UI: доступность всех элементов настройки; отклик на действия пользователя (переключатели, ползунки, чекбоксы); визуальные подсказки при активации функций (автотап, клики, триггер-зоны); корректность отображения диалогов и сообщений при отсутствии разрешений; отображение прогресса при калибровке и выполнение тапов.

На этапе производительного тестирования оценивались количественные метрики: время запуска сервисов (замер от активации до появления курсора); задержка отклика курсора при получении сигнала с сенсоров; скорость обработки жестов и кликов; влияние на энергопотребление — приложение проверялось при длительном использовании в фоне (30+ минут работы без перерыва); нагрузка на ресурсы устройства (использование ЦП, памяти, камеры,

микрофона).

Также проводилось тестирование устойчивости. Для его оценки создавались нестандартные ситуации, выполнялись: отключение разрешений во время работы сервиса (камера, микрофон, overlay); переворот экрана (смена ориентации); сворачивание приложения и возврат из фона; выгрузка приложения из памяти и автоматический перезапуск; активация режима при слабом уровне сигнала или нестабильной работе сенсоров.

Тестирование показало, что приложение AIS-A работает стабильно и предсказуемо во всех предусмотренных сценариях использования: Все четыре режима управления (HEAD, EYES, G-SENS, T-PAD) успешно запускались, останавливались и корректно реагировали на команды из пользовательского интерфейса; Курсовое управление функционировало плавно, с минимальной задержкой, благодаря реализованным алгоритмам сглаживания (экспоненциальное и псевдо-Калманов фильтр); Сервисы правильно освобождали ресурсы при завершении и не конфликтовали между собой при смене режима; Настройки корректно передавались через BroadcastReceiver и применялись в реальном времени; Пользовательский интерфейс оставался стабильным при смене ориентации, фоновом запуске, а также после смены параметров конфигурации; Отказоустойчивость была обеспечена благодаря дополнительным проверкам разрешений и механизму автопаузы.

Таким образом, проведённое тестирование подтвердило готовность приложения AIS-A к реальному применению. Все модули прошли как индивидуальную проверку, так и совместное интеграционное тестирование, что подтверждает надёжность архитектуры, адаптивность пользовательского интерфейса и устойчивость системы к типичным сбоям.

3.1 Средства тестирования

В названии темы работы используется термин "Android-устройства" который включает смартфоны, но не ограничивается ими.

Однако в дипломе реализация была выполнена на смартфоне, потому что: это наиболее массовый и доступный тип Android-устройств, на который ориентируются почти все assistive-технологии; пользователи с ОВЗ в 95% случаев используют именно смартфоны, а не планшеты или телевизоры. Смартфон даёт доступ ко всем необходимым сенсорам: камера, акселерометр, микрофон — без дополнительного оборудования. Реализация на других устройствах возможна, т.к. с технической точки зрения различий в API между смартфоном или планшетом в рамках использования AccessibilityService и SensorManager — нет. Поэтому функциональность будет сохраняться при адаптации, без изменений архитектуры — достаточно адаптировать интерфейс и

разрешения.

Стартом тестирования стал эмулятор Android Studio с различными размерами экранов и ориентациями;

Также приложение было протестировано на десятке разнообразных физических устройств, выбирались максимально непохожие устройства, с разными характеристиками. Самые примечательные показаны в таблице 3.1.

Таблица 3.1 - Устройства на которых тестировалось приложение AIS-A

Модель	Краткие технические характеристики	Версия андроид	Обоснование выбора устройства
Samsung Galaxy a30s	Процессор: 2x 1.8 GHz ARM Cortex-A73, 6x 1.6 GHz ARM Cortex-A53, Графический процессор: ARM Mali-G71 MP2, 770 МГц, ОЗУ 4 ГБ Фронтальная камера : 16Мп	Android 9.0	Оболочка One UI 1.5.
qin f21 pro	Процессор: MediaTek 6761 с четырьмя ядрами Cortex-A53 2.0GHz ГГц ОЗУ: 4 ГБ Фронтальная камера 2 Мп	Android 10	Смартфон с маленьким экраном, без PlayMarket, полным отсутствием Google-сервисов, с физическими кнопками, нестандартным расположением фронтальной камеры. Нет акселерометра.

FREETEL Model1 Retro II	Процессор MediaTek Helio G85 с четырьмя ядрами 2.0GHz ОЗУ 4 GB, фронтальная камера 13 Мп	Android 13	Смартфон в формате «раскладушки» с маленьким экраном, сенсорным экраном и физическими кнопками
Unihertz Jelly 2	MediaTek Helio P60 (MT6771), Процессор: 4x2.0 GHz ARM Cortex-A73, 4x2.0 GHz ARM Cortex-A53 ОЗУ 6 ГБ фронтальная камера 8Мп	Android 11	Самый маленький серийно выпускаемый смартфон в мире.
redmi 10 a	MediaTek Helio G25 (MT6762G) Процессор: 4x 2.0 GHz ARM Cortex-A53, 4x 1.5 GHz ARM Cortex-A53 ОЗУ 3 ГБ фронтальная камера 5 Мп	Android 11	Оболочка MIUI 12.5

3.2 Результаты тестирования

В ходе комплексного тестирования, охватывающего модульную, интеграционную, функциональную и пользовательскую проверку, все четыре режима управления в приложении AIS-A успешно продемонстрировали соответствие заявленной функциональности и устойчивую работу в рамках проектной логики. Ниже приведён обзор ключевых результатов и выявленных особенностей по каждому режиму.

Общие результаты

Все сервисы (T-PAD, HEAD, EYES, G-sens) успешно прошли

изолированное и комплексное тестирование, включая запуск, завершение, отклик на команды, корректную обработку настроек, взаимодействие с интерфейсом и визуальные отклики. Сервисы корректно освобождают ресурсы при смене режима, не конфликтуют между собой и устойчиво работают в фоне. Интерфейс реагирует на изменения параметров в реальном времени без необходимости перезапуска.

Режим HEAD в процессе тестирования при слабом уровне освещённости наблюдались дрожания и нестабильность курсора. Проблема была решена за счёт внедрения экспоненциального сглаживания и адаптивного масштабирования движения головы в зависимости от скорости и освещённости. Первоначально предполагалось, что фронтальная камера с разрешением менее 5 Мп может не справляться с задачей. Однако практика показала, что качество изображения в данном случае играет меньшую роль, чем равномерность освещения и отсутствие сильных теней на лице.

Режим EYES при стабильном освещении и правильно проведённой калибровке достигнута точность позиционирования курсора в пределах 10% от целевого элемента, что считается приемлемым для hands-free интерфейсов. Алгоритм прилипания к кликабельным элементам улучшил взаимодействие при слабом зрительном контроле. Повышенная чувствительность к теням и бликам устранена за счёт повторной калибровки и сглаживания.

Режим T-PAD обнаружена задержка обработки двойного тапа при высокоскоростном перемещении пальца по виртуальному тачпаду. Проблема решена путём увеличения временного окна интерпретации двойного нажатия до 300 мс, что улучшило отзывчивость и снизило процент ошибок.

Режим G-sens успешно реализовано прилипание курсора к элементам при замедлении движения. При высоком уровне чувствительности возникали ложные срабатывания тапов из-за микродвижений устройства. Для компенсации добавлен порог допустимого движения и механизмы подтверждения действия (зона предтапа).

Приложение проверялось при подключении смартфона к внешним экранам (телевизор, проектор, AR-очки XREAL Air 2) по протоколу Miracast.

Были выполнены сценарии: открытие и просмотр YouTube, Instagram, TikTok; поиск через браузер, открытие страниц и переходы по ссылкам; набор текста на стандартной клавиатуре Android; запуск галереи и просмотр изображений; Все действия выполнялись с использованием только альтернативных методов управления — без прикосновений.

Приложение успешно протестировано на устройствах с 3 ГБ ОЗУ и процессором Snapdragon 662 (уровень производительности — ниже среднего). При включении трекинга глаз и обработки звука наблюдается увеличение потребления CPU до 40–50%. На слабых устройствах рекомендуется использовать режимы T-PAD или G-SENS, не требующие камеры и аудиоанализа.

Для количественной оценки производительности были зафиксированы

следующие значения (на устройстве среднего уровня): средняя задержка отклика курсора после движения головы — около 120 мс; точность позиционирования (режим EYES) — до 10% от центра элемента управления; среднее время активации автотапа при неподвижности — 800–1000 мс; количество ложных срабатываний клика по звуку в шумной среде — до 25%, при использовании FFT и порогов — снижается до 5%; процент успешных действий (клик, запуск приложения, навигация) — от 85% до 97% в зависимости от режима.

Таблица 3.2 — Сравнение режимов управления по ключевым метрикам (средние значения)

Параметр	HEAD	EYES	T-PAD	G-SENS
Задержка отклика	~120 мс	~140 мс	<100 мс	~110 мс
Точность позиционирования	средняя	высокая*	высокая	средняя
Уровень утомляемости	низкий	высокий	средний	низкий
Процент ложных тапов	<2%	до 8%	<1%	до 5%
Рекомендованная поза	сидя , полусидя лежа на спине(с подушкой под головой)	полусидя, сидя , лежа на спине(голова должна быть неподвижна!)	сидя, стоя, лежа на спине,	лёжа на боку, лежа на животе, сидя
*при стабильном освещении и точной калибровке				

Проведённое тестирование подтвердило, что приложение AIS-A: стабильно функционирует при различной освещённости и уровнях чувствительности; эффективно адаптируется к условиям эксплуатации; допускает использование в комбинации с внешними экранами; обладает высокой точностью и отзывчивостью при корректно выполненной настройке; устойчиво к сбоям, резким изменениям состояния сенсоров, отключению разрешений и перевороту экрана. Система прошла этап технического тестирования и доказала готовность к пользовательским испытаниям среди целевой аудитории — людей с двигательными ограничениями.

3.3 Тестирование пользователями с двигательными ограничениями

В завершающей фазе проекта было проведено прикладное тестирование с участием реальных пользователей, имеющих стойкие ограничения моторики, но обладающих сохранным уровнем когнитивных функций.

Организация тестирования

К участию были привлечены пациенты под наблюдением врача-реабилитолога, проходящие восстановительную терапию после травм, инсульта или в связи с врождёнными двигательными нарушениями. Все участники были заранее проинструктированы по методике работы с приложением и имели базовые навыки обращения с Android-устройствами. Тестирование проводилось в условиях дневного освещения и с подключением внешнего дисплея для визуального наблюдения за действиями. Использовались сценарии: наведение курсора, калибровка, выбор метода клика, открытие приложений, ввод текста, навигация по меню.

Пользователи с разными нарушениями уверенно справлялись с навигацией в режиме T-PAD.

HEAD и EYES показали высокую эффективность при спокойной посадке и стабильном освещении. Метод автотапа оказался наиболее интуитивным для большинства участников. Клик по звуку был признан удобным, но требующим чёткой настройки эталона и тишины в помещении.

Все участники отметили удобство интерфейса, визуальные подсказки и то, что можно использовать только голову или один палец для управления всем устройством. Результаты второго этапа тестирования с участием целевой аудитории подтвердили высокий потенциал приложения AIS-A как вспомогательного средства для людей с ограниченными двигательными возможностями. Благодаря адаптивности, минимализму интерфейса и гибким настройкам, система может использоваться как в быту, так и в рамках реабилитационной практики. Результаты будут дополнительно использованы для доработки системы под реальные поведенческие сценарии.

3.4 Тестирование с пользователем, имеющим последствия тяжёлой черепно-мозговой травмы.

Общая информация о пользователе

Участником пользовательского тестирования стал М., 28 лет, находящийся на длительной реабилитации после тяжёлой черепно-мозговой травмы. Диагноз включал: спастический тетрапарез (ограничение в движении всех четырёх конечностей), афазию (нарушение речевой функции), нистагм (непроизвольные

колебательные движения глаз).

Не смотря на серьёзные ограничения, у М. наблюдалась частичная сохранность моторики в правой кисти — он мог шевелить пальцами, удерживать предметы, сжимать и слегка перемещать их. Это сделало возможным использование мобильного устройства при условии адаптации интерфейса.

Методика взаимодействия и выбор режима T-PAD

Ввиду того, что у М. была ограниченная, но стабильная подвижность большого пальца правой руки, в качестве начального способа управления был предложен режим T-PAD, реализованный в приложении AIS-A. Телефон с запущенным приложением был уложен на ладонь пользователя экраном вверх, так чтобы он мог управлять курсором движением одного пальца. Смартфон был заранее подключён к проектору через Miracast, а изображение проецировалось на потолок (см. рисунок 3.1).



Рисунок 3.1— Укладка телефона с AIS-A в режиме T-PAD для пользователя с ограниченной моторикой.

М. с интересом воспринял предложенную схему и сумел: переключать экраны; запускать приложения (YouTube, Chrome); совершать базовые действия в интерфейсе.

Несмотря на положительный опыт использования режима T-PAD, при длительном применении в предложенной позе у пользователя наблюдалось утомление, что обусловлено статичной нагрузкой на руку. Это подчёркивает необходимость индивидуального подбора позы и режима. Не смотря на это, он выразил положительное мнение об опыте и продолжал использовать режим T-PAD в течение месяца, после чего проявил интерес к альтернативным способам

управления.

3.4.1 Переход к управлению через акселерометр (режим G-sens)

В ходе реабилитации у М. появились движения в лучезапястном суставе — сгибание, разгибание, а также супинация и пронация кисти (см. рисунок 3.2).



Рисунок 3.2 — Пронация, нейтраль, супинация кисти руки у пользователя.

Это позволило перейти к тестированию режима G-sens, использующего данные акселерометра. Телефон был зафиксирован в спортивном чехле и закреплён на массажном мяче, экраном наружу (см. рисунки 3.3, 3.4). Проектор теперь направлялся на стену, создавая более привычную плоскость отображения (см. рисунок 3.5).



Рисунок 3.3 — Крепление смартфона в чехле на мяче.



Рисунок 3.4 — Крепление смартфона в чехле на полусфере.



Рисунок 3.5— Проецирование изображения на стену.

Пользователь обучился сопоставлять движения руки с движениями курсора: сгибание — движение курсора вправо, разгибание — влево, супинация — вверх, пронация — вниз.

Управление через акселерометр оказалось более удобным и менее утомительным, чем ранее протестированный режим T-PAD. М. отметил, что при G-sens затраты усилий ниже, движения проще координируются, а реакция системы — предсказуемая. По его субъективной оценке, именно этот способ оказался наиболее комфортным и доступным для длительного использования.

Дополнительно была проведена экспериментальная проверка работы системы в различных положениях тела пользователя. М. тестировал режим G-sens: лёжа на животе, размещённом на массажном мяче (см. рисунок 3.6, 3.7), при этом использовал не только движения кисти, но и помогал телом: раскачивался на мяче, отталкивался ногами и искал оптимальный угол наклона руки для ускорения перемещения курсора;



Рисунок 3.5— Положение лежа на животе на мяче



Рисунок 3.6— Проецирование изображения на пол

Сидя с выпрямленными ногами, когда экран проектора располагался справа, а голова была повёрнута в ту же сторону, что обеспечивало удобный визуальный контроль за интерфейсом(см. Рисунок 3.7).



Рисунок 3.7— Положение сидя на полу с упором на мяч

Каждая поза раскрывала разные аспекты адаптации интерфейса под индивидуальные особенности пользователя и продемонстрировала высокий потенциал для интеграции в реабилитационные практики.

Особенно интересно проявила себя позиция “лёжа на животе на мяче”, в которой пользователь интуитивно начал включать в работу не только руку, но и все тело: раскачиваться, отталкиваться ногами, балансировать. Эти действия не только способствовали управлению курсором, но и задействовали крупные мышечные группы, создавая естественную двигательную активность.

Таким образом, использование AIS-A в такой конфигурации фактически сформировало своеобразную БОС-платформу (биологической обратной связи), где результат действий (движение курсора) становился немедленным визуальным откликом, а сам процесс управления — стимулом к контролируемой двигательности. Это открывает перспективы использования подобных решений в рамках нейрореабилитации и тренировки двигательных навыков.

3.4.2 Попытка использования режимов EYES и HEAD

После успешного освоения управления через наклоны, пользователь изъявил желание попробовать управление глазами. Однако режим EYES оказался непригодным для текущего состояния пользователя из-за нестабильного нистагма, который мешал калибровке и вызывал хаотичное движение курсора. Хотя нистагм проявлялся не постоянно, он значительно затруднял распознавание направления взгляда.

Учитывая появление небольших движений головы, было принято решение протестировать режим HEAD. Телефон был установлен на штатив (или селфи-палку) на расстоянии 40–50 см от лица пользователя. М. лежал на спине и без затруднений прошёл калибровку, после чего смог управлять курсором при помощи поворотов головы: опускать и поднимать курсор;

перемещать его вправо (в этом направлении движения были более свободными ввиду спастичности).

Хотя управление было менее плавным, чем в режиме G-sens, М. отметил, что такой подход кажется ему более привычным, так как экран был расположен прямо перед глазами, без необходимости взгляда на проекцию на потолке или стене.

3.4.3 Вывод по наблюдению

В ходе длительного взаимодействия с приложением AIS-A пользователь М. продемонстрировал устойчивый прогресс в освоении различных режимов hands-free управления, не смотря на тяжёлые двигательные ограничения. Включение в работу даже минимально сохранных функций правой руки позволило ему не только осуществлять базовую навигацию по интерфейсу устройства, но и почувствовать контроль над цифровой средой, что само по себе стало значимым психологическим и мотивационным фактором.

Наиболее эффективным и комфортным оказался режим G-sens, в котором управление осуществлялось через движения кисти и предплечья. Этот способ требовал минимальных усилий, позволял задействовать биомеханику в естественных амплитудах и был успешно адаптирован под несколько телесных позиций. Особенно интересным оказался опыт использования телефона, закреплённого на мяче в положении лёжа на животе — в таком варианте формировалась своеобразная БОС-платформа (биологической обратной связи), в которой каждое движение пользователя вызывало немедленную визуальную реакцию на экране, стимулируя целенаправленную двигательную активность.

Режим HEAD также показал положительную динамику. Пользователь

сумел калиброваться, выполнять базовые перемещения курсора и выразил заинтересованность в дальнейшем использовании этого способа. Не смотря на меньшую плавность управления по сравнению с G-sens, привычное положение экрана перед глазами обеспечивало психологический комфорт.

Режим EYES, к сожалению, не дал стабильных результатов из-за нестабильного нистагма, который затруднял калибровку и вызывал хаотичное движение курсора.

Общий вывод:

Регулярное использование приложения AIS-A в адаптированной форме способно не только удовлетворить потребность в управлении устройством у людей с тяжёлыми двигательными нарушениями, но и стать инструментом стимуляции двигательной активности, формирования новых устойчивых навыков, а также дополнением к нейро- и кинезитерапии. Система продемонстрировала высокий потенциал для применения в индивидуальных реабилитационных программах.

3.5 Тестирование с пользователем, имеющим последствия спинальной травмы

Общие сведения о пользователе: Пользователь К., 65 лет, с диагнозом тетрапарез на фоне перенесённой спинальной травмы шейного отдела позвоночника, был включён в пользовательское тестирование приложения AIS-A. Ввиду отсутствия активных движений в руках и туловище на момент начала взаимодействия, был выбран наиболее подходящий режим управления — EYES, основанный на отслеживании направления взгляда.

3.5.1 Использование режима EYES

Пользователь находился в полулежачем положении, голова была зафиксирована за счёт естественного давления тела на специальное крепление по типу мягкого хомута, удерживающего смартфон перед лицом (см. рисунок 3.8). Экран устройства был расположен прямо напротив глаз пользователя, а лицо хорошо освещено для точного распознавания.

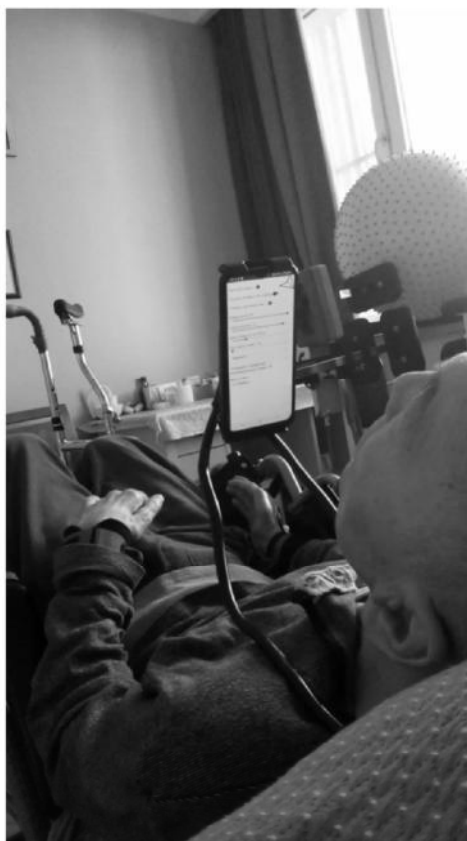


Рисунок 3.8— Установка устройства перед лицом пользователя в режиме EYES.

Из-за неподвижности головы, связанной с последствиями перелома шейных позвонков, движения глаз фиксировались приложением AIS-A с высокой точностью, что обеспечивало чёткое позиционирование курсора. Пользователь смог: запускать видео в YouTube; набирать короткие сообщения; перемещаться по интерфейсу с минимальной погрешностью.

Тем не менее, К. отметил высокую утомляемость, связанную с необходимостью длительно удерживать фокус взгляда, а также с повышенной концентрацией внимания, требуемой для выполнения команд.

3.5.2 Переход к режиму G-sens

Спустя примерно месяц регулярных упражнений у К. появилась частичная подвижность в правой руке: движения в локтевом суставе, супинация, пронация, а также сгибание и разгибание в запястье. Это позволило протестировать режим

G-sens, в котором управление осуществляется через движения устройства, зафиксированного на теле.

Телефон с заранее активированным AIS-A был закреплён на тыльной стороне ладони, при этом пользователь находился в прежнем положении, а изображение проецировалось на потолок с помощью Miracast (см. рисунки 3.9 и 3.10).



Рисунок 3.9 - Способ крепления устройства при использовании режима G-sens.



Рисунок 3.10 - Проекция экрана смартфона на потолок.

Пользователь довольно быстро освоил управление курсором, отметив, что оно стало более естественным и «приятным», а также напомнило ему упражнение для разработки руки. Он мог дольше пользоваться этим режимом без выраженной усталости, хотя по прошествии времени всё же наступала утомляемость.

3.5.3 Попытка использования режима T-PAD

В дальнейшем, с появлением некоторой активности пальцев на левой руке, было принято решение протестировать режим T-PAD. Однако, не смотря на желание пользователя, движений пальцев оказалось недостаточно, чтобы достичь сенсорной зоны экрана. Попытка была признана нецелесообразной, и этот режим временно отложен.

3.5.4 Переход к управлению через движения головы (HEAD)

Через неделю К. отметил, что может совершать минимальные движения головой — повороты, наклоны, разгибания. Это позволило протестировать режим HEAD.

Изначально возникли трудности: пользователь не мог долго сидеть без поддержки головы, а при подложенной под голову подушке амплитуда движений была ограничена. Было принято решение использовать специальный подвес для головы, по принципу балансирующего противовеса. Смартфон устанавливался на штатив на расстоянии 40–50 см от лица (см. рисунок 3.11).



Рисунок 3.11 - Крепление устройства в режиме HEAD

К. смог управлять курсором с помощью головы, но столкнулся с ограничением: недостаточный диапазон движений не позволял курсору достигать краёв экрана. Однако в центральной области интерфейса управление

было точным, и пользователь с интересом выполнял задачи, комментируя, что будет стремиться «дотянуться» до углов в будущем. Он отметил удобство прямого визуального контакта с экраном и положительное ощущение от движений головы, расценивая их как «полезные и активизирующие».

Удобство такой позиции заставило вернуться к режиму EYES и пользователь отметил что этот режим для него подходит лучше всего. Близость экрана телефона, отсутствовало лишнее напряжение мышц. Пользователь мог взаимодействовать с устройством довольно долго. Он сказал: "Я щупаю экран глазами".

3.5.5 Вывод по наблюдению

Пользователь К. показал, что даже при тяжёлых ограничениях подвижности можно подобрать доступный способ цифрового взаимодействия, способствующий как улучшению функциональных навыков, так и восстановлению движений через мотивированную активность.

Наиболее продуктивными оказались режимы EYES, G-sens и HEAD, обеспечившие реальное вовлечение руки и шеи в управление, а также положительное восприятие со стороны пользователя.

Режим EYES, хотя и оказался технически рабочим, в неудобной позе требовал высоких усилий и быстро утомлял, что ограничивает в подобной ситуации его длительное применение. Но при правильном положении пользователя, он обретает противоположные качества.

Режим T-PAD не был реализован из-за недостаточной силы и амплитуды движений пальцев, но может быть возвращён к рассмотрению при дальнейшем восстановлении.

Системный подход к тестированию, постепенное вовлечение режимов и гибкая адаптация интерфейса позволили подобрать индивидуальное решение даже для пользователя с тяжёлой формой тетрапареза. AIS-A в данном случае выступает не только как средство управления, но и как мотиватор, тренажёр и компонент восстановительной терапии.

Таким образом, использование режима EYES позволило пользователю с тетрапарезом выполнять действия, ранее недоступные без посторонней помощи. Это демонстрирует потенциал AIS-A как средства цифровой доступности в бытовых условиях при тяжёлых формах двигательной дисфункции.

3.5.6 Обобщённый вывод.

Результаты проведённого тестирования с участием пользователей с ограниченными возможностями здоровья (ОВЗ) подтверждают высокую актуальность и практическую значимость разработанного приложения AIS-A как вспомогательного средства hands-free взаимодействия с Android-устройствами.

Все участники проявили интерес ко всем режимам управления и выражали желание переключаться между ними в зависимости от текущего состояния. Нередкими были ситуации, когда пользователь самостоятельно просил сменить метод взаимодействия, объясняя это тем, что одна из частей тела «перетрудилась» или стала менее чувствительной. Такая возможность гибко подстраиваться под физическое состояние не только повышает удобство использования, но и даёт ощущение контроля и выбора, столь важное для людей в условиях функциональных ограничений.

AIS-A предоставляет широкий спектр альтернативных способов управления — через движение головы, глаз, акселерометр и сенсорную панель — каждый из которых может быть адаптирован под индивидуальные потребности. Это делает систему универсальным и гибким инструментом цифровой абилитации и реабилитации.

Приложение AIS-A продемонстрировало свойства БОС-системы (биологической обратной связи), при которых каждое действие пользователя вызывало мгновенный отклик на экране. Такая обратная связь повышает мотивацию и способствует формированию целенаправленного моторного контроля, особенно в рамках домашних реабилитационных программ.

Важно, что все тесты проводились в бытовых условиях, без специализированного медицинского оборудования, что подтверждает доступность технологии для домашнего использования. Приложение не требует навыков программирования, подключения дополнительных устройств или установки сложных систем, а значит, может быть рекомендовано для повседневного использования широким кругом людей с двигательными нарушениями.

Мой обобщённый вывод тестирования: AIS-A — это не просто интерфейс управления, а адаптивная, реабилитационно-значимая платформа, способная поддержать человека в восстановлении, адаптации и самостоятельном освоении цифровой среды. Возможность использовать приложение в домашних условиях, переключаться между режимами и наблюдать немедленный результат делает систему мощным инструментом в борьбе за независимость и качество жизни пользователей с ОВЗ.

В ходе испытаний приложение доказало, что может выступать не только как вспомогательное средство управления, но и как инструмент двигательной стимуляции, особенно в реабилитационных и домашних условиях. В отличие от

многих аналогов, AIS-A предоставляет сразу несколько способов hands-free взаимодействия, что делает его универсальным в адаптивных сценариях.

3.6 Другие реальные сценарии практического использования

Приложение AIS-A, будучи изначально ориентированным на пользователей с двигательными нарушениями, демонстрирует широкий потенциал применения и в других условиях, где традиционное взаимодействие с устройством затруднено или нецелесообразно (см.рисунок 3.12).

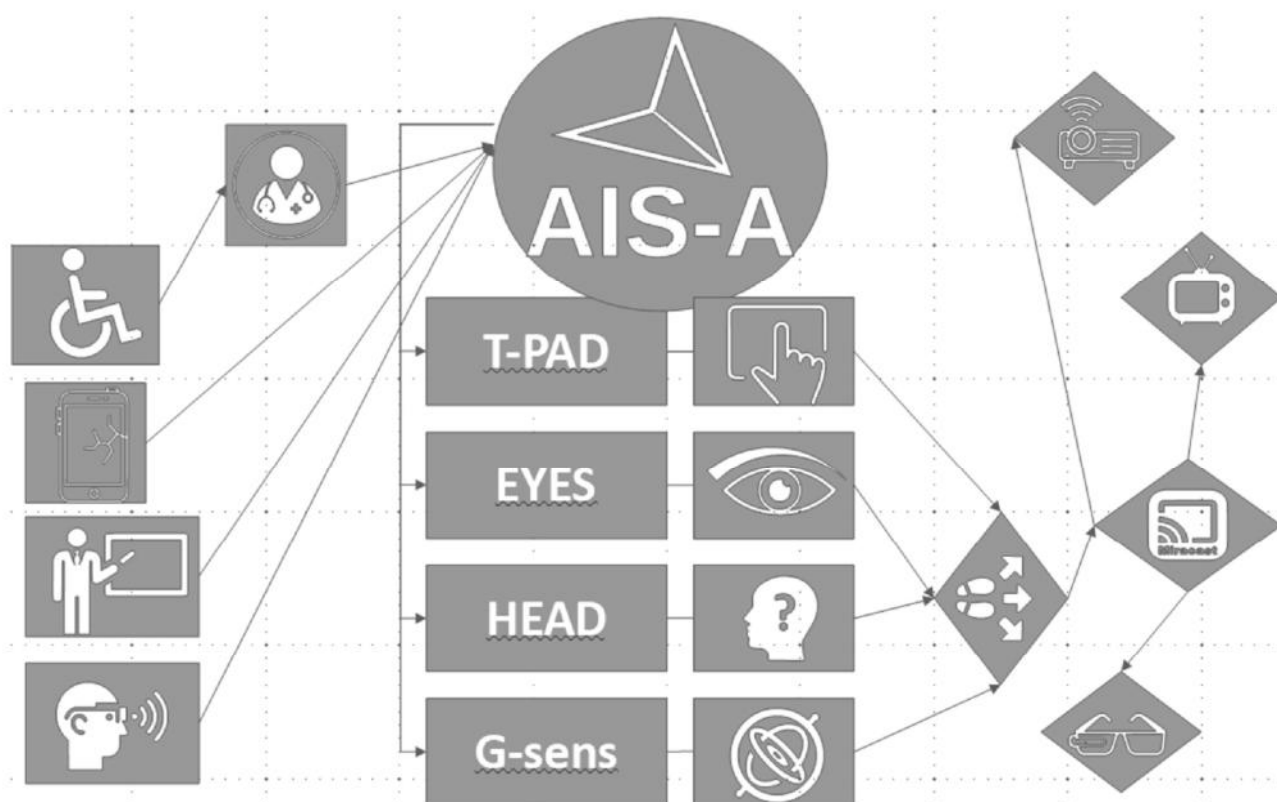


Рисунок 3.12 — Явные сценарии практического использования

Универсальный подход к архитектуре, поддержка различных режимов управления и возможность работы без дополнительных аппаратных средств делают систему востребованной как в бытовых, так и в профессиональных ситуациях.

Один из типичных примеров — ситуация с повреждённым или частично неработающим сенсорным экраном. Пользователь, столкнувшись с тем, что дисплей реагирует только на часть прикосновений, может подключить мышь через OTG-кабель, установить AIS-A и продолжить полноценно использовать

устройство в режиме виртуального курсора. Таким образом, система выполняет не только вспомогательную, но и восстановительную функцию, продлевая срок службы устройства и устраняя необходимость срочного ремонта.

В других случаях AIS-A позволяет использовать смартфон как удалённый интерфейс управления. Например, при подключении устройства по Miracast к очкам XREAL Air 2 пользователь может управлять медиаплеером или браузером с помощью режима T-PAD, не глядя на сам телефон. Благодаря визуальной индикации и отзывчивому курсору взаимодействие остаётся интуитивным и точным. Это особенно актуально для людей, находящихся в лежачем положении, в дороге или использующих вспомогательные дисплеи.

Система также находит применение в мобильной демонстрации контента: торговые представители и специалисты по продукту могут спонтанно проводить презентации с помощью мини-проектора и смартфона. Управление происходит через AIS-A в режиме T-PAD, без необходимости прямого контакта с экраном, что особенно удобно в условиях ограниченного пространства. Ещё один сценарий — использование AIS-A в домашней обстановке, когда, например, подключается телефон к проектору или телевизору, управляя воспроизведением с удобного расстояния, не отвлекаясь на экран телефона.

Таким образом, система AIS-A, разработанная для нужд доступности, демонстрирует высокую универсальность и применимость в реальных жизненных сценариях, выходящих за рамки задач только реабилитационного характера. Это подчёркивает её ценность не только как вспомогательного средства, но и как гибкого адаптивного интерфейса нового поколения.

3.7 Развёртывание и распространение

На момент завершения тестирования приложение AIS-A (Alternative Interaction System for Android) находится на завершающей стадии подготовки к распространению. Все функциональные модули полностью реализованы, проведено комплексное тестирование, и осуществляется оформление документации, необходимой для публикации в официальных и альтернативных каналах дистрибуции.

В рамках стратегии доступности и открытости система AIS-A планирует поддерживать несколько путей распространения.

В приоритетный путь публикация в Google Play Store

Основной целевой канал дистрибуции, позволяющий обеспечить охват широкой аудитории. Приложение будет размещено бесплатно, с полным открытым доступом к функциональности для всех категорий пользователей.

Проводится финальная проверка соответствия требованиям Google, включая: корректное описание разрешений; соответствие требованиям политики

конфиденциальности; безопасность обработки данных; отказ от монетизации и рекламы.

На данный момент используется ручная установка APK-файла. Метод используется на этапе тестирования и внутреннего внедрения. APK-файлы устанавливаются вручную через интерфейс Android, что позволяет тестировать сборки без привязки к маркетплейсу. Этот подход в последующем также будет актуален для организаций, не использующих Google Play по соображениям безопасности или внутренней политики.

Еще один путь – это распространение через альтернативные платформы. Рассматривается возможность публикации на F-Droid и других каталогах с открытым программным обеспечением. Это позволит заинтересованным пользователям, ориентированным на open-source-экосистему, легко получать обновления и проверять исходный код проекта.

Публикация в официальном магазине Google включает следующие этапы:

Создание аккаунта разработчика в Google Play Console и регистрация проекта. Подготовка APK-билда в Android Studio, генерация ключа подписи (keystore) и подпись приложения. Разработка медиа-контента: скриншоты приложения; демонстрационное видео; иконка и баннер для карточки продукта. Подготовка и загрузка метаданных: название, описание; поддерживаемые устройства; категории пользователей. Добавление ссылки на политику конфиденциальности и страницу технической поддержки. Прохождение модерации и проверки на соответствие правилам безопасности, контента и UI Guidelines Google.

3.7.1 Текущий статус

В настоящий момент приложение находится на завершающей стадии подготовки к публикации. Все функциональные компоненты проверены, сопровождающая документация оформляется, ведётся сбор мультимедийных материалов и юридическое оформление политик. После завершения этих процедур AIS-A будет размещено в Google Play Store, открывая доступ к системе hands-free управления Android-устройством для широкой аудитории пользователей, включая людей с ОВЗ.

Таким образом, AIS-A готово к развёртыванию как зрелое, протестированное программное решение, отвечающее техническим требованиям и этическим стандартам современной мобильной разработки. Его публикация станет логическим продолжением проекта, направленного на повышение цифровой доступности, снижение барьеров для пользователей с двигательными нарушениями и широкое внедрение вспомогательных технологий в повседневную жизнь.

3.8 Экономическое обоснование проекта.

3.8.1 Трудозатраты и оценка стоимости разработки

Разработка мобильного приложения AIS-A велась индивидуально в течение 8 месяцев. Средняя загрузка составила около 15 часов в неделю. Таким образом, совокупное количество человеко-часов можно оценить следующим образом: 15 часов в неделю $\times$ 32 недели = 480 человеко-часов

Согласно рыночным ставкам Android-разработчика начального/среднего уровня в Казахстане (в диапазоне 3500–4500 тенге/час), примем среднюю ставку: 480 часов $\times$ 4000 тенге/час = 1 920 000 тенге

Таким образом, стоимость самостоятельной разработки приложения составляет около 1 920 000 тенге (~4300 USD по курсу на май 2025 г.).

Расходы на программно-аппаратную инфраструктуру подсчитать тоже несложно. Для реализации проекта использовались бесплатные инструменты: Android Studio (официальная IDE, лицензия Apache 2.0); Язык программирования Kotlin; ML Kit (работает offline, в рамках допустимого использования для некоммерческих проектов); TarsosDSP — свободная библиотека анализа аудиосигналов; Устройства тестирования — приобретены ранее, отдельного бюджета не требовали. Дополнительные расходы на лицензии, подписки или оборудование отсутствовали.

3.8.2 Потенциальный экономический эффект

Проект AIS-A ориентирован на аудиторию, испытывающую трудности с сенсорным управлением смартфоном. Существующие решения (Tobii, AMAneo, Tecla) либо недоступны для Android, либо требуют дорогостоящего оборудования (от 1500 до 4000 USD), что делает их непригодными для массового использования в странах с низким уровнем доходов.

AIS-A работает на обычных Android-смартфонах и предоставляет 4 режима управления без необходимости покупки дополнительных устройств. Это снижает барьер цифровой доступности и может использоваться: в домашних условиях — без медперсонала; в центрах реабилитации — как бесплатная альтернатива коммерческим системам; в учебных заведениях — как инструмент цифровой инклюзии. Ориентировочная экономия на одного пользователя может составлять от 1 000 до 3 000 USD по сравнению с существующими аппаратными решениями.

3.8.3 Перспективы масштабирования и внедрения

Приложение может распространяться по модели: Open-source / Freemium — базовая версия остаётся бесплатной; Pro-версия — включает функции удалённого мониторинга, синхронизации с ПК, расширенные настройки для специалистов; Интеграция с государственными программами — цифровая инклюзия, помощь инвалидам, удалённое образование.

Возможна подача заявки на грантовое финансирование (например, через стартап-программы, инкубаторы, НПО, акселераторы при поддержке Министерства труда и соцзащиты РК).

Разработка AIS-A показала, что современное hands-free управление Android-устройствами может быть реализовано без дополнительных затрат, на основе открытых технологий. Внедрение подобного решения способно существенно сократить издержки на доступную среду, повысить цифровую самостоятельность пользователей с ограничениями и обеспечить высокий социальный эффект при минимальных инвестициях.

ЗАКЛЮЧЕНИЕ

В рамках выполнения дипломного проекта была разработана, протестирована и практически апробирована программная система AIS-A (Alternative Interaction System for Android) — мобильное приложение для hands-free управления Android-устройствами, ориентированное на пользователей с двигательными ограничениями. Основной задачей являлось создание надёжного, интуитивно понятного и функционально гибкого средства взаимодействия с мобильной системой без использования рук — для людей, которым недоступны традиционные методы управления через сенсорный экран.

Проект охватил все стадии жизненного цикла программного обеспечения: от анализа требований и проектирования архитектуры до реализации, многоуровневого тестирования и подготовки к распространению. Ключевой особенностью стало внедрение модульной архитектуры, позволяющей каждому режиму управления (HEAD, EYES, G-SENS, T-PAD) функционировать как автономный компонент с собственной логикой, интерфейсом и системой параметров. Такой подход обеспечивает масштабируемость, расширяемость и простоту поддержки.

Техническая реализация включила:

- современные интерфейсные решения, соответствующие концепциям Material You и Glassmorphism;

- адаптивную и интуитивную систему пользовательских настроек;

- управление через Broadcast-интенты, позволяющее гибко переключать режимы и параметры;

- алгоритмы фильтрации и сглаживания, обеспечивающие стабильность и точность движения курсора;

- защиту от ложных срабатываний (в том числе при звуковом вводе);

- визуальную и аудиальную обратную связь во всех режимах.

Система прошла модульное, интеграционное и пользовательское тестирование, в том числе с участием людей с различными формами двигательной дисфункции. Эти испытания подтвердили устойчивость работы, адаптивность интерфейсов, практическую применимость решений и выявили направления для дальнейшего улучшения. Результаты тестов позволили оптимизировать параметры взаимодействия, повысить удобство использования и создать более комфортные сценарии для конечного пользователя.

На момент завершения проекта AIS-A находится на завершающей стадии подготовки к публикации в Google Play Store. Завершена проработка соответствия требованиям маркетплейса, оформляется сопровождающая документация, подготавливается медиаконтент. В числе ближайших направлений развития проекта: интеграция дополнительных способов управления (включая голосовые команды, ЭМГ, сигналы дыхания),

взаимодействие с внешними ассистивными устройствами, публикация исходного кода в открытом доступе и развитие проекта в рамках open-source-сообщества.

Разработка AIS-A наглядно показала, что современные технологии hands-free управления могут быть реализованы на базе открытых решений, без необходимости приобретения дорогостоящего оборудования. Такой подход снижает входной порог цифровой доступности, делает мобильные устройства более универсальными и даёт возможность людям с ограниченными возможностями использовать смартфон полноценно — для связи, работы, учёбы и реабилитации. Проект демонстрирует не только техническую состоятельность, но и высокий социальный потенциал, способный внести вклад в развитие доступной среды при минимальных вложениях.

Итог: AIS-A представляет собой зрелое, практически ориентированное программное решение, которое может быть использовано как в научных разработках, так и в прикладных медицинских, социальных и образовательных сценариях. Приложение может служить как инструмент абилитации и реабилитации [17], так и платформой для развития пользовательских интерфейсов нового поколения — доступных каждому.

Таблица 8 – Термины и сокращения, используемые в работе

ОДА	опорно-двигательный аппарат
ВОЗ	Всемирная организация здравоохранения
ОВЗ	ограниченные возможностями здоровья
Спастический тетрапарез	это комплекс двигательных нарушений верхних и нижних конечностей, возникающий в следствие различных неврологических заболеваний.
Афазия	отсутствие или расстройство уже сформировавшейся речи с нарушением производства или восприятия речи при сохранении слуха.
Нистагм	непроизвольные колебательные движения глаз высокой частоты
БОС	биологическая обратная связь

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Журналы

- 1 Laureys S., Owen A.M., Schiff N.D. The Locked-in Syndrome: What is it like to be conscious but paralyzed and voiceless? // *Progress in Brain Research*. – 2009. – Vol. 177. – P. 131–140.
- 2 Cramer S.C., Sur M., Dobkin B.H., et al. Harnessing neuroplasticity for clinical applications // *Brain*. – 2011. – Vol. 134, № 6. – P. 1591–1609.
- 3 Wijdicks E.F.M. The Diagnosis of Locked-In Syndrome: An Evidence-Based Review // *Neurology*. – 2002. – Vol. 58, № 6. – P. 853–857.
- 3 Ortiz J.F., et al. Locked-in Syndrome: A Systematic Review of Long-Term Management and Prognosis // *Cureus*. – 2021. – Vol. 13, № 3. – P. e13937.
- 4 Canny E., et al. The Feasibility of Combining Communication BCIs with FES for Individuals with Locked-In Syndrome // *arXiv preprint arXiv:2306.03159*. – 2023.
- 5 Castaldi E., Lunghi C., Morrone M.C. Neuroplasticity in Adult Human Visual Cortex // *arXiv preprint arXiv:1905.02405*. – 2019.
- 6 Zimmerman M. Devices: Development and Direction // *American Journal of Occupational Therapy*. – 1960. – Vol. 14, № 6. – P. 273–279.
- 7 Zhang Y., Chen S., Fan L., et al. Automated and Context-Aware Repair of Color-Related Accessibility Issues for Android Apps // *arXiv preprint arXiv:2308.09029*. – 2023.
- 8 Krishnavajjala A., Mansur S.H., Jose J., Moran K. MotorEase: Automated Detection of Motor Impairment Accessibility Issues in Mobile App UIs // *arXiv preprint arXiv:2403.13690*. – 2024.
- 9 Su Y., Chen C., Wang J., et al. The Metamorphosis: Automatic Detection of Scaling Issues for Mobile Apps // *arXiv preprint arXiv:2212.04388*. – 2022.

Книги

- 1 World Health Organization. World Report on Disability. – Geneva: WHO Press, 2011. – 352 p. – URL: <https://www.who.int/publications/i/item/9789241564182> (дата обращения: 19.05.2025).
- 2 Методические указания по написанию и оформлению дипломного проекта. – Алматы: КазННТУ им. К.И. Сатпаева, 2024. – 32 с.
- 3 International Classification of Functioning, Disability and Health (ICF). – Geneva: World Health Organization, 2001. – 299 p.
- 4 Шапиро И.С., Ильин Е.П. Ассистивные технологии и средства реабилитации: учебное пособие. – М.: Академия, 2019. – 256 с.
- 5 Грачёв Г.В. Технологии абилитации и реабилитации в цифровой среде. – СПб.: Питер, 2021. – 304 с.

- 6 Kyberd P. Making Hands: A History of Artificial Hands and Their Makers. – London: Springer, 2021. – 320 p.
- 7 Cooper R.A., et al. Rehabilitation Engineering Applied to Mobility and Manipulation. – Boca Raton: CRC Press, 1995. – 400 p.
- 8 Коржинский С.Н. Разработка мобильных приложений. – М.: КноРус, 2025. – 432 с.
- 9 Нужный А.М., Гребенникова Н.И. Разработка мобильных приложений на языке Java с использованием Android Studio: учебное пособие. – Воронеж: ВГТУ, 2020. – 120 с.
- 10 Меджидов З.У. Разработка мобильных приложений: учебное пособие. – Махачкала: ДГУНХ, 2019. – 77 с.
- 11 Виноградов Д.В. Разработка мобильных приложений и облачные сервисы: учебное пособие. – Владимир: ВлГУ, 2022. – 150 с.

Электронные ресурсы

- 1 Android Developers. Build Accessible Apps [Электронный ресурс]. – URL: <https://developer.android.com/guide/topics/ui/accessibility> (дата обращения: 24.05.2025).
- 2 Google Accessibility. Resources for Developers and Publishers [Электронный ресурс]. – URL: <https://www.google.com/accessibility/for-developers/> (дата обращения: 24.05.2025).
- 3 Level Access. Assistive Technology for Android [Электронный ресурс]. – URL: <https://www.levelaccess.com/assistive-technology-for-android/> (дата обращения: 24.05.2025).
- 4 Reading Rockets. Assistive Technology That's Built Into Mobile Devices [Электронный ресурс]. – URL: <https://www.readingrockets.org/topics/assistive-technology/articles/assistive-technology-thats-built-mobile-devices> (дата обращения: 24.05.2025).
- 5 HandTalk. Android Assistive Technology: Check Out Free Services and Apps! [Электронный ресурс]. – URL: <https://www.handtalk.me/en/blog/android-assistive-technology/> (дата обращения: 24.05.2025).
- 6 Verywell Health. Adaptive Technology for Improving Life With MS [Электронный ресурс]. – URL: <https://www.verywellhealth.com/ms-technology-5225481> (дата обращения: 24.05.2025).
- 7 Teen Vogue. For Disabled People, Accessible Technology Is Finally Getting Better [Электронный ресурс]. – URL: <https://www.teenvogue.com/story/for-disabled-people-accessible-technology-is-finally-getting-better> (дата обращения: 24.05.2025).
- 8 Axios. Houston Museum of Natural Science Introduces App for People with Low Vision [Электронный ресурс]. – URL:

<https://www.axios.com/local/houston/2025/02/11/houston-museum-of-natural-science-app-low-vision> (дата обращения: 24.05.2025).

9 Accessibility Spark. Top 10 Types of Assistive Technology Devices [Электронный ресурс]. – URL: <https://accessibilityspark.com/types-of-assistive-technology/> (дата обращения: 24.05.2025).

10 Accessibility Checker. The Different Types of Assistive Technologies [Электронный ресурс]. – URL: <https://www.accessibilitychecker.org/blog/types-of-assistive-technologies/> (дата обращения: 24.05.2025).

Accessibility Checker

11 PDQ. 8 Assistive Technologies for Your Workplace [Электронный ресурс]. – URL: <https://www.pdq.com/blog/assistive-technologies-for-your-workplace/> (дата обращения: 24.05.2025).

12 Minnesota.gov. Types of Assistive Technology [Электронный ресурс]. – URL: <https://mn.gov/admin/at/getting-started/understanding-at/types/> (дата обращения: 24.05.2025).

13 Android Developers. Accessibility Overview [Электронный ресурс]. – URL: <https://developer.android.com/guide/topics/ui/accessibility> (дата обращения: 24.05.2025).

14 Material Design 3. Accessibility Overview [Электронный ресурс]. – URL: <https://m3.material.io/foundations/overview/assistive-technology> (дата обращения: 24.05.2025).

Material Design

Приложение А

Техническое задание

А.1.1 Техническое задание на разработку мобильного приложения для альтернативного hands-free управления Android-устройством (AIS-A)

Настоящее техническое задание распространяется на разработку мобильного приложения для альтернативного hands-free управления Android-устройством (AIS-A). Предполагается что использовать его будут специалисты в области реабилитации, абилитации, пользователи с ОВЗ, пользователи без ОВЗ.

А1.1.1 Основание для разработки

Основанием для разработки дипломного проекта является учебный план направления подготовки «Информационные технологии» и утверждённая тема дипломной работы: «Разработка мобильного приложения для альтернативного hands-free управления Android-устройством (AIS-A)».

А.1.1.2 Назначение

Разрабатываемое приложение AIS-A предназначено для пользователей с ограниченными двигательными возможностями, а также для ситуаций, когда классический сенсорный ввод невозможен (повреждение экрана, использование проектора, необходимость hands-free). Приложение предоставляет четыре альтернативных способа управления Android-устройством: через тачпад (T-PAD), движения головы (HEAD), направление взгляда и мимику (EYES), а также с использованием акселерометра (G-sens).

А.1.1.3 Требования к функциональным характеристикам

- Поддержка четырёх режимов управления: T-PAD, HEAD, EYES, G-sens;

- Универсальный виртуальный курсор;
- Настраиваемые триггеры: задержка, звук, открытие рта, двойной тап;
- Калибровка координат взгляда и центра головы;
- Возможность переключения между режимами без перезапуска приложения;
- Использование Android AccessibilityService и работы поверх всех окон.

A.1.1.4 Нефункциональные требования:

- Поддержка Android 8.0 и выше;
- Энергоэффективность и оптимизация потребления ресурсов;
- Интуитивный интерфейс и адаптация под разные размеры экрана;
- Стабильность и отказоустойчивость при переключении режимов.

A.1.1.5 Требования к надежности

Приложение должно функционировать на смартфонах и планшетах под управлением ОС Android версии 8.0 и выше, с фронтальной камерой и встроенными датчиками (акселерометр, микрофон). Должно поддерживаться управление как в портретной, так и в ландшафтной ориентации.

A.1.1.6 Требования к составу и параметрам технических средств

- Смартфон или планшет под управлением Android 8.0 и выше;
- Наличие фронтальной камеры (от 2 Мп);
- Наличие акселерометра и микрофона;
- Оперативная память от 3 ГБ.

A.1.1.7 Требования к программной документации

Программная документация должна включать пояснительную записку, содержащую: описание архитектуры, алгоритмы работы сервисов, примеры использования, инструкции по установке и настройке, а также результаты тестирования.

Приложение Б

Листинг (код) программы

Основные фрагменты реализации TouchpadAccessibilityService

1. Подключение сервиса и инициализация курсора:

```
override fun onServiceConnected() {
    super.onServiceConnected()

    windowManager      =      getSystemService(WINDOW_SERVICE)      as
WindowManager
    val metrics = DisplayMetrics()
    windowManager.defaultDisplay.getMetrics(metrics)

    screenWidth = metrics.widthPixels
    screenHeight = metrics.heightPixels

    cursorView = ImageView(this).apply {
        setImageBitmap(defaultCursorBitmap)
    }

    layoutParams = WindowManager.LayoutParams(
        cursorSize, cursorSize,
        WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY,
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
        PixelFormat.TRANSLUCENT
    )

    windowManager.addView(cursorView, layoutParams)
}
```

Фрагмент отвечает за отображение виртуального курсора поверх всех окон Android и его позиционирование.

2. Обработка касаний для перемещения курсора:

```
private val overlayTouchListener = View.OnTouchListener { _, event ->
    when (event.actionMasked) {
        MotionEvent.ACTION_MOVE -> {
            val dx = event.rawX - lastTouchX
            val dy = event.rawY - lastTouchY
            moveCursorIfNeeded(dx, dy)
        }
    }
}
```

```

MotionEvent.ACTION_UP -> {
    handleTap()
}
}
true
}

```

Этот обработчик отслеживает движение пальца и перемещает курсор соответствующим образом.

3. Обработка одиночного и двойного тапа:

```

private fun handleTap() {
    val currentTime = System.currentTimeMillis()
    if (currentTime - lastTapTime < doubleTapPause) {
        tapCount++
    } else {
        tapCount = 1
    }
    lastTapTime = currentTime

    if (tapCount == 2) {
        performDoubleTapAction()
        tapCount = 0
    } else {
        handler.postDelayed({
            performSingleTapAction()
        }, tapResponseTime)
    }
}

```

Метод определяет, является ли касание одиночным или двойным тапом, и запускает соответствующее действие.

4. Выполнение жеста "тап" в заданной координате (имитация нажатия):

```

@RequiresApi(Build.VERSION_CODES.N)
private fun performClickGesture(x: Float, y: Float) {
    val path = Path().apply { moveTo(x, y) }

    val gesture = GestureDescription.Builder()
        .addStroke(GestureDescription.StrokeDescription(path, 0, 100))
        .build()

    dispatchGesture(gesture, null, null)
}

```

Функция имитирует нажатие на экране в нужной точке — это ключ к реализации взаимодействия с UI.

5. Пример команды изменения размера курсора через широковещательное сообщение:

```
private val cursorSettingsReceiver = object : BroadcastReceiver() {  
    override fun onReceive(context: Context?, intent: Intent?) {  
        if (intent?.action == "UPDATE_CURSOR_SIZE") {  
            val newSize = intent.getIntExtra("NEW_CURSOR_SIZE", 40)  
            setCursorSize(newSize)  
        }  
    }  
}
```

Позволяет менять поведение курсора "на лету" по команде из пользовательского интерфейса.

2.1.2 Основные фрагменты реализации HeadAccessibilityService

Режим управления головой задействует камеру, ML Kit Face Detection, а также может использовать звук в качестве триггера для тапа.

Захват координат и калибровка

Для определения направления взгляда/движения головы используется анализ координат центра лица:

```
val face = faces[0]  
val faceCenterX = face.boundingBox.centerX().toFloat()  
val faceCenterY = face.boundingBox.centerY().toFloat()
```

Перед началом работы запускается калибровка, в ходе которой фиксируется центр лица:

```
val avgX = calibrationSamples.map { it.first }.average().toFloat()  
val avgY = calibrationSamples.map { it.second }.average().toFloat()  
calibrationData["centerX"] = avgX  
calibrationData["centerY"] = avgY
```

Управление курсором

Положение курсора рассчитывается на основе отклонения головы от калиброванного центра и масштабируется:

```
val offsetX = -(rawX - centerX)  
val offsetY = (rawY - centerY)  
val finalX = screenW / 2f + offsetX * scaleFactor  
val finalY = screenH / 2f + offsetY * scaleFactor
```

Затем курсор плавно перемещается с учётом экспоненциального сглаживания и псевдо-Калмана.

Выполнение жестов

Для нажатий используется метод `dispatchGesture()`:

```
val path = Path().apply { moveTo(x, y) }  
val gesture = GestureDescription.Builder()  
.addStroke(GestureDescription.StrokeDescription(path, 0, 100))  
.build()  
dispatchGesture(gesture, callback, null)
```

Двойной тап может вызывать действия `BACK`, `HOME` или `CLOSE_APP`.

4.2 «Зона предтапа» (`confirm-marker`) защита от ложных тапов прежде чем `HeadAccessibilityService` окончательно нажмёт на экран, он выводит прозрачный кружок-маркер, просит курсор выйти из него и вернуться назад.

Пока этого не произошло — клика не будет.

Если пользователь надумал передумать, маркер сам исчезнет по тайм-ауту.

```
private fun processTap() {
```

```
...
```

```
showConfirmMarker(cursorLayoutParams.x, cursorLayoutParams.y)
```

```
}
```

`processTap()` вызывается двумя механизмами -- `idle-tap` (когда курсор стоит дольше

`idleZoneDurationMs`) и `auto-tap`.

Для звукового клика (`clickMethod == 1`) этот шаг пропускается – звук уже был подтверждением.

```
markerView = View(this).apply {  
    setBackgroundResource(R.drawable.marker_bg)    // жёлтый кружок с  
    обводкой
```

```
}
```

```
windowManager.addView(markerView, lp)
```

```
pendingTap = true    // «клик запрошен»
```

```
markerArmed = false    // пока не вышли за круг — не «вооружён»
```

```
markerShownAt = now()
```

```
confirmHandler.postDelayed({ removeMarker() }, confirmTimeoutIdleMs)
```

Логика в `checkConfirmHit()` – «ВЫЙТИ → ВОЙТИ»

```
val dist = distance(cursor, markerCenter)
```

```
if (!markerArmed && dist > markerRadius) {
```

```
    markerArmed = true    // шаг 1: вышли ⇒ маркер «вооружён»
```

```
    extendTimeout(confirmTimeoutArmedMs)
```

```
    return
```

```
}
```

```
if (markerArmed && dist <= markerRadius) {
```

```
performClickGesture(...)    // шаг 2: вернулись внутрь ⇒ настоящий клик
removeMarker()
}
```

Реализация "тапа по звуку"

Если выбран метод `clickMethod = 2`, система слушает микрофон, сравнивает полученный спектр с эталонным:

```
val similarity = computeSpectralSimilarity(refSpectrum, liveSpectrum)
if (similarity >= 0.80) {
    performClickAtPosition()
}
```

2.1.3 Интерфейс и настройки

Каждый сервис получает параметры из `SharedPreferences` и слушает команды от интерфейса через `BroadcastReceiver`, например:

```
"com.bignerdranch.android.mycursorcontrol.UPDATE_CURSOR_SPEED" ->
{
    val newSpeed = intent.getFloatExtra("NEW_CURSOR_SPEED", cursorSpeed)
    setCursorSpeed(newSpeed)
}
```

Сервис на «лету» обновляет курсор, скорость, способ клика, изображение и зоны действия, не требуя перезапуска.

Основные фрагменты реализации `EyeAccessibilityService`.

Сервис `EyeAccessibilityService` реализует hands-free управление курсором с использованием движений глаз и мимики. Он использует модуль `CameraX` и `ML Kit Face Detection` для захвата изображения с фронтальной камеры и распознавания ключевых точек лица.

Сервис поддерживает три основных метода взаимодействия:

Фиксация взгляда: положение курсора рассчитывается на основе координат глаз. Реализованы сглаживание, предсказание и фильтрация траектории движения курсора с использованием экспоненциального сглаживания и псевдо-Калмана.

Открытие рта: клик срабатывает при открывании рта на определённое количество кадров подряд. Перед этим пользователь проходит этап калибровки (собираются значения с закрытым и открытым ртом).

Задержка на месте: при неподвижности курсора в течение заданного времени (по умолчанию 2 секунды), включается таймер автотапа. При

сохранении стабильности курсора и низкой скорости взгляда, происходит автоматический клик.

Дополнительно реализована поддержка:

Динамической калибровки (глаза и рот)

Прилипания к элементам интерфейса

Индикации и визуальной обратной связи

Триггер-зон по краям экрана (например, "назад", "домой", "скроллинг")

Двойного тапа (в зависимости от настроек — действие «назад», «выход» и др.)

Сервис адаптируется под выбранный метод клика (взгляд, рот, звук) и обеспечивает удобство управления без использования рук.

EyeAccessibilityService: управление глазами, клик ртом и автотап

Подключение камеры и распознавание лица

```
val options = FaceDetectorOptions.Builder()
```

```
.setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_FAST
```

```
)
```

```
.setLandmarkMode(FaceDetectorOptions.LANDMARK_MODE_ALL)
```

```
.setContourMode(FaceDetectorOptions.CONTOUR_MODE_ALL)
```

```
.build()
```

```
val detector = FaceDetection.getClient(options)
```

Используется библиотека ML Kit Face Detection для определения координат глаз и рта [9] на каждом кадре, поступающем с фронтальной камеры через CameraX.

Отслеживание взгляда [9] и перемещение курсора

```
val leftEye = face.getLandmark(FaceLandmark.LEFT_EYE)?.position
```

```
val rightEye = face.getLandmark(FaceLandmark.RIGHT_EYE)?.position
```

```
val rawEyeX = (leftEye.x + rightEye.x) / 2f
```

```
val rawEyeY = (leftEye.y + rightEye.y) / 2f
```

```
val normX = ((rawEyeX - minX) / (maxX - minX)).coerceIn(0f, 1f)
```

```
val normY = ((rawEyeY - minY) / (maxY - minY)).coerceIn(0f, 1f)
```

```
val screenX = (1f - normX) * screenWidth
```

```
val screenY = normY * screenHeight
```

вычисляется средняя координата между глазами, которая затем нормализуется и масштабируется под экран устройства. Это позволяет перемещать курсор в зависимости от направления взгляда пользователя.

Псевдо-Калманов фильтр для сглаживания

```
val smoothX = (1 - alpha) * prevX + alpha * screenX
```

```
val smoothY = (1 - alpha) * prevY + alpha * screenY
```

используется экспоненциальное сглаживание, чтобы компенсировать дрожание взгляда и сделать движение курсора более стабильным и предсказуемым.

Клик по открытию рта

```
val mouthDist = getMouthHeight(face)
if (mouthDist > mouthThreshold && !hasTappedThisOpen) {
    hasTappedThisOpen = true
    processTap()
}
```

если выбран способ клика "по открытию рта", сервис сравнивает текущее расстояние между губами с пороговым значением, заданным при калибровке. При превышении — выполняется тап.

Автотап по неподвижности

```
if (eyeSpeed < 50 && distanceMoved < threshold && timeStationary > 2000) {
    startAutoTapCountdown(x, y)
}
```

если взгляд пользователя неподвижен более 2 секунд, запускается обратный отсчёт, по завершении которого автоматически совершается клик — удобно для пользователей с ограниченной подвижностью.

Выполнение тапа через AccessibilityService

```
val path = Path().apply { moveTo(x, y) }
val gesture = GestureDescription.Builder()
    .addStroke(GestureDescription.StrokeDescription(path, 0, 100))
    .build()
```

```
dispatchGesture(gesture, null, null)
```

сервис имитирует касание в заданной точке экрана, используя доступные средства AccessibilityService, без необходимости прямого управления экраном.

Основные фрагменты реализации GsensorAccessibilityService.

Инициализация акселерометра

```
sensorManager = getSystemService(Context.SENSOR_SERVICE) as
SensorManager
    accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
```

Получаем SensorManager и основной акселерометр.

Регистрация слушателя акселерометра

```
sensorEventListener = object : SensorEventListener {
    override fun onSensorChanged(event: SensorEvent) {
        if (event.sensor.type == Sensor.TYPE_ACCELEROMETER) {
            handleAccelerometer(event)
        }
    }
}
override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {}
```

```

    }
    sensorManager.registerListener(sensorEventListener, accelerometer,
SensorManager.SENSOR_DELAY_GAME)
    Регистрируем SensorEventListener, вызывающий handleAccelerometer().
    Обработка движения (handleAccelerometer())
    val dx = -x * sensitivity
    val dy = y * sensitivity

    if (isCursorLocked) {
    if (movementStrength > unlockThreshold) {
    isCursorLocked = false
    } else return
    }
    Обновление позиции курсора
    updateCursorPositionWithSnapping(lastX.toInt(), lastY.toInt())
    Курсор перемещается и при необходимости прилипает к ближайшему
ClickableElement.
    Механизм прилипания
    if (snappedElement == null) {
    val nearest = findNearestElement(x, y)
    if (dist < dynamicSnapR) {
    snappedElement = nearest
    finalX = nearest.centerX
    finalY = nearest.centerY
    }
    }

    Если курсор близко к кликабельному элементу

    Тап по неподвижности (Idle)
    if (currentTime - lastMovementTime >= cursorIdleThreshold) {
    performClickAtPosition()
    }
    Если курсор долго не двигается — совершается тап по текущей позиции.
    Управление звуком
    if (clickMethod == 2 && hasAudioPermission) {
    startListeningForSound()
    }
    Поддерживается активация тапа по звуку, если включен соответствующий
режим и получено разрешение.

    Trigger-зоны по краям экрана
    handleZoneEnter(CursorAction.SCROLL_DOWN)

```

Попадание курсора в край экрана активирует определённое действие (Back, Scroll, Home и др.) после небольшой задержки с прогресс-баром.

Поддержка двойного тапа

```
if (tapCount == 2) {  
    performDoubleTapAction()  
}
```

Реализована логика: одинарный тап → обычный клик, двойной тап → back, close или другое действие из настроек.

Поддержка смены настроек через BroadcastReceiver

```
"com.bignerdranch.android.mycursorcontrol.UPDATE_CURSOR_SPEED" ->  
{ ... }
```

Настройки курсора можно динамически обновлять — скорость, размер, способ клика, видимость триггер-зон и пр.

Вспомогательные элементы:

highlightView — визуальная подсветка тапа.

snapProgressView — прогресс-бар прилипания.

zoneIndicatorView — прогресс-бар действия в зоне.

clickableElements — список ближайших к курсору кликабельных элементов.

2.3 Структура данных и хранение настроек

Данные о текущем состоянии и предпочтениях пользователя сохраняются и извлекаются двумя основными способами:

1. SharedPreferences — Локальное хранилище пользовательских настроек

Назначение: для хранения примитивных настроек: скорость курсора, размер, способ клика, изображение курсора, включение триггер-зон и др.

Пример использования:

// Получение SharedPreferences

```
sharedPreferences = getSharedPreferences("cursor_prefs", MODE_PRIVATE)
```

// Чтение значений

```
cursorSize = sharedPreferences.getInt("cursor_size", 40)
```

```
cursorSpeed = sharedPreferences.getFloat("cursor_speed", 1.0f)
```

```
clickMethod = sharedPreferences.getInt("click_method", 0)
```

// Сохранение значений

```
sharedPreferences.edit().putFloat("cursor_speed", newSpeed).apply()
```

Применение:

В `GSensorAccessibilityService` — настройки скорости, размера, способа клика и др.

В `TouchpadAccessibilityService`, `EyeAccessibilityService` — общие параметры курсора.

Динамически обновляются через `BroadcastReceiver`, что позволяет интерфейсу сохранять состояние даже при перезапуске сервисов.

Преимущества: Простота и мгновенный доступ к данным. Подходит для небольших, часто используемых параметров

XML-файлы конфигурации интерфейса — Статическая настройка интерфейсов

Назначение: определение визуального и интерактивного интерфейса для каждого режима управления.

Пример — `accessibility_service_eye.xml`

```
<androidx.constraintlayout.widget.ConstraintLayout>
```

```
<!-- Камера (предпросмотр) -->
```

```
<androidx.camera.view.PreviewView
```

```
android:id="@+id/previewView"
```

```
... />
```

```
<!-- Точка и текст калибровки -->
```

```
<TextView android:id="@+id/calibrationText" ... />
```

```
<View android:id="@+id/calibrationPoint" ... />
```

```
<!-- Прогресс и управление -->
```

```
<ProgressBar android:id="@+id/calibrationProgress" ... />
```

```
<Button android:id="@+id/nextStepButton" ... />
```

```
<Button android:id="@+id/trackButton" ... />
```

```
<Button android:id="@+id/recordSoundButton" ... />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Применение:

Для каждого модуля управления (EYES, G-Sensor, Touchpad) используется отдельный XML-файл.

Определяет UI компоненты: кнопки, текст, элементы калибровки и т.д.

XML-файл подгружается в соответствующем Activity или Service, и связывается с обработчиками через `findViewById()`.

Примеры взаимодействия между `SharedPreferences` и XML-интерфейсами:

Пример: при изменении размера курсора в настройках:

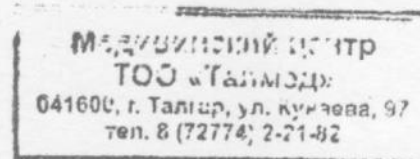
XML-интерфейс вызывает `setCursorSize()` по кнопке или `SeekBar`.

Метод сохраняет размер в `SharedPreferences`:

```
fun setCursorSize(newSize: Int) {  
    cursorSize = newSize  
    sharedPreferences.edit().putInt("cursor_size", newSize).apply()  
    loadCursorImage()  
}
```

Загружается изображение и пересоздается `ImageView` курсора с новым размером.

Приложение В.
Акт о внедрении результатов дипломного проектирования



АКТ О ВНЕДРЕНИИ

Мы, нижеподписавшиеся, составили настоящий акт о том, что результаты дипломной работы студента университета им. К. И. Сатпаева. КазНУТУ Эйсмонта Вячеслава Владимировича, выполненной по теме:

«Разработка мобильного инструмента для альтернативного взаимодействия с Android-смартфонами и планшетами в ограниченных условиях», были внедрены в рамках практической деятельности индивидуального предпринимателя Эйсмонта В.В. при проведении занятий абилитации с пациентами, имеющими двигательные ограничения, а также при проведении занятий абилитации в условиях стационара на базе медицинского центра ТОО «ТАЛМЕД».

Программная система AIS-A, созданная в рамках дипломного проекта, использовалась как вспомогательное средство для пациентов проходящих процесс абилитации, для hands-free взаимодействия с мобильными устройствами. В ходе применения была подтверждена практическая значимость решения, удобство его настройки и эффективность взаимодействия для пользователей с ограниченными возможностями.

Научные результаты, полученные в рамках дипломной работы, нашли практическое применение и используются в текущей профессиональной деятельности.

Дата: « 1 » февраля 2025 г.

Директор ТОО «ТАЛМЕД»
Сулейменов Б.С.

БИН 001240014603
Адрес: Алматинская Область, Г.Талгар,
Проспект Д.Қонаев, Дом 97

Подпись:



ИП «Реабилитолог Эйсмонт»

ИИН 900804301007
Адрес: Г.Алматы, У. Варламова 1/3,
корпус В, офис 99
КР "ReEis"

Подпись:



печать