MOBILE PROJECT PART A

SUBMISSION 1

DAN

GRIZLI777

Contents

Introduction

This document contains the result of my research into various aspects of mobile application development. It covers some of the features of different mobile platforms, programming environments for mobile application development, programming languages and libraries.

Features of Different Mobile Platforms

Features of mobile phones:

- Portability
- Self contained unit with battery
- Connectivity, wireless internet (3g, 4g, 5g), Wi-Fi
- Hi resolution touch screens
- Camera
- Integrated CPU
- Storage space
- Sensors

Pocket computers:

Pocket computers are largely obsolete nowadays as mobile phones have evolved into pocket computers in and of themselves. What the makers of Pocket PC's intended to accomplish is now superseded by a standard smart phone.

Tablets:

Features of tablets tend to be very similar to that of a mobile phone with the primary difference being a larger screen size. As a result, they are somewhat less portable than a mobile phone, yet more portable than a laptop. They utilise a touch screen instead of a keyboard, although often a keyboard can be plugged in by the user if desired.

Digital assistants:

Digital assistants are computer programs designed to facilitate the user interacting with the computer by using voice commands. The most widely know digital assistants are Siri which is used for iPhones, and Cortana which is Microsoft's digital assistant for the computer.

Self contained units such as the "Google Home" and "Amazon Echo" are always listening devices designed to be placed in the home, which will respond to pre-determined commands; allowing the user to interact access the internet and retrieve information, connect with other devices such as TV's and play songs, without the need for physical input such as a mouse or keyboard.

Target Platform

The target platform for our application is Android; this selection is primarily due to hardware availability. To develop an application for an iOS device, access to a Mac is required, which is something I do not currently have access to. In addition, I have access to an Android mobile phone, so testing can be performed on a real device.

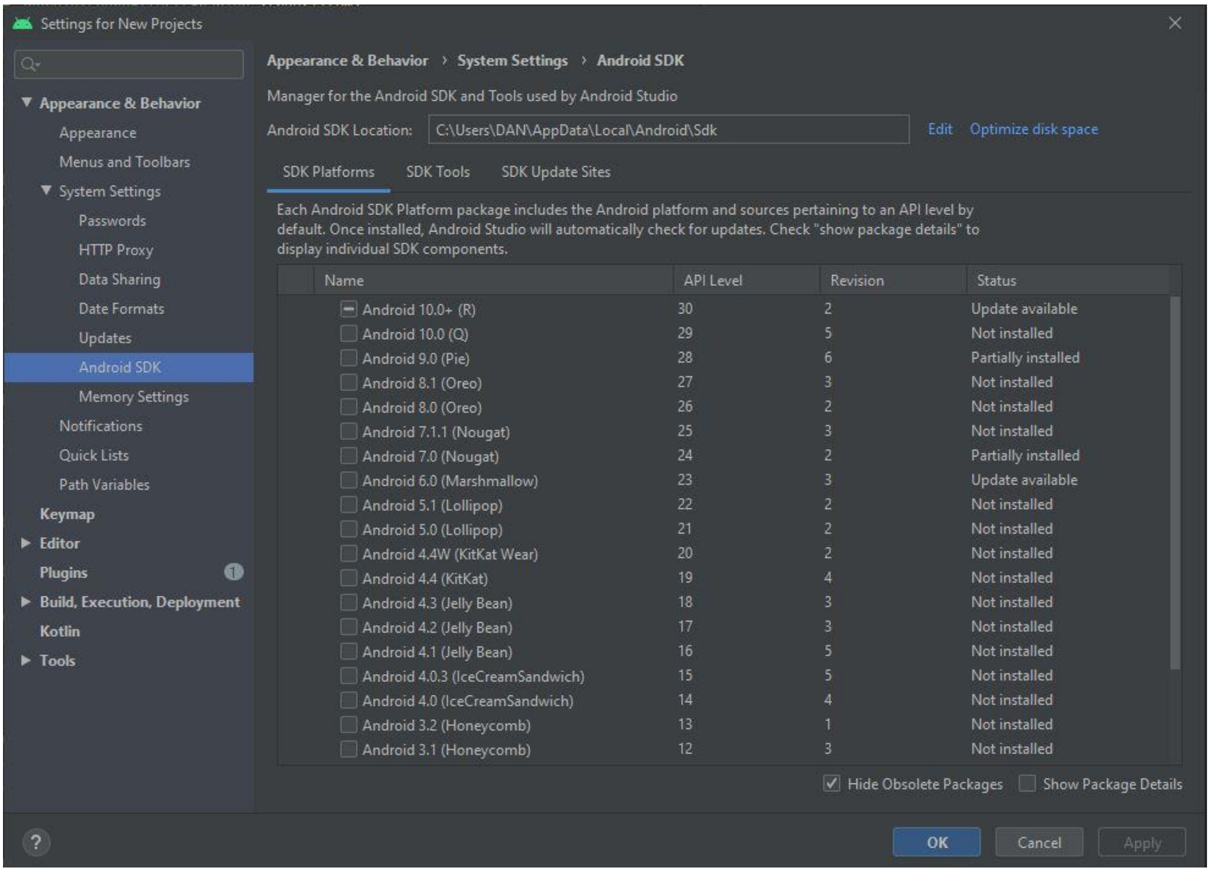
Chosen IDE

Android Studio is the native environment for the development of Android apps. While Android apps *can* be built in Visual Studio, Android Studio is superior for the development of apps that are targeting *only* Android.

Both choices have inbuilt debuggers and compilers. Android Studio is geared towards Android app creation and is the native choice for Android app development. As such it is a little bit easier to make an Android app using Android Studio.

Another reason for my choice is simply that I have not yet used Android Studio or Java to build something and I would like to learn a new environment.

IDE Setup



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

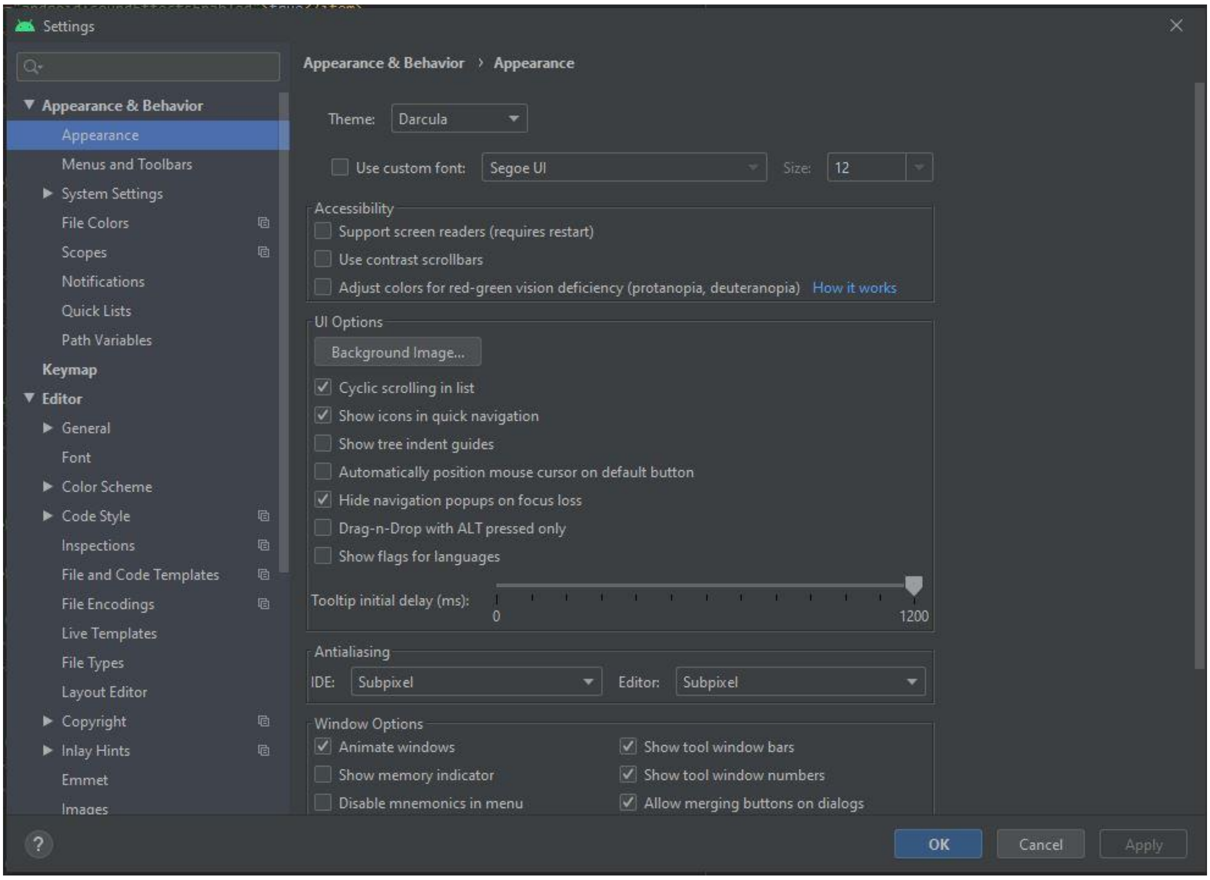
Android SDK Location: C:\Users\DAN\AppData\Local\Android\Sdk Edit Optimize disk space

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
⊟ Android 10.0+ (R)	30	2	Update available
☐ Android 10.0 (Q)	29	5	Not installed
☐ Android 9.0 (Pie)	28	6	Partially installed
☐ Android 8.1 (Oreo)	27	3	Not installed
☐ Android 8.0 (Oreo)	26	2	Not installed
☐ Android 7.1.1 (Nougat)	25	3	Not installed
☐ Android 7.0 (Nougat)	24	2	Partially installed
☐ Android 6.0 (Marshmallow)	23	3	Update available
☐ Android 5.1 (Lollipop)	22	2	Not installed
☐ Android 5.0 (Lollipop)	21	2	Not installed
☐ Android 4.4W (KitKat Wear)	20	2	Not installed
☐ Android 4.4 (KitKat)	19	4	Not installed
☐ Android 4.3 (Jelly Bean)	18	3	Not installed
☐ Android 4.2 (Jelly Bean)	17	3	Not installed
☐ Android 4.1 (Jelly Bean)	16	5	Not installed
☐ Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
☐ Android 4.0 (IceCreamSandwich)	14	4	Not installed
☐ Android 3.2 (Honeycomb)	13	1	Not installed
☐ Android 3.1 (Honeycomb)	12	3	Not installed

☑ Hide Obsolete Packages ☐ Show Package Details

The above screenshot shows the various SDK's that are available. The necessary SDK's will be determined by your target API level. Too high and the app may not function on very many devices; only the very latest. Too low will restrict the functionality of the app.



Appearance & Behavior › Appearance

Theme: Darcula

□ Use custom font: Segoe UI Size: 12

Accessibility

□ Support screen readers (requires restart)
□ Use contrast scrollbars
□ Adjust colors for red-green vision deficiency (protanopia, deuteranopia) How it works

UI Options

Background Image...

☑ Cyclic scrolling in list
☑ Show icons in quick navigation
□ Show tree indent guides
□ Automatically position mouse cursor on default button
☑ Hide navigation popups on focus loss
□ Drag-n-Drop with ALT pressed only
□ Show flags for languages

Tooltip initial delay (ms): 0 1200

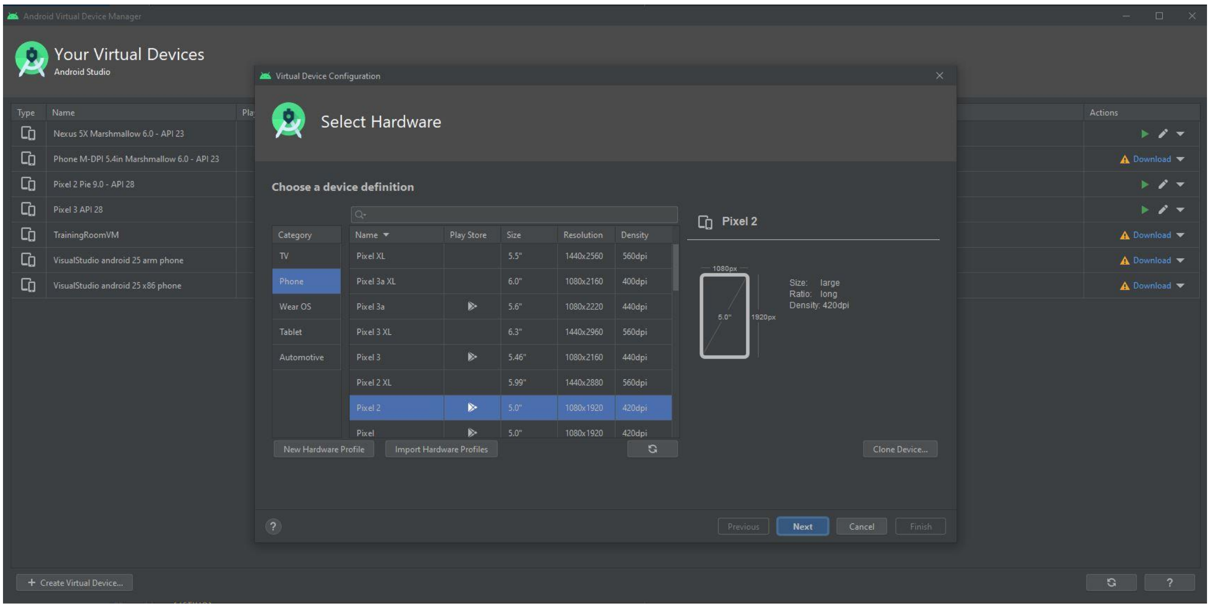
Antialiasing

IDE: Subpixel Editor: Subpixel

Window Options

☑ Animate windows ☑ Show tool window bars
□ Show memory indicator ☑ Show tool window numbers
□ Disable mnemonics in menu ☑ Allow merging buttons on dialogs

The above screenshot shows some general settings for the IDE's display.



Type	Name
	Nexus 5X Marshmallow 6.0 - API 23
	Phone M-DPI 5.4in Marshmallow 6.0 - API 23
	Pixel 2 Pie 9.0 - API 28
	Pixel 3 API 28
	TrainingRoomVM
	VisualStudio android 25 arm phone
	VisualStudio android 25 x86 phone

Choose a device definition

Category
TV
Phone
Wear OS
Tablet
Automotive

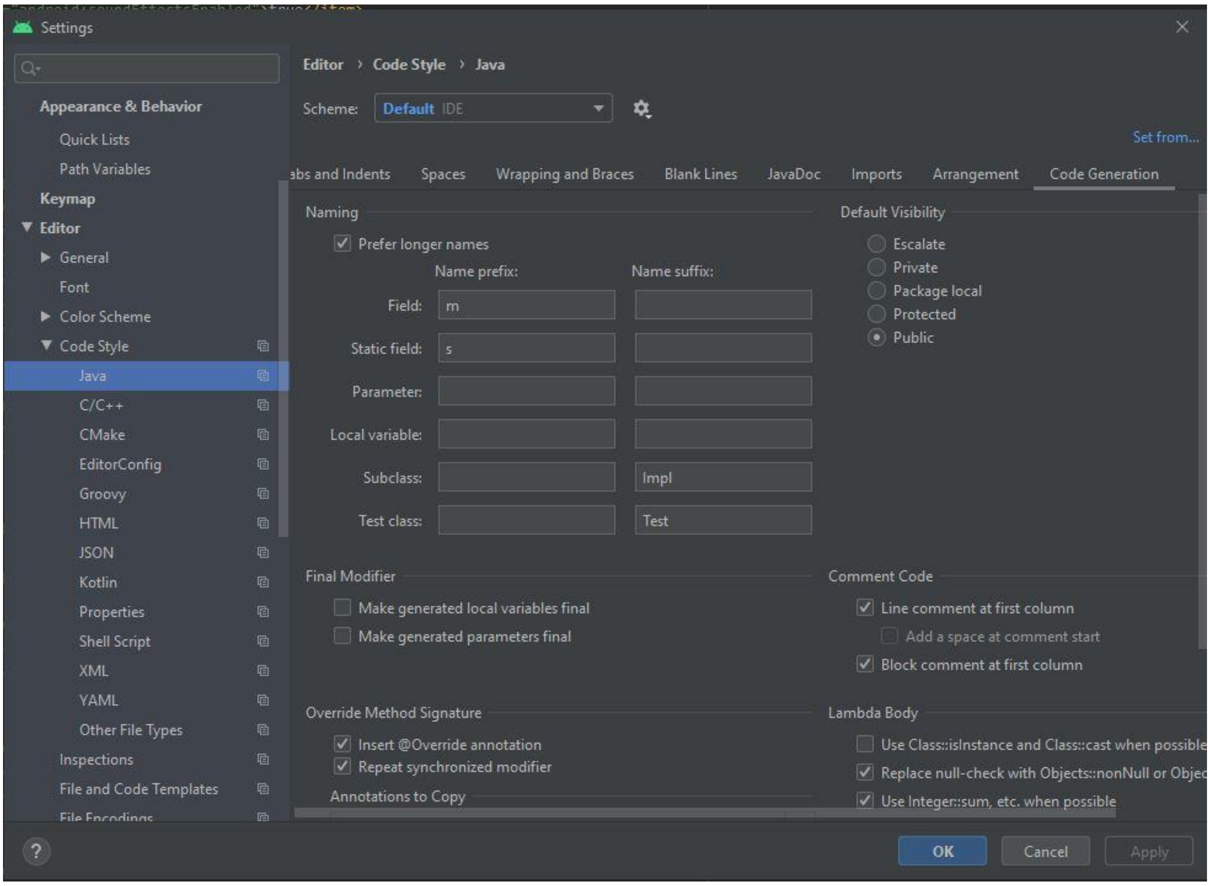
Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2560	560dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi
Pixel 2 XL		5.99"	1440x2880	560dpi
Pixel 2	▶	5.0"	1080x1920	420dpi
Pixel	▶	5.0"	1080x1920	420dpi

Pixel 2

1080px 1920px 5.0"

Size: large
Ratio: long
Density: 420dpi

This shows the AVD (Android Virtual Device) manager, where you can manage the emulators used to test your application.



Scheme: Default IDE

Naming

☑ Prefer longer names

	Name prefix:	Name suffix:
Field:	m	
Static field:	s	
Parameter:		
Local variable:		
Subclass:		Impl
Test class:		Test

Default Visibility

○ Escalate
○ Private
○ Package local
○ Protected
● Public

Final Modifier

□ Make generated local variables final
□ Make generated parameters final

Comment Code

☑ Line comment at first column
□ Add a space at comment start
☑ Block comment at first column

Override Method Signature

☑ Insert @Override annotation
☑ Repeat synchronized modifier

Annotations to Copy

Lambda Body

□ Use Class::isInstance and Class::cast when possible
☑ Replace null-check with Objects::nonNull or Objec
☑ Use Integer::sum, etc. when possible

This screen allows you to adjust your code style, including default prefixes for field types. This can save a lot of time during the development phase.

Object Oriented Principles

The four main object-oriented principles are Abstraction, Polymorphism, Inheritance and Encapsulation.

In keeping with these principles, We will create classes for the creation of our objects (abstraction), we will reuse the same code for different purposes where we can (polymorphism), we will use abstract classes and inherit where we can which will help avoid code duplication (inheritance) and we will create methods with inputs and outputs to decouple the code and keep things tidy (encapsulation).

Language Integration

In web design, HTML is the language which is responsible for defining the structure of a website. CSS is used to style the HTML while JavaScript is used to add interactivity and make the website more dynamic. Using a combination of these three languages is fundamental to producing a quality, engaging website.

AJAX is a set of techniques which allows data to be transferred to a server without interfering with the presentation layer of the application. It runs asynchronously and in this way it effectively decouples the data layer from the presentation layer of the application.

UI Design Principles

When designing the app, considerations will be made to keep the user interface simple, employing only what is necessary. It will be organised in a way that makes as much logical sense as possible. When a user performs an action, the application will respond to the user what it is doing, so that the user is informed of what is happening. It will be tolerant by allowing re-input if mistakes are made and considerations will be made to ensure that if incorrect data is attempted to be passed in, it will inform the user of the user's mistake.

Web Services

Web services are systems used to transfer information using the internet. Using HTTP we can make our application directly access an application via the internet and request data from a web server.

XML is the code used to communicate to and from a web service.

To briefly explain how this works in practice; our application will send an XML message to a server using the hypertext transfer protocol (HTTP) and the server will reply with an XML message. Our application will then decode the XML and utilize the data. For this application, the XML will end up being used to create Java objects.

Libraries

Android Libraries are structurally similar to an application module, with the main difference being that they do not compile into an Android Application Package (APK). They contain methods and resources which can be re-used.

Our application is going to utilize the Gson library; which will help us to use the data we request from a server and convert it into Java objects.

We are also going to use the Http3 library which contains components to help us connect to the server via http.

Both libraries are free for use and will achieve the desired goal. Both Okhttp3 and Gson are licensed under the Apache 2.0 licence which allows commercial use, modification, distribution, patent use and private use.

References

https://en.wikipedia.org/wiki/Principles_of_user_interface_design

https://www.tutorialspoint.com/webservices/what_are_web_services.htm

https://developer.android.com/studio/projects/android-library