CrimeDatabase

```
package com.example.criminalintent_website.database

import androidx.room.Database
import androidx.room.RoomDatabase
import androidx.room.TypeConverters
import androidx.room.migration.Migration
import androidx.sqlite.db.SupportSQLiteDatabase
import com.example.criminalintent_website.Crime

@Database(entities = [ Crime::class ], version = 2)
@TypeConverters(CrimeTypeConverters::class)
abstract class CrimeDatabase : RoomDatabase() {

    abstract fun crimeDao(): CrimeDao

}

// Adding a column to the database. fun addMigrations(migration_1_2) is called in CrimeRepository
val migration_1_2 = object : Migration(1, 2) {
    override fun migrate(database: SupportSQLiteDatabase) {
        database.execSQL(
            "ALTER TABLE Crime ADD COLUMN suspect TEXT NOT NULL DEFAULT ''"
        )
    }
}
```

CrimeTypeConverters

```
package com.example.criminalintent_website.database

import androidx.room.TypeConverter
import java.util.*

class CrimeTypeConverters {

    // Converts the date to a Long to store in database
    @TypeConverter
    fun fromDate(date: Date?): Long? {
        return date?.time
    }

    // Converts the Long back to Date
    @TypeConverter
    fun toDate(millisSinceEpoch: Long?): Date? {
        return millisSinceEpoch?.let { Date(it) }
    }

    // Converts the UUID to a String to store in database
    @TypeConverter
    fun fromUUID(uuid: UUID?): String? {
        return uuid?.toString()
    }

    // Converts the String back to UUID
    @TypeConverter
    fun toUUID(uuid: String?): UUID? {
        return UUID.fromString(uuid)
    }

}
```

CrimeDao

```
package com.example.criminalintent_website.database

import androidx.lifecycle.LiveData
import androidx.room.*
import com.example.criminalintent_website.Crime
import java.util.*

@Dao
interface CrimeDao {

    @Query("SELECT * FROM crime")
    fun getCrimes(): LiveData<List<Crime>>

    @Query("SELECT * FROM crime WHERE id = (:id)")
    fun getCrime(id: UUID): LiveData<Crime?>

    @Update
    fun updateCrime(crime: Crime)

    @Insert
    fun addCrime(crime: Crime)

}
```

CrimeRepository

```
package com.example.criminalintent_website

import android.content.Context
import androidx.lifecycle.LiveData
import androidx.room.Room
import com.example.criminalintent_website.database.CrimeDatabase
import com.example.criminalintent_website.database.migration_1_2
import java.io.File
import java.lang.IllegalStateException
import java.util.*
import java.util.concurrent.Executors

private const val DATABASE_NAME = "crime-database"

class CrimeRepository private constructor(context: Context) {

    // Members to reference database and DAO classes
    private val database: CrimeDatabase = Room.databaseBuilder(
        context.applicationContext,
        CrimeDatabase::class.java,
        DATABASE_NAME
    ).addMigrations(migration_1_2).build()

    private val crimeDao = database.crimeDao()
    private val executor = Executors.newSingleThreadExecutor()
    private val filesDir = context.applicationContext.filesDir

    // DAO functions
    fun getCrimes(): LiveData<List<Crime>> = crimeDao.getCrimes()
    fun getCrime(id: UUID): LiveData<Crime?> = crimeDao.getCrime(id)

    fun updateCrime(crime: Crime) {
        executor.execute {
            crimeDao.updateCrime(crime)
        }

    }

    fun addCrime(crime: Crime) {
        executor.execute {
            crimeDao.addCrime(crime)
        }
    }

    // Return file objects that point to the right location of a photo
    fun getPhotoFile(crime: Crime): File = File(filesDir, crime.photoFileName)

    // Initialize and access the Repository
    companion object {
        private var INSTANCE: CrimeRepository? = null

        fun initialize(context: Context) {
            if (INSTANCE == null) {
                INSTANCE = CrimeRepository(context)
            }
        }
        fun get(): CrimeRepository {
            return INSTANCE ?:
            throw IllegalStateException("CrimeRepository must be initialized")
        }
    } // End getPhotoFile
}
```

Crime

```
package com.example.criminalintent_website

import androidx.room.Entity
import androidx.room.PrimaryKey
import java.util.*

@Entity
data class Crime (@PrimaryKey val id : UUID = UUID.randomUUID(),
                  var title: String = "",
                  var date: Date = Date(),
                  var isSolved: Boolean = false,
                  var suspect: String = "") {
    val photoFileName
        get() = "IMG_$id.jpg"
}
```