
REPORT ON MINI PROJECT

3rd Semester

Computer Science and Technology

IEST, Shibpur

Date of Submission: December 7, 2017

Topic: Android App Development

Group Members ~

Ishika Mitra

Kumar Subhodeep Guin

Mentor ~

Prof. Ashish Kumar

Layek

TABLE OF CONTENTS

- CSTServerChecker: **Abstract**
- What does **Pinging** a server mean?
- About Android
- Working of **Activities** and **Layouts**
- The Activity Life Cycle
- Effects of Good Implementation of Life Cycle
- **Design** of the CSTServerChecker App
- **Functionality** of our App
- Bibliophile

CSTServerChecker: Abstract

The app is useful to check the running status of a server once connected to a Wi-Fi. The basic function of the app is to ping IP of a Server and determine if the server is down or up and running. The user can enter an IP to check the status of the server. The app also allows the user to save an IP so that he/she doesn't have to enter the IP again and again and can directly check the status of the server by selecting the required IP.

What does **Pinging** a Server Mean?

The Ping command allows you to test the connection speed between you and another network mode i.e. it enables you to find out a server IP address and determines the accessibility of the server.

The Ping test uses your internet connection to send out some packets of data to a specific address. These packets are then sent back to your system. The test records the amount of time it took for the packets to reach the address, and whether or not any packets were lost in the process.

About Android

In today's world, everyone wants a smartphone or a tablet, and Android devices are hugely popular. Android is the world's most popular mobile platform. It is a powerful development framework that includes everything one needs to build great apps using a mix of Java and XML.

A typical android app has mainly 2 components:

- **Layouts** – Define the appearance of the app. In a multiscreen app, one defines what each screen looks like using a layout to define its appearance. Layouts are usually defined using XML, and can include GUI components such as buttons, text fields etc.
- **Java Code** – You define what the app does by writing a Java code. A special Java class called an Activity decides which layout to use and tells the app how to respond to the user. For example, if the layout includes a button, one needs to write Java code in the activity to define what the button should do when pressed.

Certain terms related to android -

- **Android SDK:** The Android Software Development Kit contains the libraries and tools you need to develop Android apps.

- **Android Studio:** Android Studio is a version of IDEA that includes a version of Android SDK and extra GUI tools to help in the development of the app. It is the IDE we have used to build our app.
- **API Level:** Our app supports all Android Package Index (API) levels of API 19 (Android 4.4) or higher.

Working of **Activities** and **Layouts**

- Let's look more closely at how activities and layouts work together to create a user interface.
- The device launches the app and creates an activity object.
- The activity object specifies a layout.
- The activity tells Android to display layout on the screen.
- The user interacts with the layout displayed on the device.
- The activity responds to these interactions by running application code.
- The activity updates the display which the user sees on the device.

The Activity Lifecycle

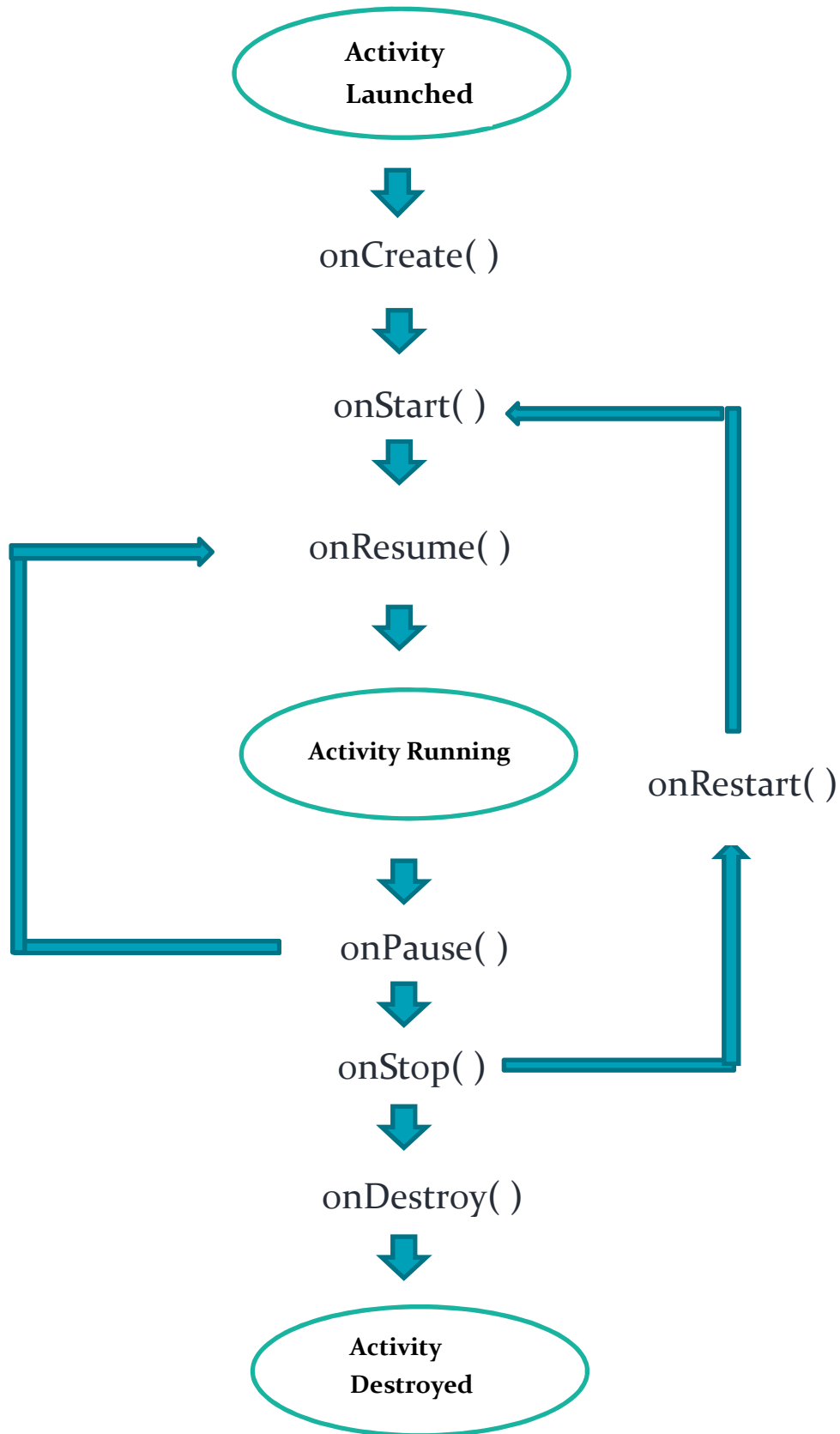
As a user navigates through, out of, and back to the app, the Activity instances in the app transition through different states in their lifecycle.

The Activity class provides a number of callbacks that allow the activity to know that a state has changed: that the system is creating, stopping or resuming an activity, or destroying the process in which the activity resides.

Within the lifecycle callback methods, you can declare how your activity behaves when the user leaves and re-enters the activity.

To navigate transitions between stages of activity lifecycle, the Activity class provides a set of six callbacks: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` and `onDestroy()`.

The system invokes each of these callbacks as an activity enters a new stage.



THE ANDROID LIFECYCLE

The activity gets launched, and the onCreate() method runs.
The activity isn't yet visible as no call to onStart() has been made.

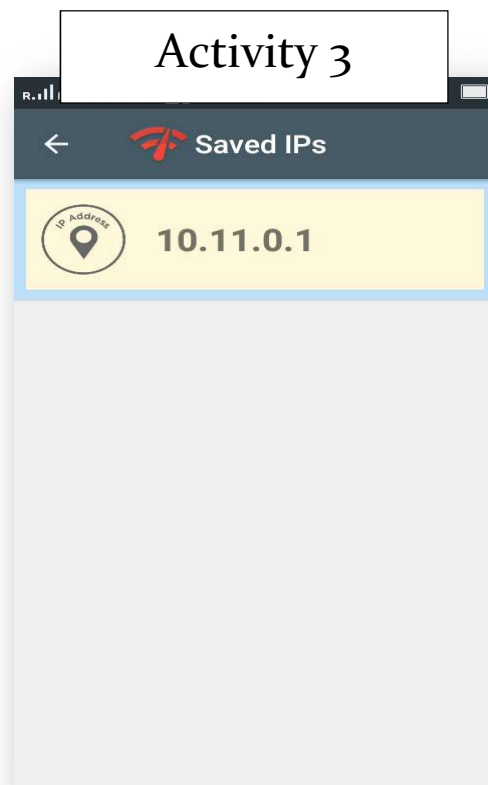
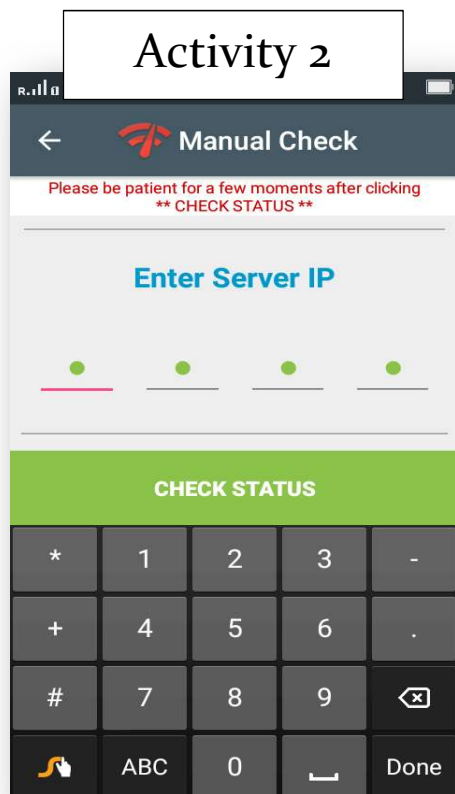
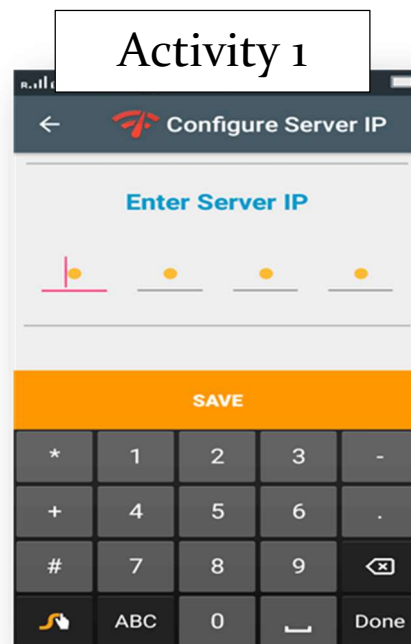
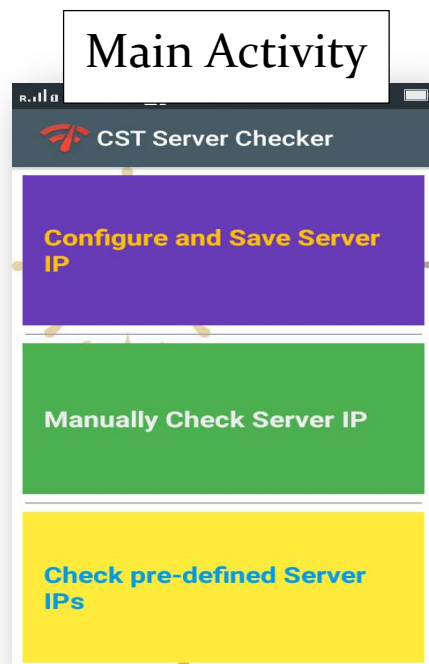
1. **The onStart() method runs after the onCreate() method.** It is called when the activity is about to become visible. After the onStart() method has run, the user can see the activity on the screen.
2. **The onResume() method runs after the onStart() method.** It gets called when the activity is about to move to the foreground. After the onResume() method has run, the activity has the focus and the user can interact with it.
3. **The onPause() method runs when the activity stops being in the foreground.** After the onPause() method has run, the activity is still visible but doesn't have the focus.
4. **If the activity moves into foreground again, the onResume() method is called.** The activity may go through this cycle many times if the activity repeatedly loses and regains focus.
5. **The onStop() method runs when the activity stops being visible to the user.** After the onStop() method has run, the activity is no longer visible.

6. **If the activity becomes visible to the user again, the `onRestart()` method gets called followed by `onStart()` and `onResume()`.** The activity may go through this cycle many times if the activity repeatedly becomes invisible and visible again.
7. **Finally, the activity is destroyed.** The `onPause()` method gets called before the activity is destroyed. The `onStop()` method will usually be called before `onDestroy()`, but it may get bypassed if the device is extremely low on memory.

Effects of Good Implementation of Life Cycle

- App avoids crashing if the user receives a phone call or switches to another app while using the app.
- App avoids losing the user's progress if they leave the app and return to it after some time.
- App avoids consumption of valuable system resources when the user is not actively using it.

DESIGN OF THE CSTSERVERCHECKER APP



Functionality of our App

Our app supports the following functions:

1. Configure and Save Server IP

- The user is required to enter an IP address.
- If already present, the IP won't be saved and displayed a Toast message quoting "IP Already Present". This is a measure to *restrict duplicate data entry* in the app.
- The App saves the IP address for future checks.

2. Manually Check Server IP

- The user is required to enter an IP address.
- The App runs the Ping test to check the status of the server to report one of the following:
 - Server Not Active
 - Problem Checking Status (Waiting Condition)
 - Server Up and Running

3. Check pre-defined Server IPs

- It populates a list and sets the display, fetching data from the IPAdapter which is a custom ArrayAdapter class.
- IPAdapter takes input objects of the IP class.
- IP class contains an image, IP text and an indicator which changes color as green, orange and red based on the status of the server.
- Every item in the list is clickable and the click triggers the code to ping the server and turn the indicator accordingly.

Additional capabilities of the app include ~

- ✚ If connected to a Wi-Fi having normal data connection, then it can be used to check servers like Amazon and Google servers(8.8.8.8).
- ✚ If not connected to a Wi-Fi the app will open and you can save IP too, but rest of the features cannot be accessed as it is a server checker app, it is considered a good practice to connect to Wi-Fi and then check, hence prompting the user to connect to Wi-Fi only.
- ✚ Prevent you from entering invalid data in the IP. (no alphabets and numbers between 0 to 255 only.)
- ✚ Furthermore, you are not at all allowed to enter more than 3 digits in the IP field. Keep pressing buttons, it won't.
- ✚ Cool app Icon on all its activities header.
- ✚ Branding colors-
 - Save IP page uses the branding color as Orange. The buttons and the ip field hints all are in orange color.
 - Manually Check IP uses the branding color as Green. Reason being same as above. Plus, it displays the total data fetched from the ping result and you can 'scroll' the page and read through it all.
- ✚ Animation in the buttons when you click on it. You can see a small tint in the color of the buttons when clicked.

- ✚ Button always visible – when you enter the ip the button does not remain with the page but adjusts according to the page area covered by the keyboard of the phone/tablet but the button stays above the keyboard.
- ✚ A little bit of *French* support is also included in the app in the development process for users who use their phones in French Language settings.
- ✚ A faint click *sound* can be heard when clicking buttons and lists in the app.

Bibliophile

- <https://stackoverflow.com>
- <https://developer.android.com/index.html>
- <https://en.wikipedia.org/>
- <https://udacity.com>
- Multiple Google Searches for various small queries

--- THANK YOU ---