Requirements : -

1. App should open with splash screen with logo
2. It should able to track location of mobile GPS , if GPS is not enable then should ask to enable.
3. After that should show list of doctors available nearby by brief information
4. After clicking on that detail of doctor with speciality , fees,GPS location of clinic and should have option for schedule oppointment .
5. After clicking and scheduling appointment confirmation message should go to doctor , patient and company owner .

Possible Names : -

1. Medicare (rejected)
2. GoHealthy(rejected)
3. Curesure(rejected)
4. CureIsSure (CIS)(finalized)

**Splash Screen Background color :** #13866f

**Logo Design Site :** logomakr.com/4h56zM

Important links :

<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>

http://stackoverflow.com/questions/29339565/calling-rest-api-from-an-android-app

<http://stackoverflow.com/questions/26470117/can-we-connect-remote-mysql-database-in-android-using-jdbc>

<http://stackoverflow.com/questions/18188373/connecting-android-app-to-amazon-rds>

http://loopj.com/android-async-http/

**WiFi – GPS Location :**

After coming from splash screen to Main Activity is should first check for GPS location , if GPS location is not on then give pop up to on GPS , then check for net connection if no Internet then ask to on wifi or data pack .

After successful gps and wifi connection , get pin code and access list of all doctors based on pin code

based on Doctors and Hospitals clicked.

If clicked on button (Doctors or Hospitals) then again do the same above .

**List View :**

Message to both doctor and patient about appointment

Customer detail

Photo , Rating , Name of Doc/Hos/test , specialization , location ,

Distance in km

Detail , Link for Book appointment .

Photo , Rating , Name of Doc/Hos/test , specialization , location ,

Distance in km

**Tables :**

**Doctor table**

|  |  |
| --- | --- |
| **Primary key** | **CIS\_DOC\_ID** |
|  | **LONGITUDE** |
|  | **LATITUDE** |
|  | **FULL ADDRESS** |
|  | **DESCRIPTION (DETAIL)** |
|  | **MAIL ID** |
|  | **CONTACT** |
|  | **EXPERIENCE (SINCE)** |
|  | **SPECIALIZATION** |
|  | **DOCTOR NAME** |

**Hospital Table :**

|  |  |
| --- | --- |
| **Primary key** | **CIS\_HOS\_ID** |
|  | **LONGITUDE** |
|  | **LATITUDE** |
|  | **FULL ADDRESS** |
|  | **ABOUT HOSPITAL** |
|  | **MAIL ID** |
|  | **CONTACT** |
|  | **ESTABLISHED** |
|  | **SPECIALIZATOIN (FACILITY)** |
|  | **HOSPITAL NAME** |

**Test Center Table :**

|  |  |
| --- | --- |
| **Primary key** | **CIS\_TEST\_ID** |
|  | **LONGITUDE** |
|  | **LATITUDE** |
|  | **FULL ADDRESS** |
|  | **DESCRIPTION** |
|  | **MAIL ID** |
|  | **CONTACT** |
|  | **ESTABLISHED** |
|  | **SPECIALIZATION (TYPES OF TEST)** |
|  | **TEST CENTER NAME** |

**Medical shop :**

|  |  |
| --- | --- |
| **Primary key** | **CIS\_MED\_ID** |
|  | **LONGITUDE** |
|  | **LATITUDE** |
|  | **FULL ADDRESS** |
|  | **MAIL ID (OPTIONAL)** |
|  | **CONTACT** |
|  | **TEST CENTER NAME** |

**Appointment table :**

|  |  |
| --- | --- |
| **Primary key** | **UNIQUE\_KEY\_APPOINTMENT** |
|  | **NAME OF PATIENT** |
|  | **CIS\_DOC\_ID/CIS\_HOS\_ID/CIS\_TEST\_ID** |
|  | **STATUS ( BOOKED OR COMPLETED OR COMPLETED TREATMENT)** |
|  | **TIME AND DATE OF APPOINTMENT** |
|  | **PAID OR NOT** |

**S3 –**

**Thumbnail image : For showing in listview**

**Collection of image : For detail**

**Mapping using CIS\_DOC\_ID , CIS\_HOS\_ID**

**Flow for connection with database :**

1. Wrong method database can be hacked

AWS Database

Android app (client)

1. Right method using open api

AWS Database

Web services/app open api restful

Service provider

Android app (client)

Service consumer

**Web Services can be implemented by using java or php .**

**Will do web services in java .**

**How do deploy java web application / api / services on live server (not local host).**

**APIs call :**

1. **On app opened Main Activity it should call “/detailDoctor” using “longitude” and “latitude” as params to get list of doctors nearby to user’s “longitude” and “latitude”.**
2. **On click of “Hospitals” it will call “/detailHospital” using “longitude” and “latitude” as params to get list of hospitals nearby to user’s “longitude” and “latitude”.**
3. **On click of “Doctors” it will call “/detailDoctor” using “longitude” and “latitude” as params to get list of doctors nearby to user’s “longitude” and “latitude”.**
4. **On click of “Test Center” it will call “/detailTestCenter” using “longitude” and “latitude” as params to get list of test centers nearby to user’s “longitude” and “latitude”.**
5. **On click of “Medical shop” it will call “/detailMedicalShop” using “longitude” and “latitude” as params to get list of medical shops nearby to user’s “longitude” and “latitude”.**
6. **On click of “Schedule” it will call “/detailschedule” using “unique\_key\_appointment” to get schedule detail of patient.**
7. **On click any doctor in list of doctors it will call “/detailIndividualDoctor” using “cis\_doc\_id” to get detail of doctor.**
8. **On click any hospital in list of hospitals it will call “/detailIndividualHospital” using “cis\_hos\_id” to get detail of hospital.**
9. **On click any test center in list of test centers it will call “/detailIndividualTestCenter” using “cis\_test\_id” to get detail of test center.**
10. **On click any medical shop in list of medical shops it will call “/detailIndividualMedicalShop” using “cis\_med\_id” to get detail of medical shop**