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](#_Toc2771809)

[2 Theoretical Background 1](#_Toc2771810)

[2.1 Open Data 1](#_Toc2771811)

[2.2 Model View ViewModel (MVVM) 2](#_Toc2771812)

[2.3 Android Components 2](#_Toc2771813)

[2.3.1 Activity 2](#_Toc2771814)

[2.3.2 Intent 3](#_Toc2771815)

[2.3.3 ViewPager 5](#_Toc2771816)

[2.3.4 RecyclerView and Adapter 5](#_Toc2771817)

[3 Implementation 5](#_Toc2771818)

[3.1 Application Outline 5](#_Toc2771819)

[3.1.1 Description 5](#_Toc2771820)

[3.1.2 Material Design and Layout 5](#_Toc2771821)

[3.1.3 Development and Process 5](#_Toc2771822)

[3.2 Project Requirements 5](#_Toc2771823)

[3.2.1 Feature A 5](#_Toc2771824)

[3.2.2 Feature B 5](#_Toc2771825)

[3.2.3 Feature C 5](#_Toc2771826)

[3.3 Technology Stacks 5](#_Toc2771827)

[3.3.1 Data Flow 6](#_Toc2771828)

[3.3.2 Kotlin over java 6](#_Toc2771829)

[3.3.3 Unit Testing 6](#_Toc2771830)

[4 Results 6](#_Toc2771831)

[5 Conclution 6](#_Toc2771832)

[6 References 6](#_Toc2771833)

[References 7](#_Toc2771834)

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

## Open Data

There must always be text or a new subheading below each heading. Do not place a figure or table below a heading with no text in between.

Label each figure and table appropriately. Provide a number, title and reference (if needed) below each figure and above each table. Make sure to mention all figures and tables in the text. Each figure and table must be explained in the text and referred to by its number (… as figure 1 illustrates. /as summarized in table 1.).

Apply the Figure style for each image. This is necessary in order to prevent a page break from occurring between the figure and its caption. The Figure caption style is applied for the figure’s caption. This causes the figures to be numbered automatically.

1. Virtual studies completed by Metropolia students in the academic year 2009-2010.

There must always be text between a figure or table and a new heading.

## 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.

## Android Components

### 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.

### Intent

There must always be text or a new subheading below each heading.

Use the Quotation style for an indented quotation. For the last paragraph immediately before the quotation, use the Body Text before Quotation or List style.

If a direct quotation is several lines long, indent the quotation and use single (1.0) line spacing. No quotation marks are used then. Always provide a reference to the source. If the direct quotation is shorter than two lines, include it in the body of the text in quotation marks, and provide a reference to the source.

After an indented direct quotation, continue the text at the left margin using the Body text style.

Use the Bulleted list style for an in-text list:

* This is the first list item.
* The second item of the list contains a long text that spans more than one row. The left margin will be automatically justified.
* This is the third list item.
* This is the fourth list item.

The items on the bulleted list begin with a capital letter. An item ends in a full stop if each item on the list is a full sentence.

The list items begin with a lower-case letter if the list items are not sentences. The last item is followed by a full stop. Thus, a thesis consists of

* words
* sentences
* paragraphs
* sections.

After the list, the text continues from the left margin in the Body text style.

You can insert numbered formulas that are displayed on separate rows:

(1)

Insert a new formula by selecting Insert/Quick parts/Formula (using the formula function). If you want to use Microsoft equation editor instead of the newer formula function, select Insert/Quick parts/Formula (using MS Equation).

An example of a listing is given below. Use Code line style to mark code lines, and create indentations with the Tab key. The caption should follow the Listing caption style.

def inventory():

cur = db.cursor()

sql = "SELECT Description FROM OBJECT WHERE Location='PLAYER'"

cur.execute(sql)

if cur.rowcount>=1:

print("You carry the following items:")

for row in cur.fetchall() :

print (" - " + row[0])

else:

print("You don't carry anything.")

return

1. A Python subroutine that outputs information about objects in possession of a player.

### ViewPager

### RecyclerView and Adapter

# 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

## Project Requirements

### Feature A

### Feature B

### Feature C

## 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.