BACH KHOA UNIVERSITY

COMPUTER SCIENCE AND COMPUTER ENGINEERING



Assignment 1 Report

**Computer Network**

Lecturer: Nguyễn Mạnh Thìn  
Student: Nguyễn Chí Bình Nguyên (1552253)

Nguyễn Tấn Sang (1552314)

1. **Characteristics of the application**

* This is a simple IoT application on Android smartphone. This application can run on Android 4.0 and above, it also request permission for GPS, and Internet.
* Android device uses the GPS sensor to collect data about longitude and latitude then send to Cloud. From Cloud, it will handle and send back all related location information which based on user’s position in real-time.

1. **Functions of the application**

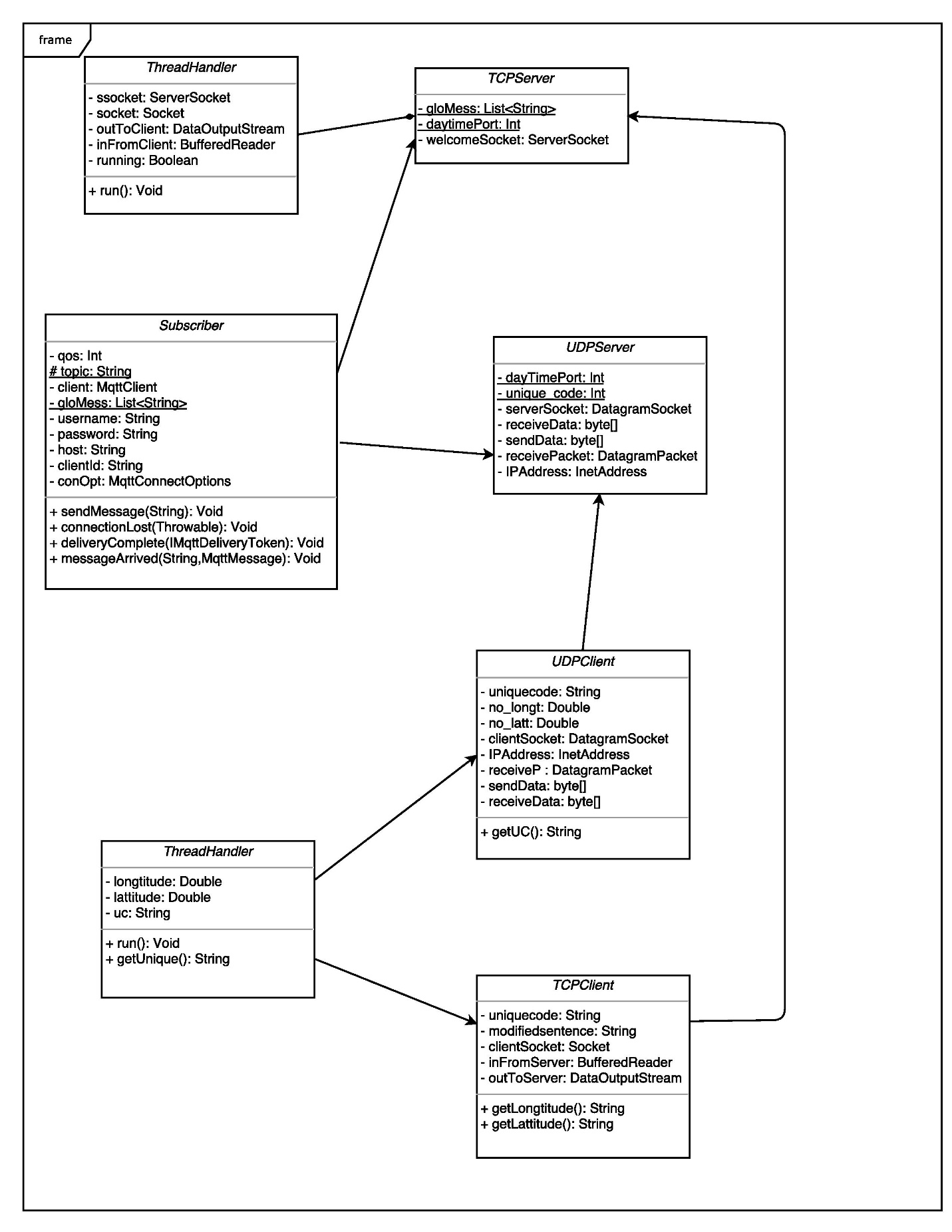
* Collect GPS data from the users.
* Display the latitude and longitude.
* Display the address.
* Display the speed.

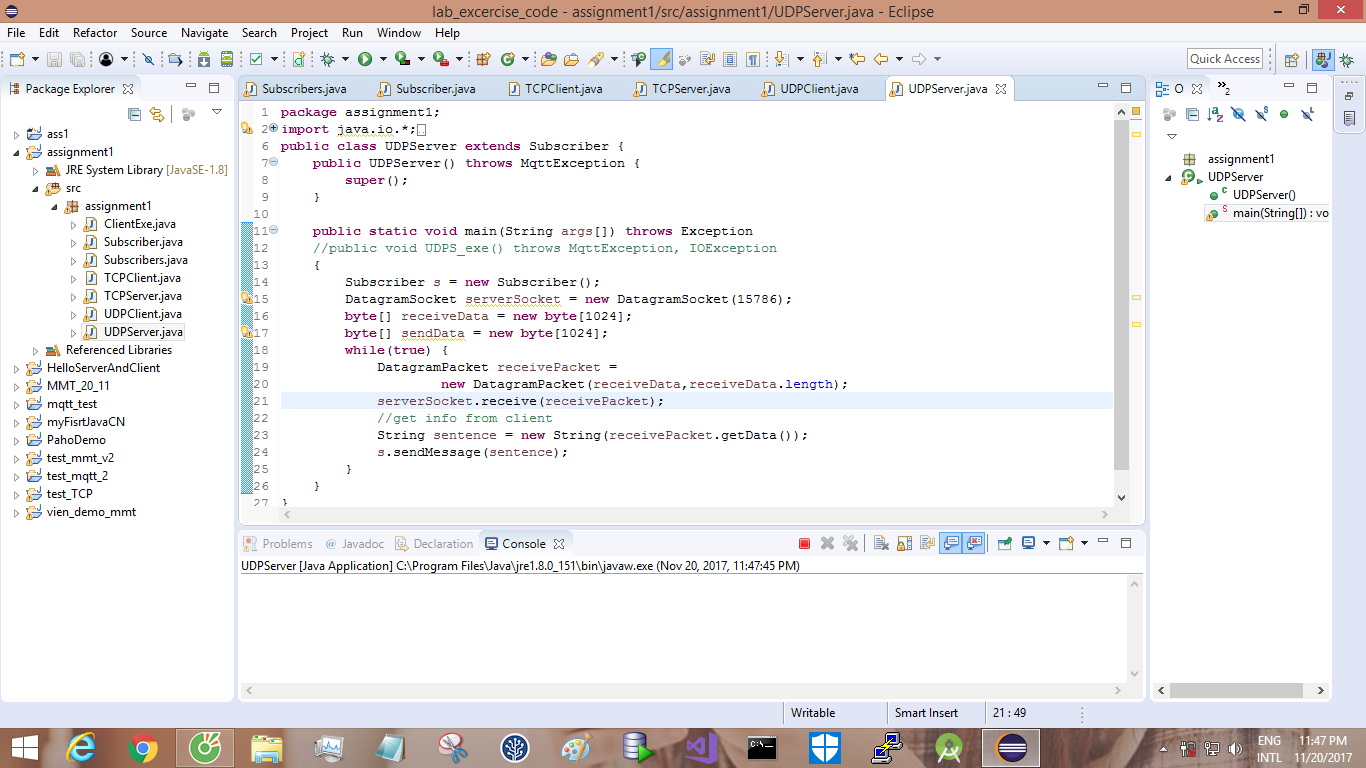
1. **Detail structure of system**
   * Client: This is your application on the Android mobile collecting data from sensor devices as request. All collected data will be transferred to a Gateway through UDP protocol.
   * Gateway: Data collecting station. There can be one or many Gateway, depend on the situation. Each Gateway can manage a group of one or many client. Data of client will be transferred directly to its manager (Gateway) through UDP protocol.
   * CloudMQTT: Data center. All collected data at all Gateway will be transferred to Data center and stored there. You can create one by register on cloudmqtt.com. Gateway can transfer data to CloudMQTT using MQ Telemetry Transport Protocol (MQTT).
   * Analytic Application: this one will handle all stored data of CloudMQTT. It will analyze the data and give back useful information for users. This job will be done in a different server. The result will be transferred to client (mobile application) through your defined protocol (using TCP protocol in transport layer) and display it on the mobile.
2. **Detail of all protocols using in the application**

* UDP:
* In [computer networking](https://en.wikipedia.org/wiki/Computer_network), the **User Datagram Protocol** (**UDP**) is one of the core members of the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite). With UDP, computer applications can send messages, in this case referred to as [*datagrams*](https://en.wikipedia.org/wiki/Datagram), to other hosts on an [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP) network. Prior communications are not required in order to set up [communication channels](https://en.wikipedia.org/wiki/Communication_channel) or data paths.
* UDP uses a simple [connectionless communication](https://en.wikipedia.org/wiki/Connectionless_communication) model with a minimum of protocol mechanism. UDP provides [checksums](https://en.wikipedia.org/wiki/Checksum) for data integrity, and [port numbers](https://en.wikipedia.org/wiki/Port_numbers) for addressing different functions at the source and destination of the datagram. It has no [handshaking](https://en.wikipedia.org/wiki/Handshaking) dialogues, and thus exposes the user's program to any [unreliability](https://en.wikipedia.org/wiki/Reliability_(computer_networking)) of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection.

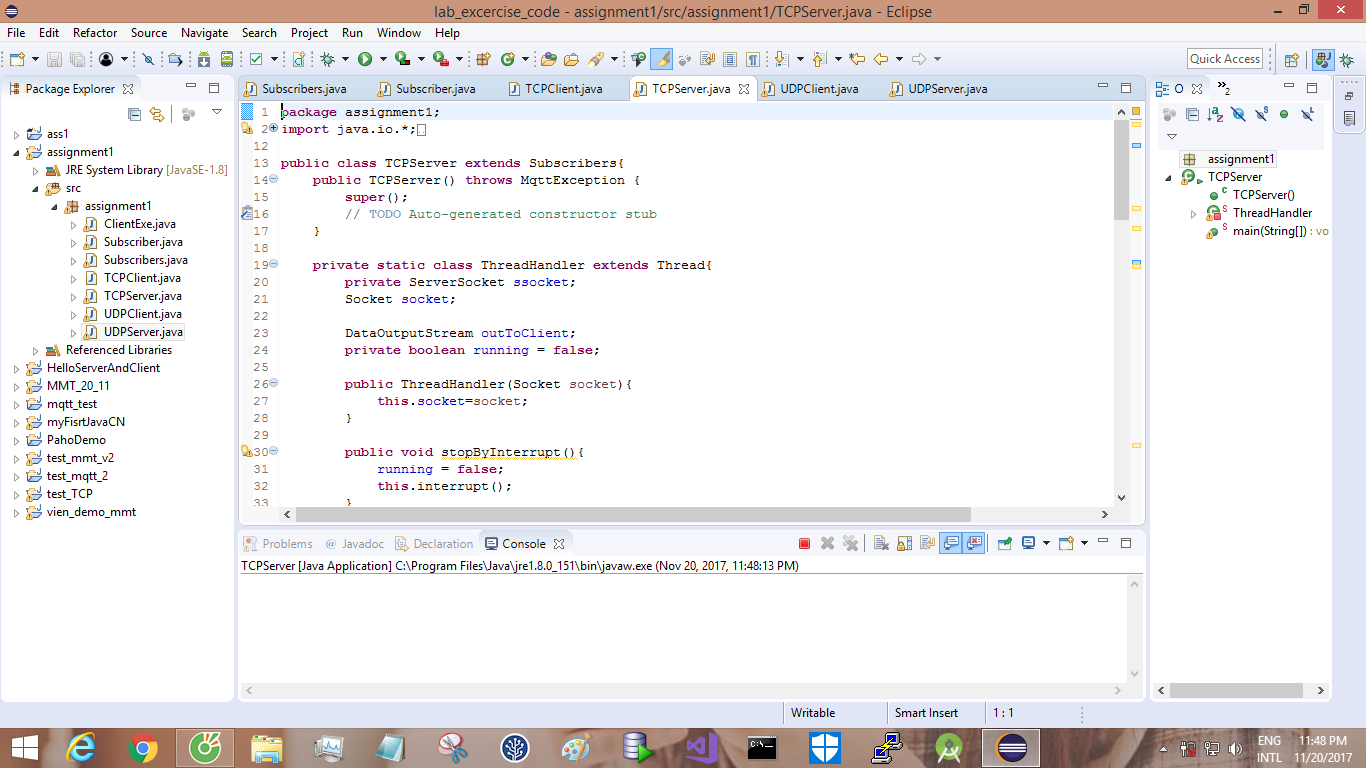
# In our assignment, we use UDP for sending the longitude and latitude. Particularly, we have two class for UDP: UDPClient and UDPServer. UDPClient takes the data of longitude and latitude from the GPS, then transform that data into packet and send that packet to UDPServer. UDPServer now can be known as a gateway, and after taking the packet from UDPClient, UDPServer take the responsibility for sending the information from that received packet to CloudMQTT and UDPServer also create an “Unique code” which attach with the message before send it to CloudMQTT. After that, UDPServer send that “Unique code” to client packet(UDPClient and TCPClient) for checking later.

* TCP:
* The **Transmission Control Protocol** (**TCP**) is one of the main [protocols](https://en.wikipedia.org/wiki/Communications_protocol) of the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite). It originated in the initial network implementation in which it complemented the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP). Therefore, the entire suite is commonly referred to as [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP). TCP provides [reliable](https://en.wikipedia.org/wiki/Reliability_(computer_networking)), ordered, and [error-checked](https://en.wikipedia.org/wiki/Error_detection_and_correction) delivery of a stream of [octets](https://en.wikipedia.org/wiki/Octet_(computing)) between applications running on hosts communicating by an IP network. Major Internet applications such as the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), [email](https://en.wikipedia.org/wiki/Email), [remote administration](https://en.wikipedia.org/wiki/Remote_administration), and [file transfer](https://en.wikipedia.org/wiki/File_transfer) rely on TCP.
* We also have two classes for TCP in our assignment, including TCPClient and TCPServer. TCPServer take the message (which contain information we already sent from UDPClient) from CloudMQTT, TCPSever also play a role as “Analytic Application”, it will handle with the information of the message(analyzing the string and comparing between the “Unique code” of message with the “Unique code” of client packet) and give the useful information after finish. The final result will be transferred to TCPClient. TCPClient is the final link of a whole chain, it receive the result and handle the result (such as display the current address or city for user).

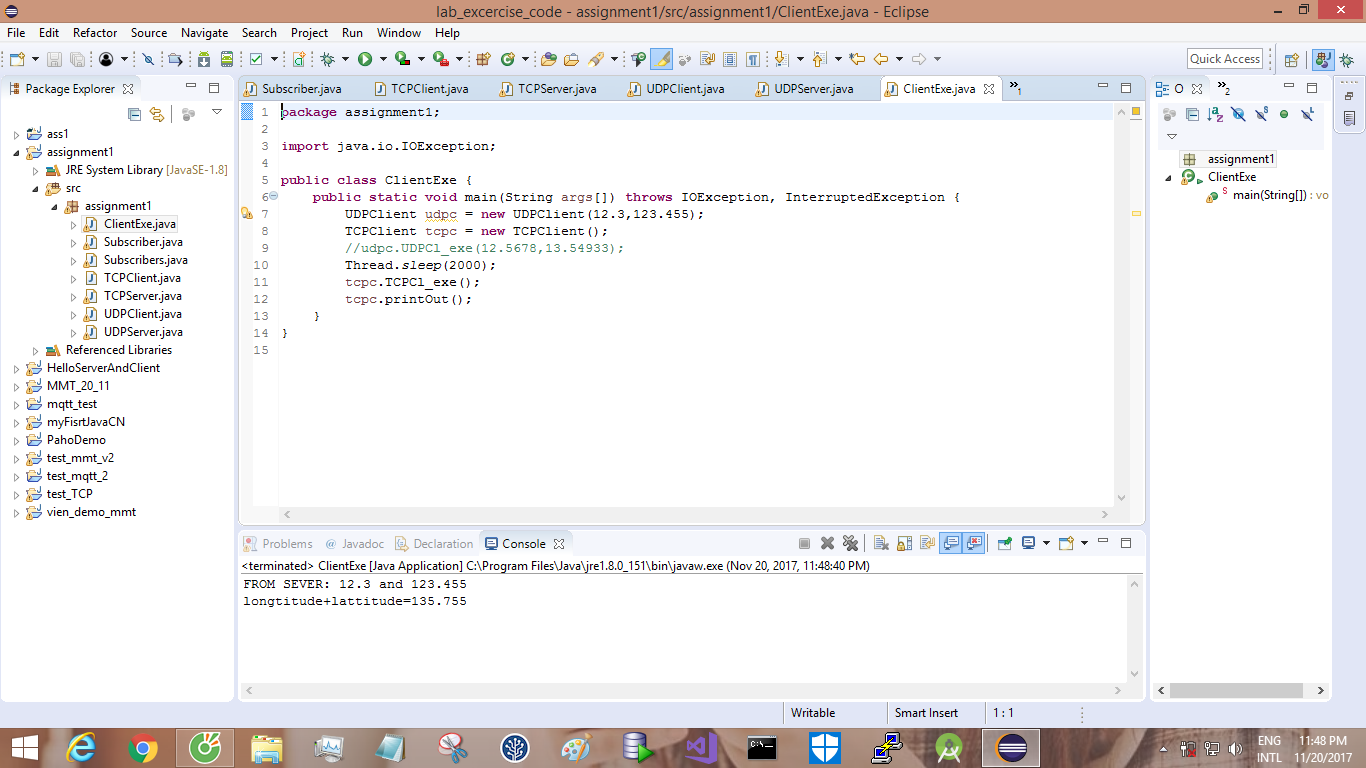
1. **Class diagram**
2. **Achieved results**
   * We can send and receive message between client-server through both UDP and TCP connection. This work is done on Eclipse.



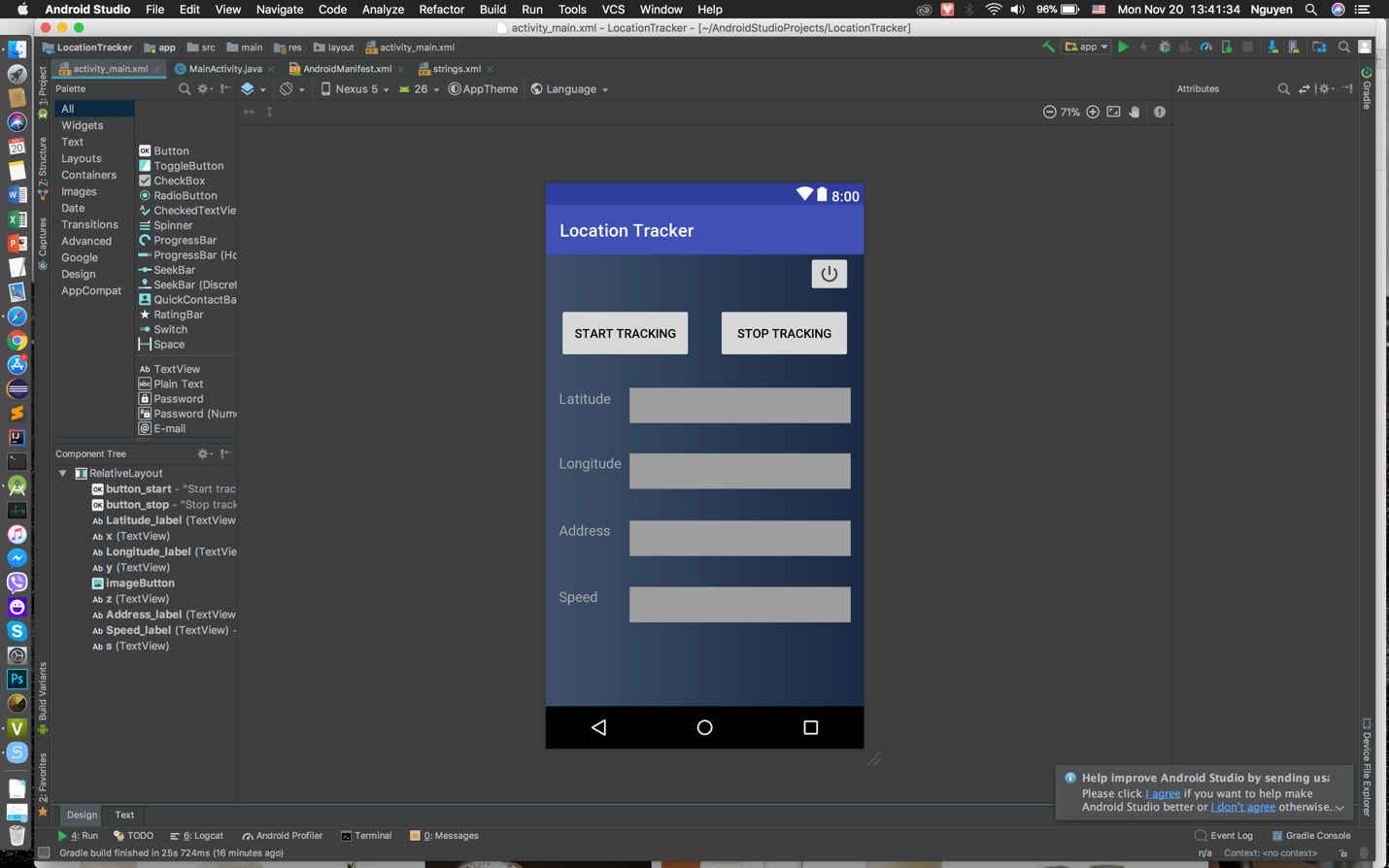
* Above is UDP server is running.



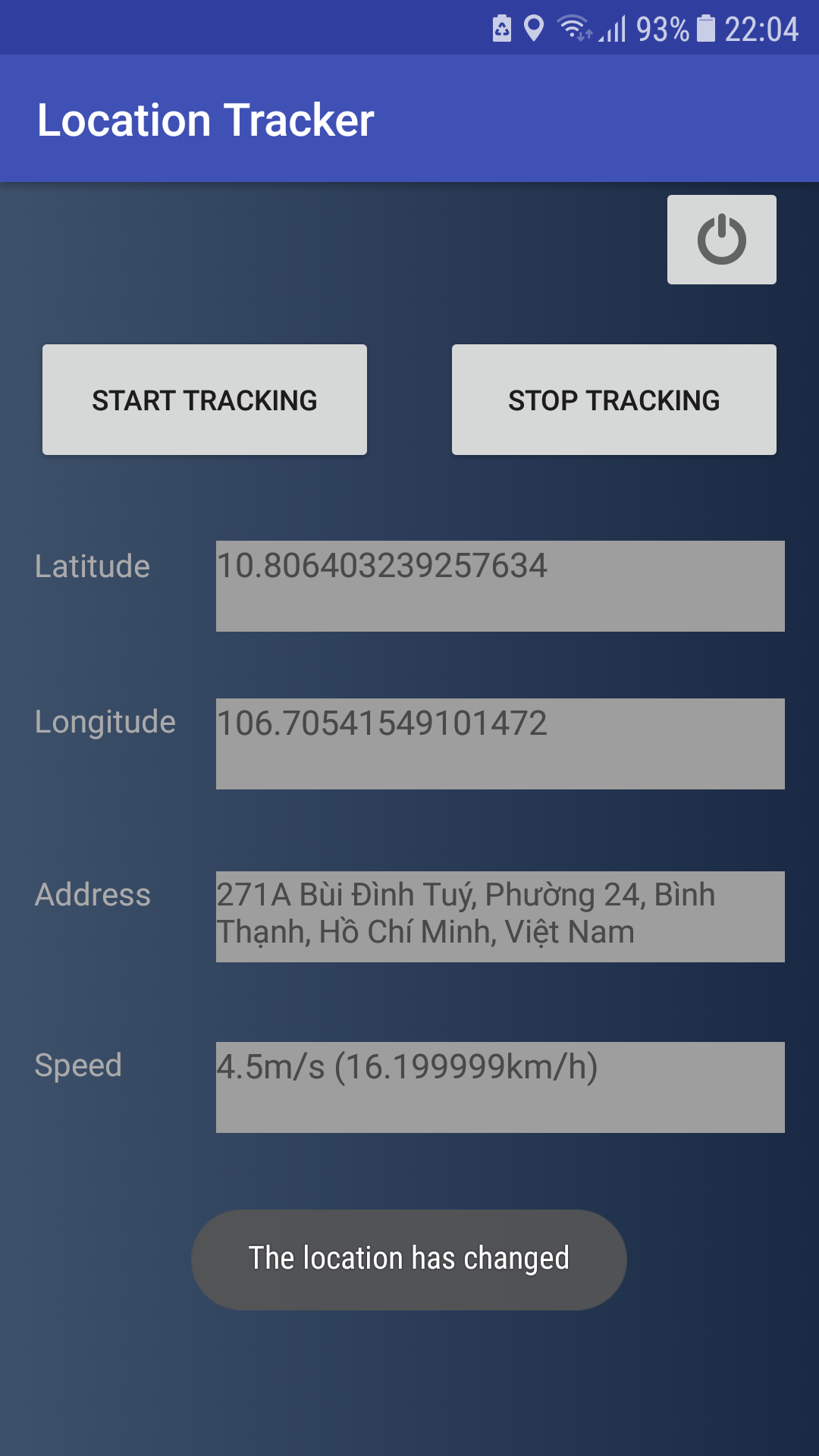
* Above is TCP server is running.



* Above is a demo class for proving that they work.
* We create the users interface on Android device that matches our descriptions above. This work is done on Android Studio.



* + Then, we compile the app and run likes the picture below.



1. **Extras function of the application**
   * Just need to start the system one time, it will refresh every time the users change their location.
   * When the users press “START TRACKING” or “STOP TRACKING” a small Toast also appears to notify.
   * When the users change their location, a small Toast appears to notify them and the application automatically updates new information.
   * Terrain Google map with satellite data, surrounded 300m around the users.
2. **Manual of the application**
   * Install the APK file manually from Android Studio.
   * The application will automatically open for you.
   * Press the “START TRACKING” button for the application to start searching the location and also information which is related to the location.
   * The application will begin its services after a few seconds then displays the result directly onto the device’s screen.
   * Press the “STOP TRACKING” to stop searching location.
   * Press the icon above left corner to exit.