Countdown For Android

Technical Manual

Gheorghe – Vasile Burac & Bartlomiej Kiraga

[Year]

## **0. Table of contents**

Contents

[**0. Table of contents** 1](#_Toc34630032)

[**1. Introduction** 1](#_Toc34630033)

[**2. System Architecture** 2](#_Toc34630034)

[**3. High-Level Design** 3](#_Toc34630035)

[**4. Problems and Resolution** 4](#_Toc34630036)

[**5. Installation Guide** 5](#_Toc34630037)

[Testing 10](#_Toc34630038)

# 1. Introduction

## 1.1 Overview

This technical manual provides a description of the finished implementation of the Countdown for Android application. It contains the architecture, high level design and development issues of the project.

Countdown for Android is a multiplayer mobile game designed for Android OS. Its game mechanics mirror the TV show Countdown. The game is divided into three rounds: the letters round, the numbers round and the conundrum round. In the letters round players receive six numbers and a random target number ranging from 101 to 999. They are then tasked with applying mathematical operations to their letters in a way that results in the target number. In the letters round the players receive nine letters and are tasked with using their letters to come up with the longest possible valid word. The conundrum round functions similarly to the letters round with the added condition that the player’s word must use up all the letters available to them. The game implements solver algorithms for all the rounds allowing users to see a generated viable answer. Countdown for Android supports both single and multiplayer game mode, with the multiplayer mode supporting up to four players per match.

## 1.2 Glossary

Android OS - Operating system for mobile devices (phones, tablets, TV, etc)

Firebase Realtime database - Real time database system hosted in the cloud

Kotlin - Cross Platform Programming Language (IOS & Android)

Java - General Purpose Programming Language (runs on Android OS)

Android Studio - Integrated Development Environment for Android OS

System - The Android Application and its associated components or services

Firebase Cloud Functions - Code that is executed on the server side

Firebase Authentication - Authentication service provided by Google Firebase

Android Application - a program that is developed to be run specifically on the Android OS

# 2. System Architecture

The following depicts the system architecture of Countdown for Android.

The user has the option to play single or multiplayer. Each option provides the three rounds of the game Countdown – Letters, Numbers and Conundrum. Each round has a solver associated that returns the solution to the game challenge. The game rounds and solver code is reused in the single player and multiplayer.

The components are:

1. Android app

* Letters and Solver
* Numbers and Solver
* Conundrum and Solver

1. Firebase services

* Cloud functions
* Realtime Database
* Authentication

The android app is what the user interacts with directly.

(fig.1)A screenshot of a computer

Description automatically generated

# 3. High-Level Design

The following diagrams detail the high-level system architecture of the project. The following components of single player are reused in the multiplayer functionality of the game:

* Scoreboard
* Game Logic
* Score Comparer
* Solver
* Local Game Data

The single player functionality is detailed in (fig.1). This encompasses the core functionality of the application.

(fig.2)

A map of a lot of snow

Description automatically generated

(fig.3)

A close up of text on a white surface

Description automatically generated

# 4. Problems and Resolution

The problems encountered during the development of this project were:

1. Performance issues when inserting more than 121000 words in a custom Trie object.

Solution: Divide the work and complete it parallel using multithreading techniques.

1. Mobile development specific threading synchronisation and communication issues.

Solution: Use coroutines and semaphores to enable communication and synchronisation between threads.

1. Searching for the best solution when given a set of nine letters. This was an issue because the approach used was to generate a list of all possible nine letter words. The resulted list of possible nine letter words was more than 320000 words. Finding the best solution resulted in performance issues.

Solution: Simplify the search algorithm to reduce the search space of the problem.

# 5. Installation Guide

Countdown for Android supports a variety of smartphones that run Android OS. The system can be installed on a physical device or on a virtual device. The installation on a virtual device is almost identical to the installation on a physical device.

The required software packages include:

* Android Studio (version min. 3.5) – required in order to install the app
* Android OS (version min. 4.4 – Android KitKat, or later) – required to run the app

The require hardware components include:

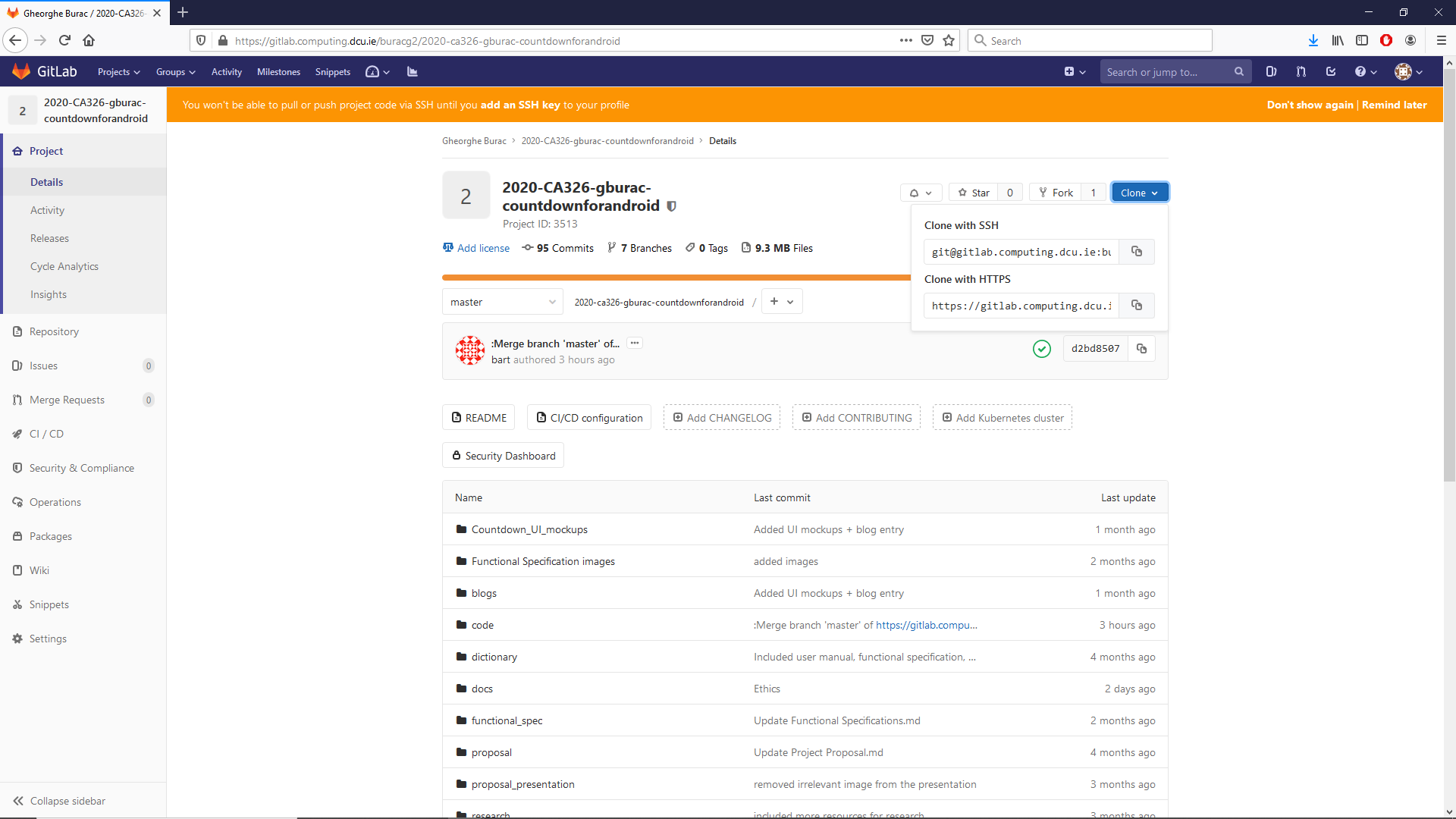
* A CPU that supports Intel Hardware Accelerated Execution Manager (HAXM), this is required in order to emulate the Android Virtual Device (A.V.D.) (is required to be able to run Countdown for Android on an Android Virtual Device)
* A smartphone running Android OS (version 4.4 – KitKat, or later)

Countdown for Android can be installed by following the following steps:

1. Obtain a copy of the system from the Gitlab repository: <https://gitlab.computing.dcu.ie/buracg2/2020-ca326-gburac-countdownforandroid.git>

This can be done by cloning the repo.

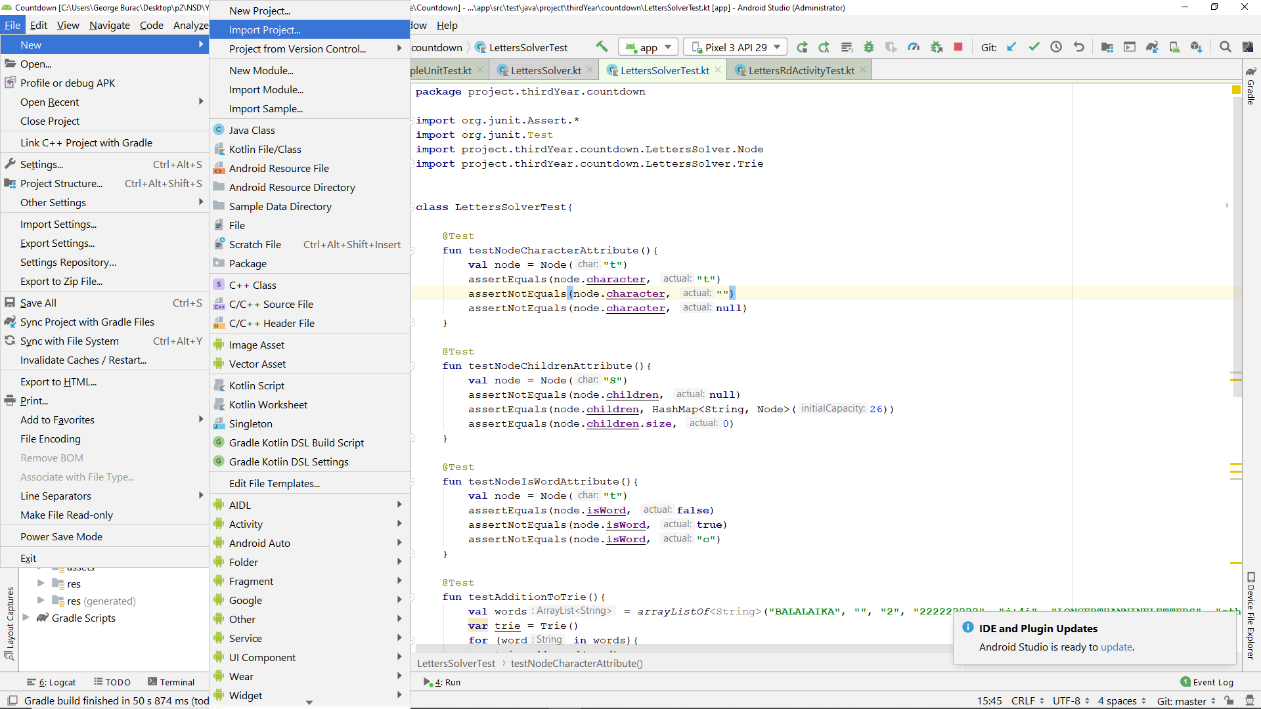
(fig.4)



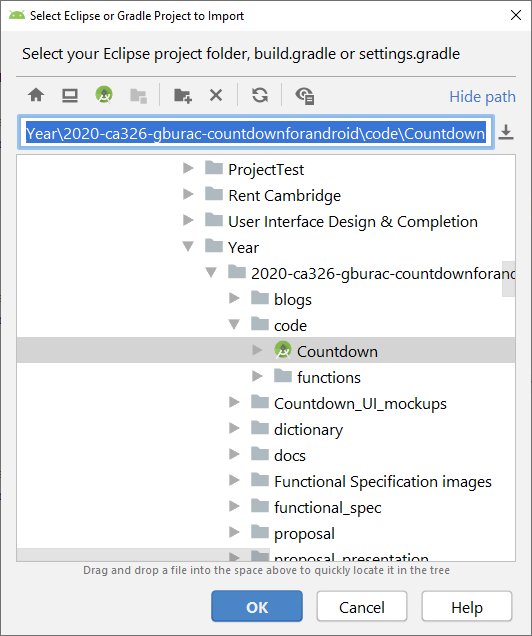
1. Import the system into Android Studio by starting Android Studio on your computing device from which you want to initiate the installation and navigate to File and then hover with the mouse or navigate with the keyboard to New > Import Project…

Following the import wizard allows us to locate the system that we want to import and import it from the file system of the machine into Android Studio. This is depicted by the following:

(fig.5)



(fig.6)



1. Connect the target device to Android Studio, this can be achieved by either running the emulator from Android Studio, in case of virtual devices; or by plugging in a data transfer cable which has Developer Options enabled. Developer Options must be enabled in order to allow the installation of applications from another device.

* Physical devices may need to have Developer Options enabled. This varies depending on manufacturer. The general case is that is present in Settings and can be enabled by repeatedly tapping some section of the Settings, depending on manufacturer.
* On devices manufactured by Samsung, Developer Options can be enabled by navigating to Settings > About Phone > Software Information. In Software Information menu, the Build Number section must be tapped seven times in rapid succession. Developer Options can be turned on by toggling the On button from Off to On. This is detailed by the following:

(fig.7) (fig.8) (fig.9) (fig.10)A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

(fig.11) (fig.12)

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

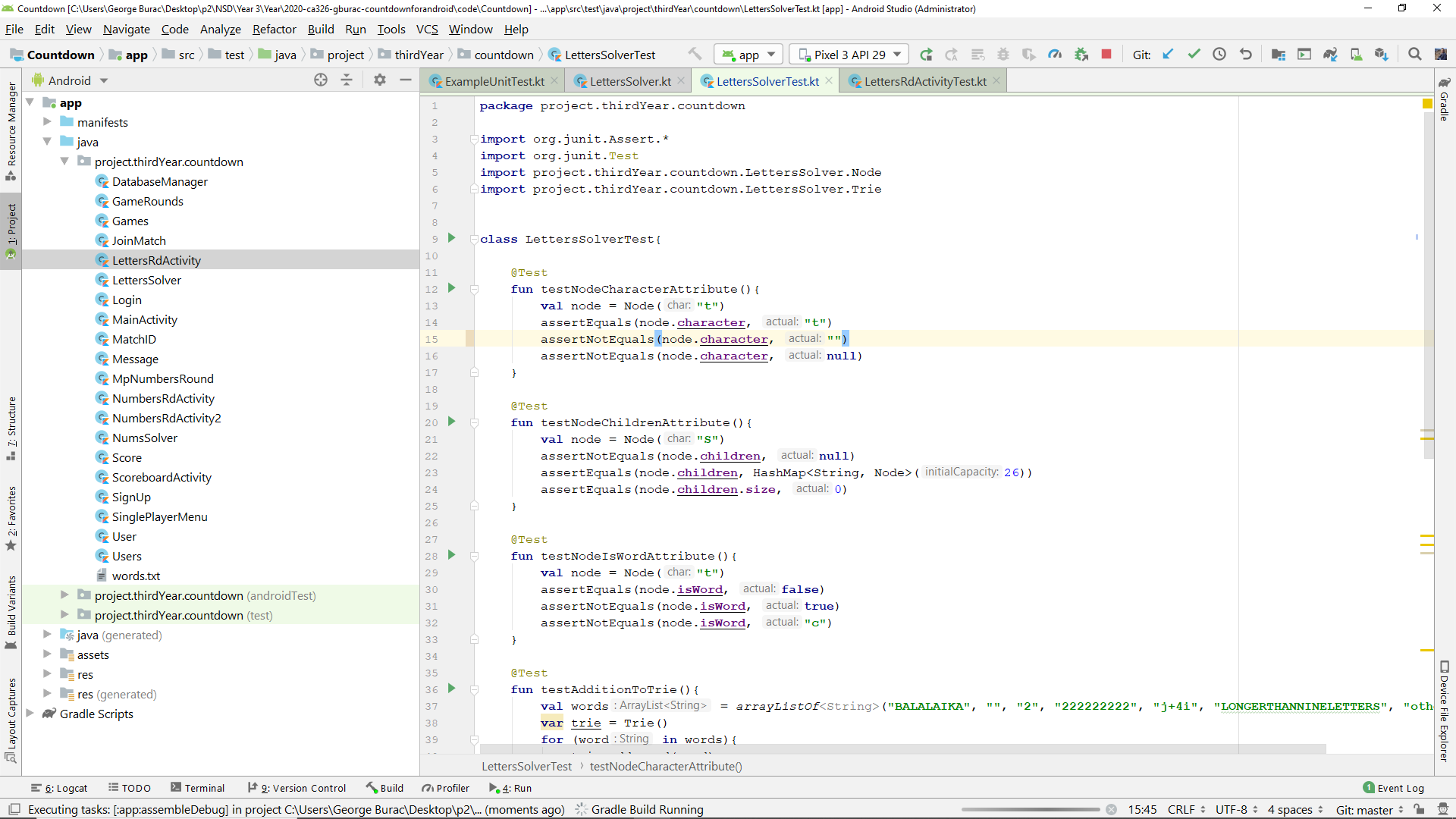
Description automatically generated

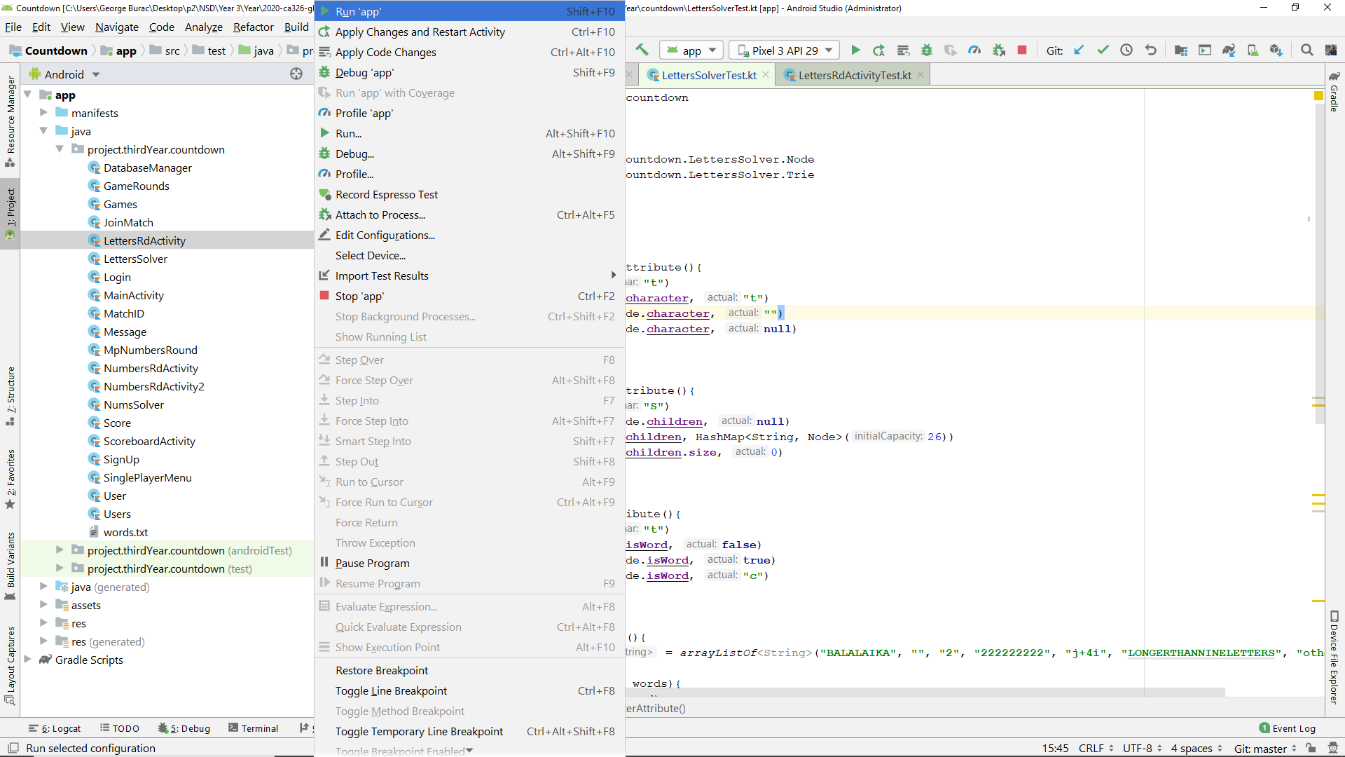
* On similar devices manufactured by Huawei, there is one extra step required. USB (Universal Serial Bus) Debugging must be enabled also. The following details this:

A screenshot of a computer screen

Description automatically generated (fig.13)

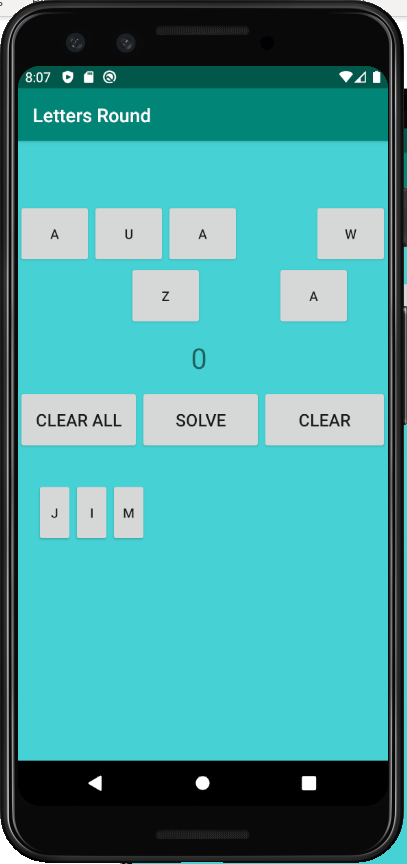
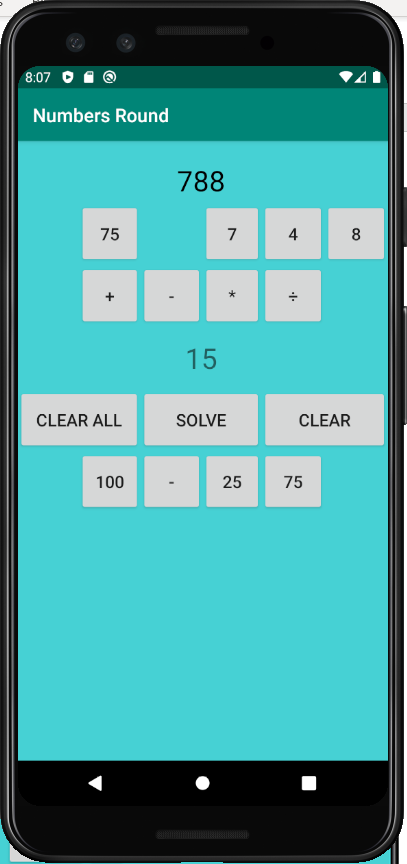
1. Run the application on the selected device. This step can be completed by navigating to Run and then Run ‘app’ as illustrated below:

 (fig.14)

 (fig.15)

1. Enjoy the game experience.

(fig.16)



# Testing

The test strategy implemented was to have unit tests (implemented using Junit 4), integration tests (instrumented tests using AndroidX Test) and end to end tests.

The following figure depicts our strategy.

To achieve our testing goals, a CI/CD pipeline was implemented on Gitlab. Manual testing was also required.

For API testing (Cloud Functions), Postman, Python unit testing and manual testing was implemented.

The app was tested on several different devices (Google Pixel 3, Huawei P20, etc).

User testing was also employed to gain insight into potential issues the users might face and to understand the possible improvements that can be implemented.

The app works on a number of devices.

The solvers have some very limited edge cases which they don’t support for performance reasons.

User testing demonstrated that the users cared in particular about the user interface, thus an emphasis was placed on the user interface and user experience development.

(fig.17)A close up of text on a black background

Description automatically generated