**Recorder Service**

Initialize sensors and variables

Initialize available location service (GPS provider or network provider)

Sync local database with backend

While service active

Get live jobs from local database

If job found choose first record

While job status is alive

If upload timeout reached upload data

If job is in valid location range and time range

persist sensor reading data into local database

thread sleep for the job frequency till next reading needed

else

if job expired or out of location range

update the correct job status

sync local db with backend

get live jobs from local db

if there is any live job select the first job

else if job is yet to start

sleep the thread till it start it

update the job status

else

sleep thread for constant amount of time

sync local db with backend

**Get Jobs**

Get jobs is one of the important operations on the back end. Because it is the one who select the most appropriate mobile node for a particular job. Following point are considered when selecting a proper job.

1. When selecting the appropriate job it is very important to check the location range of the mobile node with job location. Therefore to find distance between two GPS location is calculated using following formula.

location1 = (lat1, lon1)

location2 = (lat2, lon2)

(these longitude, latitude values need to be from radians.)

dist = arccos(sin(lat1) \* sin(lat2) + cos(lat1) \* cos(lat2) \* cos(lon1 - lon2)) · R

1. Number of nodes that needed to complete the job, should be greater than the node\_count which is the number nodes currently do that job. That is this job still needs more nodes to do its work.
2. Job expire date must be satisfied. It should fall in future.
3. Job start time should satisfied following rules

If job start\_time is null

start\_time = now

if job start\_time is not null and its already in past

start\_time = now

if still start\_time is in future

it will be the start\_time

1. Job expire time should satisfied following rules

If expire\_time is null

expire\_time will be (start\_time + time\_period)

if expire\_time is not null and (start\_time + time\_period) < expire\_time

expire\_time = start\_time + time\_period

else

expire\_time will be itself

1. First order the record set by start time ascending order to find the closest jobs to the current time. And then order records by user rank descending, to get the highest rank users job with priority within the closest jobs.
2. DateTime on sql can not assign as null it will always be "0000-00-00 00:00:00" therefore we need to check for "UNIX\_TIMESTAMP(field)=0". Because UNIX timestamp of a time will 0 when it has 0 values.

**Upload Data**