Mobile Application Development

Orientation

Week1 Android Architecture & Android Studio

Yi SUN

Kobe Institute of Computing

Overview of this course

This course offers a comprehensive introduction to mobile application development, with a focus on the Android platform. Participants will learn how to set up the Android Studio development environment, program using Kotlin, and design user interfaces for Android applications. The course covers essential concepts such as the MVVM (Model-View-ViewModel) architecture, integration of RESTful APIs, working with location services, and Firebase for backend data management. By the end of the course, students will complete a project-based application that integrates these features.

Course Objectives

1. Set up and configure the Android Studio development environment.
2. Understand the basics of Kotlin programming language.
3. Build user interfaces (UI) using Android components.
4. Apply the MVVM architecture in mobile app development.
5. Work with JSON, HTTP requests, and integrate RESTful APIs.
6. Implement location services and map systems in applications.
7. Use Firebase for real-time database management.
8. Work with the Canvas component to manipulate images.
9. Develop a fully functioning mobile app through hands-on practice.

Learning Methods

- On-demand video lectures.
- Hands-on coding exercises and assignments.
- Forum discussions for doubt clearance and peer interaction.

Course Outline (1)

- week1: Introduction to Mobile Development & Android Studio Setup
- week2: Introduction to Kotlin Programming
- week3: Advanced Kotlin Concepts
- week4: Building User Interfaces with Android UI Components
- week5: Handling User Input and Interactions
- week6: Introduction to MVVM Architecture
- week7: Building a Simple MVVM-based Application

Course Outline (2)

- week8: Networking: Working with JSON & HTTP Requests
- week9: Introduction to RESTful APIs
- week10: Working with Location and Maps in Android
- week11: Firebase Database Integration
- week12: Advanced Firebase Features
- week13: Working with Android Canvas
- week14: Mobile App Security and Best Practices
- week15: Project Workshop 1: Building a Complete Mobile Application
- week16: Project Workshop 2: Finalizing and Deploying the Mobile Application

Preparation and Review

Students are expected to spend approximately 2 hours outside of class for every class attended.

This will involve:

- Reviewing video lectures.
- Working on assignments and exercises.
- Participating in forum discussions and self-study.

Evaluation Methods

- Assignments & Exercises: 60%
- Final Project: 30%
- Participation in Forum Discussions: 10%

Requirements for participants

- Basic knowledge of computer programming concepts.
- Basic knowledge of Java programming.
- Basic knowledge of object-oriented programming concepts.
- A Windows computer with internet access and no less than 16GB memory.
- Android Studio development environment.
- Commitment to regular study.

Textbook, Notes, and Resources

Textbook will not be used. Instead, PPT materials and subject's related materials would be shared to the participants prior to the class.

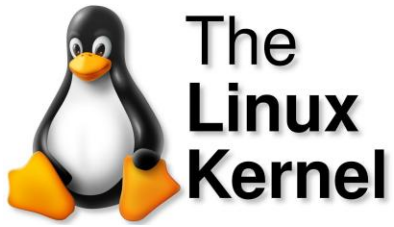
Reference Resource

- Android Developer Documentation
- Kotlin Documentation
- Android Developer Guides
- Teach Computer Science with Kotlin

Android Development

Introduction to Android Architecture

- **Overview**
  - Android is an open-source operating system mainly used for mobile devices, created by Google.
  - It is based on a modified version of the Linux kernel.



Five Main Layers of Android Architecture

1. **Linux Kernel**
    - Core part of the Android operating system.
    - Manages hardware and system level functions.
2. **Native Libraries**
    - Libraries specific to Android, such as OpenGL, media frameworks, etc.
3. **Android Runtime (ART)**
    - Manages the execution of the app, and performs the translation of the application bytecode into native instructions.

4. **Application Framework**
    - Provides many Java classes as the structural role, which helps in building the actual application.

5. **Applications**
    - The top layer where all the apps like Contacts, Browser, Calculator, etc., resides.

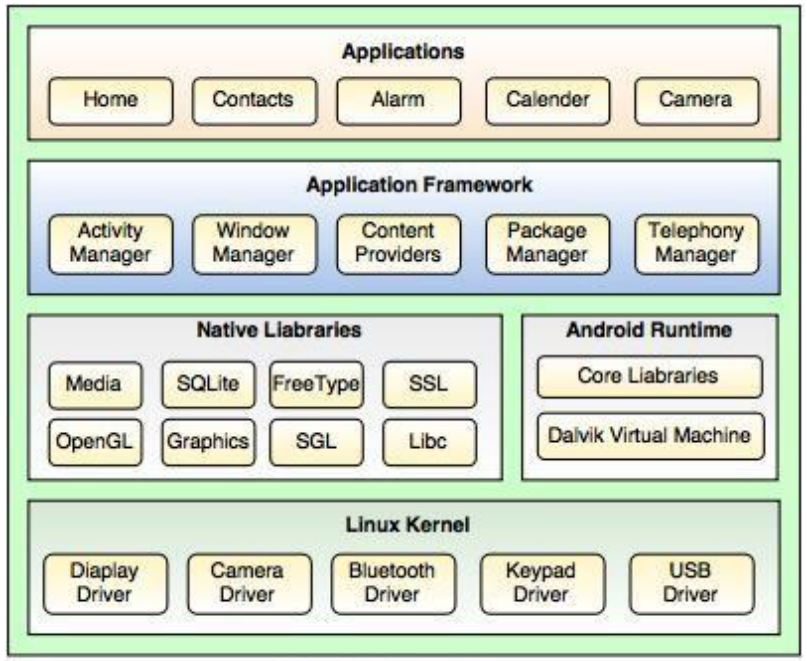


Fig. Android Architecture

Key Android System Components (1)

1. **Activities**
    - Single, focused screen user interacts
    - Manages user interface and interactions
    - Example: Email app's compose screen
2. **Services**
    - Long-running operations in background
    - No user interface
    - Example: Music playback, network operations

Key Android System Components (2)

3. **Broadcast Receivers**
    - Responds to system-wide broadcast messages
    - Enables communication between apps
    - Example: Battery low notification, incoming calls

System-Level Features (1)

1. **Content Providers**
    - Manages shared app data
    - Enables data sharing between applications
    - Example: Contacts, media files
2. **System Services**
    - Location Services
    - Notification Manager
    - Package Manager
    - Window Manager

System-Level Features (2)

3. **Hardware Access**
    - Camera API
    - Bluetooth
    - Sensors (GPS, Accelerometer)
    - NFC

Android Studio Installation Guide

Step 1: Download Android Studio

1. Visit the official download page: https://developer.android.com/studio
2. Click the "Download Android Studio" button
3. Accept the Terms and Conditions

Step 2: Run the Installer

1. Double-click the downloaded installation file
2. For Windows users, select "Yes" to allow the app to make changes
3. Click "Next" to start the installation wizard

Step 3: Choose Components

Select the following components to install:

- Android Studio
- Android SDK
- Android Virtual Device
- Performance (Intel ® HAXM)

Step 4: Installation Location

1. Choose installation directory
2. Ensure sufficient disk space (minimum 8GB recommended)
3. Click "Next" to continue

Step 5: Installation Process

The installation may take some time depending on your internet speed and computer performance.

Step 6: First Launch Setup

1. Choose whether to import previous settings
2. Select UI theme (Dark/Light)
3. Complete the initial setup

Troubleshooting Tips

Common issues and solutions:

- **RAM Issues**: Ensure your computer has at least 8GB RAM
- **SDK Download Fails**: Check internet connection or use offline SDK
- **HAXM Installation**: Enable virtualization in BIOS
- **Gradle Sync Issues**: Check firewall settings

System Requirements

Windows

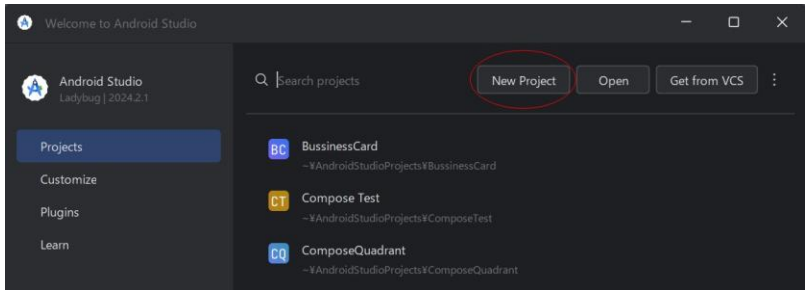
- 64-bit Microsoft® Windows® 10/11
- 8GB RAM minimum (16GB recommended)
- 8GB disk space minimum
- 1280 x 800 minimum screen resolution

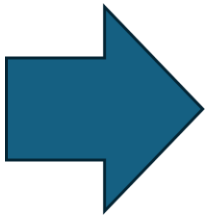
Creating Your First Android Project

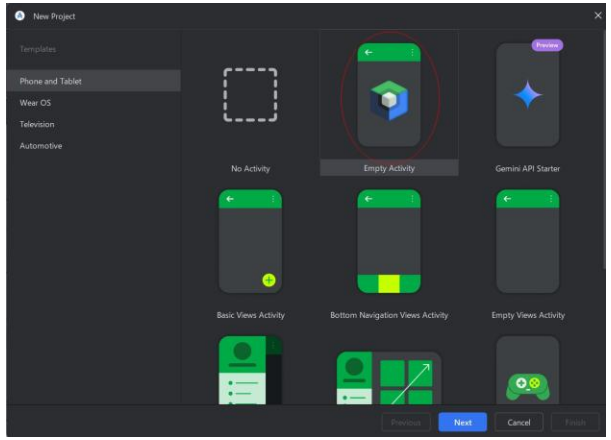
Hello World in Android Studio

Step 1: Create New Project

1. Open Android Studio
2. Click "New Project" from welcome screen
3. Select "Empty Activity" template
4. Click "Next"

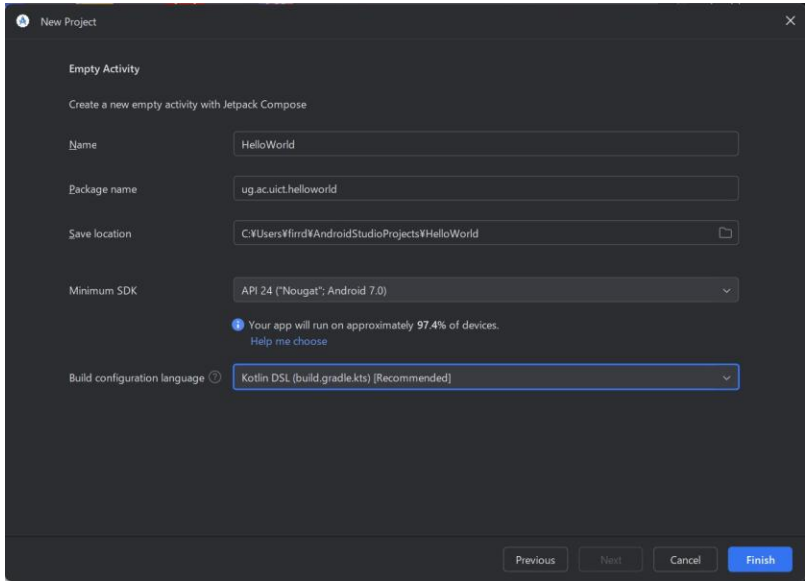






Step 2: Configure Project

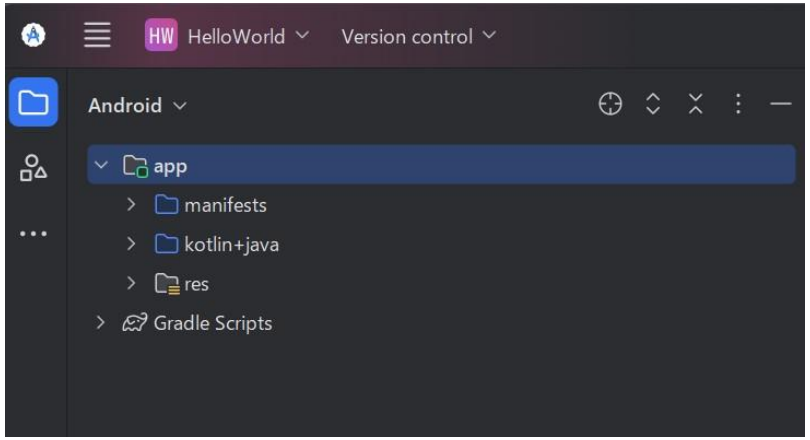
1. **Name**: HelloWorld
2. **Package name**: ug.ac.uict.helloworld
3. **Save location**: Choose project directory
4. **Minimum SDK**: API 24 (Android 7.0)
5. **Build configuration language**: Kotlin DSL (build.gradle.kts)



Project Structure Overview

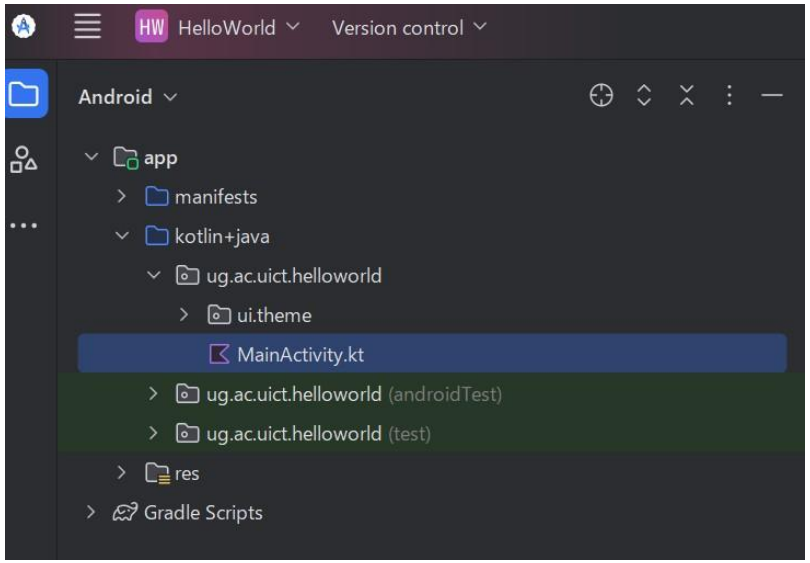
Key project files and directories:

- **app > manifests**: AndroidManifest.xml
- **app > kotlin+java**: Contains source code
- **app > res**: Resources (layouts, images)
- **Gradle Scripts**: Build configuration



Main Activity Layout

Located in: `app > kotlin+java > ug.ac.uict.helloworld > ui.theme > MainActivity.kt`



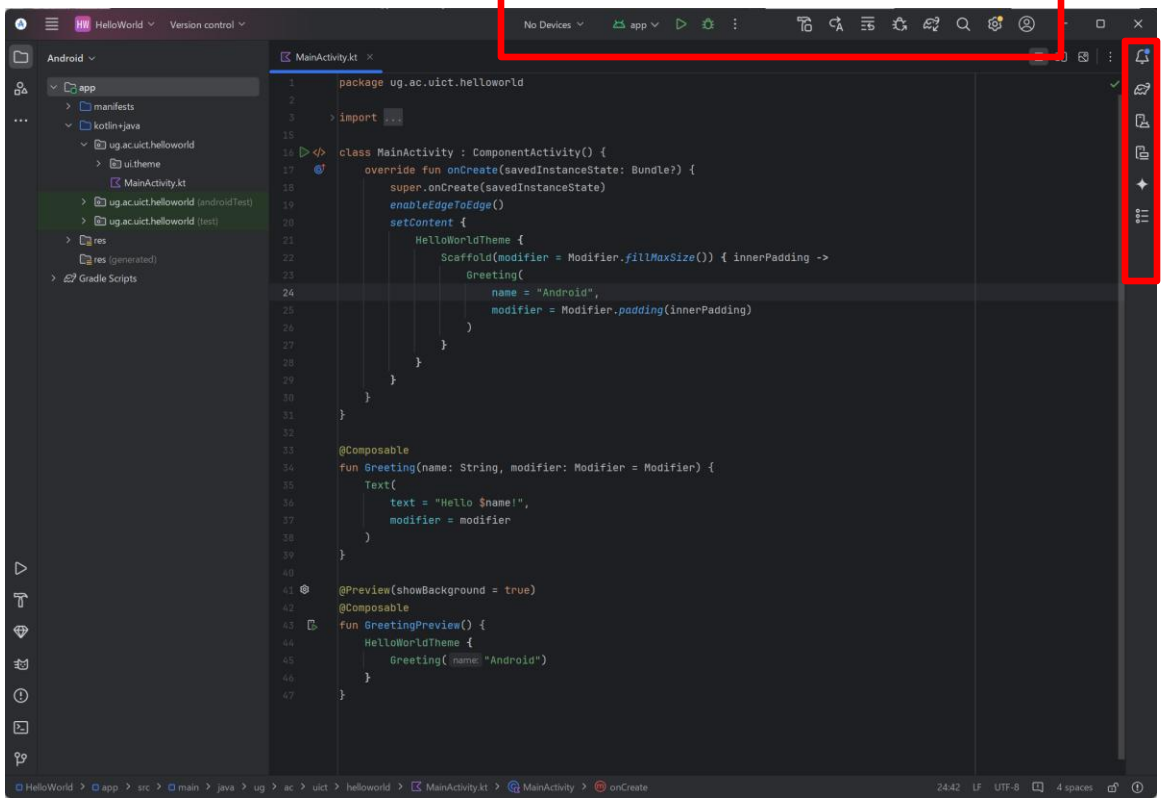
The interface of Android Studio

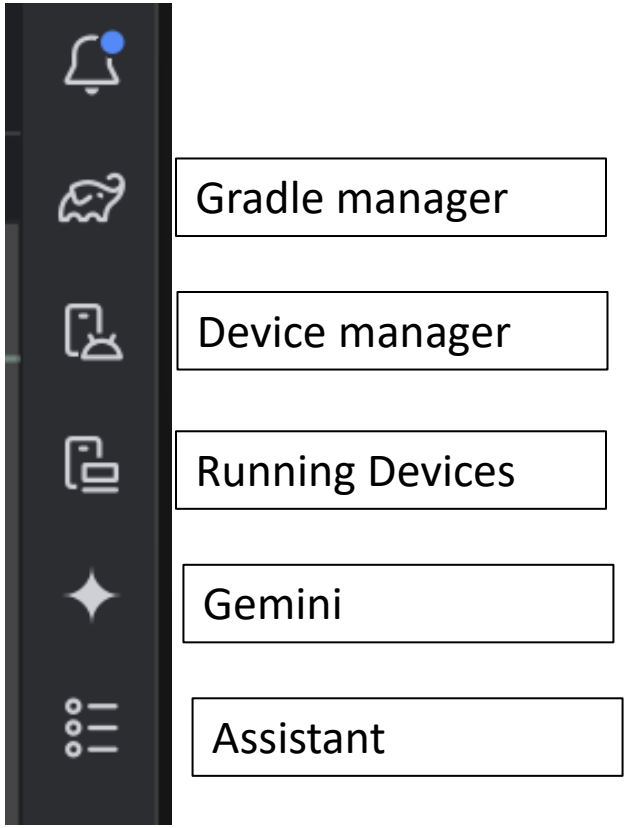
Connect to physical device

run app

debug

build





Gradle manager

Device manager

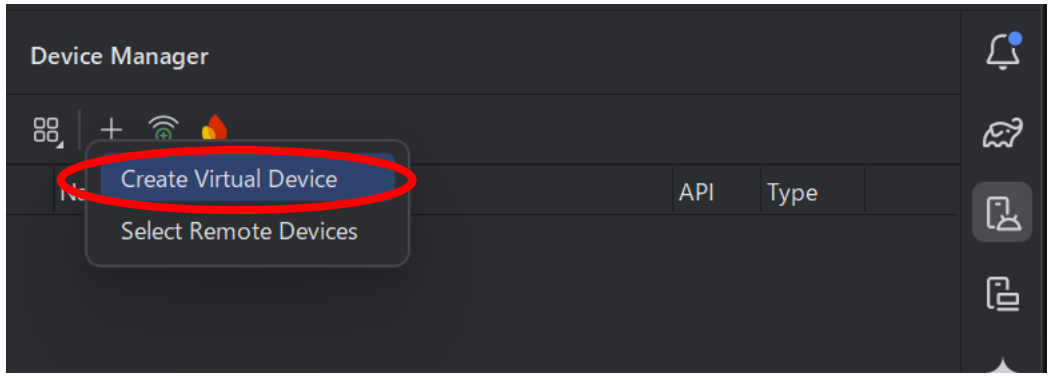
Running Devices

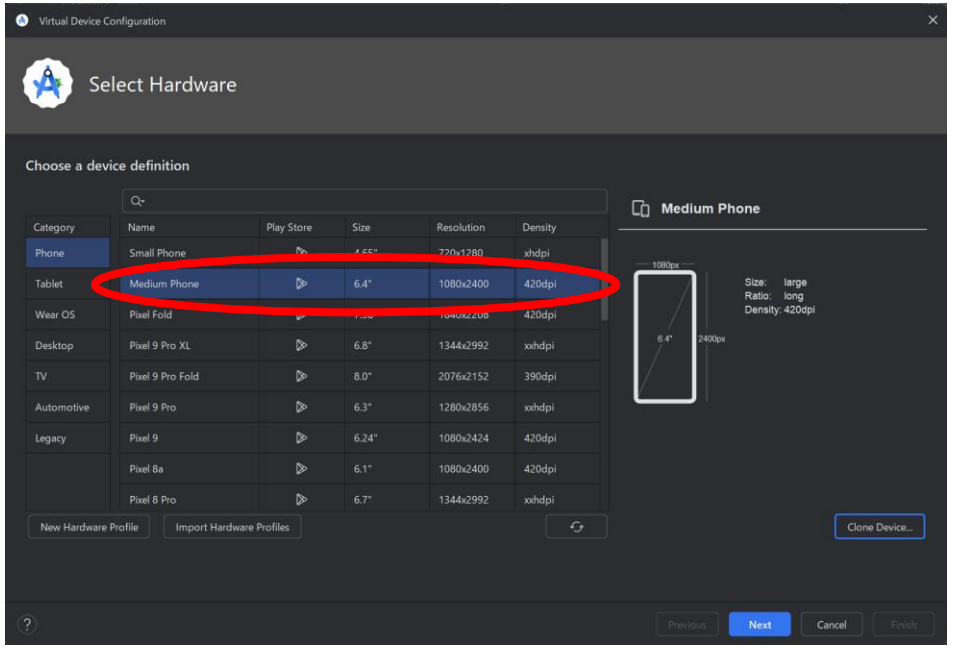
Gemini

Assistant

Create a new Virtual Device

1. Click the Device manager
2. Click the + and select the “Create Virtual Device”
3. Select Hardware “Medium Phone”



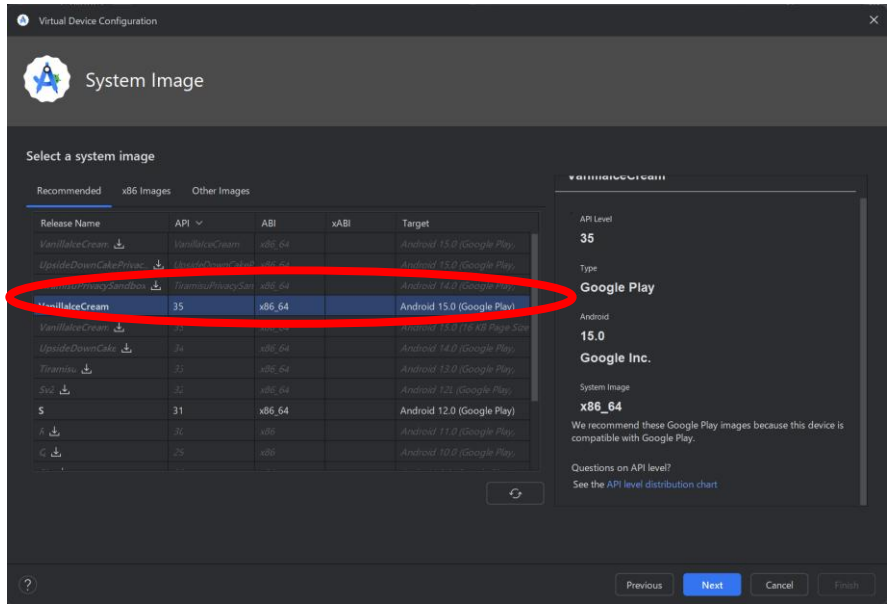


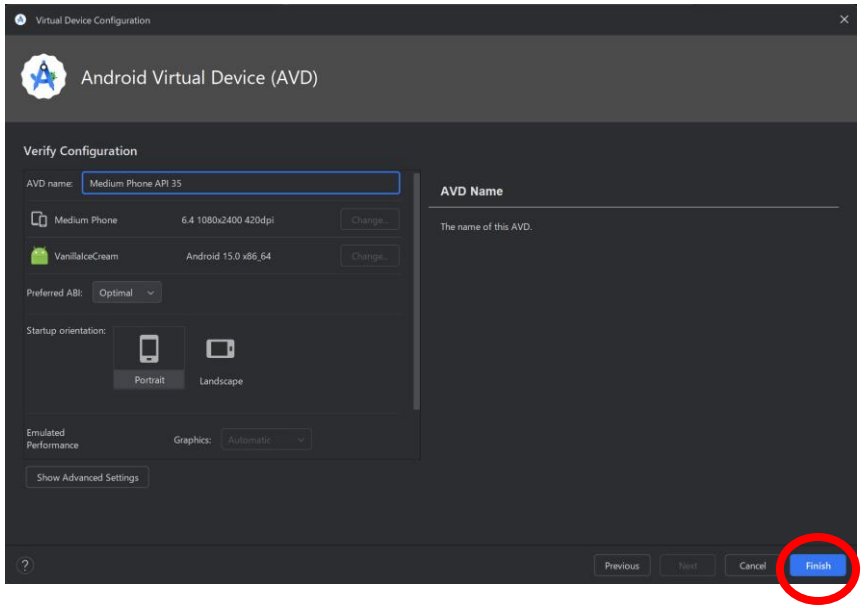
Create a new Virtual Device

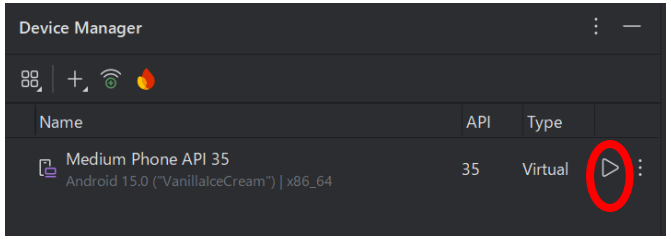
4. Select the System Image “VanillalceCream” API 35 Android 15.0

5. Set up the AVD name and Click “Finish” Button.

6. From the Device Manager find the AVD name and Click the ▷ button to start up the Virtual Android Device.







Run the app

1. Click the “Running Devices” check the AVD already running.
2. Click the “ run app” button to run the application.

6. From the Device Manager find the AVD name and Click the ▷ button to start up the Virtual Android Device.