# workflow diagram - New Page.png

# Application Architecture

The users of this app will have it downloaded on there device firstly, they would be brought to a Login page, if they are unable to login it means that they will need to register on this app to create their own account after registering they will be brought back to login again where this time there attempt will be successful. Upon logging in they will be prompted to pick the boundries around their geo-location if they are a returnning user and do not wish to pick new boundries it means then they may want to look at stored geofences in their database, once either one of these steps are completed they will move forward to set up the Push notification that will be sent out if the boundry is breached, this step is a must and the user will not be able to move on to the next step unless this is done, when this step is been done in the background steps 1 – 3 are been completed of the push notifications chart;

1. The android device will send its sender id and application id to GCM server for registration.
2. When Registration is complete, GCM server issues registration id to android device.
3. After receiving registration id, the device will send registration id to the database for further use.

After the Push notification have been set up and boundries are all picked the GPS on the device monitors the location, it keeps checking to see if the boundries have been broken and if they have not been it keeps going in a loop until it has been broken, this then triggers the Push Notification to be sent out to the device that has broken the boundry, this is done by completing step A & B of the push notification chart;

1. Whenever push notification is needed, our database sends a message to GCM server along with device registration id
2. GCM server will deliver that message to the android device using the device registration id.

Once this step is then done the app automatically jumps back up to monitor the area and thus going in a constant loop untill the user logs out of the app.

# How Push notifications work

1. The android device will send its sender id and application id to GCM server for registration.
2. When Registration is complete, GCM server issues registration id to android device.
3. After receiving registration id, the device will send registration id to the database for further use.
   1. Whenever push notification is needed, our database sends a message to GCM server along with device registration id
   2. GCM server will deliver that message to the android device using the device registration id.

# What is GeoFencing?

Geo-fencing is a feature in a software program that uses the global positioning system or better known as [GPS](http://searchmobilecomputing.techtarget.com/definition/Global-Positioning-System) or radio frequency identification to define geographical boundaries.

Geo-fencing allow an administrator to set up [triggers](http://searchsqlserver.techtarget.com/definition/trigger) so when a device enters or exits the boundaries defined by the administrator, an alert is issued. Many geo-fencing applications incorporate Google Earth, allowing administrators to define boundaries on top of a [satellite](http://searchmobilecomputing.techtarget.com/definition/satellite) view of a specific geographical area.  Other applications define boundaries by longitude and latitude or through user-created and [Web-based maps](http://whatis.techtarget.com/definition/Virtual-Earth).

Geofence virtual barriers can be active or passive. Active geofences require an end user to opt-in to [location services](http://searchnetworking.techtarget.com/definition/location-based-service-LBS) and a mobile app to be open. Passive geofences are always on; they rely on [Wi-Fi](http://searchmobilecomputing.techtarget.com/definition/Wi-Fi) and [cellular data](http://searchmobilecomputing.techtarget.com/definition/cell) instead of GPS or RFID and work in the background.

# What I expect to see as a cutomer of this product?

As a customer of this product and an active member of the Sub Aqua Club what I would like to to see from this product is if I was out diving with the Sub Aqua Club and there was a group of us Diving in a built up area maybe a wreck for instance that we could set up a geo-location boundy around us so if any other boats come into that boundry it will send an alert to the other vesile so they know there is divers in the water.

# How it is going to be implimented and programs needed?

Firstly I will check that i have all the necessary pre-requisites. These include will:

* Android Studio
* An Android device that runs Android or a configured Android emulator. However in order to test entering and leaving a GeoFence, a real device that can be moved around is needed
* The latest version of the Android SDK including the SDK tools component.
* The Google Play Services SDK.

After I have done this step I will then go on to setting up my Android studio and beginning to write my code

# What OS and Why?

The Operating System (OS) im going to be implimenting is Android, there is no clear winner as to which OS is better they are both very evenly matched I have just choosen Adroid as I am an Android OS phone/tablet user and also because I have coded in Android studio before so I will be comleting this project in Android

# What is an API and what ones will be used in the implimentation of the program?

An API (Application Programming Interface) is a set of methods and tools that can be used for building software applications.

Google Maps API will be used in the implimentation of theis project they allow you to display maps on your web site.

Google Cloud Messaging (GXM) will be used to store information needed for push notifications to be sent out