Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по УП

Задание 4. Приложение CriminalIntent

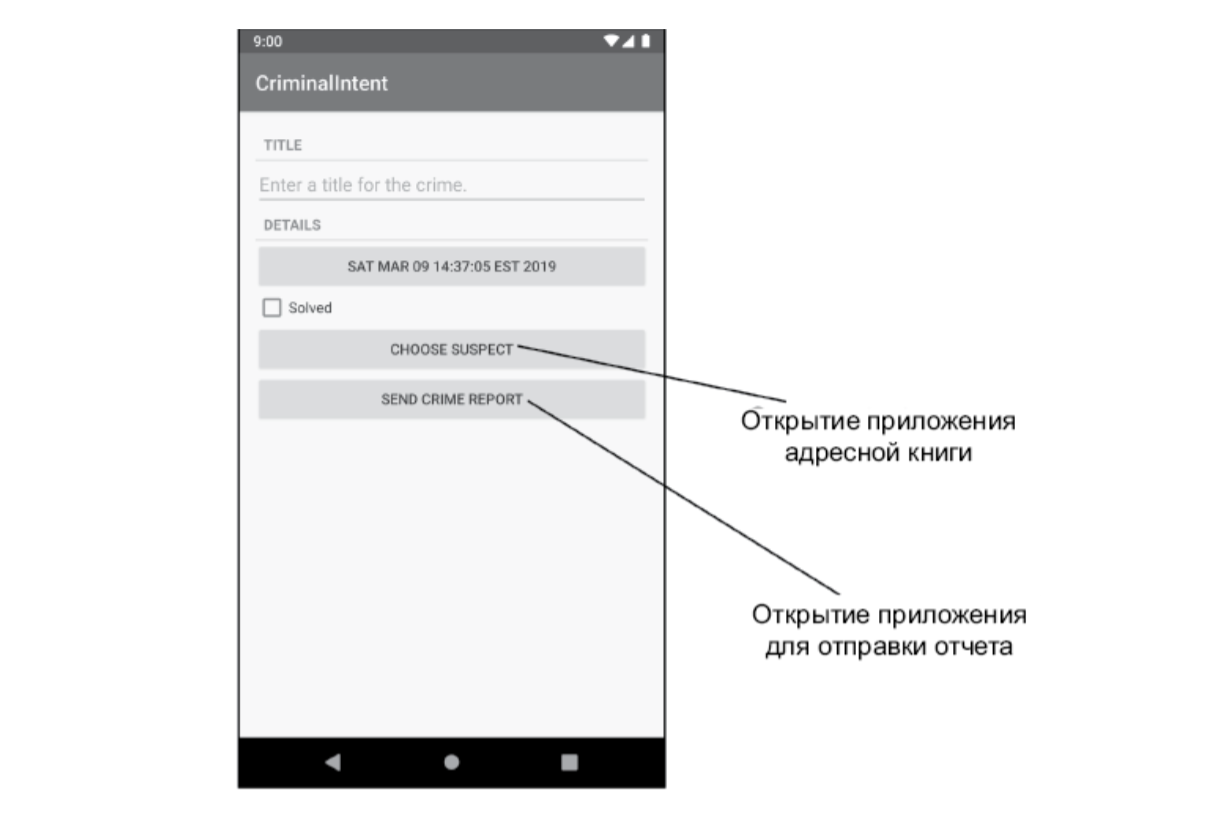
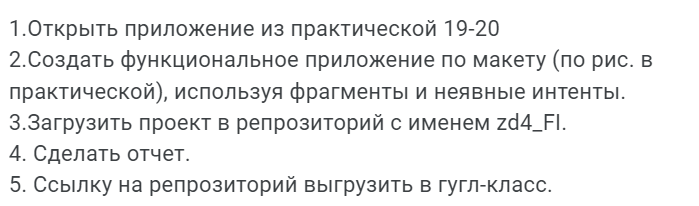
Выполнила: Маринина Светлана Романовна

Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2024

1. **Описание задачи:**

****

**Входные данные:** titleField - поле для ввода преступления, name - поле для выбора подозреваемых

**Выходные данные:** message – сообщение о преступлении и подозреваемом

1. **Структура проекта:**

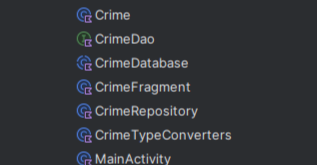


Рис.1. Схема классов

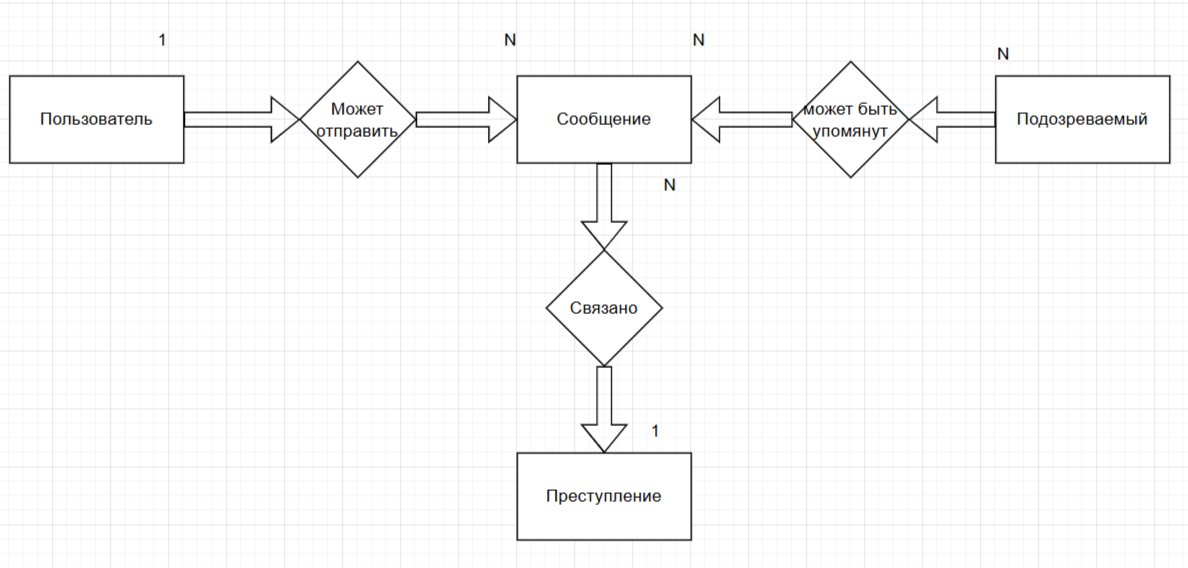


Рис.2. ER - диаграмма

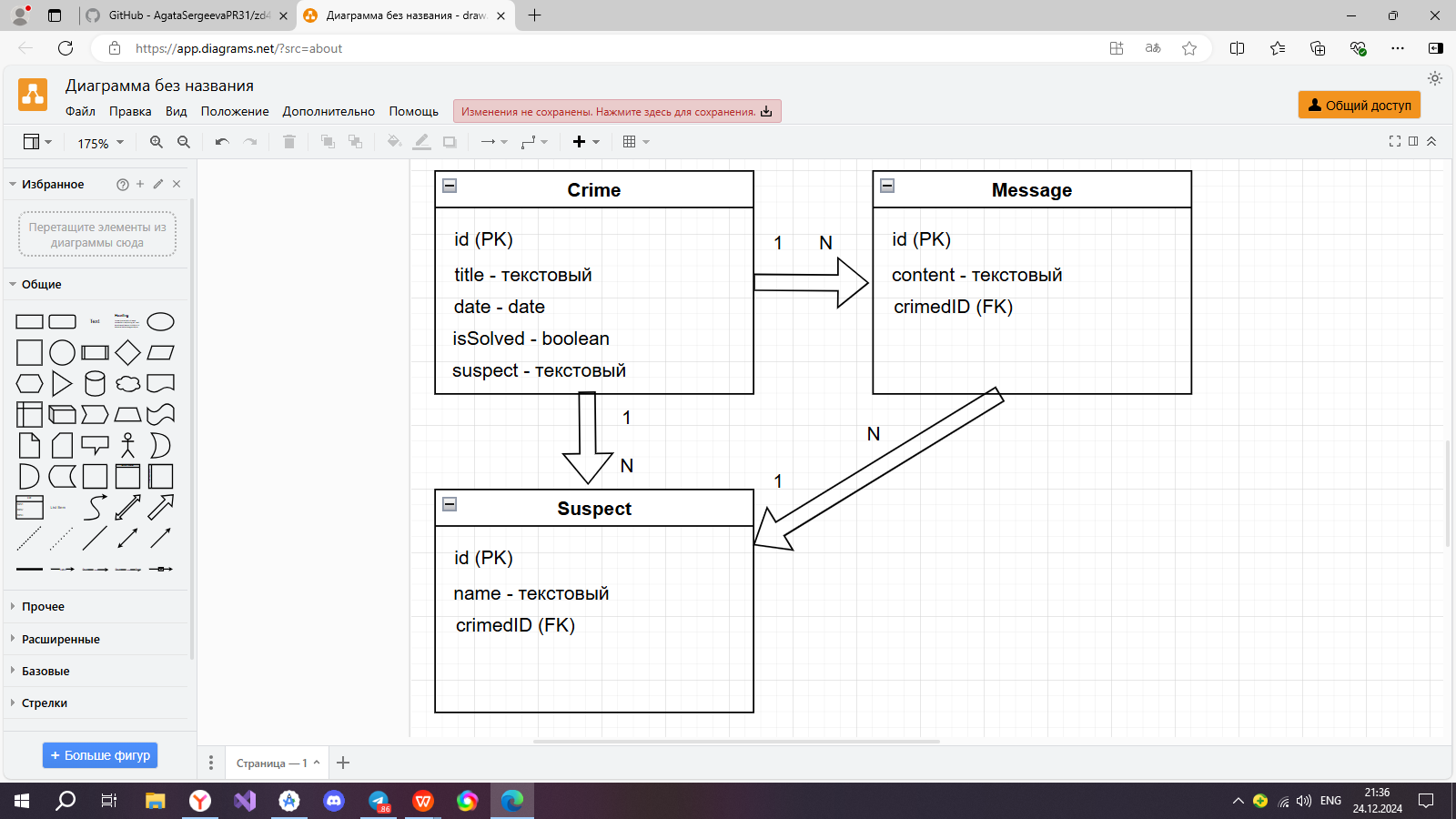


Рис.3. ER - диаграмма

1. **Листинг программы:**

Код дизайна:

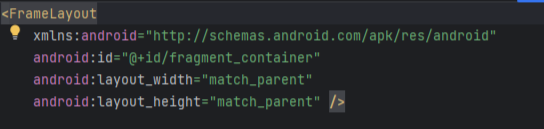


Рис.4. «activity\_main»



Рис.5. «fragment\_crime»



Рис.6. «fragment\_crime»

Код классов:

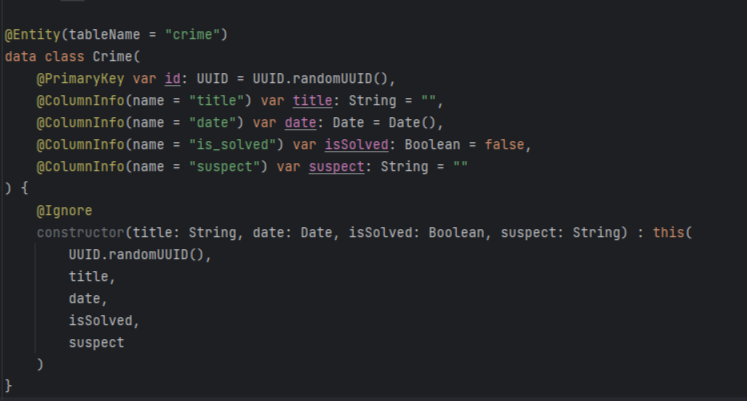


Рис.7. «Crime»

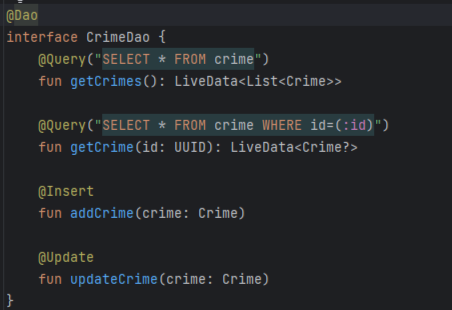


Рис.8. «CrimeDao»



Рис.9. «CrimeDatabase»

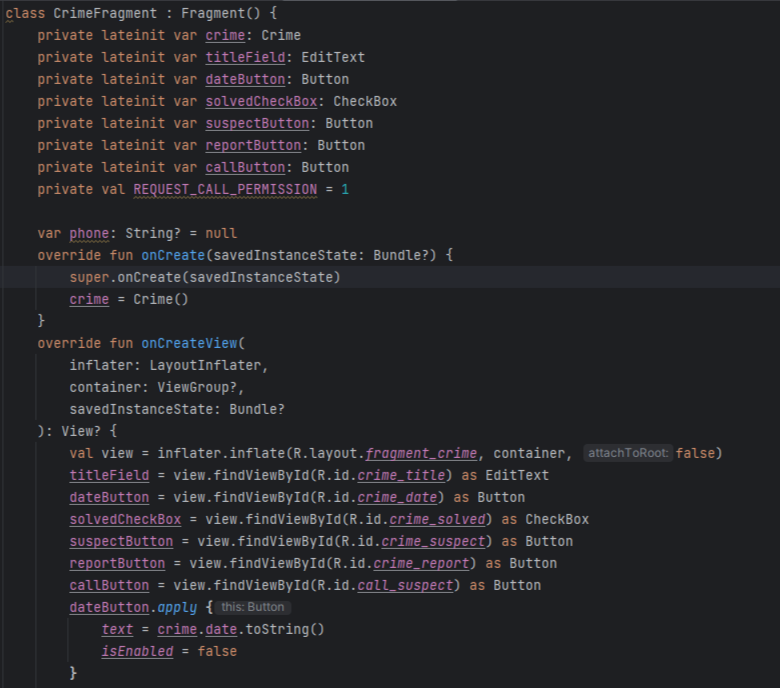


Рис.10. «CrimeFragment»

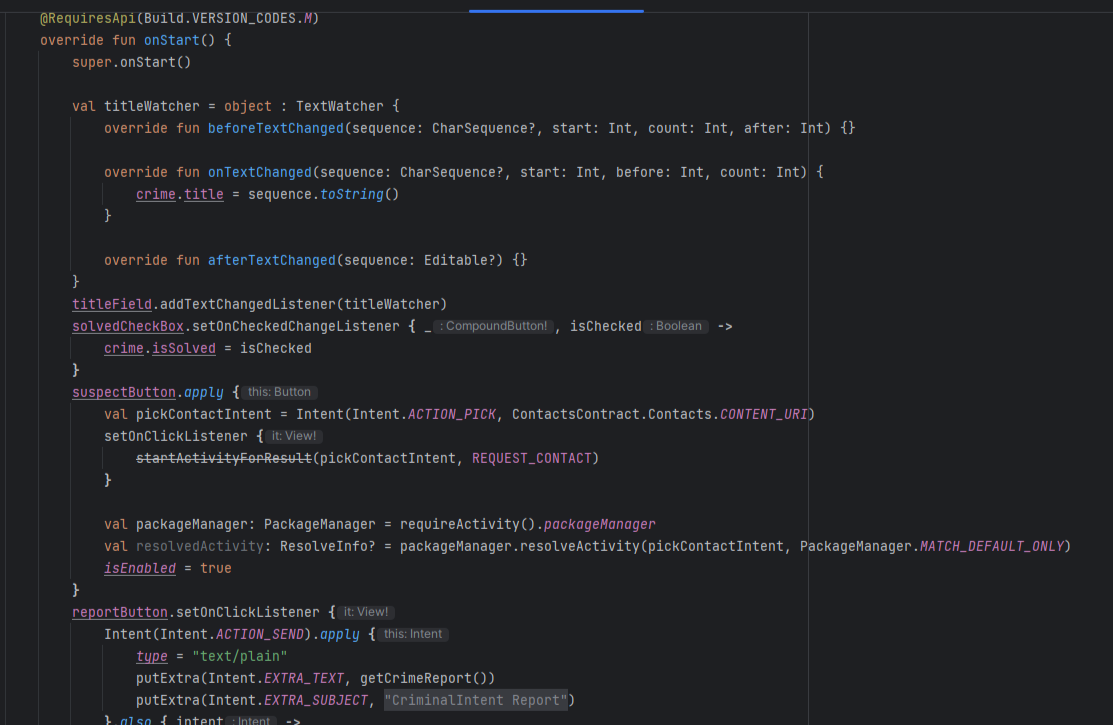


Рис.11. «CrimeFragment»

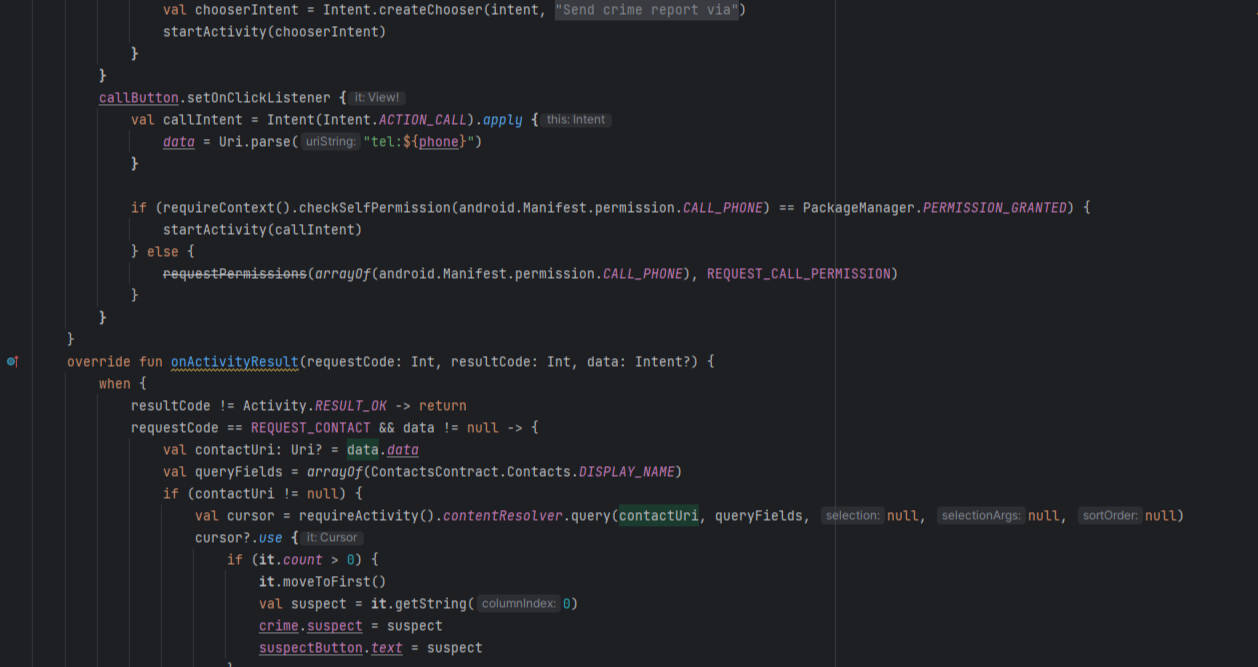


Рис.12. «CrimeFragment»



Рис.13. «CrimeFragment»

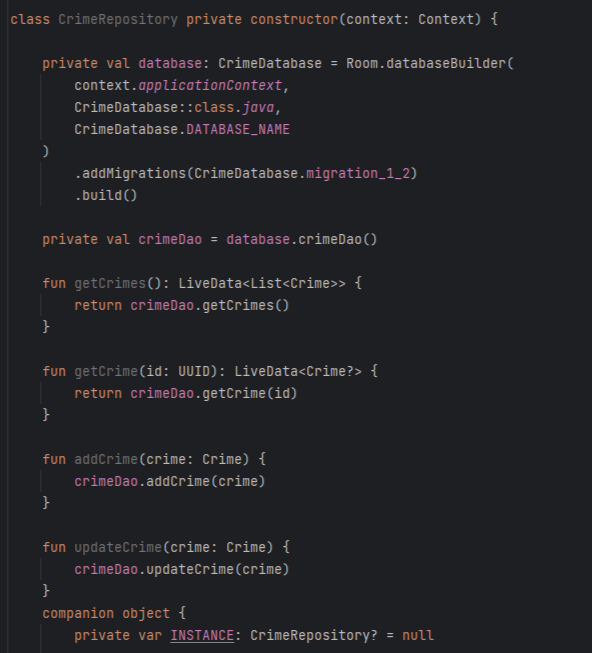


Рис.14. «CrimeRepository»

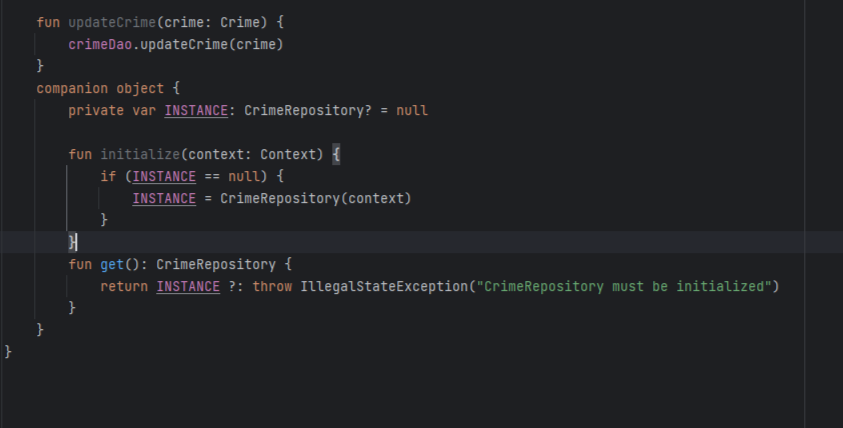


Рис.15. «CrimeRepository»

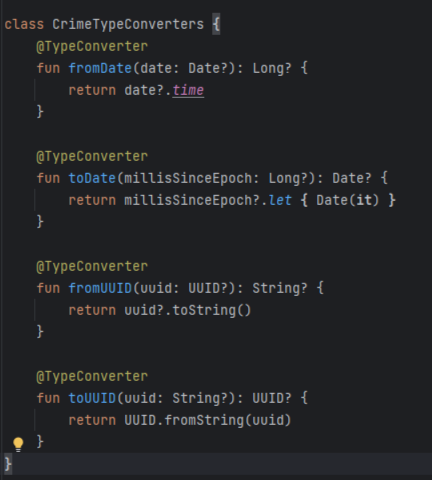


Рис.16. «CrimeTypeConverts»

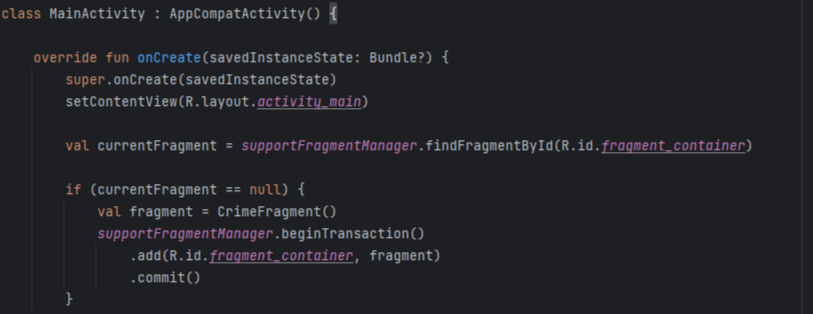


Рис.17. «MainActivity»

1. **Описание разработанных методов:**

Методы класса CrimeFragment:

#### **onStart()** - метод вызывается, когда фрагмент становится видимым для пользователя. Нужен для обновления заголовка, статуса, а также взаимодействие с кнопками

#### **onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) -** метод, который обрабатывает результат выбора контакта. Если пользователь выбрал контакт, мы извлекаем его имя и обновляем текст кнопки suspectButton

#### **getCrimeReport(): String -** метод, который формирует строку отчета о преступлении (берет всю информацию)

Методы класса CrimeDao:

****getCrimes() -****  метод, который возвращает список всех преступлений

****getCrime(id: UUID) -**** метод, который получает конкретное преступление по его уникальному id

****addCrime(crime: Crime) -**** метод, который добавляет новое преступление в базу данных

****updateCrime(crime: Crime)**** - метод, который обновляет существующее преступление в базе данных

Методы класса CrimeDatabase

****abstract fun crimeDao(): CrimeDao -**** метод, который возвращает экземпляр интерфейса CrimeDao, содержащий методы для взаимодействия с базой данных(получение, добавление и обновление записей о преступлениях)

Методы CrimeRepository:

**getCrimes() -** метод, который возвращает список всех преступлений

****getCrime(id: UUID) -**** метод, который получает конкретное преступление по его уникальному id

****addCrime(crime: Crime) -**** метод, который добавляет новое преступление в базу данных

****updateCrime(crime: Crime)**** - метод, который обновляет существующее преступление в базе данных

**get() - м**етод возвращает текущий экземпляр CrimeRepository. Если экземпляра нет, то выбрасывается исключение

Методы класса CrimeTypeConverts:

**fromDate(date: Date?): Long?**: - преобразует Date в значение типа Long

**toDate(millisSinceEpoch: Long?): Date?**: - принимает значение типа Long и возвращает объект Date (обратное преобразование)

**fromUUID(uuid: UUID?): String?**: - преобразует UUID в строку

**toUUID(uuid: String?): UUID?**: - принимает значение строки и возвращает объект UUID (обратное преобразование)

1. **Алгоритм решения:**



Рис.18. Алгоритм класса CrimeFragment

1. **Используемые библиотеки:**

### ****androidx.core:core-ktx:1.7.0 -**** библиотека предоставляет расширения Kotlin для основных компонентов Android

### ****androidx.appcompat:appcompat:1.6.1 -** библиотека для** совместимости с различными версиями Android

### ****androidx.room:room-runtime:2.5.2-**** это библиотека для работы с базами данных в Android

### ****androidx.room:room-ktx:2.5.2 -**** библиотека предоставляет расширения Kotlin для Room, позволяя использовать функции Kotlin для асинхронного доступа к базе данных

### ****androidx.fragment:fragment-ktx:1.6.1-** б**иблиотека включает в себя расширения для работы с фрагментами

### ****androidx.lifecycle:lifecycle-runtime-ktx:2.8.7**** - библиотека предоставляет компоненты для управления жизненным циклом приложения

### **androidx.compose.ui:ui библиотека для использования** компонентов и инструментов для построения UI

### ****androidx.test.espresso:espresso-core:3.6.1 -**** библиотека для тестирования пользовательского интерфейса в Android. Она позволяет писать тесты, которые взаимодействуют с UI-компонентами

1. **Используемые инструменты:**

Android Stusio, Kotlin

1. **Описание пользовательского интерфейса:**

Для начала пользователь должен ввести тему преступления. К примеру он ввёл (dead).

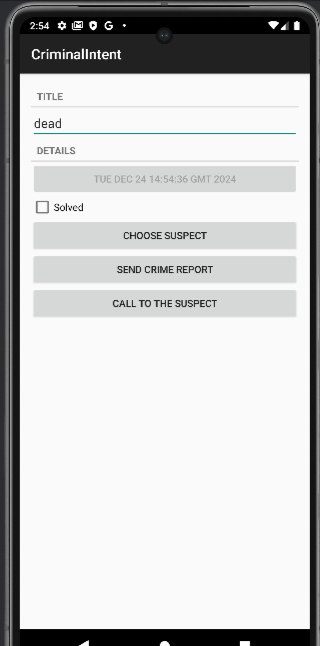


Рис.19. «Экран»

Далее пользователь может выбрать контакт, который является подозреваемым. Также он может выбрать решено ли это преступление (Solved)

Пример на экране:

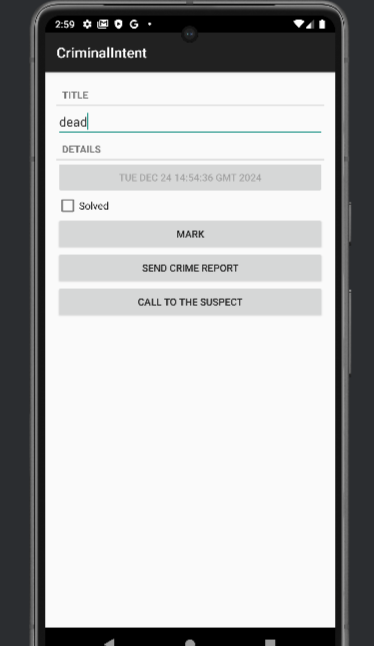


Рис.20. «Экран»

После этого пользователь может отправить любому нужному контакту всю информацию о подозреваемом и его преступлении. Пример на экране:

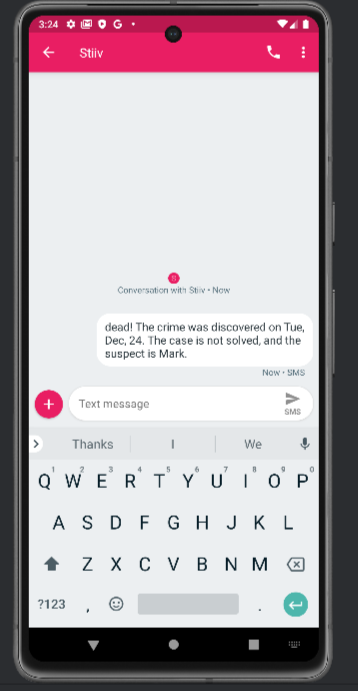


Рис.21. «Экран»

Также можно позвонить подозреваемому:

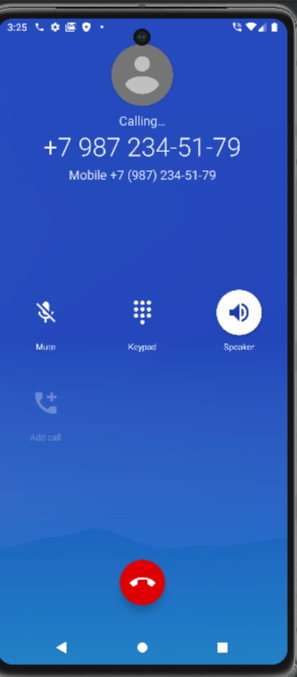


Рис.22. Экран»