GLOBAL POSITIONING SYSTEM LOGGER INSTALLATION AND PREFERENCE SETTINGS FOR MOBILE DEVICES

UCI Population Health and Disease Prevention

spatial epidemiology
PARKER GROUP
global health

http://www.parkerlabgroup.com/

Table of contents

A. Introduction

This protocol describes methods for installing the GPSlogger application on a mobile smart phone and setting the application preferences for a research project. The GPSlogger application is an Android application that uses a mobile smart phone's locational capabilities to record ("log") locational information about the mobile phone device. A number of preferences can be set so that recording intervals, log file type, log file location, etc .can be modified as needed. The protocol was written for project staff who will be involved in using or distributing mobile devices with the GPSlogger application.

For this protocol a Samsung Galaxy J2 Prime mobile smart phone was used. This protocol should be relevant for other android based devices, with slight modification for the location of the default folder on the device.

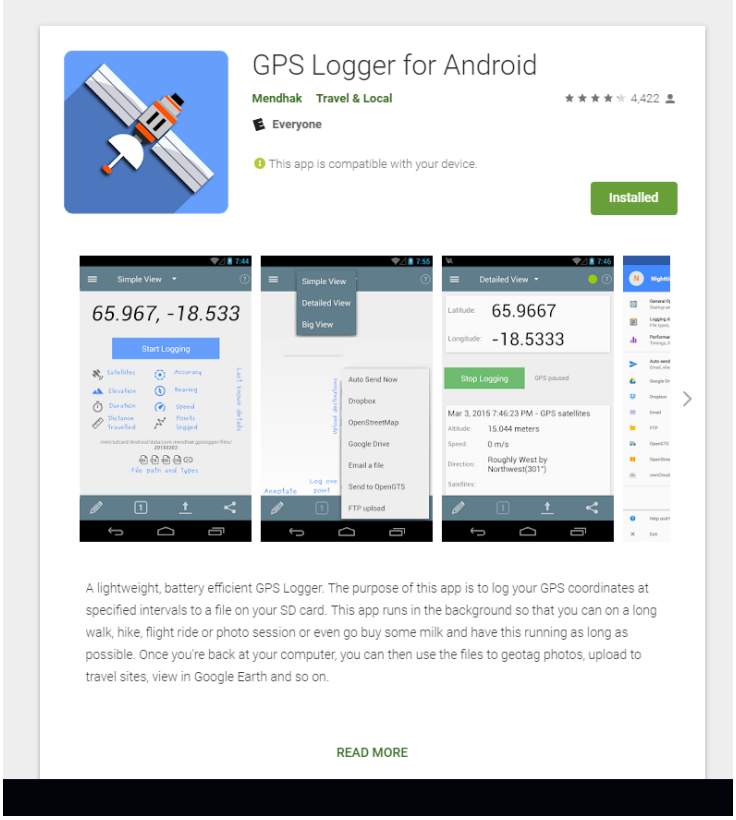
Section B of the protocol describes installation of the application in two ways: I.) via the Google Play store, and II.) via an APK file. Installation time is approximately the same through both approaches. Installation using Google Play requires internet access for each device. Downloading the APK file from GitHub requires an internet connection, but afterwards the file can be distributed across many devices without an internet connection.

Section C describes setting the preferences (time between readings, logging file type, etc.) for the application. Again, two approaches are described: I.) manually setting preferences using the application, and II.) using Java to preset preferences via a file that is loaded on the mobile device. It is recommended for someone who is first becoming familiar with this application to first set the preferences on a personal device manually. When installing the application across multiple devices, the use of the preset preferences file will be necessary. Using the preset file, it should take less than 1.5 minutes to install and set up an application on a single mobile smart phone device.

B. Installing the application on a mobile smart phone device

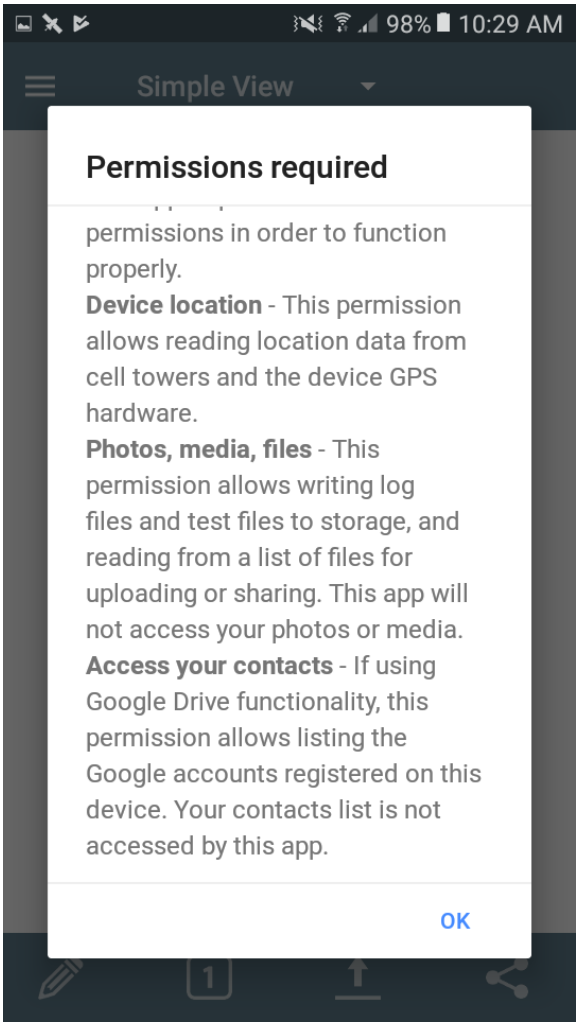
I.) Installing the application from Google Play (requires internet connection for each install)

1.) Find the application on Google Play. The **GPSlogger** application can be installed directly on a smart phone from Google Play. Look for "**GPS Logger for Android**" by **Mendhak**. Click "**Install**".

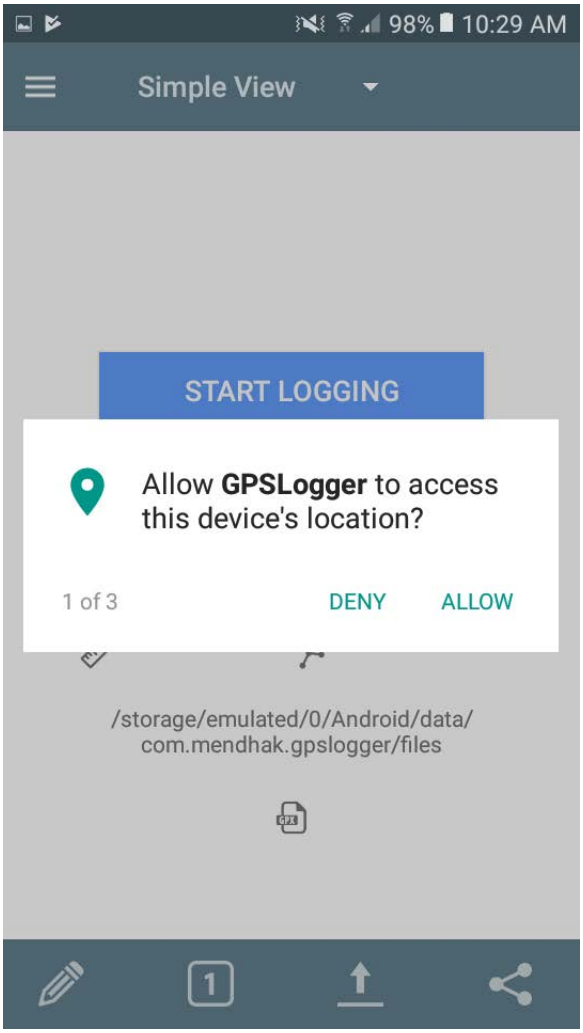


2.) Provide permissions for the application. Once the application has been installed, it will be necessary to allow it to access other parts of your phone (location, files, contacts). Select "**OK"** to provide permissions (A) and then select "**ALLOW**" for each of the three permission types (B).

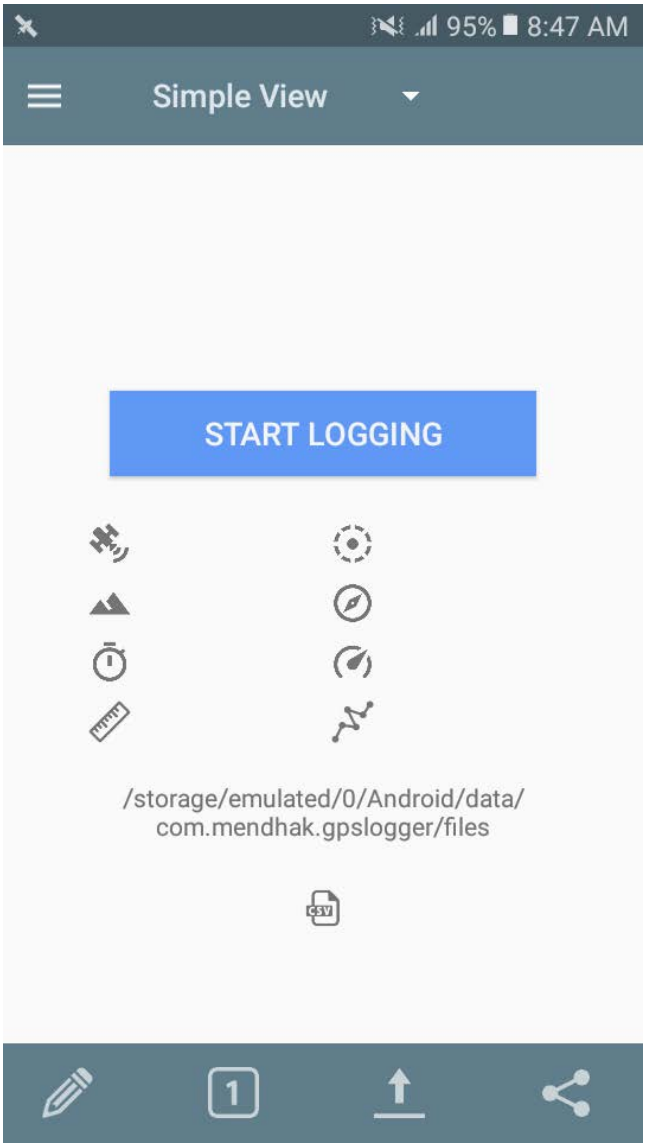
A



B



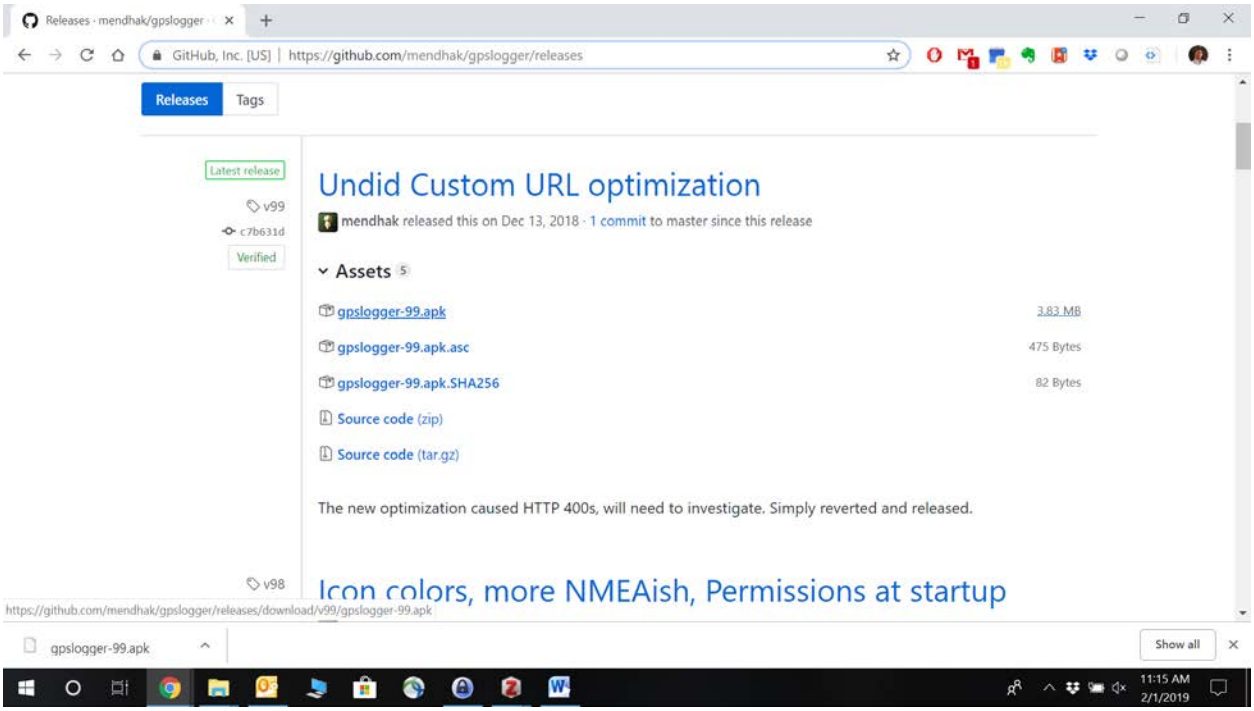
After providing the necessary permissions, you should see the screen shown below. Note that there is a location indicated (on this device, the location is: /storage/emulated/0/Android/data/com.mendhak.gpslogger/files). This indicates the default folder where logging files will be saved.



II.) Install the application using an APK file (does not require use of internet for each install)

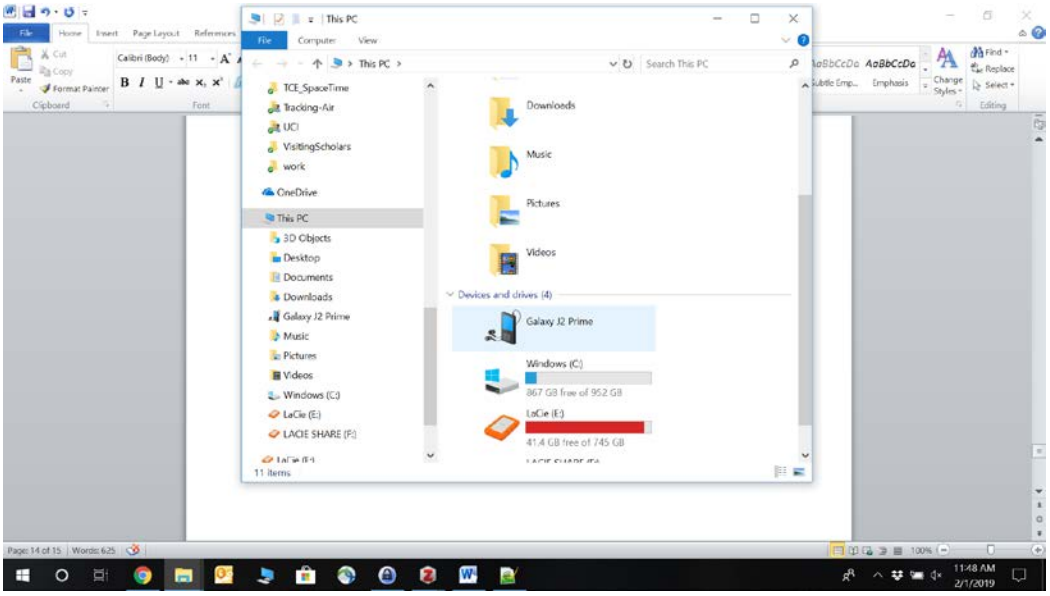
The application (as an APK file) can be downloaded directly from github:
https://github.com/mendhak/gpslogger/releases

Release "99" has been used for this example. After you've clicked on the link you should find the downloaded APK file in your "**downloads**" folder. Move it to another location that is clearly named for this project.

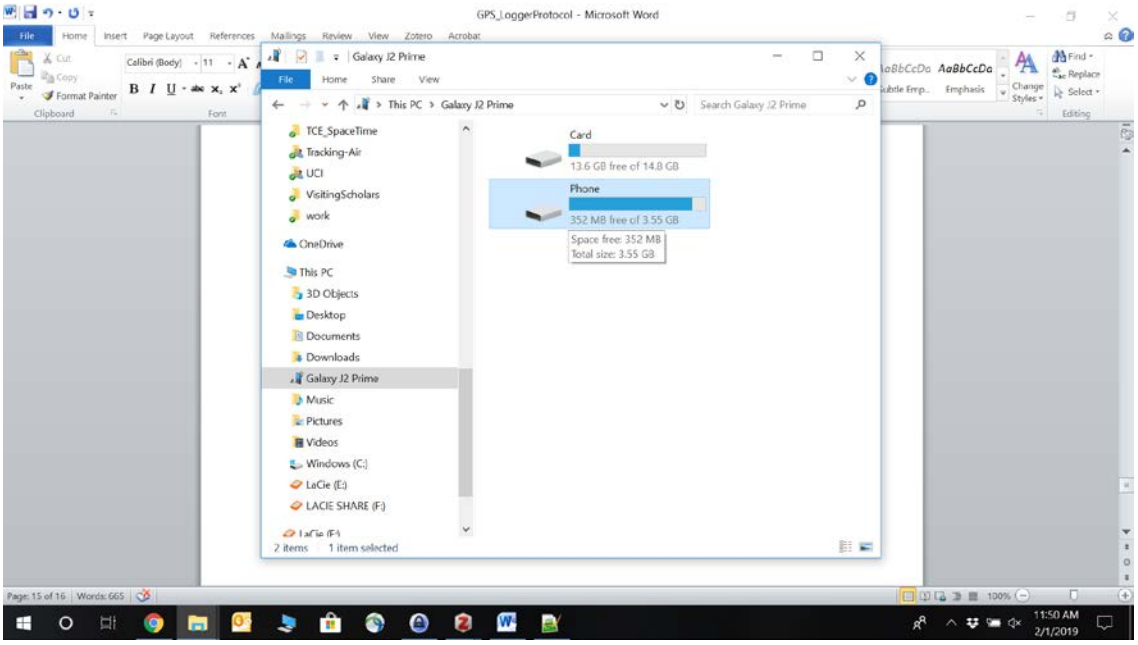


Next, connect a mobile device to a personal computer (using a USB port) that will be used for distributing the application. To install the application on the device:

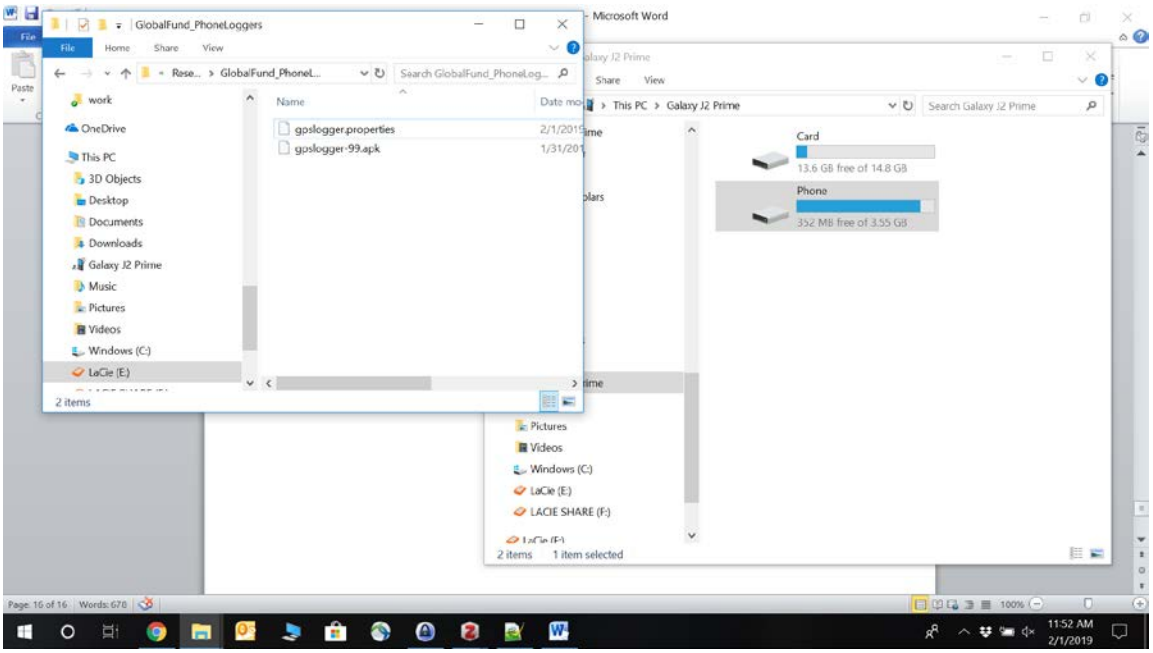
1.) Find the mobile device on your personal computer. Select and open it.



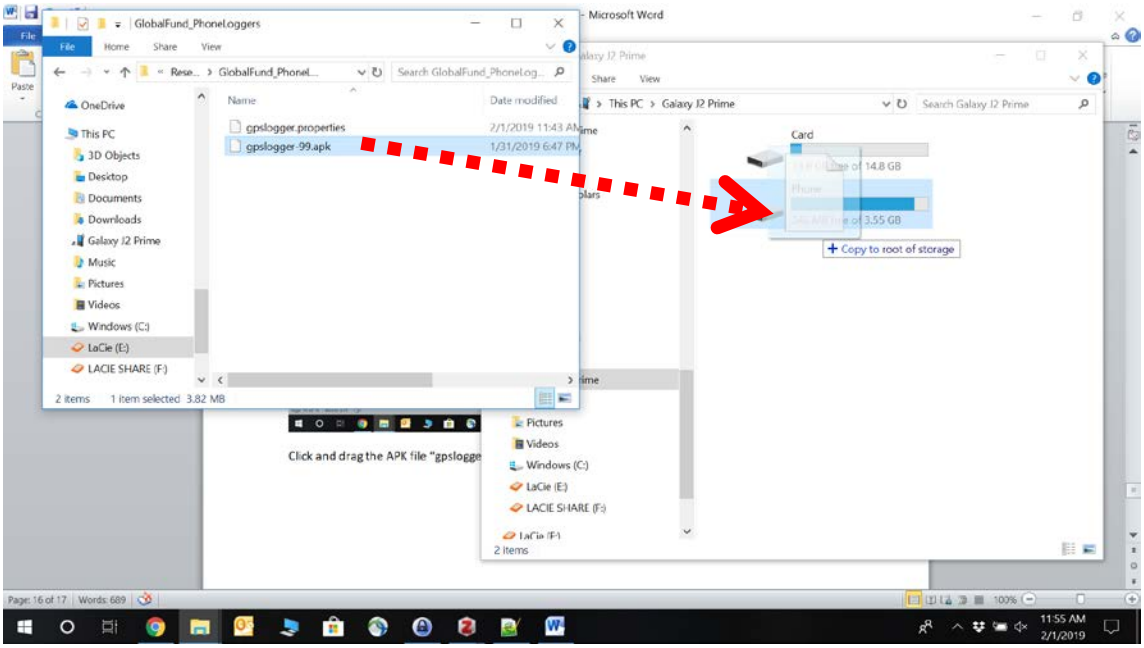
This particular device has both a "**phone**" and a portable "**card**". Either could be used, but for this example we will use the "**phone**" as this is where the application automatically chooses for its default location.



2.) In another window, open the folder/location where you've placed the APK file.



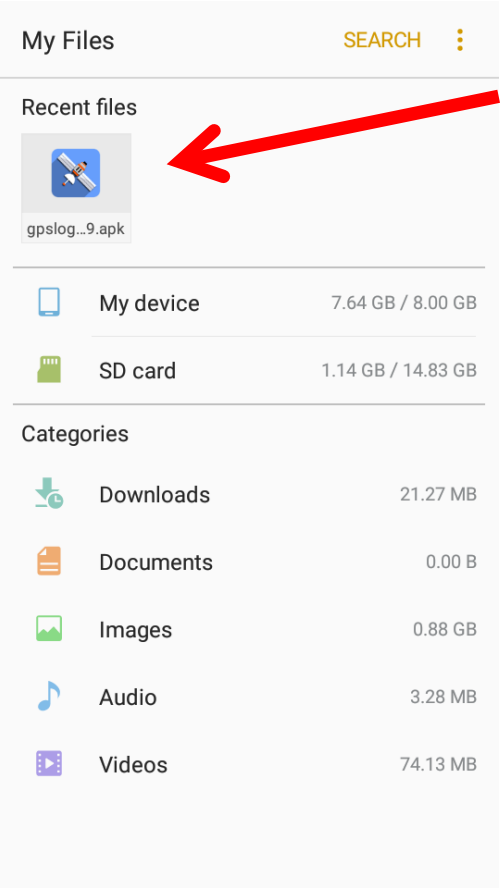
3.) Click and drag the APK file "**gpslogger-99.apk**" into the "**Phone**" folder.



4.) On the phone/device, find the file manager or file explorer. On the Samsung Galaxy this is in the Samsung folder and is named "**My Files**" (A). Select the "**gpslogger-99.apk**" file to begin installation (B).

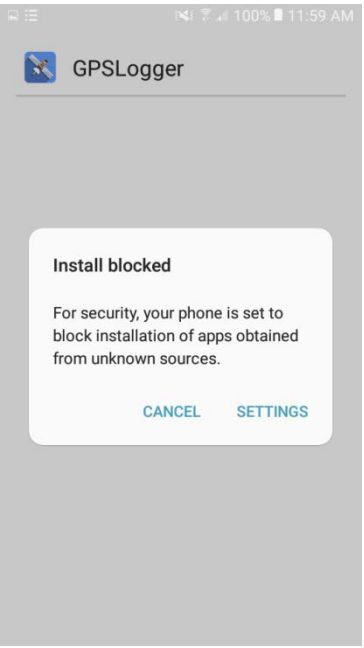
A

B

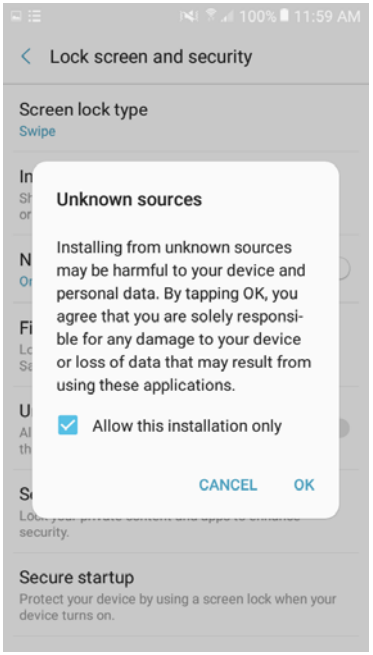


The mobile device could have a safety mechanism with regard to installing applications in this way. If so you'll receive a message as below (A). If so, you'll need to go into settings, select "**Unknown sources**", and allow the device to be installed (B). Then you should see the prompt to install the application (C).

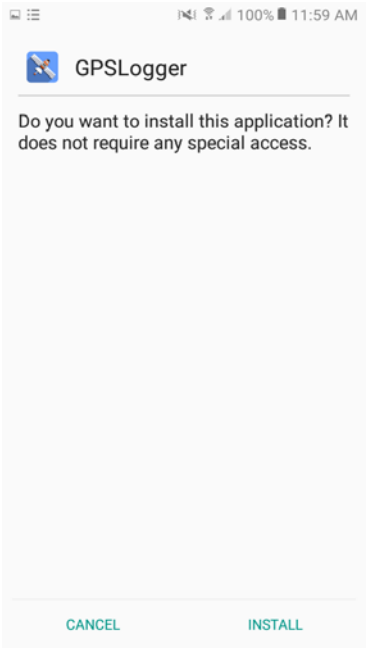
A



B



C



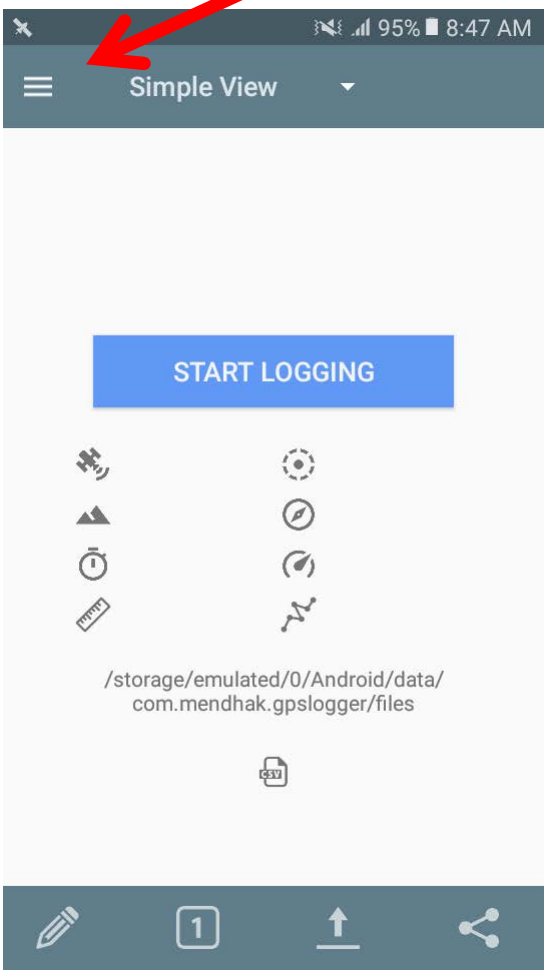
Open the application and provide the permissions (as in **Page 4**). Now the application and its default folders are installed.

C. Setting the application preferences

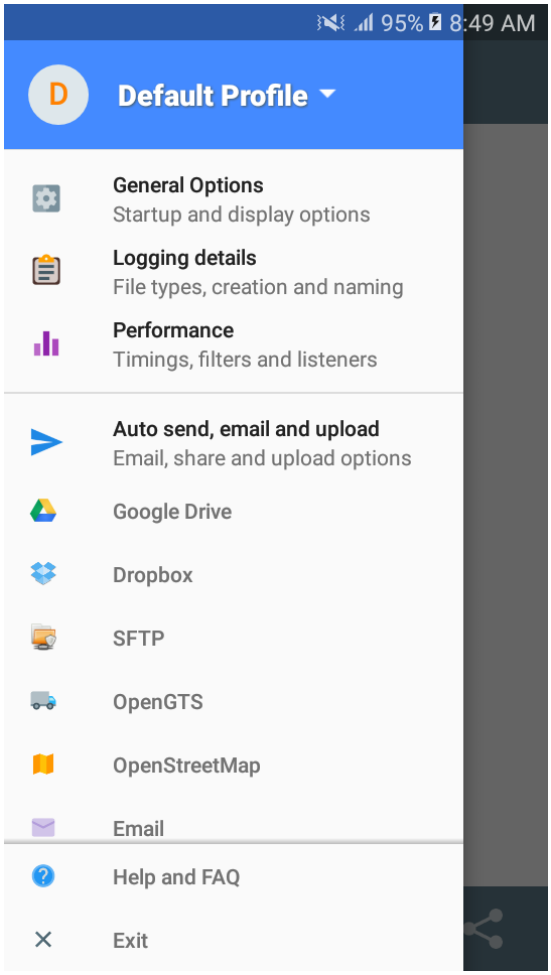
I. Setting the application preferences manually

1.) Select the list icon on the top left (A) and you will be provided several options (B) for setting up the logger. The first important options are in "**General Options**"; "**Logging details**"; and "**Performance**".

A



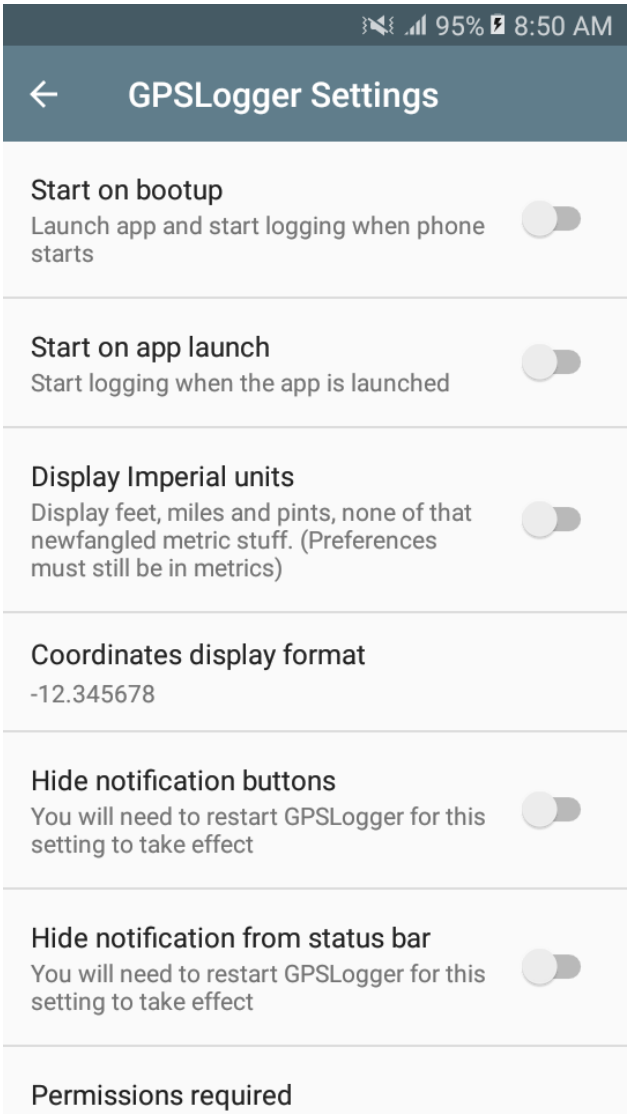
B



2.) Set options for the "**General Options**"

The "**General Options**" window should look similar to the image below. Turn on the options for:

- Start on bootup
- Hide notification buttons
- Hide notification from status bar

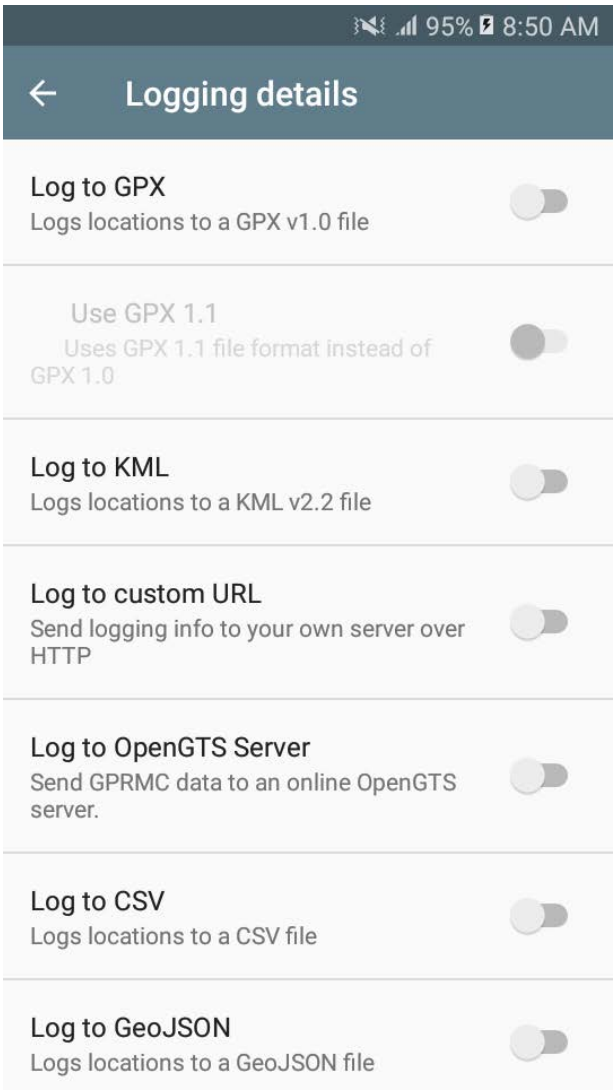


3.) Set **Logging** options

Go to the "**Logging details**" window (which should look similar to the image below). Turn on the options for:

- Log to CSV
- Prefix the device serial number to the file name

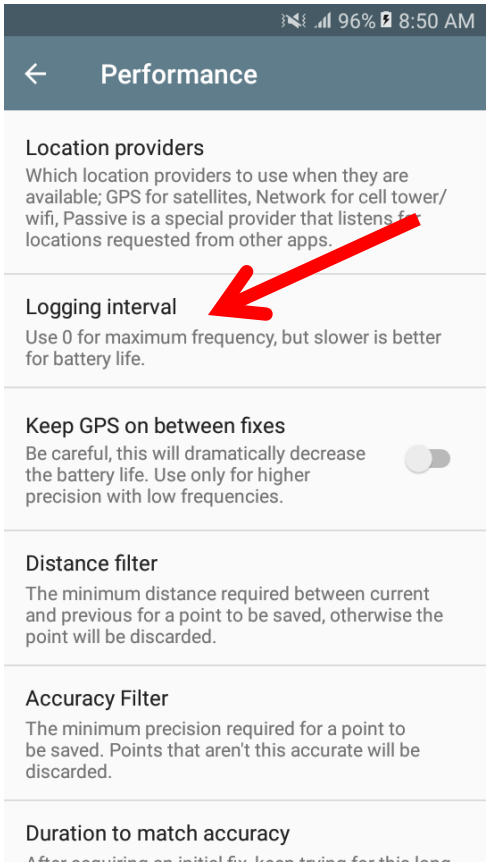
Default for the application is to "Log to GPX". If this is on (switch to the right and green indicator light) then turn it off.



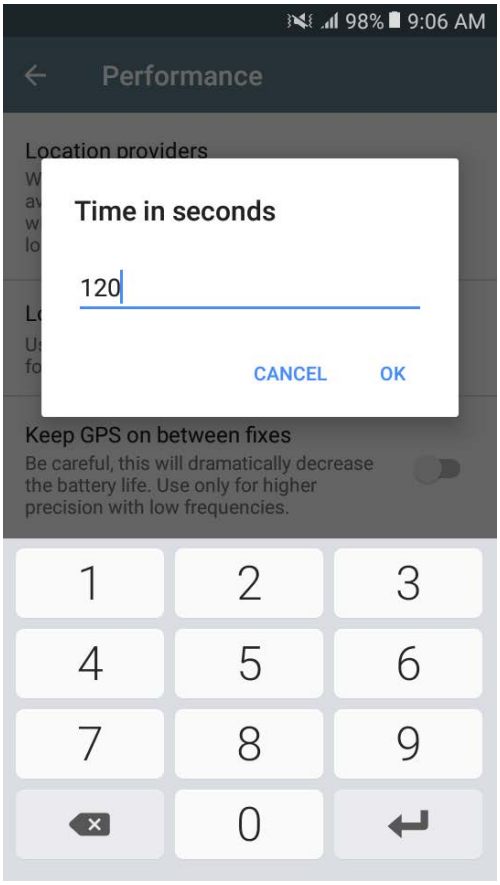
4.) Set **performance** options

Go to the "**Performance**" window (shown below). For this example, set the **Logging interval** to 120 seconds (two minutes.) This will change for the actual study as logging the coordinates this frequently will lead to enormous datasets of long periods of time (a reading every 2 minutes for 2-3 months).

A

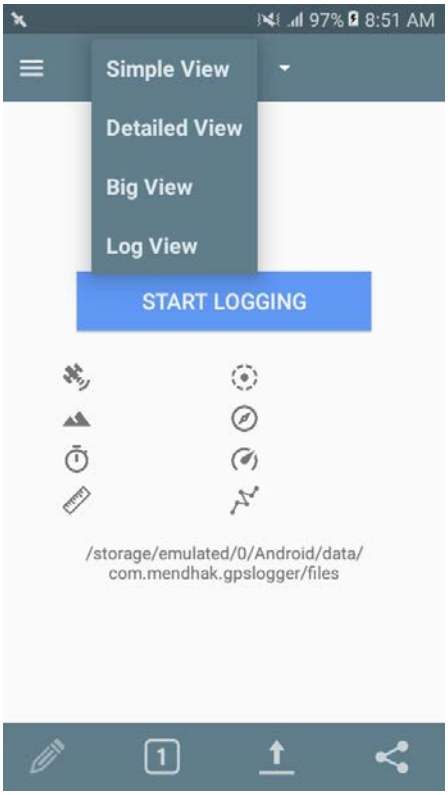


B



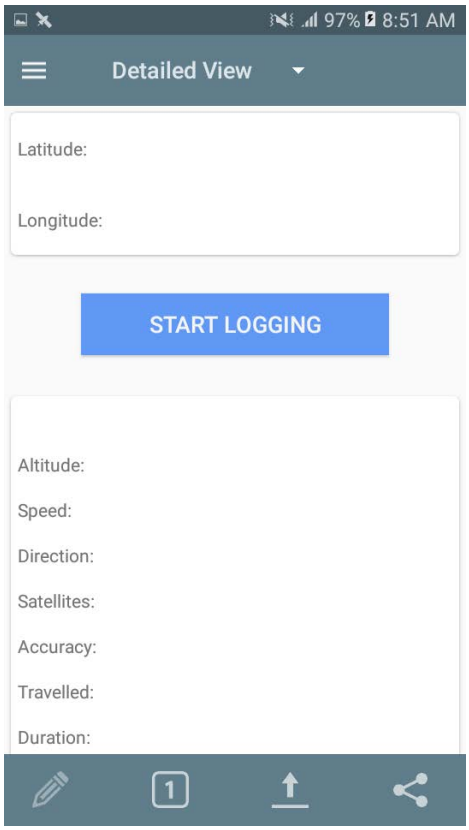
Select "**Simple View**" from the top middle of the home screen. You will see 4 options:

- Simple View (default)
- Detailed View
- Big View
- Log View

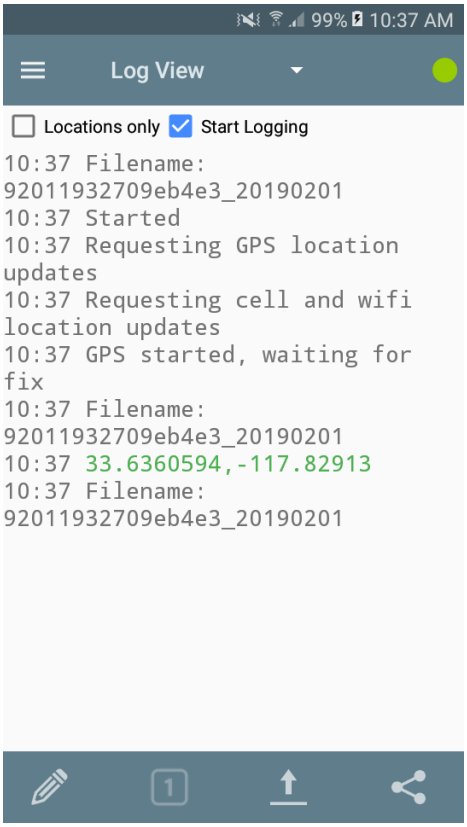


Detailed View (A) and **Log View (B)** are both useful for seeing how the device is functioning. Select "Start Logging" and then move to Log View. You should then see the actions being taken by the application on your device.

A



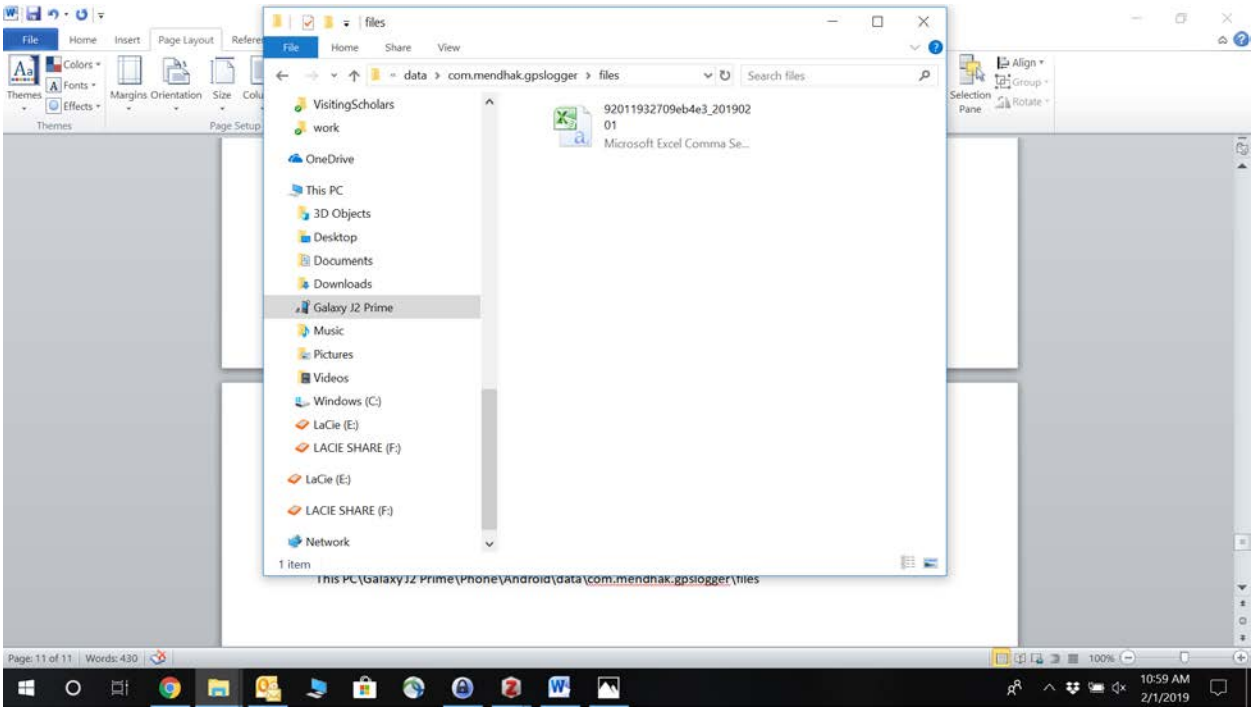
B



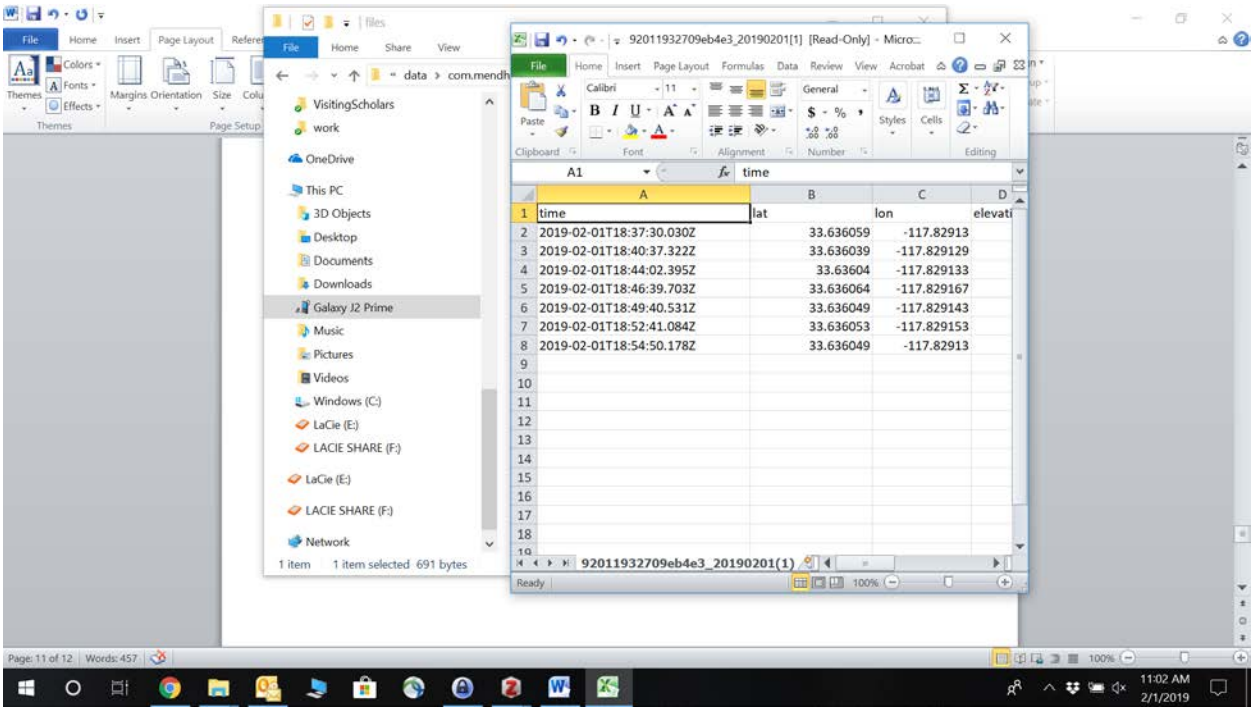
For this example, the application has been set to log geographic coordinates to a CSV file every 2 minutes. With the device connected to a personal computer (i.e. through a USB port), we can navigate to the folder in which the logs are being recorded. The default folder can vary by phone type (the example here is a Samsung Galaxy J2 Prime). The default location on this example phone is:

This PC\Galaxy J2 Prime\Phone\Android\data\com.mendhak.gpslogger\files

You can see that the CSV file is named based on the device ID and the date (YYYYMMDD).



It is also possible to have this file stored to something like Dropbox, sent to a server, emailed etc. The opened file should appear as the image below. This CSV file can be loaded to mapping or statistical software, or can be processed and formatted for loading into mapping or statistical software. Each new row (new reading) adds approximately 100bytes of data to the file.

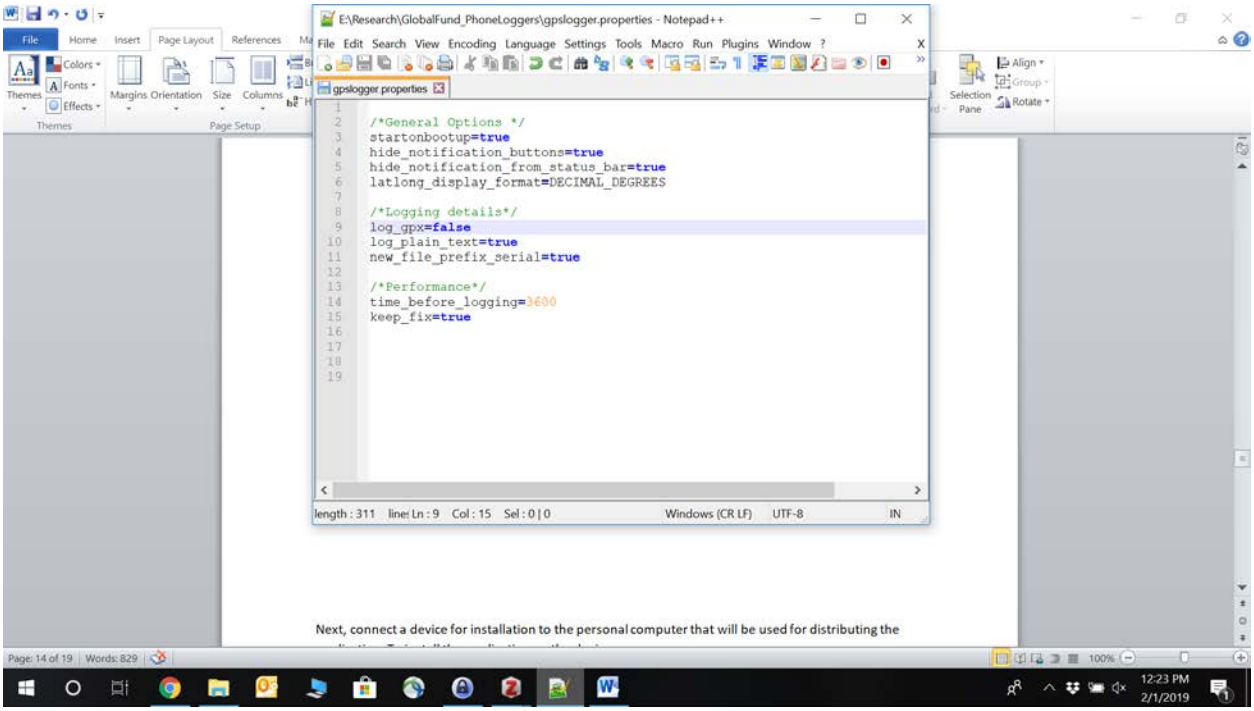


II. Presetting application preferences using Java

1.) A ".properties" file, written in Java language, can be prepared (with preset properties for the application) and provided at installation. The file should look as below (opened in the Notepad++ environment). A list of preference names can be found at the following link:

https://github.com/mendhak/gpslogger/blob/master/gpslogger/src/main/java/com/mendhak/gpslogger/common/PreferenceNames.java

More details on application preference settings will be provided in **Section D**. The file should be named "**gpslogger**" or else it may not be recognized and read by the application.



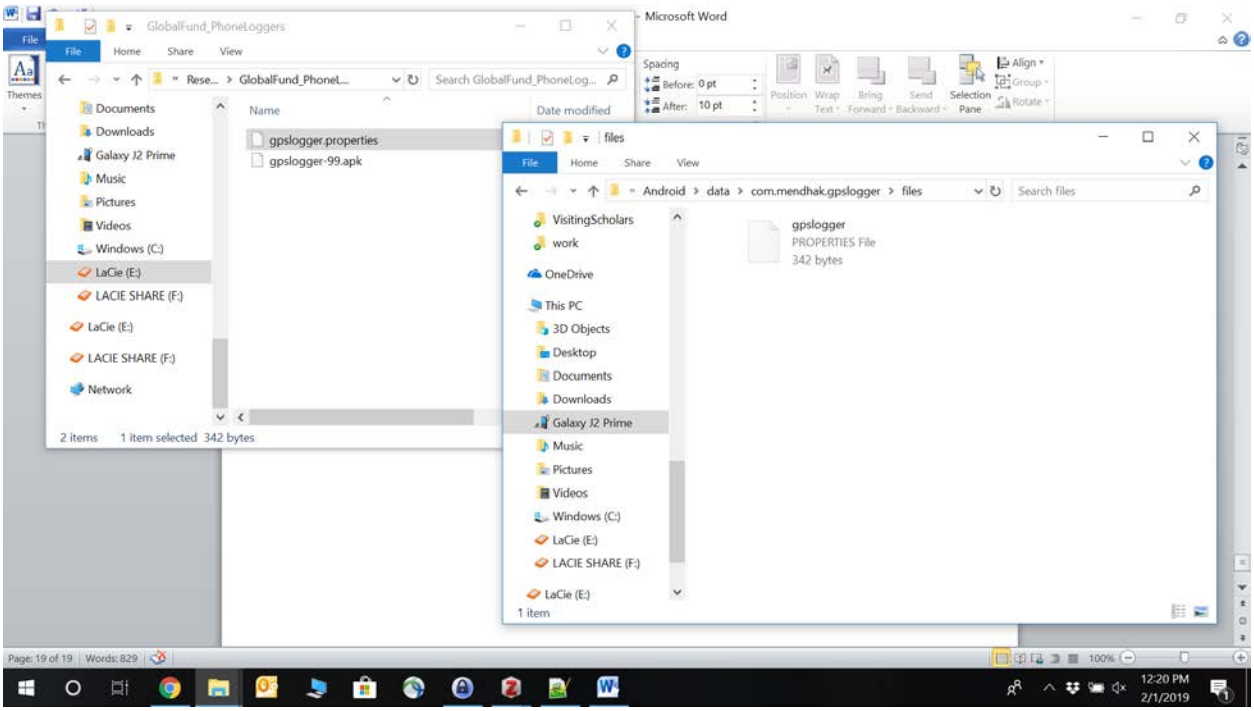
*in this protocol "PROPERTIES file" is used to indicate a file type with the file extension: ".properties"

2.) Save the **PROPERTIES** file in a clearly named folder for this project.

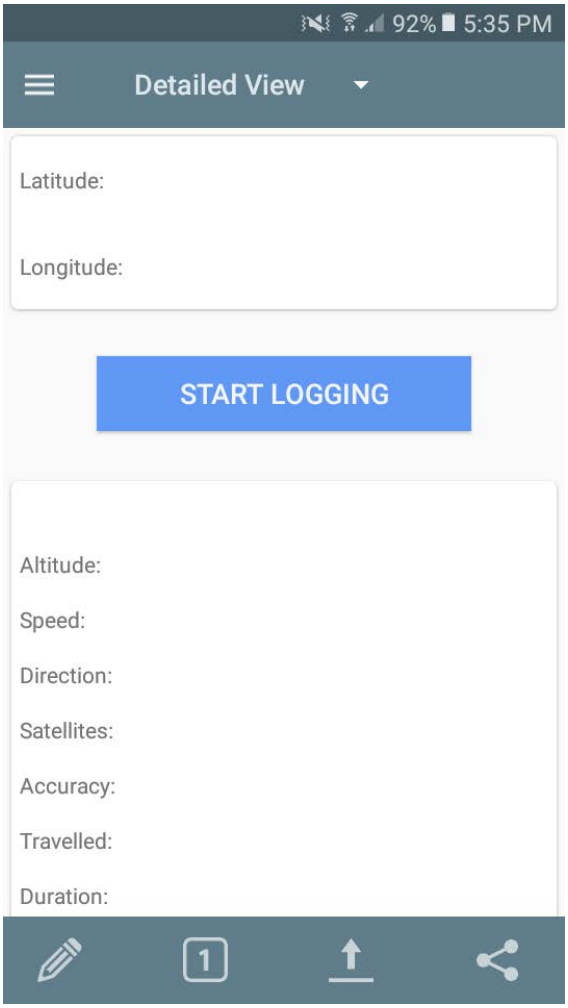
3.) Place the **PROPERTIES** file in the default folder for the application. For this phone the default location was:

This PC\Galaxy J2 Prime\Phone\Android\data\com.mendhak.gpslogger\files

You can copy and paste or click and drag the file into this folder. Afterwards, unplug the mobile device from the computer.



4.) Open the application and start the logger by pressing "**START LOGGING**". You can check the status by changing the view to **Log View**. Restart the mobile device. After it has restarted, the logging application should automatically update based on the preferences listed in the **PROPERTIES** file.



D. Application preference details

A list of application preferences and their names in Java can be found at:

https://github.com/mendhak/gpslogger/blob/master/gpslogger/src/main/java/com/mendhak/gpslogger/common/PreferenceNames.java

It does not appear to be necessary to pass all possible preferences to the application. The only preferences that are listed below are either: a.) those that are desired for use, and b.) those that come switched on by default, but that we want to turn off.

Preferences are listed below (in ***bold blue italics***), with their values and a brief description for the reasoning behind the preference values. They are ordered by the Options windows shown on pages 12 – 14.

Application preferences:

/*General Options */

startonbootup=true

This tells the application to begin logging when the mobile device is started. This is set to "true" so that if the device is turned off, the logger will automatically begin once it is turned back on.

hide_notification_buttons=true

This tells the application to hide the application notification buttons so that it isn't always apparent that the device is on and so that it doesn't become a nuisance to the user.

hide_notification_from_status_bar=true

This tells the application to hide the application notification symbol from the status bar so that it isn't always apparent that the device is on and so that it doesn't become a nuisance to the user.

latlong_display_format=DECIMAL_DEGREES

This tells the application to log geographic references in decimal degrees.

/*Logging details*/

log_gpx=false

This tells the application to turn off logging of GPX files, which will take up space on the device and which we are unlikely to use for this project.

log_plain_text=true

This tells the application to log geographic references in a CSV file.

new_file_prefix_serial=true

This tells the application to log geographic references in file that is labeled by the mobile device's identification number.

/*Performance*/

listeners=list.add(gps, network)

This tells the application to use GPS devices when possible and network information when not. Network locations are based off of cell towers or WiFi access points. The network location is less precise than the GPS location, but offers useful locational information in the absence of GPS satellite based locations. More information on geographic referencing through android devices is provided in the reference section. The type of locational reading that has been recorded (GPS or network) is noted in the log file.

time_before_logging=3600

This tells the application to take a reading every 3600 seconds (every hour). There isn't an easy, perfect choice for this setting. More frequent readings will lead to a drain on the mobile device's battery, will result in massive datasets, and will result in redundant data. Infrequent readings can result in missing important human movement patterns. Error in GPS readings (for example, from GPS drift) are more easily detected when there are frequent readings because it is possible to make assumptions about how far an individual could possibly move in a given amount of time, if that time unit is relatively small.

keep_fix=true

This tells the application to keep the GPS device on in between readings. It can take time for a GPS device to "wake up". While keeping the GPS device on will lead to battery drain, it will increase

locational accuracy. It is expected that since the mobile device is a phone, participants will be likely to keep it charged when possible.

absolute_timeout=900

This tells the application to quit trying to find a GPS signal after 900 consecutive seconds (15 minutes) of trying. It will try again based on the *time_before_logging* setting time.

/*begin actual script*/

```
/*General Options */

startonbootup=true
hide_notification_buttons=true
hide_notification_from_status_bar=true
latlong_display_format=DECIMAL_DEGREES

/*Logging details*/

log_gpx=false
log_plain_text=true
new_file_prefix_serial=true

/*Performance*/

listeners=list.add(gps, network)
time_before_logging=3600
keep_fix=true
absolute_timeout=900
```

/*end actual script*/

****still need to work out collecting the data on a server. There are several ways to accomplish this****

One option is to: `LOG_TO_URL`

Another option is to: `AUTOSEND_GOOGLEDRIVE_ENABLE` or `AUTOSEND_FTP_ENABLED`

E. References and miscellaneous

get the “apk” file from: https://github.com/mendhak/gpslogger/releases

or get the app from the google play store:

https://play.google.com/store/apps/details?id=com.mendhak.gpslogger&hl=en_US

learn GPSlogger app basics: https://gpslogger.app/

the GPSlogger github page: https://github.com/mendhak/gpslogger/

Some info on android location providers (GPS, network, passive):
https://developerlife.com/2010/10/20/gps/

GPS explanations: https://www.e-education.psu.edu/geog862/node/1407

Periodically force a question about whether or not the participant has a fever?

For now keeping logs in CSV files. Is it better to only have this sent to a server or to the cloud? Good to have a “hard” copy on the device in case the cloud or transmission to the cloud fails? I’m currently thinking that it won’t accumulate too much data (probably around 230kb, see calculations below). The files can both be sent to a central server and kept on the phone as a backup.

bytes per reading/log	100
hours/day	24
days/month	31
months participation	3

223200	bytes accumulated
223.2	kilobytes accumulated