

УТВЕРЖДЕН
RU.17701729.10.03-01 81 01-1-ЛУ

**ПРОГРАММА, ОЦЕНИВАЮЩАЯ СПОСОБНОСТЬ СЛЕДИТЬ ЗА
НЕСКОЛЬКИМИ ДВИЖУЩИМИСЯ ОБЪЕКТАМИ ОДНОВРЕМЕННО
«MULTITRACKER»**

Пояснительная записка

RU.17701729.11.01-01 81 01-1

Листов 13

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
1. ВВЕДЕНИЕ	3
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	4
2.1. Функциональное назначение	4
2.2. Эксплуатационное назначение.....	4
2.3. Краткая характеристика области применения	4
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....	5
3.1. Постановка задачи	5
3.2. Мобильные устройства.....	5
3.3. Операционная система Android	6
3.4. Язык программирования Kotlin	6
3.5. Архитектура системы	6
3.5.1. Общее архитектурное решение.....	6
3.5.2. LiveData	7
3.5.3. Navigation API.....	8
3.5.4. База данных.....	8
3.5.5. Сопрограммы	9
3.5.6. Storage Access Framework.....	9
3.5.7. TimeAnimator	10
3.6. Описание и обоснование метода организации входных и выходных данных	10
3.6.1. Организация входных данных.....	10
3.6.2. Организация выходных данных	10
ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	11
ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ	12

1. ВВЕДЕНИЕ

Наименование темы разработки: «Программа, оценивающая способность следить за несколькими движущимися объектами одновременно “MultiTracker”».

Условное обозначение темы разработки: «MultiTracker»

Основанием для разработки является учебный план программы «Мобильная разработка на Kotlin».

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Согласно ВОЗ, за 2022 год в России у более чем 700 детей и подростков (в возрасте от 0 до 19 лет) диагностировали рак мозга, таким образом он занимает второе место по частоте новых случаев среди всех видов рака в нашей стране^[1]. Несмотря на то, что за последние несколько лет в России достигнуты значительные успехи в излечении нейроонкологических заболеваний^[2], после длительного, агрессивного лечения пациенты нуждаются в реабилитации для восстановления когнитивных функций мозга^[3]. Программа «MultiTracker» предоставляет возможность восстанавливать навык слежения за несколькими объектами одновременно, посредством тренировки и постепенного усложнения задачи, а также собирать статистику, которая позволит реабилитологу отслеживать динамику развития данной способности у пациента.

2.2. Эксплуатационное назначение

Программа «MultiTracker», оценивающая способность следить за несколькими движущимися объектами одновременно, предназначена для использования в реабилитационных центрах, занимающихся восстановлением когнитивных функций людей (преимущественно детей и подростков), перенесших рак мозга.

2.3. Краткая характеристика области применения

Программа «MultiTracker» – приложение, разрабатываемое под руководством НИИ развития мозга и высших достижений РУДН, с целью диагностики и реабилитации пациентов (преимущественно детей и подростков), перенесших нейроонкологические заболевания. Под присмотром реабилитологов пациенты проходят тестирование на способность следить за несколькими движущимися объектами одновременно и, тренируясь, восстанавливают и развивают этот навык. Реабилитологи, в свою очередь, получают возможность просматривать статистику по каждому из пациентов, оценивать по ней их восстановление и составлять дальнейший план реабилитации.

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи

Задача: реализовать программный продукт, позволяющий проводить тестирование и оценивать способность следить за несколькими движущимися объектами одновременно, а также получать статистику по пройденным тестированиям. На рис. 1 представлена диаграмма прецедентов, отображающая типичный способ взаимодействия пользователей с системой.

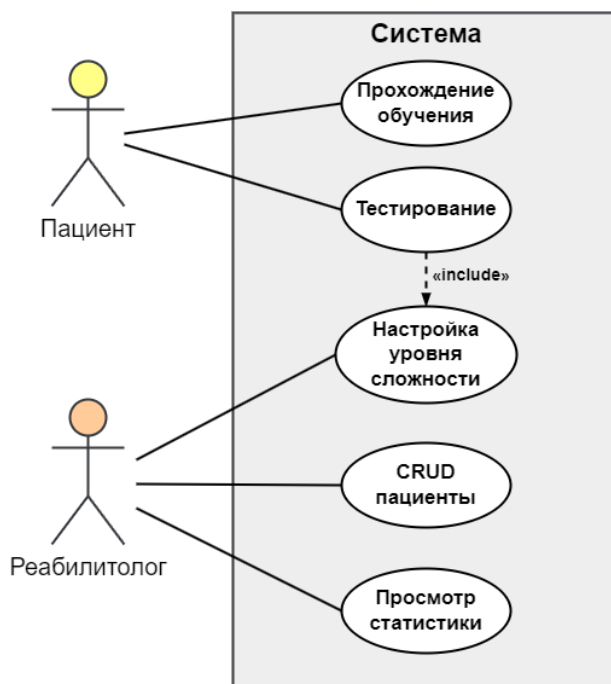


Рисунок 1 – Модель прецедентов

3.2. Мобильные устройства

В ходе анализа области применения, было принято решение о написании программы для мобильных устройств. Так как предполагается, что с системой могут взаимодействовать дети дошкольного возраста, которым может быть сложно взаимодействовать с системой по средством компьютерной мыши и(или) клавиатуры. В тоже время, взаимодействие при помощи сенсорного экрана интуитивно понятно и не требует особых навыков. Также мобильные устройства имеют небольшой размер и вес, что тоже делает их использование детьми более удобным и доступным.

В первую очередь разработка направлена на планшетные компьютеры, так как, в отличии от смартфонов, они имеют большой дисплей, что увеличивает площадь касания и способствует уменьшению нагрузки на зрение детей.

3.3. Операционная система Android

В качестве операционной системы была выбрана ОС Android, так как она является наиболее популярной мобильной операционной системой в России с долей рынка около 72%^[4], что делает программу более доступной.

3.4. Язык программирования Kotlin

В качестве языка программирования был выбран Kotlin. Это современный статически типизированный язык программирования, используемый более чем 60% профессиональных разработчиков Android.^[5]

Выбор был сделан в связи со следующими факторами:

- 1) Kotlin полностью поддерживает разработку под Android
- 2) Kotlin лаконичен, он позволяет писать удобный для чтения код, уменьшает использование шаблонного кода
- 3) Kotlin позволяет писать более безопасный код.
Благодаря разделению типов на nullable и не nullable, включенному в систему типов Kotlin, он помогает избежать исключений NullPointerExceptions. Вероятность сбоя приложений Android, использующих Kotlin, на 20% ниже.^[5]
- 4) Современный инструментальный пользовательского интерфейса Android построен на Kotlin, что позволяет быстро создавать пользовательский интерфейс с помощью мощных и интуитивно понятных API.
- 5) Структурированный параллелизм
Сопрограммы Kotlin оптимизируют асинхронное программирование, делая обычные задачи, такие как сетевые вызовы и обновления баз данных, простыми и производительными.

3.5. Архитектура системы

3.5.1. Общее архитектурное решение

Разработка велась в соответствии с рекомендованной для приложений под Android архитектурой^[6].

Система состоит из двух слоев (рис. 2):

- 1) UI Layer – отвечает за взаимодействие с пользователем
Состоит из элементов пользовательского интерфейса и ViewModel-ей (классов, которые содержат необходимую логику для задания состояний элементам пользовательского интерфейса).
- 2) Data Layer – обращается к источнику данных
Состоит из классов, обеспечивающих общение с источником данных, и репозиториями, отвечающих за связь этих классов с остальным приложением).

Диаграмма пакетов представлена на рис.3.

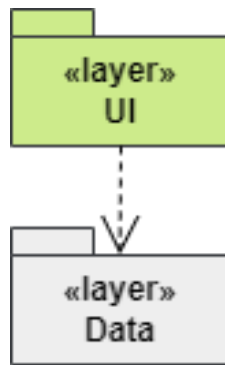


Рисунок 2 - Слои системы

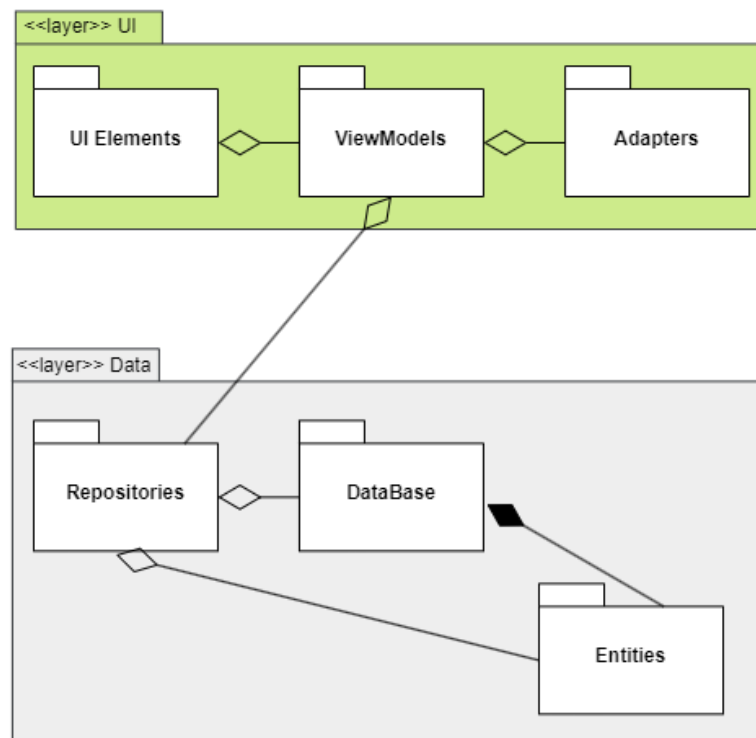


Рисунок 3 – Диаграмма пакетов

3.5.2. LiveData

Класс LiveData – хранилище информации, работающее по принципу шаблона проектирования Observer (наблюдатель).

LiveData используется для обмена данными между их источников и остальными объектами приложения с разными жизненными циклами. Основной обязанностью ViewModel-ей является загрузка данных, связанных с пользовательским интерфейсом, и управление ими, в них хранятся объекты типа LiveData.

LiveData имеет ряд следующих преимуществ:

- 1) предотвращает утечку памяти, когда наблюдатель привязан к жизненному циклу;
- 2) предотвращает сбои из-за остановки активности;
- 3) автоматически обрабатывает жизненный цикл;

- 4) упрощает тестирование кода.^[7]

3.5.3. Navigation API

Использование современного Navigation API входящего в набор библиотек Android Jetpack, имеет следующие преимущества:

- 1) создание единого графического представление всего графа переходов;
- 2) контроль переходов в приложении на соответствии построенной схеме;
- 3) унификация и упрощения перехода к новой активности или фрагменту, а также передаче им данных;
- 4) модифицированные навигационные элементы UI теперь интегрированы с новым механизмом навигации, вследствие чего нужно писать значительно меньше кода при их реализации.

3.5.4. База данных

В качестве хранилища данных было принято решение использование локальной базы данных, по следующим причинам:

- 1) Локальная база данных хранит данные на устройстве, обеспечивая конфиденциальность и безопасность, что важно для данной системы, так как она подразумевает хранение личных данных пациентов
- 2) Быстродействие и производительность, для обеспечения быстрого отклика интерфейса
- 3) Возможность работать с приложением без подключения к интернету
- 4) Более простая разработка и поддержка

Для создания базы и взаимодействия с ней применялась библиотека Room, предоставляющая уровень абстракции поверх SQLite, позволяющий осуществлять свободный доступ к базе данных. В частности, Room предоставляет следующие преимущества:

- 1) Проверка SQL-запросов во время компиляции.
- 2) Удобные аннотации, которые сводят к минимуму повторяющийся и подверженный ошибкам стандартный код.
- 3) Упрощенные пути миграции базы данных

На рисунке 4 представлена модель базы данных.

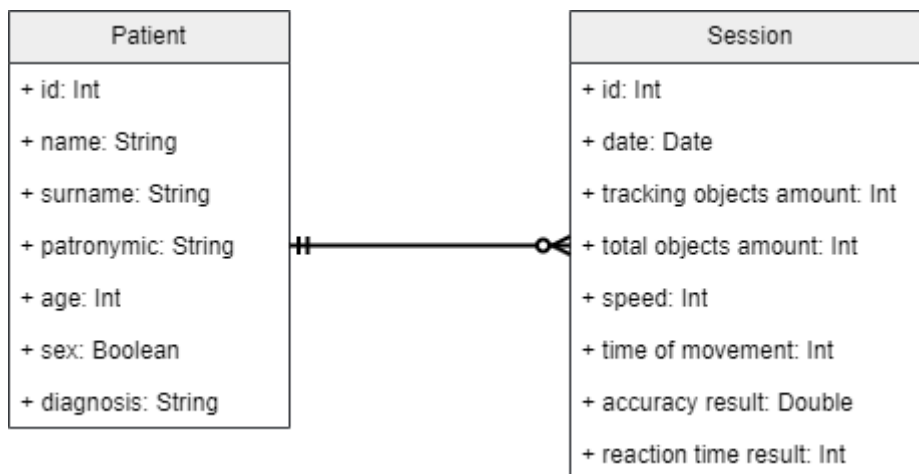


Рисунок 4 – Модель базы данных

3.5.5. Сопрограммы

Сопрограмма - это шаблон проектирования параллелизма, используемый в Android, для упрощения асинхронно выполняющегося кода.

Сопрограммы помогают управлять длительными задачами, которые в противном случае могут заблокировать основной поток и привести к тому, что приложение перестанет отвечать на запросы. Более половины разработчиков, использующих сопрограммы в своей работе, заметили увеличение производительности своих приложений.^[8]

Помимо прочего сопрограммы имеют следующие преимущества:

- 1) Возможность запускать множество сопрограмм в одном потоке благодаря поддержке приостановки, которая не блокирует поток, в котором выполняется сопрограмма. Приостановка экономит память по сравнению с блокировкой, поддерживая множество одновременных операций.
- 2) Сопрограммы следуют принципу структурированного параллелизма, который означает, что новые сопрограммы могут быть запущены только в определенной области, которая ограничивает время жизни сопрограммы, что уменьшает вероятность утечки памяти.
- 3) Многие библиотеки Jetpack включают расширения, обеспечивающие полную поддержку сопрограмм.

3.5.6. Storage Access Framework

Для сохранения файлов с данными в формате .csv в общее хранилище использовался Storage Access Framework. Он предоставляет стандартный, простой в использовании пользовательский интерфейс, не требует системных разрешений, поскольку пользователь участвует в выборе каталогов, к которым может получить доступ приложение, при этом повышается пользовательский контроль и конфиденциальность. Кроме того, файлы будут сохраняться вне каталога конкретного

приложения и вне хранилища мультимедиа и, следовательно, останутся на устройстве в случае удаления приложения.^[9]

3.5.7. TimeAnimator

Для анимации объектов был выбран класс TimeAnimator. Он отправляет события для каждого кадра анимации в свой слушатель с информацией об аниматоре, общем затраченном времени и времени, прошедшем с момента предыдущего кадра анимации.^[10]

Данный способ анимации был выбран, так как необходимо было на каждом кадре рассчитывать направление движения шариков и проверять их на столкновение друг с другом и краями экрана.

3.6. Описание и обоснование метода организации входных и выходных данных

3.6.1. Организация входных данных

Пользователь вводит текст с клавиатуры мобильного устройства в предназначенные для этого текстовые поля. Входными данными при работе с программой также служат касания экрана мобильного устройства в ходе пользования программой.

Выбор обоснован тем, что детям будет легче взаимодействовать с системой по средством касаний, так как этот способ взаимодействия наиболее понятен и не требует каких-либо особых навыков.

3.6.2. Организация выходных данных

Выходными данными является отображение на экране мобильного устройства элементов интерфейса и отчетов.

ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Данный проект не предусматривает расчет экономической эффективности.

Ожидается, что разработанное ПО будет использоваться в реабилитационных центрах, предоставляя удобный способ оценки и развития способности следить за несколькими объектами одновременно.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Number of new cases, both sexes, age [0-19], in 2022, Russian Federation — URL: https://gco.iarc.who.int/today/en/dataviz/tables?mode=cancer&key=total&cancers=31&age_end=3&group_populations=0&multiple_populations=1&populations=643&types=0 (режим доступа: свободный, дата обращения 08.02.2024)
2. Информационный портал об опухолях мозга (проект Благотворительного Фонда Константина Хабенского) — URL: <https://oncobrain.ru/rehabilitation.html> (режим доступа: свободный, дата обращения 08.02.2024)
3. Chipeeva, N.; Deviaterikova, A.; Glebova, E.; Romanova, E.; Karelin, A.; Kasatkin, V. Comparison of Neurocognitive Functioning and Fine Motor Skills in Pediatric Cancer Survivors and Healthy Children. *Cancers* 2022, 14, 5982. — URL: <https://doi.org/10.3390/cancers14235982> (режим доступа: свободный, дата обращения 08.02.2024)
4. Mobile Operating System Market Share Russian Federation — URL: <https://gs.statcounter.com/os-market-share/mobile/russian-federation#monthly-202401-202402> (режим доступа: свободный, дата обращения 24.03.2024)
5. Kotlin and Android — URL: <https://developer.android.com/kotlin> (режим доступа: свободный, дата обращения 25.03.2024)
6. Recommended app architecture — URL: <https://developer.android.com/topic/architecture#recommended-app-arch> (режим доступа: свободный, дата обращения 25.03.2024)
7. LiveData overview — URL: <https://developer.android.com/topic/libraries/architecture/livedata> (режим доступа: свободный, дата обращения 25.03.2024)
8. Kotlin coroutines on Android — URL: <https://developer.android.com/kotlin/coroutines> (режим доступа: свободный, дата обращения 25.03.2024)
9. Open files using the Storage Access Framework — URL: <https://developer.android.com/guide/topics/providers/document-provider> (режим доступа: свободный, дата обращения 15.06.2024)
10. TimeAnimator — URL: <https://developer.android.com/reference/android/animation/TimeAnimator> (режим доступа: свободный, дата обращения 15.06.2024)