Beacon Management:

* Need to know where each beacon is and know how to keep track of the time of the devices connected to the beacon.
* Beacons should be able to pinpoint a specific location sing geo-fencing (beacons set in different locations).
* Time the duration of each session when a device is connected to the beacon.
* One active day per day, but all events are stored on the events DB.
* All events are stored, even when the active day has been achieved
  + Availability:
    - The system is not 24/7. The system is not available at night (read only) – solution is during down time store all data on the phone and upload when the system is available still (perhaps a push notification system should be in place during the upload time).

16 active days a month -benefits.

First Priority: Giving active day – reward policy for later is to reward based on active days

Skills/Knowledge to be Acquired:

* Amazon cloud services, java -Java EE, spring-Boot, wild-Fly. How these use REST

**Functional Requirements**

* Beacons have id and id need to be given a location – beacon management
* Timing of devices within the beacon location ( e.g. gyming ), how many devices have connected to the beacons in a day for uses like shop data (No. of customers).
* When user enters a location containing beacons they should be notified that they are in a particular area and additional info should be given for that event if necessary.
* Web interface deals with partner management, and beacon management – choose non-relational DB eg graph DB – (More research needs to be done in this regard).
* Gamification of the system.
* Geo-fencing:
  + Beacons communicate with the phone and phone to AMZN, beacon management
  + GPS secondary priority.
  + Google analytics, firebase (good idea)
* User communication – target user notifications eg (target platinum users).
  + Base is 120000 client
* Amazon scalability, services should be able to handle request/receive data
* Usability must be good, no user manual should be needed but make one.
  + Proper design process should be implemented including prototyping and usability tests, using saturated sampling or otherwise.
* Programming languages:
  + Java 8 + JavaEE, Docker
  + Security:
    - openID, authentication, job token