

Configuration Manual

MSc Research Project
MSc in Cyber Security

Nash John
Student ID: 21154341

School of Computing
National College of Ireland

Supervisor: Dr. Vanessa Ayala-Rivera

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Nash Jacob John
Student ID: 21154341
Programme: MSc in Cyber Security **Year:** 2022
Module: MSc Research Project
Lecturer: Dr. Vanessa Ayala-Rivera
Submission Due Date: 15th December 2022
Project Title: Detection and Prevention of data transfer through Bluetooth by unauthorized devices on Android OS 8,9 & 10
Word Count: 794 **Page Count:** 4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nash Jacob John
Date: 15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Nash John
Student ID: 21154341

1 System Requirements

In this configuration manual, we will be looking into the software and hardware prerequisites needed to build and run the Android application. We will be looking in detail at the Software and Hardware requirements below sections:

1.1 Software:

- JDK - **11.0.13**
- Kotlin Version - **213-1.7.20-release-for-android-studio-AS6777.52**
- Gradle Version - **7.4**
- Gradle Plugin Version - **7.3.1**
- Android Minimum SDK - **API 26 (Android Oreo 8.0)**
- Android Maximum SDK – **API 31 (Android 12)**

1.1.1 External Library

To Implement this project the Android default framework doesn't provide the necessary library for our use case. I have used the below-mentioned library to build this application.

- Android-BLE-Library by NordicSemiconductors version – 2.6.0-alpha03
- Android Coroutine version – 1.6.4
- Android ViewModel version – 2.5.1
- Android LiveData version – 2.5.1

1.2 Hardware

The minimum requirement to build and run this application are mentioned below:

1.2.1 PC requirements

- RAM – **16 GB**
- Storage SSD – **512 GB**
- Processor - **11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz 3.11 GHz**
- Operating System - **Windows 11 Home Single Language, 64-bit operating system, x64-based processor**

1.2.2 Mobile Requirements (Android)

- RAM – **8 GB**

- ROM – **64 GB**
- Processor – **Dimensity 1200-AI Octa-Core**
- Operating System – **Android 12**

2 Software Tools

2.1 Android Studio

Android Studio, which is based on IntelliJ IDEA, is the preferred Integrated Development Environment (IDE) for building Android applications. In addition to IntelliJ's powerful code editor and developer tools, Android Studio now offers additional features that boost your productivity when creating Android apps, including a flexible build system based on Gradle, Apply You can modify the code and resources used by your running program without having to restart it. With the help of code templates, GitHub integration, tools, and frameworks for thorough testing, you may construct common app features and import example code, Lint tools can detect performance, usability, version compatibility, and other concerns, and since Google Cloud Platform is already supported and C++ and NDK are supported, it is straightforward to connect App Engine with Google Cloud Messaging.

2.1.1 Installation

Android is an open-source software developed and maintained by Google. We can download the software from the official android studio website. After downloading the software, we can install the software to our preferred Operating System using the default installation wizard.

2.1.2 Environment Step up

To successfully run compile and android applications, the Operating System should have java installed in the system. The minimum JDK version should be JDK 11. The JDK should be defined as HOME_VARIABLE in the windows environment, So the java version is accessible by the android studio during the compilation process.

2.1.3 Build Project

The code which was written for the application must be compiled. In Android, Gradle will compile our kotlin high-level code to machine-readable code. To perform this operation, we have the option called to build in Android Studio.

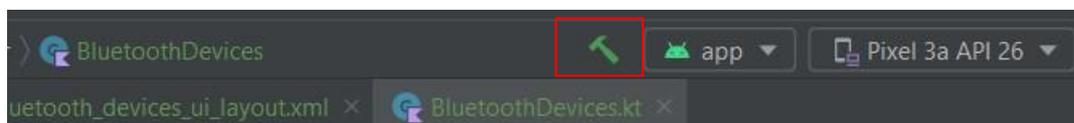


Figure 1: Build Button

2.1.4 Run Project

After generating a successful Gradle build by the Android studio. We can deploy our application into a testing environment. In our case, the testing environment is an android device. We have two options to attach a device. One is using the android emulator provided by the android studio or by plugging in our own device by USB cable to our PC or laptop.

After pressing the Run button as shown in Fig. 2, we could see our application running on a live device.



Figure 2: Run Button

3 Project Execution

In this section, I will explain how we could run the application on an Android Device

1. First, we need to get the project from GitHub:
<https://github.com/ni8522/Bluetooth-Packet-Filter.git>
2. Assuming we have, Android studio pre-installed on the device, or else follow the instructions from 2.1.1 to install android studio. On Opening up the Android studio, we will be asked to few options, from those options select “choose from the existing project” and direct the path to the “Bluetooth-Packet-Filter” file from the File explorer.
3. After opening the project, the android studio takes a couple of minutes to download and configure the Gradle.
4. After seeing the message “Gradle Build Successful”, we could Build the project as mentioned in 2.1.3.
5. According to the PC/Laptop capability we could choose to create an android emulator or attach any physical device. (My personal choice is using a physical device).
6. Attach an Android smartphone, which should be running Android OS 8. The android studio will show the attached device if the Android Debug Bridge (ADB) is working properly on the device.
7. We could the run button from Fig.2, after pressing the button, The android studio automatically sends the .apk file to our android device and automatically install’s our package, and Runs the application.

4 Results

After following the above steps thoroughly, the application should be successfully deployed on the android device, which is as shown below in Fig.3



Figure 3: Bluetooth Packet Filter

References

Meet Android Studio (no date) *Android Developers*. Available at: <https://developer.android.com/studio/intro> (Accessed: 1 December 2022).