National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Emmanuel Okechukwu Weje
Student ID: 19122411

School of Computing
National College of Ireland

Supervisor: Manuel Tova-Izquierdo

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Emmanuel Okechukwu Weje
Student ID:	19122411
Programme:	Cloud Computing
Year:	2019
Module:	MSc Research Project
Supervisor:	Manuel Tova-Izquierdo
Submission Due Date:	17/08/2020
Project Title:	Configuration Manual
Word Count:	1018
Page Count:	7

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.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
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.	☐

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

Emmanuel Okechukwu Weje
19122411

1 Introduction

In our research paper, we introduced a context aware heterogeneous mobile cloud computing approach which was aimed at augmenting the performance of smart mobile devices. In this configuration manual, we present how to set up the system and approaches we applied in order to replicate results achieved. This document is structured as follows. Section 2 presents the system environments and configurations, Section 3 shows how to set up Remote Configuration, Section 4 presents the Android Studio IDE Setup, Section 5 explains offloading options, Section 6 shows the experiments carried out and in Section 7, we show how to monitor performance metrics.

2 System Environment and Configurations

In this section, we present the various minimum requirements needed to set up our approach in order to replicate experiments. All have to be set up before progress can be made with setting up the experiments.

2.1 Software Requirements

- Windows 10 64-bit Operating System
- Java SE [1]
- Android Studio IDE [2]

2.2 Hardware Requirements

- Core i5 CPU or equivalent
- 8 GB RAM
- 500 GB HDD

[1]Java SE Development Kit: `https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html`

[2]Android Studio IDE: `https://developer.android.com/studio`

3 Remote Configuration

For remote configurations, we use Google Firebase Remote Config[3]. To set this up, a Firebase account is needed which can be created with the following steps.

1. Create account on Firebase[4]
2. Navigate to the console (https://console.firebase.google.com/) and add a new project
3. Download generated google-services.json file
4. Open up submitted application code named "CodeOffloadingApp" and add the downloaded file inside the "app" directory in the folder

After this has been done, the next step is to open up Firebase and navigate to Remote Config in the menu bar. When this has been done, add the default parameters for the app as shown in figure 1. This parameters can be changed at any time and will be reflected in the app.

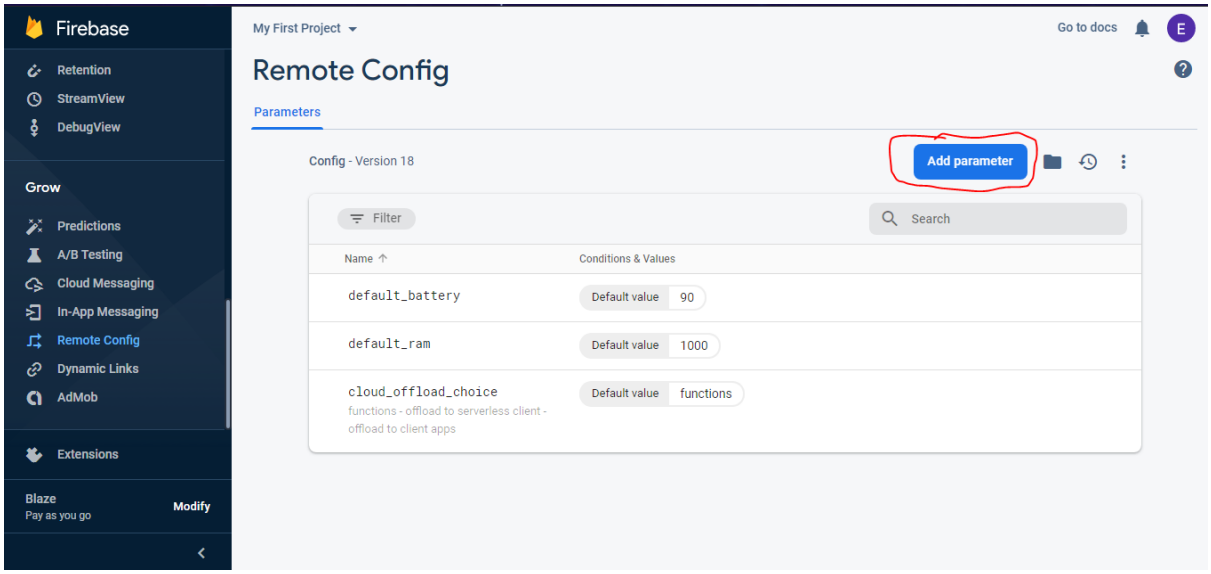


Figure 1: Remote Configuration

4 Android Studio IDE Setup

In order to run our experiments, Java and Android Studio IDE (ASI) have to be installed as per the software requirements. After these have been installed, we have to set up an Android Virtual Device (AVD) to run the experiments. The following steps are needed to create an AVD:

1. Open Android Studio IDE
2. Open the AVD manager by navigating to the top menu bar then Tools > AVD Manager. This is shown in figure 2.

[3]Firebase Remote Config: `https://firebase.google.com/docs/remote-config`
[4]Firebase: `https://firebase.google.com/`

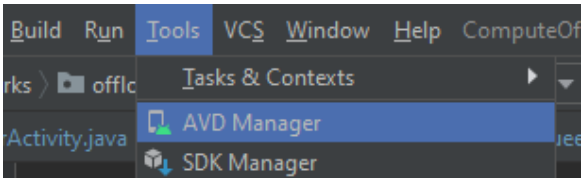


Figure 2: Open AVD Manager

3. Click on "Create Virtual Device" at the bottom left corner as shown in figure 3.

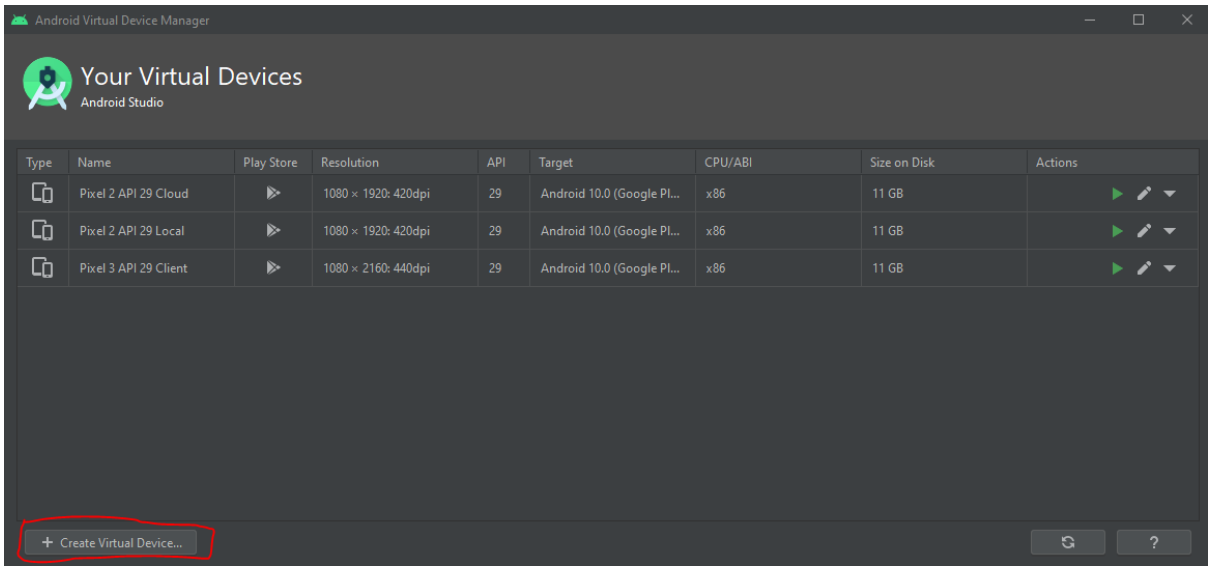


Figure 3: AVD Manager

4. Follow prompt and choose device configurations

5. Start the newly created AVD by clicking the green play button shown shown under actions. This is shown in figure 4

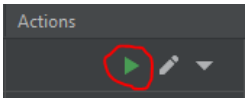


Figure 4: Start AVD

5 Offloading Options

For our offloading options, we have serverless functions and remote mobile clients. The serverless functions have been deployed already so in this section, we show how to set up the remote mobile client. Set up the remote mobile client by taking the following steps:

1. Unzip submitted code titled "Code_Offload_Client"

2. Double click "build.gradle" file in root folder to open application in ASI

3. Run the application by clicking on the green play button on the top menu bar. This will run the application in any available AVD

After the following steps have been taken, the device will be ready to receive offloaded tasks.

6 Experiments

For our experiments, we built a mobile application that contains both an Optical Character Recognition (OCR) system and an N-Queens Problem. In this section, we show steps on how to set up and run this application.

1. Unzip submitted code titled "CodeOffloadingApp"
2. Double click on "build.gradle" file in the root folder of the application in order to open it in ASI
3. Click on the green play button on the top menu bar to run the application. This is shown in figure 5.

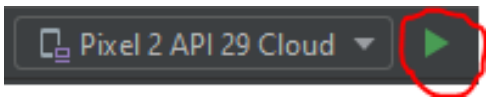


Figure 5: Run Project in ASI

The above steps will open up the application in the available AVD. This is shown in figure 6.

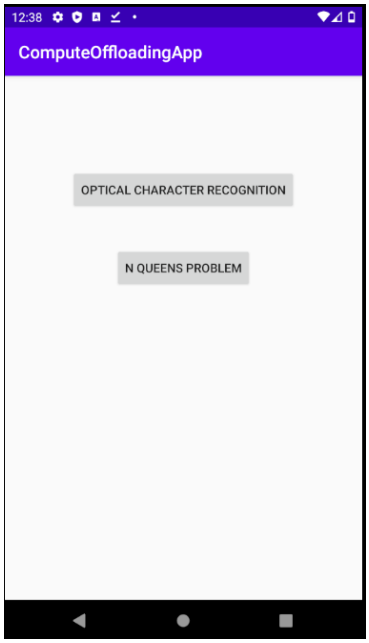
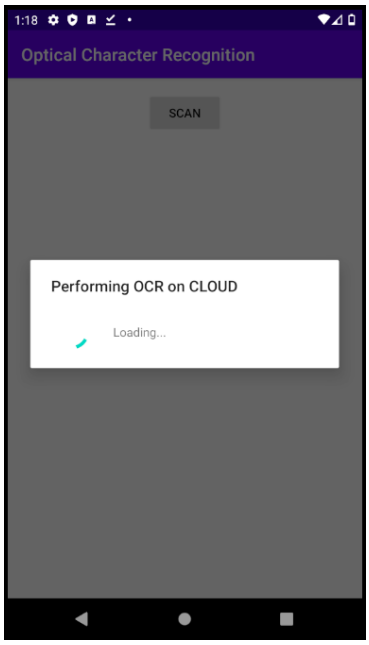


Figure 6: Code Offloading Application

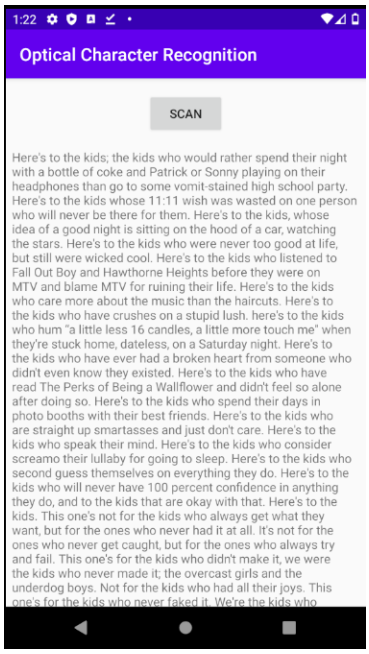
6.1 Experiment 1 / Optical Character Recognition

To run this experiment, click on the "Optical Character Recognition" button as shown in figure 6. This leads to the OCR screen as shown in figure 7a. In order to proceed with the experiment, click on the scan button to select an image to scan. Depending on the context configuration of the AVD, our decision making algorithm will decide on whether to offload the task or not. For our current device, the battery level is 20%, 700 MB RAM and excellent network. With this context, the expected offload decision should be cloud and the results are shown below in figures 7b and 7c.

(a) Optical Character Recognition Test.



(b) Running task using cloud.

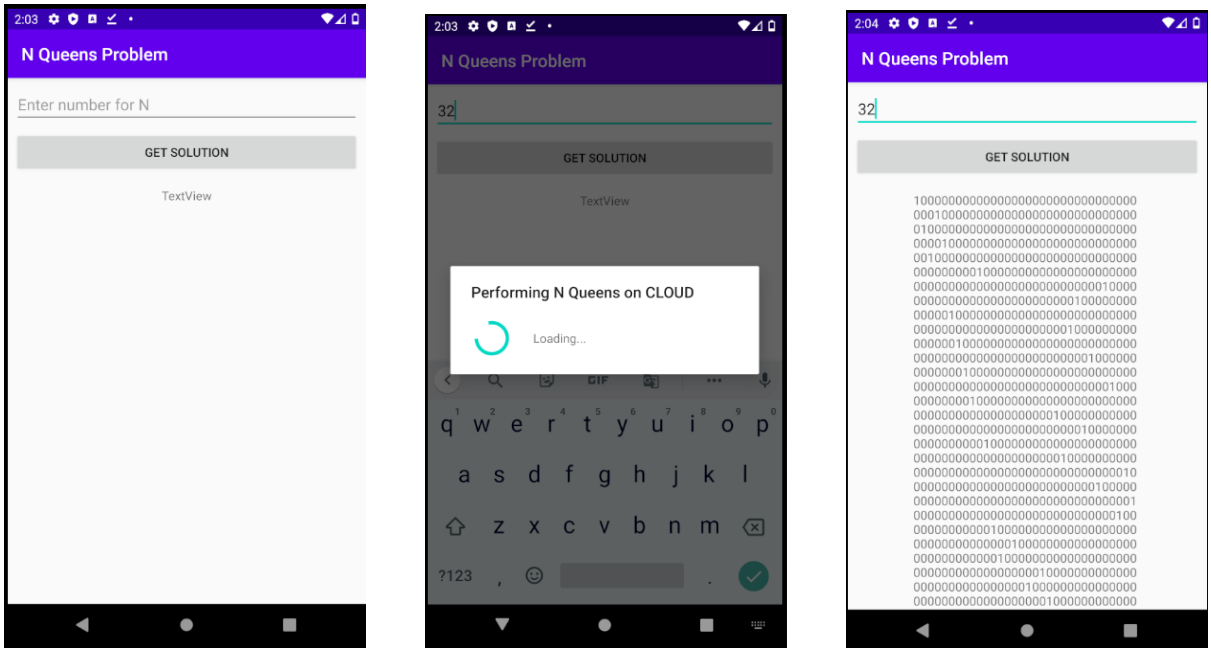


(c) OCR cloud task result.

Figure 7: OCR Experiment

6.2 Experiment 2 / N-Queens Problem

To run this experiment, click on the "N Queens Problem" button as shown in figure 6. This leads to the N Queens Problem screen as shown in figure 8a. The next step is to enter the number for N which is the number of queens we want to place on the chessboard. Once this number is provided, press the "Get Solution" button. Depending on device context parameters, this action will either be performed on the cloud or performed locally. The loading screen and result screen are shown in figures 8b and 8c respectively.



(a) N Queens Problem Test. (b) Running task using cloud. (c) N Queen cloud task result.

Figure 8: N Queen Problem Experiment

7 Performance Monitoring

In order to capture performance of experiments, we used the Android Profiler tool which comes with ASI. The profiler can be opened by navigating to the top menu bar and clicking on View > Tools Windows > Profiler. This is shown in figure 9.

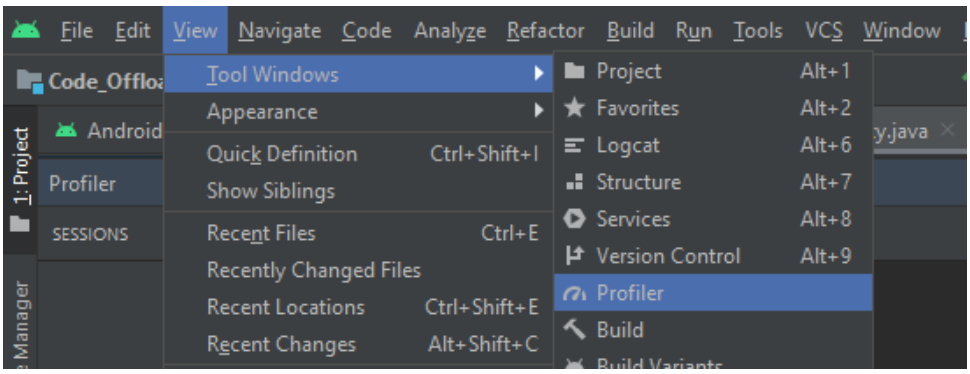


Figure 9: Open Android Profiler

When the Profiler dialog opens, click on the add session button. This is used to select an AVD to profile. This is shown in figure 10 below.

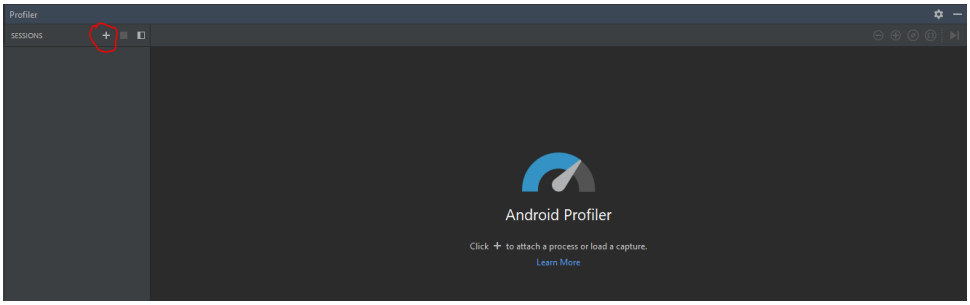


Figure 10: Add device to Profiler

After a device has been selected, the following screen in figure 11 is shown. This screen is used to monitor resource usage and execution time.

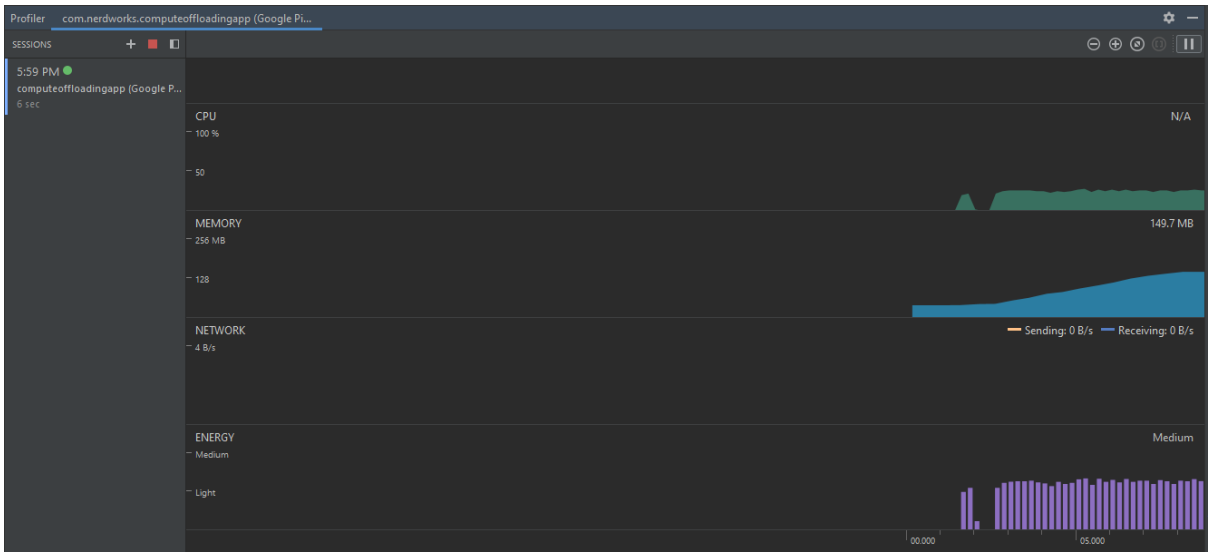


Figure 11: Android Profiler