Android Application- Big Brother 3000



The diagram above illustrates the states of the Android application:

* **Initialize**; the main activity is launched
* **Activity -1**; this is the welcome screen for the user and the press of the begin button will move the user into the second activity.
* **Activity -2**; this is the settings screen, will prompt the user for all necessary information before starting the watch and will validate all of the values entered before proceeding
* **Activity-3**; this is the main part of the program that will set all the user specified settings and will start sending the location updates to the server until the User presses the Stop button or if an error is encountered. In case of error the User will be sent back to the Settings activity.
* **Terminated**; the activity is no longer running.

**Android App Pseudocode**

Activity One XML - Welcome

{

A picture showing the Application name

Some text to lead the user

“Begin Watch” button to move to the next activity

}

Activity One Java – Welcome

{

Start the activity on Create

If (Begin Watch button is pressed)

{

Create an intent and move to the next Activity - Settings

}

}

Activity Two XML - Settings

{

Text indication Settings page

Text Fields for the IP Address and Port to be entered.

Radio Buttons so the user can choose between updates by Time or Distance

Text Field for the frequency desired by the user

Checkboxes to check if the user wants to use data (charges) or Wi-Fi

GPS is used by default.

“Start Watch” Button to continue, “Clear” Button to clear the fields

}

Activity Two Java – Settings

{

Start the activity on Create

Get all the values out of the text fields, radio button and checkboxes using their respective ID’s

If (Clear button is pressed)

{

Clear all the text in the text fields.

}

If (Start Watch button is pressed)

{

Use all the values from the text fields, checkboxes and radio buttons to validate them

If (IP address is not a valid address) => toast Error and return

If (Port number is not a valid number) => toast Error and return

If (frequency is not a valid number) => toast Error and return

If all the values are okay put all this information into a Bundle/Intent.

Pass the intent into the next activity – Send Data

}

}

Activity Three XML – Send Data

{

A picture showing that it is now sending data.

A gif image to mimic the sending of data to let the user knows it is running.

“Stop Watch” button to stop sending data and go back to the Settings

Activity.

}

Activity Three Java – Send Data

{

Start the activity on Create

Get all the values out of the bundle received from the Settings Activity. (IP Address, port, frequency, preferences)

Attempt Connection to the server.

If (Connection)

{

Start the Thread to send Data.

Get the mac address from the device and send it to the server to be used as a unique ID

Set the Location preferences criteria

Create the Location Listener

{

On Location Changes Send Data to the Server (New Location)

}

Request Location Updates based on the user Preferences (time/distance) and this will call the location listener on Updates.

}

Else

{

Let the user know

Return

}

If (Stop Watch button is pressed)

{

Close the Connection

Finish the Activity

}

Connection Thread

{

Connect to Server

If (there is a location update)

{

Send Data to Server

}

If (Connection is shut down/ Errors)

{

Free Resources;

Finish the activity;

}

}

}