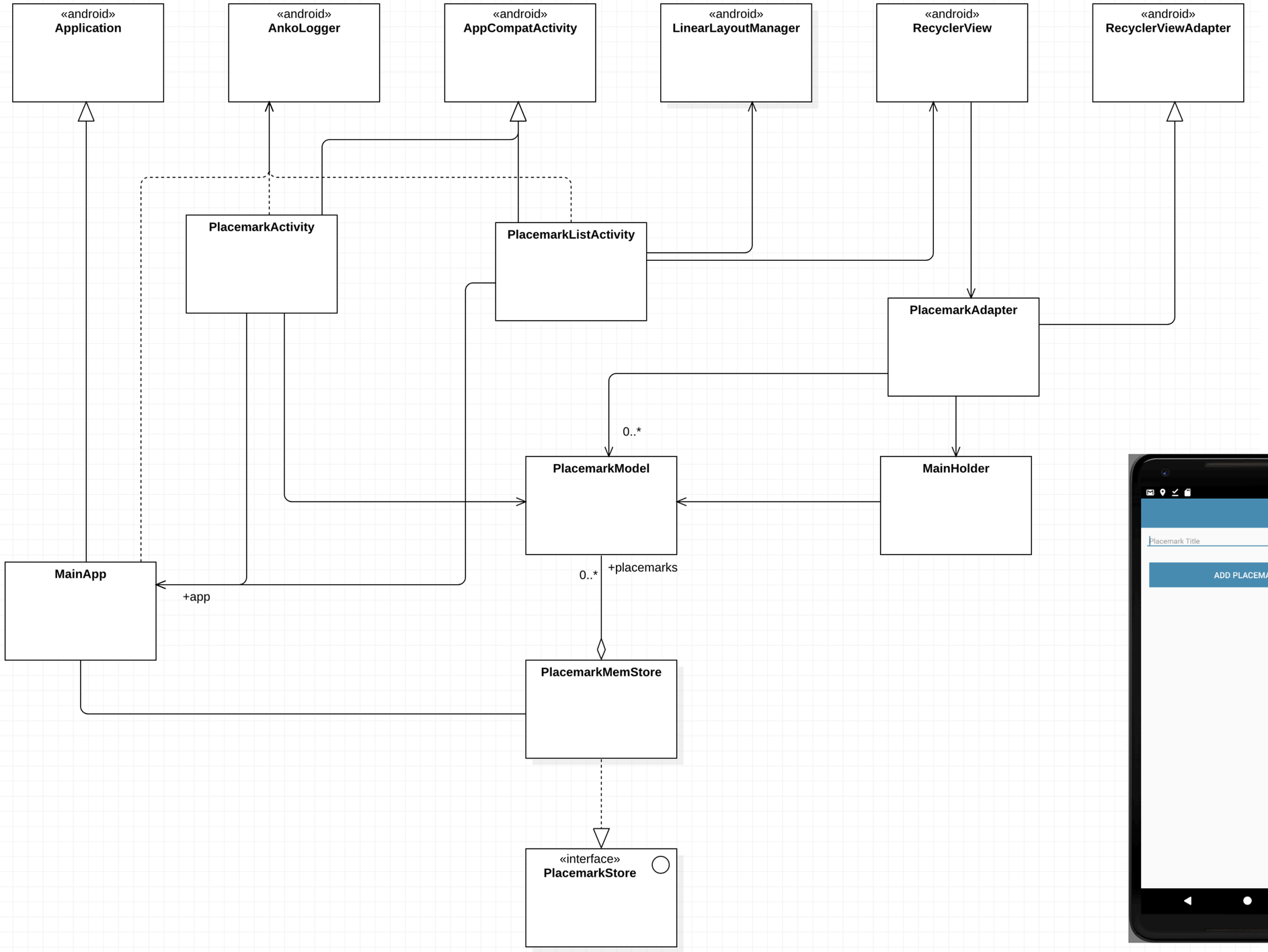
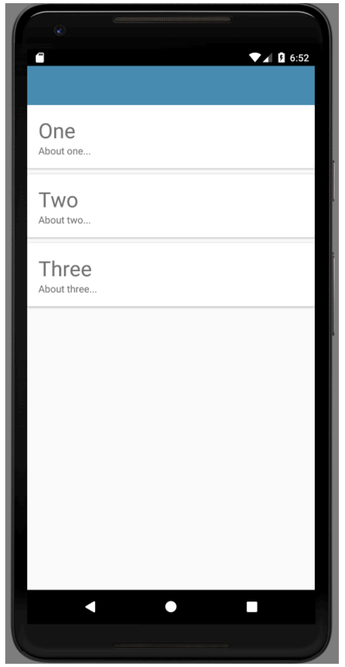
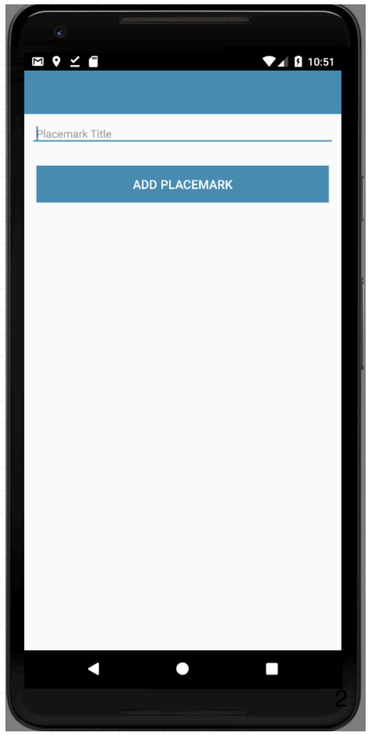
Adapter + Listener

Adapter + Listener

Equip the Adapter with a Listener interface. Use this interface to communicate from the adapter to the host Activity

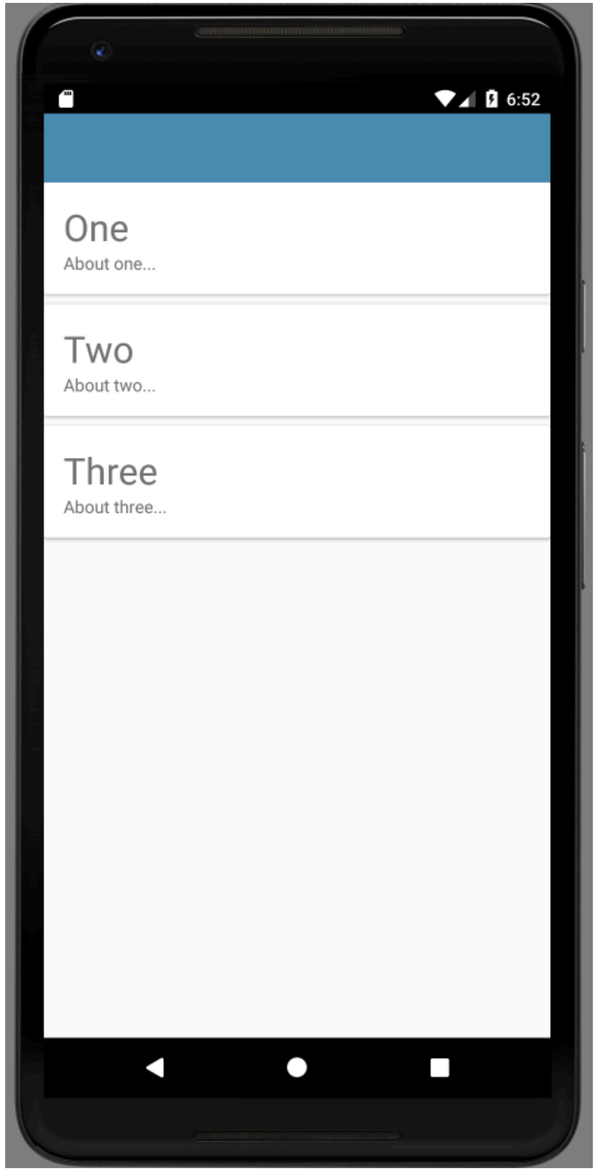

«android»
Application
«android»
AnkoLogger
«android»
AppCompatActivity
«android»
LinearLayoutManager
«android»
RecyclerView
«android»
RecyclerViewAdapter
PlacemarkActivity
PlacemarkListActivity
PlacemarkAdapter
0..*
PlacemarkModel
MainHolder
MainApp
0..*
+placemarks
+app
PlacemarkMemStore
«interface»
PlacemarkStore


6:52
One
About one...
Two
About two...
Three
About three...


10:51
Placemark Title
ADD PLACEMARK

activity_placemark_list

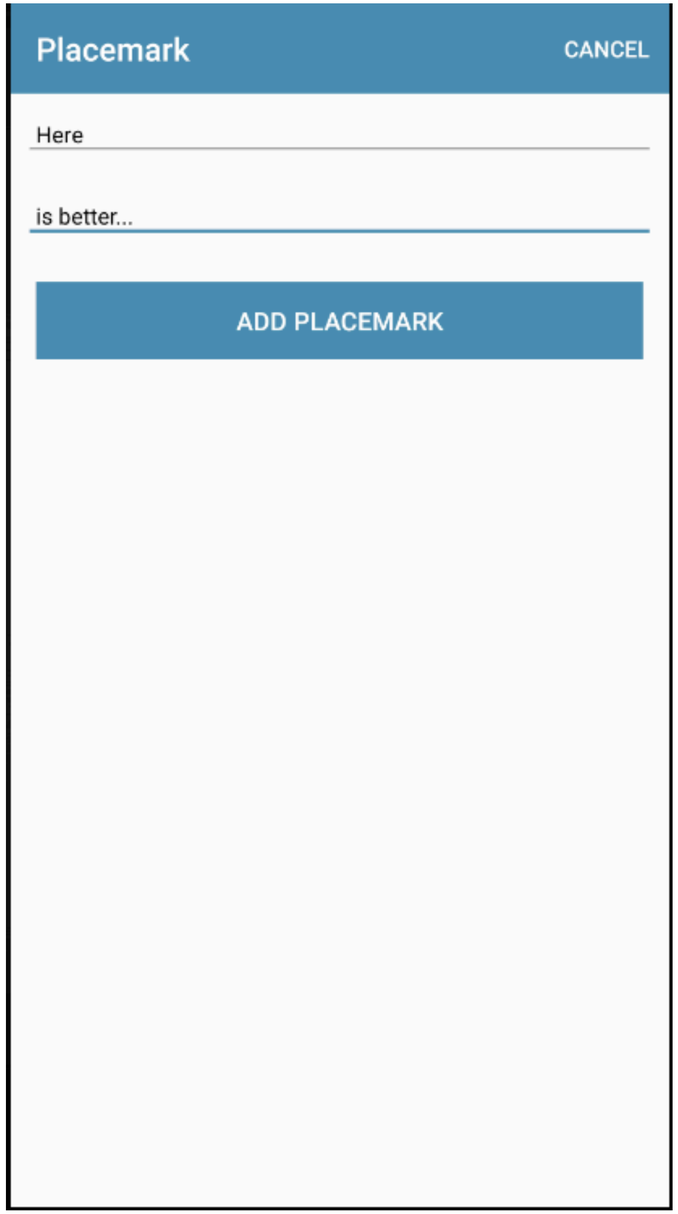
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="org.wit.placemark.activities.PlacemarkListActivity">

  <android.support.design.widget.AppBarLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="@color/colorAccent"
      android:fitsSystemWindows="true"
      app:elevation="0dip"
      app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbarMain"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:titleTextColor="@color/colorPrimary"/>
  </android.support.design.widget.AppBarLayout>

  <android.support.v7.widget.RecyclerView
      android:id="@+id/recyclerView"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:scrollbars="vertical"
      app:layout_behavior="@string/appbar_scrolling_view_behavior"/>

</android.support.design.widget.CoordinatorLayout>
```



card_placemark

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:elevation="24dp">

  <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:padding="16dp">

    <TextView
        android:id="@+id/placemarkTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:textSize="30sp"/>

    <TextView
        android:id="@+id/description"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/placemarkTitle"/>

  </RelativeLayout>
</android.support.v7.widget.CardView>
```

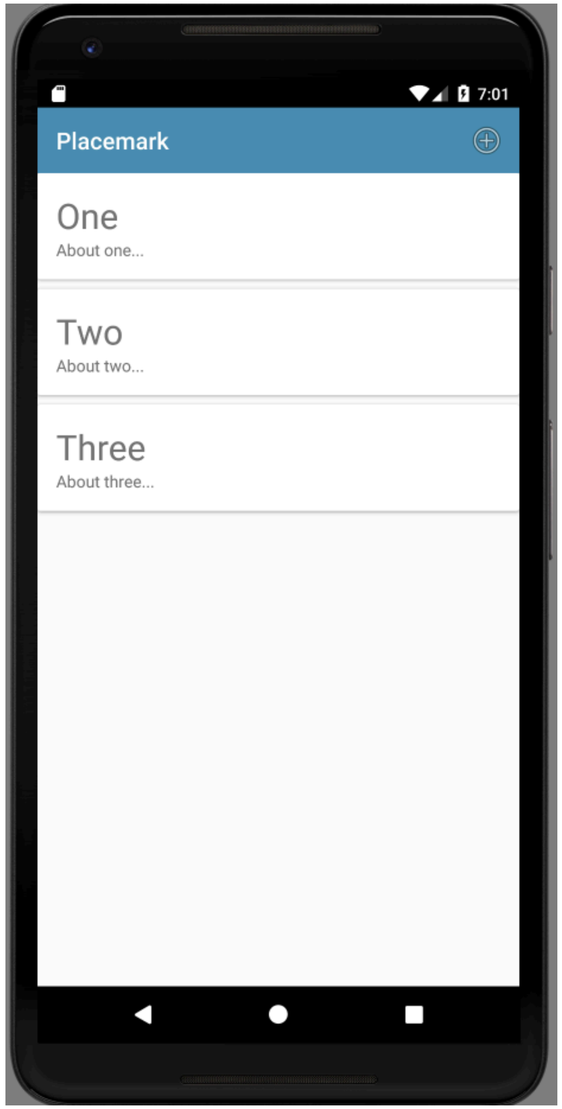
activity_placemark




```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="org.wit.placemark.activities.PlacemarkActivity">

  <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appBarLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorAccent"
        android:fitsSystemWindows="true"
        app:elevation="0dip"
        app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

      <android.support.v7.widget.Toolbar
          android:id="@+id/toolbarAdd"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          app:titleTextColor="@color/colorPrimary"/>
    </android.support.design.widget.AppBarLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/appBarLayout"
        android:fillViewport="true">

      <LinearLayout
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:orientation="vertical">

        <android.support.design.widget.TextInputEditText
            android:id="@+id/placemarkTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:hint="@string/hint_placemarkTitle"
            android:inputType="text"
            android:maxLength="25"
            android:maxLines="1"
            android:padding="8dp"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="14sp"/>

        <android.support.design.widget.TextInputEditText
            android:id="@+id/description"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:hint="@string/hint_placemarkDescription"
            android:inputType="text"
            android:maxLength="25"
            android:maxLines="1"
            android:padding="8dp"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="14sp"/>

        <Button
            android:id="@+id/btnAdd"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"
            android:background="@color/colorAccent"
            android:paddingBottom="8dp"
            android:paddingTop="8dp"
            android:stateListAnimator="@null"
            android:text="@string/button_addPlacemark"
            android:textColor="@color/colorPrimary"
            android:textSize="16sp"/>
      </LinearLayout>
    </ScrollView>
  </RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

PlacemarkListActivity

```
class PlacemarkListActivity : AppCompatActivity() {

  lateinit var app: MainApp

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark_list)
    app = application as MainApp

    toolbarMain.title = title
    setSupportActionBar(toolbarMain)

    val layoutManager = LinearLayoutManager(this)
    recyclerView.layoutManager = layoutManager
    recyclerView.adapter = PlacemarkAdapter(app.placemarks.findAll())
  }

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_main, menu)
    return super.onCreateOptionsMenu(menu)
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_add -> startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
    }
    return super.onOptionsItemSelected(item)
  }
}
```

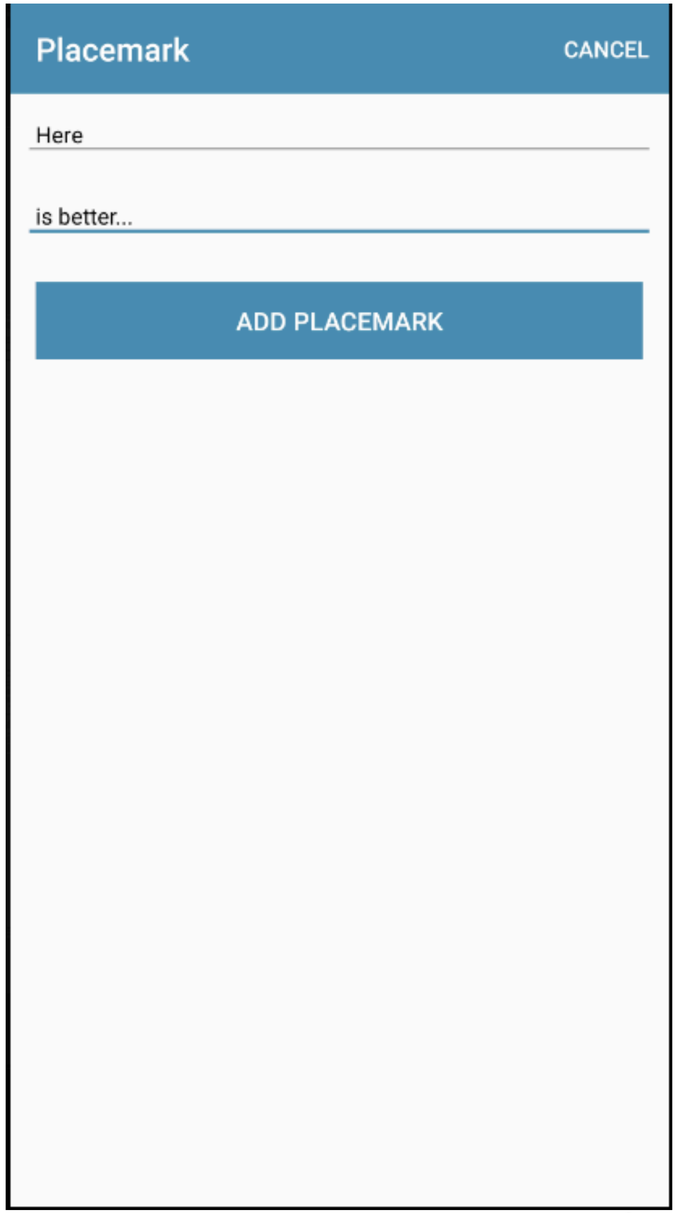


PlacemarkAdapter

```
class PlacemarkAdapter constructor(private var placemarks: List<PlacemarkModel>)
                                                    : RecyclerView.Adapter<PlacemarkAdapter.MainHolder>() {

  override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): MainHolder {
    return MainHolder(LayoutInflater.from(parent?.context).inflate(R.layout.card_placemark, parent, false))
  }

  override fun onBindViewHolder(holder: MainHolder, position: Int) {
    val placemark = placemarks[holder.adapterPosition]
    holder.bind(placemark)
  }

  override fun getItemCount(): Int = placemarks.size

  class MainHolder constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {

    fun bind(placemark: PlacemarkModel) {
      itemView.placemarkTitle.text = placemark.title
      itemView.description.text = placemark.description
    }
  }
}
```

```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  var placemark = PlacemarkModel()
  lateinit var app: MainApp

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    btnAdd.setOnClickListener() {
      placemark.title = placemarkTitle.text.toString()
      placemark.description = description.text.toString()
      if (placemark.title.isNotEmpty()) {
        app.placemarks.create(placemark.copy())
        setResult(AppCompatActivity.RESULT_OK)
        finish()
      }
      else {
        toast("Please Enter a title")
      }
    }
  }

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_placemark, menu)
    return super.onCreateOptionsMenu(menu)
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_cancel -> {
        setResult(RESULT_CANCELED)
        finish()
      }
    }
    return super.onOptionsItemSelected(item)
  }
}
```

PlacemarkActivity



PlacemarkListener

```
interface PlacemarkListener {
  fun onPlacemarkClick(placemark: PlacemarkModel)
}
```

PlacemarkAdapter (without Listener)

```
class PlacemarkAdapter constructor(private var placemarks: List<PlacemarkModel>)
                                                    : RecyclerView.Adapter<PlacemarkAdapter.MainHolder>() {

  override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): MainHolder {
    return MainHolder(LayoutInflater.from(parent?.context).inflate(R.layout.card_placemark, parent, false))
  }

  override fun onBindViewHolder(holder: MainHolder, position: Int) {
    val placemark = placemarks[holder.adapterPosition]
    holder.bind(placemark)
  }

  override fun getItemCount(): Int = placemarks.size

  class MainHolder constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {

    fun bind(placemark: PlacemarkModel) {
      itemView.placemarkTitle.text = placemark.title
      itemView.description.text = placemark.description
    }
  }
}
```

Revised PlacemarkAdapter (with Listener)

```
class PlacemarkAdapter constructor(private var placemarks: List<PlacemarkModel>,
                                   private val listener: PlacemarkListener)
                                                        : RecyclerView.Adapter<PlacemarkAdapter.MainHolder>() {

  override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): MainHolder {
    return MainHolder(LayoutInflater.from(parent?.context).inflate(R.layout.card_placemark, parent, false))
  }

  override fun onBindViewHolder(holder: MainHolder, position: Int) {
    val placemark = placemarks[holder.adapterPosition]
    holder.bind(placemark, listener)
  }

  override fun getItemCount(): Int = placemarks.size

  class MainHolder constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {

    fun bind(placemark: PlacemarkModel,  listener : PlacemarkListener) {
      itemView.placemarkTitle.text = placemark.title
      itemView.description.text = placemark.description
      itemView.setOnClickListener { listener.onPlacemarkClick(plac
    }
  }
}
```

Revised PlacemarkAdapter

```
class PlacemarkAdapter constructor(private var placemarks: List<PlacemarkModel>,
                                   private val listener: PlacemarkListener)
                                                : RecyclerView.Adapter<PlacemarkAdapter.MainHolder>() {

  override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): MainHolder {
    return MainHolder(LayoutInflater.from(parent?.context).inflate(R.layout.card_placemark, parent, false))
  }

  override fun onBindViewHolder(holder: MainHolder, position: Int) {
    val placemark = placemarks[holder.adapterPosition]
    holder.bind(placemark, listener)
  }

  override fun getItemCount(): Int = placemarks.size

  class MainHolder constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {

    fun bind(placemark: PlacemarkModel,  listener : PlacemarkListener) {
      itemView.placemarkTitle.text = placemark.title
      itemView.description.text = placemark.description
      itemView.setOnClickListener { listener.onPlacemarkClick(placemark) }
    }
  }
}
```

PlacemarkListActivity

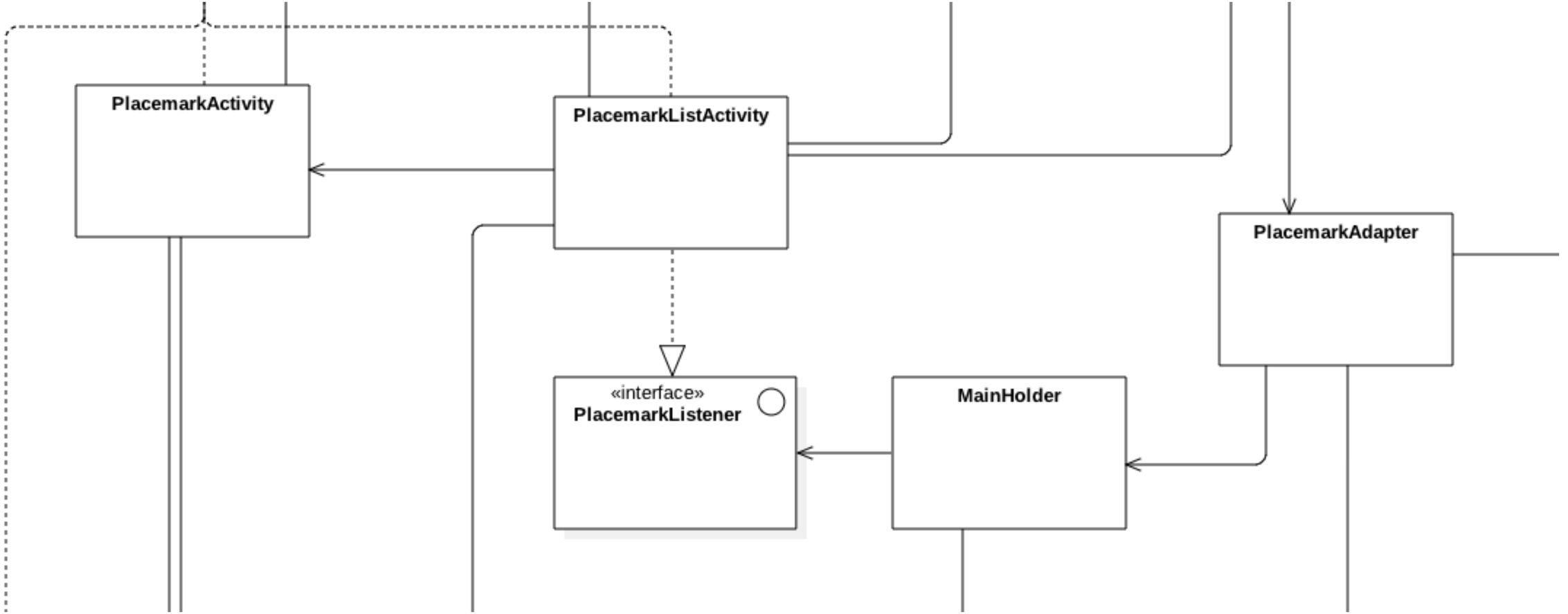
```
class PlacemarkListActivity : AppCompatActivity() {

  lateinit var app: MainApp

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark_list)
    app = application as MainApp

    toolbarMain.title = title
    setSupportActionBar(toolbarMain)

    val layoutManager = LinearLayoutManager(this)
    recyclerView.layoutManager = layoutManager
    recyclerView.adapter = PlacemarkAdapter(app.placemarks.findAll())
  }

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_main, menu)
    return super.onCreateOptionsMenu(menu)
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_add -> startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
    }
    return super.onOptionsItemSelected(item)
  }
}
```

Revised PlacemarkListActivity

```
class PlacemarkListActivity : AppCompatActivity(), PlacemarkListener {

  lateinit var app: MainApp

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark_list)
    app = application as MainApp

    toolbarMain.title = title
    setSupportActionBar(toolbarMain)

    val layoutManager = LinearLayoutManager(this)
    recyclerView.layoutManager = layoutManager
    recyclerView.adapter = PlacemarkAdapter(app.placemarks.findAll(), this)
  }

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_main, menu)
    return super.onCreateOptionsMenu(menu)
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_add -> startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
    }
    return super.onOptionsItemSelected(item)
  }

  override fun onPlacemarkClick(placemark: PlacemarkModel) {
    startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
  }
}
```

Revised PlacemarkListActivity

```
class PlacemarkListActivity : AppCompatActivity(), PlacemarkListener {

  lateinit var app: MainApp

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark_list)
    app = application as MainApp

    toolbarMain.title = title
    setSupportActionBar(toolbarMain)

    val layoutManager = LinearLayoutManager(this)
    recyclerView.layoutManager = layoutManager
    recyclerView.adapter = PlacemarkAdapter(app.placemarks.findAll(), this)
  }

  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_main, menu)
    return super.onCreateOptionsMenu(menu)
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_add -> startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
    }
    return super.onOptionsItemSelected(item)
  }

  override fun onPlacemarkClick(placemark: PlacemarkModel) {
    startActivityForResult<PlacemarkActivity>(AppCompatActivity.RESULT_OK)
  }
}
```


PlacemarkActivity
PlacemarkListActivity
PlacemarkAdapter
«interface»
PlacemarkListener
MainHolder