CrimeFragment	
package com.example.criminalintent_website import android.app.Activity import android.content.Intent import android.content.pm.PackageManager	
<pre>import android.content.pm.ResolveInfo import android.net.Uri import android.os.Bundle import android.provider.ContactsContract import android.provider.MediaStore</pre>	
<pre>import android.text.Editable import android.text.TextWatcher import android.view.LayoutInflater import android.view.View import android.view.View</pre>	
<pre>import androidx.fragment.app.Fragment import androidx.lifecycle.ViewModelProviders import androidx.lifecycle.Observer import android.text.format.DateFormat import android.widget.*</pre>	
<pre>import androidx.core.content.FileProvider import getScaledBitmap import java.io.File import java.util.*</pre>	
<pre>private const val REQUEST_DATE = 0 private const val REQUEST_CONTACT = 1 private const val REQUEST_PHOTO = 2 private const val ARG_CRIME_ID = "crime_id" private const val DIALOG_DATE = "dialogDate"</pre>	
<pre>private const val DATE_FORMAT = "EEE, MMM, DD" class CrimeFragment: Fragment(), DatePickerFragment.CallBacks {</pre>	
<pre>private val crimeDetailViewModel: CrimeDetailViewModel by Lazy { ViewModelProviders.of(this).get(CrimeDetailViewModel::class.java) } private lateinit var crime: Crime</pre>	
private lateinit var titleField: EditText private lateinit var dateButton: Button private lateinit var solvedCheckBox: CheckBox private lateinit var reportButton: Button	
<pre>private lateinit var suspectButton: Button private lateinit var photoButton: ImageButton private lateinit var photoView: ImageView private lateinit var photoFile: File private lateinit var photoUri: Uri</pre>	
<pre>override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) crime = Crime() val crimeId: UUID = arguments?.getSerializable(ARG_CRIME_ID) as UUID</pre>	
<pre>crimeDetailViewModel.loadCrime(crimeId) } override fun onCreateView(inflater: LayoutInflater,</pre>	
<pre>container: ViewGroup?, savedInstanceState: Bundle?): View? { val view = inflater.inflate(R.layout.fragment_crime, container, false)</pre>	
<pre>titleField = view.findViewById(R.id.crime_title) as EditText dateButton = view.findViewById(R.id.crime_date) as Button solvedCheckBox = view.findViewById(R.id.crime_solved) as CheckBox reportButton = view.findViewById(R.id.crime_report) as Button suspectButton = view.findViewById(R.id.crime_suspect) as Button</pre>	
photoButton = view.findViewById(R.id.crime_camera) as ImageButton photoView = view.findViewById(R.id.crime_photo) as ImageView return view } // End onCreateView	
<pre>override fun onViewCreated(view: View, savedInstanceState: Bundle?) { super.onViewCreated(view, savedInstanceState) crimeDetailViewModel.crimeLiveData.observe(</pre>	
<pre>viewLifecycleOwner, Observer { crime -> crime?.let { this.crime = crime photoFile = crimeDetailViewModel.getPhotoFile(crime)</pre>	
<pre>photoUri = FileProvider.getUriForFile(requireActivity(),</pre>	
<pre>}) } override fun onStart() { super.onStart()</pre>	
<pre>// Create TextWatcher for titleField EditText val titleWatcher = object : TextWatcher { override fun beforeTextChanged(sequence: CharSequence?, start: Int, count: Int, after: Int) {</pre>	
<pre>} override fun onTextChanged(sequence: CharSequence?, start: Int, before: Int, count: Int) { crime.title = sequence.toString() } override fun afterTextChanged(sequence: Editable?) {</pre>	
<pre>// Also intentionally left blank for the same reason }</pre>	
<pre>// Add listeners to widgets titleField.addTextChangedListener(titleWatcher) solvedCheckBox.apply{ setOnCheckedChangeListener { _, isChecked -> crime.isSolved = isChecked</pre>	
} } dateButton.setOnClickListener{ DatePickerFragment.newInstance(crime.date).apply {	
<pre>DatePickerFragment.newInstance(crime.date).apply {</pre>	
} } reportButton.setOnClickListener { Intent(Intent.ACTION_SEND).apply { type = "text/plain"	
<pre>type = "text/plain" putExtra(Intent.EXTRA_TEXT, getCrimeReport()) putExtra(</pre>	
<pre>}.also { intent -> val chooserIntent = Intent.createChooser(intent, getString(R.string.send_report)) startActivity(chooserIntent) }</pre>	
<pre> suspectButton.apply { val pickContactIntent = Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI) </pre>	
<pre>setOnClickListener { startActivityForResult(pickContactIntent, REQUEST_CONTACT) } // Check with PackageManager to make sure there is a Contacts app on the devise val packageManager: PackageManager = requireActivity().packageManager</pre>	
<pre>val packagemanager: Packagemanager = requireActivity().packagemanager val resolvedActivity: ResolveInfo? = packageManager.resolveActivity(</pre>	
<pre> photoButton.apply { val packageManager: PackageManager = requireActivity().packageManager val captureImage = Intent(MediaStore.ACTION_IMAGE_CAPTURE) </pre>	
<pre>val resolvedActivity: ResolveInfo? = packageManager.resolveActivity(</pre>	
<pre>if (resolvedActivity == null) { isEnabled = false } setOnClickListener {</pre>	
<pre>captureImage.putExtra(MediaStore.EXTRA_OUTPUT, photoUri) val cameraActivities: List<resolveinfo> = packageManager.queryIntentActivities(captureImage, PackageManager.MATCH_DEFAULT_ONLY</resolveinfo></pre>	
<pre>for (cameraActivity in cameraActivities) { requireActivity().grantUriPermission(cameraActivity.activityInfo.packageName, photoUri,</pre>	
<pre>Intent.FLAG_GRANT_WRITE_URI_PERMISSION) } startActivityForResult(captureImage, REQUEST_PHOTO) }</pre>	
} } // End OnStart // Save the user's input when they press the back button	
<pre>override fun onStop() { super.onStop() crimeDetailViewModel.saveCrime(crime) }</pre>	
<pre>override fun onDetach() { super.onDetach() requireActivity().revokeUriPermission(photoUri,</pre>	
// Update the UI to show the date the user selected override fun onDateSelected(date: Date) { crime.date = date undateUT()	
<pre>updateUI() } override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) { when {</pre>	
<pre>resultCode != Activity.RESULT_OK -> return requestCode == REQUEST_CONTACT && data != null -> { val contactUri: Uri? = data.data // Specify which fields to query</pre>	
<pre>val queryFields = arrayOf(ContactsContract.Contacts.DISPLAY_NAME) // Perform the query val cursor = contactUri?.let { requireActivity().contentResolver.query (it, queryFields, null, null)</pre>	
<pre>cursor?.use { // Verify cursor has at least one result if (it.count == 0) { return</pre>	
<pre>} // Pull out the first column of the first row - the contact name it.moveToFirst() val suspect = it.getString(0) crime.suspect = suspect</pre>	
<pre>crimeDetailViewModel.saveCrime(crime) suspectButton.text = suspect } } requestCode == REQUEST_PHOTO -> {</pre>	
requireActivity().revokeUriPermission(photoUri,	
<pre>private fun updateUI() { titleField.setText(crime.title) dateButton.text = crime.date.toString()</pre>	
<pre>solvedCheckBox.apply { isChecked = crime.isSolved jumpDrawablesToCurrentState() } if(crime.suspect.isNotEmpty()) {</pre>	
<pre>suspectButton.text = crime.suspect } updatePhotoView() }</pre>	
<pre>private fun updatePhotoView() { if(photoFile.exists()) { val bitmap = getScaledBitmap(photoFile.path, requireActivity()) photoView.setImageBitmap(bitmap) } else {</pre>	
photoView.setImageDrawable(null) } }	
<pre>private fun getCrimeReport(): String { val solvedString = if(crime.isSolved) { getString(R.string.crime_report_solved) } else { getString(R.string.crime_report_unsolved) } }</pre>	
<pre> } val dateString = DateFormat.format(DATE_FORMAT, crime.date).toString() val suspect = if(crime.suspect.isBlank()) { getString(R.string.crime_report_no_suspect) } else { </pre>	
<pre>getString(R.string.crime_report_suspect, crime.suspect) } return getString(R.string.crime_report, crime.title, dateString, solvedString, suspect) </pre>	
<pre>companion object { fun newInstance(crimeId: UUID): CrimeFragment { val args = Bundle().apply { putSerializable(ARG CRIME ID, crimeId) }</pre>	
<pre>putSerializable(ARG_CRIME_ID, crimeId) } return CrimeFragment().apply { arguments = args } </pre>	
} } // End CrimeFragment	
fragment_crime xml version="1.0" encoding="utf-8"?	
<pre><linearlayout <="" android:layout_height="match_parent" android:layout_width="match_parent" android:orientation="vertical" pre="" xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools"></linearlayout></pre>	
android:layout_margin="16dp"> <linearlayout <="" android:layout_height="wrap_content" android:layout_width="match_parent" td=""><td></td></linearlayout>	
android:layout_neight= wrap_content android:orientation="horizontal" android:layout_marginStart="16dp" android:layout_marginTop="16dp"> <linearlayout< td=""><td></td></linearlayout<>	
android:layout_width="wrap_content" android:layout_height="wrap_content" android:orientation="vertical"> <imageview< td=""><td></td></imageview<>	
android:id="@+id/crime_photo" android:layout_width="80dp" android:layout_height="80dp" android:scaleType="centerInside"	
<pre>android:cropToPadding="true" android:background="@android:color/darker_gray"/> <imagebutton <="" android:id="@+id/crime_camera" android:layout_width="match_parent" pre=""></imagebutton></pre>	
<pre>android:layout_height="wrap_content" android:src="@android:drawable/ic_menu_camera"/> </pre>	
<pre>linearLayout android:orientation="vertical" android:layout_width="0dp" android:layout_height="wrap_content" android:layout_weight="l"></pre>	
<pre><textview android:layout_height="wrap_content" android:layout_width="match_parent" android:text="@string/crime_title_label" style="?android:listSeparatorTextViewStyle"></textview></pre>	
<pre><edittext <="" android:id="@+id/crime_title" android:layout_height="wrap_content" android:layout_width="match_parent" pre=""></edittext></pre>	
<pre>android:hint="@string/crime_title_hint"/> </pre>	
<pre>style="?android:listSeparatorTextViewStyle" android:layout_width="match_parent" android:layout_height="wrap_content" android:text="@string/crime_details_label"/></pre>	
<pre> <button android:id="@+id/crime_date" android:layout_height="wrap_content" android:layout_width="match_parent" tools:text="Wed Nov 14 11:56 EST 2018"></button> </pre>	
<pre><checkbox <="" android:id="@+id/crime_solved" android:layout_height="wrap_content" android:layout_width="match_parent" pre=""></checkbox></pre>	
<pre>android:text="@string/crime_solved_label"/> <button <="" android:id="@+id/crime_suspect" android:layout_width="match_parent" pre=""></button></pre>	
<pre>android:layout_height="wrap_content" android:text="@string/crime_suspect_text"/> <button <="" android:id="@+id/crime_report" pre=""></button></pre>	
android:layout_width="match_parent" android:layout_height="wrap_content" android:text="@string/crime_report_text"/>	