**DESIGN AND IMPLEMENTATION OF LOCATION TRACKING APPLICATION**

**OVERVIEW**:

In this application, we will be using GPS and accelerometer to determine the current location of a device. Initially, we display the current location and display it on the map and then later use accelerometer to calculate approximate location using the initial position, accelerometer, time and current location in the map. The initial locations are used to calculate the future locations and accelerometer data is used to improve accuracy. To save battery power, instead of going for the highest accuracy we can trade-off between power and accuracy and use other techniques as mentioned above to obtain better location accuracy.

**COMPONENTS USED:**

| Maps | Sensor | Manual Functions | Data elements |
| --- | --- | --- | --- |
| Fragment | Sensor Manager | On Click Listener | Override Methods |
| Location Manager | Location Listener | Sensor Event Listener | UI Elements |

* The main components used in the application will be a sensor accelerometer, map fragment and google play services implementations are used in the background to allow certain functionalities.
* A Sensor Manager variable is used to handle the sensor and any mathematical functions needed to process the sensor data.
* We will also be using a timer in the application because it’s necessary to calculate the values every few seconds to update the current location on the screen.
* Apart from this, we will be using a handler to update the UI whenever there is a change in the location to avoid the race condition.
* Manual functions are created to calculate the location of the device which includes all the additional processing.
* Since the application only starts running when the switch is on, we keep it on by default as it will immediately start the application without manual inference and perform all the necessary calculation behind the screen.
* The Sensor Event listener checks whenever there is a change in each sensor and triggers the data processing.
* The override methods used are Onstart(), onPause(), and onStop() to register and unregister the listeners respectively according to their requirement.

**UI**:

The UI of this application mainly consists of three numeric fields and a map fragment to display the location, latitude, and longitude values describing the current location of the device. We add a switch which when on will be able to kickstart the application and the necessary calculation is performed in the background to display the values in the above-mentioned fields and in the map as well.

**ALGORITHM**:

In simple terms, we will be implementing a current location tracker high pass filtering of the data which is obtained from the gyroscope and low pass filtering will be done for combined values obtained from accelerometer and magnetometer. Before this calculation is performed, each of these sensors’ data will undergo additional processing.

We can divide the algorithm into different stages:

1. In the first stage, we build the project and the necessary components are created in order to display the user’s location.
2. By using location manager and Location Listener, we can display the GPS based user location and use requestLocationUpdates which will continuously get the current location updates.
3. Once this is done, we will monitor the accelerometer data and use SensorEventListener to get the acceleration values.
4. We can apply the dead reckoning algorithm in the next phase wherein we have the user’s intial location, time and acceleration to calculate the current location.
5. Now, we update this location on the Map Fragment.
6. The above steps will be looped for every few seconds based on the input Time\_interval.
7. We require location updates in a less frequent manner as we are using low power sensors like accelerometer to calculate the current location.