Hao Zhang

Design and Implementation of Social Event Application Based on Android

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s Thesis

30 April 2019

|  |  |  |
| --- | --- | --- |
| Author  Title  Number of Pages  Date | Hao Zhang  Design and Implementation of Social Event Application Based on Android  xx pages + x appendices  30 April 2019 | |
| Degree | Bachelor of Engineering | |
| Degree Programme | Information Technology | |
| Professional Major | Mobile Solutions | |
| Instructors | Kari Salo, Head of Degree Programme | |
| Over the years, more and more tourists come to Helsinki to travel, in order to let more people know about the events happening in Helsinki area, with the help of the “City of Helsinki” organization, a social events viewing application based on Android platform is born. The main goal of this thesis is to produce a public events information platform based on Android to make it easier for people to find activities happening in the moment or in the future and make people participate in activities, integrate into local life and learn about Helsinki culture.  During the development process, back-end data is provided from the organization's open data which covers public data in the Helsinki region. The major application case used in this thesis is a completed social event application written in Kotlin and the specific location of the event will be marked on Google Maps. Besides, RecyclerView is wildly used in this application to display specific event information, such as date, price, event publisher and so on.  Design pattern, as an essential part of computer science, is beneficial for keeping projects architecture scalable and testable. This thesis introduces Model-View-ViewModel, a design pattern encouraged for Android development. In addition, MVVM design pattern will be demonstrated along with the extracted code from the application case.  In summary, this thesis implements a social event application based on MVVM design pattern and the UI of application conforms to the “Material Design” specification. | | |
| Keywords | | Material Design, Model-View-ViewModel, MVVM, Android Development, Kotlin |

Contents

List of Abbreviations

[1 Introduction 1](#_Toc2973009)

[2 Theoretical Background 1](#_Toc2973010)

[2.1 Android Widgets 1](#_Toc2973011)

[2.1.1 Fragment 1](#_Toc2973012)

[2.1.2 ViewPager 2](#_Toc2973013)

[2.1.3 RecyclerView 3](#_Toc2973014)

[2.2 Android Components 4](#_Toc2973015)

[2.2.1 Activity 4](#_Toc2973016)

[2.2.2 Service 5](#_Toc2973017)

[2.2.3 Broadcast Receiver 6](#_Toc2973018)

[2.2.4 Content Provider 6](#_Toc2973019)

[3 Implementation 7](#_Toc2973020)

[3.1 Application Outline 7](#_Toc2973021)

[3.1.1 Description 7](#_Toc2973022)

[3.1.2 Material Design and Layout 7](#_Toc2973023)

[3.1.3 Development and Process 8](#_Toc2973024)

[3.2 Model View ViewModel (MVVM) 8](#_Toc2973025)

[3.3 Project Requirements 9](#_Toc2973026)

[3.3.1 Feature A 9](#_Toc2973027)

[3.3.2 Feature B 9](#_Toc2973028)

[3.3.3 Feature C 9](#_Toc2973029)

[3.4 Technology Stacks 11](#_Toc2973030)

[3.4.1 Data Flow 11](#_Toc2973031)

[3.4.2 Kotlin over java 11](#_Toc2973032)

[3.4.3 Unit Testing 12](#_Toc2973033)

[4 Results 12](#_Toc2973034)

[5 Conclution 12](#_Toc2973035)

[6 References 12](#_Toc2973036)

[References 13](#_Toc2973037)

Appendices

Appendix 1. Title of the Appendix

Appendix 2. Title of the Appendix

List of Abbreviations

ORM Object-relational mapping. The set of rules for mapping objects in a programming language to records in a relational database, and vice versa.

DBMS Database management system. Software for maintaining, querying and updating data and metadata in a database.

# Introduction

Write the introduction of your thesis here. Use the Body text style for normal text. Likewise, use styles Heading 1, Heading 2, and Heading 3 for headings. As you apply the predefined styles, the text will automatically format correctly: the line spacing will be 1.5, both edges will be justified, and the text will be hyphenated.

The last line of a paragraph can be left hanging, that is, it does not have to reach the right margin.

Begin a new paragraph at the left margin, that is, do not indent the first line.

# Theoretical Background

## Android Widgets

### Fragment

One of the powerful features added to Android 3.0 (API 11) is Fragment, which is designed to provide more dynamic and flexible UI design support for large screens such as tablets. Fragment is a component that can be embedded in an activity. It can combine multiple fragments in one activity to build a multi-pane UI, have its own life cycle, and can have or have no user interface.

Fragments must always be embedded in the Activity, and their lifecycle is directly affected by the lifecycle of the host activity. For example, when an activity is paused, all of its clips are also paused; when the activity is destroyed, all clips are also destroyed. However, when the Activity is running, you can manipulate each fragment independently, such as adding or removing them. When you execute such a fragment transaction, you can also add it to the return stack managed by the Activity.

### ViewPager

ViewPager is a component of SupportV4, mainly a View container that can realize a card-style left and right sliding. Using this class is similar to ListView, you need to use a custom adapter PagerAdapter, the difference is the way to get a View each time.

ViewPager is a page container, except that each page is a whole screen, and the sliding screen is similar to the card type. At present, WeChat 5.1 and the overall look are all viewed in this way.

The whole process is that the ViewPager control goes to the adapter implementation class to get a View every time it is flipped, and then caches a view of the sliding direction. The role of PagerAdapter is to let the developer implement the data itself, fill it with a custom single-page View layout, and then implement some lifecycle methods of PagerAdaper, returning the View to the ViewPager display. Usually user data is a list, stored with a List, and data items are also a variety of complex objects, List can be used as a constructor of the PagerAdapter inheritance class.

        ViewPager always holds three View instances. Except for the first time, each time the pagerAdapter's instantiateItem method is called to get the View instance, the passed position is the forward page of the current page position, which is to cache a page. Then destroy the latter item of the latter item. The initialization process is to call getCount() twice to initialize 0, 1 items. The user turns the page to 1 item and generates 2 items. At this time, there are three items. After that, page through 2 items, destroy 0 items, and generate three items. At this time, it is 1, 2, and 3 items, as shown in the figure.

PagerAdapter is an abstract class that needs to inherit several important methods:

1.getCount(); how many items are there?

2.instantiateItem(View container, int position) instantiates a project, that is, the ViewPager object dynamically retrieves a View from the adapter interface. The current page number is passed in, which is related to the getCount() method. To get the corresponding View from a custom data collection

3.destroyItem(View container, int position, Object object) Using the ViewPager.remove method

4.isViewFromObject must be implemented, it determines whether View is a strong turn, the input is View and Object, you can directly compare it. In the process of optimization, the system will pre-interpret one page before and after, and then judge whether it is from the existing memory.

About the data source:

The data source used in the PagerAdapter is also the user data. Usually, the ArrayList is used to carry the data items, and each item is the data to be carried in the corresponding display space in the page. Similar to the adapter baseAdapter of ListView. The only difference is that getting the ListView control to take the View requires manual implementation of the container.addView() method. Usually the data List is passed as the constructor of the Adapter.

### RecyclerView

RecyclerView is one of the components in Android5.0materials design, and there are CardView, Palette and so on. Looking at the name we can see a little clue, yes, its main feature is reuse. We know that ViewHolder can be reused in the Adapter in Listview. RecyclerView provides a less coupled way to reuse ViewHolder and easily implement ListView, GridView, and waterfall streams.

RecyclerView is a new component in the support-v7 package. It is a powerful sliding component. Compared with the classic ListView, it also has the function of item recycling and multiplexing, but directly encapsulates the implementation of the viewholder. Users only need to implement their own viewholder. Yes, this component will automatically recycle and reuse each item for you.

Advantages of RecyclerView:

1. It doesn't care about where and how the item is displayed ---> layoutManager We can switch the style of RecyclerView in just one sentence using layoutManager.

2. It doesn't care how item is split--->ItemDecoration. We can customize our own dividing line according to the needs of our project.

3. It does not care about the effect of item addition and deletion animation--->ItemAnimator we can define item addition and deletion animation according to our own needs.

4. Just care about how to take the view--->ViewHolder google forces us to use viewHolder, we need to define our own ViewHolder.

## Android Components

The Android application is mainly composed of four components: Activity, Service, Broadcast Receiver and Content Provider [50]. The development of any Android application is inseparable from this. Four major components, in the realization process of the scenic intelligent tour guide service system, with the different functions of the four components, the functions of map display, visitor location monitoring, component communication and so on are realized. The relationship between the various components is shown in Figure 2.2:

### Activity

The most frequently used in Android, the most well-known is the Activity interface. Activity is all the interfaces that can be seen on the mobile phone. These interfaces basically belong to or depend on the Activity. The main role of Activity is to show the user information, provide user and system interaction interface, and monitor the interactive controls such as buttons on the interface.

In order to weaken the language of the admonition program and the characteristics of the process, the Android system designer designed the Activity component into a system control with a lifecycle form. The Activity mainly has seven lifecycles: onCreate, onStart, onResume, onPause, onStop, onDestroy, onRestart. These cycles are between the app running from the click and the last time the app is closed. When developers design the functionality of an app, they only need to match the life cycle according to the business to determine what needs to be done at different times.

----------------------------------------

An activity can be thought of as a user window that can be implemented programmatically. It is an important component of an Android application. It is the basis for all Android application development and provides a visual interface for interacting with users. In an activity, you can add multiple components and use different components to take care of different functions. Activities mainly include interface display and event processing. Developers can load all components of the user interface into the active lifecycle method onCreate, which is displayed on the device screen. Activities can handle the events of all components on the user interface, and complete the system functions by writing event listener code. For example, a developer can load a Button button on the interface and set the button's onClick event to implement other actions after the user clicks the button.

An Android application can have only one activity, or multiple activities can be combined. These activities form the activity stack. The current activity is at the top of the stack, and the previous activities are suppressed. These activities strictly follow the activity life cycle limits. In the actual development, the activity life cycle method should be focused on, and the implementation of each method directly affects the function and performance of the program.

### Service

The service component is similar to the active component and is the functionality required to complete the developer. The only difference is that the service component does not have a user interface. Developers only need to write the function code of the service, not involving the loading and display of the interface. The service can run long hours in the background, completing some of the developer's time-consuming operations that cannot be implemented in the activity. Other application components can launch a designed service, even when the user switches to another application, the service can still run in the background.

    Services can be divided into two types, startup and binding. Startup, when the active component starts the service by calling the startService method, the service is turned on, even if the activity that started it has been destroyed, the service can still run in the background; binding, when the active component is bound to the service by calling the bindService method, The service is in a bound state. Multiple components can be bound to the same service, and the service is destroyed when they are untied.

Developers generally perform long-running functions such as network upload and download, music playback, and user positioning that do not require an interface display through the service.

### Broadcast Receiver

A broadcast receiver is a component that handles external events of an application, similar to a service component, and has no user interface. After the broadcast receiver is defined, other applications call it according to its defined rules and send a broadcast to it. After receiving the broadcast, you can start an activity or a service to start the follow-up function. The use of broadcast receivers enables communication between different components within the same program and communication between different program components. An application can define multiple broadcast recipients, either dynamically by inheriting from the BroadcastReceiver class or statically registered in the AndroidManifest file.

### Content Provider

Content providers are used to share a given data set of one application to other applications, and other applications can obtain data from a content provider. The relational database SQLite provided in Android system will create its own data set for each application level, and only the content provider can share data between each application. The content provider includes the addition, deletion, and modification of data.

# Implementation

## Application Outline

### Description

### Material Design and Layout

In the era of disorder, Android is full of the style of the old Google period of freedom and laissez-faire. Android is like a wasteland opened by Google that everyone can use freely. Google does not impose any restrictions. Developers can upload any application designed according to their own ideas to the store without review. Android does not provide developers with a unified design specification in the era of disorder. The application products are mostly referenced to the materialized, flat design norm or self-contained style. The products of that period have no user experience, and people are forced to adapt to various Different interaction styles.

Google launched the new design language Material Design at I/O 2014. Google rethinks the user experience on the Android platform, trying to bring the physical world back through metaphors of the real world

The rules of the feed migrated to the digital world, and plans to extend the design language building to all platforms, including Android, Chrome OS and web pages, starting with the most popular Google Maps and Google Now. Clean typography and a simple unified layout provide a unified interactive experience for Google's many technology products, although they are not identical in form UI and physical world, but have the same interaction rules.

Material Design is based on three principles:

1. The entity sense metaphor is the central principle of Material Design. Starting from exploring the elements that make up the application, Google uses "paper" as the basis for Material Design to explore real paper. Material Design combines paper metaphor with ink coating metaphors. When the user touches a button, it will feed back to the user by expanding to a global mode.

### Development and Process

## Model View ViewModel (MVVM)

In the development process of a large software system, if you pay attention to the architecture of the program, you don't pay attention to the developer's modular design of the code, and do not pay attention to the decoupling of the function modules coupled by the program, which may be generated during the later iterative development and maintenance of the program. Some undetectable and difficult to locate errors, especially when the program reaches a certain scale, it will need to refactor the program. However, due to the coupling of function and business, the reconstruction will be difficult. Finally, the software can only be redesigned. To achieve, and this is a failed software development process. The best solution to the above problems is to design the system in the early development process, and to design functions and data for the functions and services that the software may have, thereby improving development efficiency and laying the foundation for future software maintenance.

As the software development process matures, many developers have summarized and created a number of framework design patterns suitable for large-scale software development processes, including MVC, MVP and MVVM. These three framework ideas were originally applied to Web applications. Mode, which shines in the era of rapid Internet development, the Android Framework incorporates these into the design of the system; MVP evolved from MVC, originally developed by IBM for object-oriented programming (java and C++) The framework model, after learning Android for a period of time, borrowed to structure the Android App architecture, and got a lot of praise; MVVM framework model was first proposed and applied by Microsoft in software development, with Google in 2015 developers After the Data Binding technology was proposed at the conference and the standard class library of Data Binding was provided to support Android development, many developers used this technology to introduce the MVVM architecture pattern into the mobile development field, and benefited from the introduction of MVVM after development. Data Binding is convenient, many Android developers Give a high rating.

Architecture Patterns The MVVM model was first proposed by Microsoft engineers, but at the 2015 Google Developers Conference, Google engineers introduced MVVM to the development of Android App. MVVM is called Model View ViewModel. MVVM is based on MVP, because Google easily transforms MVP into MVVM through Data Binding technology. The relationship between specific MVVMs is shown in Figure 2.3.

    MVVM is actually a further optimization of MVP. MVP is implemented by the Presenter layer when the data between the Model layer and the View is passed, but it is implemented by the ViewModel in MVVM, but the specific implementation process is realized by Data Binding technology. This is where Data Binding technology drives the implementation of the MVVM framework model. The Data Binding technology is that when the Model layer changes, the latest data is automatically filled in by the attributes written in the XML file in advance, without the developer taking the time to re-update the data for the View layer, thereby simplifying operations and improving development and testing efficiency.

## Project Requirements

### Feature A

### Feature B

### Feature C

#### Google Map调用的实现

The Google Maps API is a personalized map programming API provided by Google Inc. for map software developers based on various platforms. Given the open source and free features of the Google Maps API, using the Google Maps API, developers can embed all the data from Google Maps into their own development software and use Google Maps data to provide location services to users without having to It takes a lot of manpower, material and financial resources to build your own map server.

In addition to helping developers embed maps into web applications, the Google Maps API allows developers to use Javascript scripts for application development, add annotations and paths to maps, and other layer overlays. Click on the action and display a bubble alert window with content information. The operability of the Google Maps API gives mobile phone users a sense of software use, so the travel information service application developed this time uses the Google Maps API to achieve the goal of personalized maps.

Specific steps to invoke the implementation of the Google Maps API;

1. Create the project and select "Goolge Apis" for the SDK.

2. Modify the Android Manifest.xml file.

This is because the data about the map needs to be obtained by moving the path: <uses-permission android; name="android.permission.INTERNET>

3. Create a MapView to create a MapView according to the prompts in the map. The apiK value used in the android system is the key value of the previous application.

4, to achieve Map Activity

Map Activity is mainly used to manage the MapView created before. Of course, this part of the program needs to come from the Map Activity class to implement the MapView function.

MapView can mainly provide W kinds of ground circumference, and their respective setting methods are as follows. mMapView.setTraffic(true);//Set to traffic mode mMapView.setSate...te(tme);//Set to satellite mode mMapView.setStreetView(false);//Set to Street View mode 42 Chapter 5 Travel Information Service Application Software The set Built Zoom Controls method can be used to reduce and enlarge the map display. 5, Ovelay use Overlay can be used in advance on the map to express their own language, that is, text records. Before doing this work, you need to use the transformation between the coordinates to achieve the goal.

## Technology Stacks

### Data Flow

### Kotlin over java

The app with the Android operating system is developed in the Java language, which is an object-oriented programming language. Since Java was created, it has been honed for many years and the use of many developers, feedback, maintenance, Java is now very mature, with many excellent features compared with other languages, such as Java is a simple Object-oriented languages no longer use C++'s multi-inheritance, operator overloading and other language features that are difficult to understand. Java also has distributed features. Java supports network programming. Java guarantees through language-level exception trapping mechanism. Its robustness, the Java language is also portable, because the Java language provides the specified basic type data, the Java language provides multi-threading at the same time, and provides a complete set of mechanisms for multi-threading to ensure the correctness of the data. With the further development and improvement of Java, the performance of Java is getting higher and higher on different platforms, and the speed of running on some platforms has not lost C++.

### Unit Testing

# Results

# Conclution

# References

References

Details of the references are given here. Use the referencing system required in your degree programme or as agreed with your supervisor.

Layout of this page in the author-date (Harvard) referencing system:

Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

Layout of this page in the number (Vancouver) referencing system:

1. Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.
2. Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

**Title of the Appendix**

The contents of the appendix are placed here. Below are the instructions for removing and adding appendices in a way that maintains the headers and footers in their correct form.

Instructions for removing an unwanted appendix:

1. Select the entire page(s) that form the appendix and delete the contents by hitting the Delete key.
2. As you are in the beginning of the empty appendix page (see the image below), double-click the header of the empty page and press Link to Previous button in the ribbon. The following dialogue window opens:



Click Yes.

1. If necessary, make hidden format information visible by pressing .
2. Delete the section break immediately before the appendix to be removed (see image below).



Instructions for adding a new appendix:

1. Place the cursor at the end of the last appendix.
2. Select Page Layout from the menu bar. From the ribbon, select Breaks/Section Breaks/Next Page. This causes a new appendix to appear, but the appendix number in the header is not yet correct.
3. Double click the header of the new appendix with the wrong appendix number. If the option “Link to Previous” is selected, click the corresponding button to deselect it.
4. Replace the appendix number with the correct number.

Note that the appendices need to be updated in the table of contents manually.

**Title of the appendix**

The contents of the appendix are placed here.