Mobile Health Monitor Documentation

1. Purpose of Application

This project is an implementation of a personalized health monitor application for managing the user's wellbeing and health related situations. The health situation can be divided into Hypotension, Hypertension (Stage 1, 2, 3), Heat Stroke and Healthy.

The application will inform the user which of these situations occurring in real time using simple rules. Four sensory data will be generated to measure health situation, such as heart rate (HR) (bpm), systolic blood pressure (SBP) and diastolic blood pressure (DBP) (mmHg), and room temperature. For simplicity, we just use Data Generator to generate the health data, which is a Java Object implements Runnable to generate data continuously.

2. Important Design Consideration

- **Publish/Subscribe model**
  - Used for publishing the generated data and allowing other classes to subscribe to the generated data
  - DataGenerator extends Publisher
  - Rule class and GraphData class implments Subscriber
- **Writing data to database**
  - Generated data written into the SQLite Database.
- **Deploying web service**
  - Google Places API: a service that returns information about Places — defined within this API as establishments, geographic locations, or prominent points of interest — using HTTP requests. Place requests specify locations as latitude/longitude coordinates.
  - Geocoding API: is the process of converting addresses (like "Monash University Caulfield") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map.
- **Android AsyncTask**
  - Used when deploying web service
  - Allow to assign parameters (such as location, maker on map) in the background process with doInBackground() function
  - Use doPostExecute() function to generate the result
- **Android Broadcast**
  - In order to make the graph of health data refreshed continuously.
  - MonitorScreen class is the one who broadcast data, and SimpleGraph is the broadcast receiver.

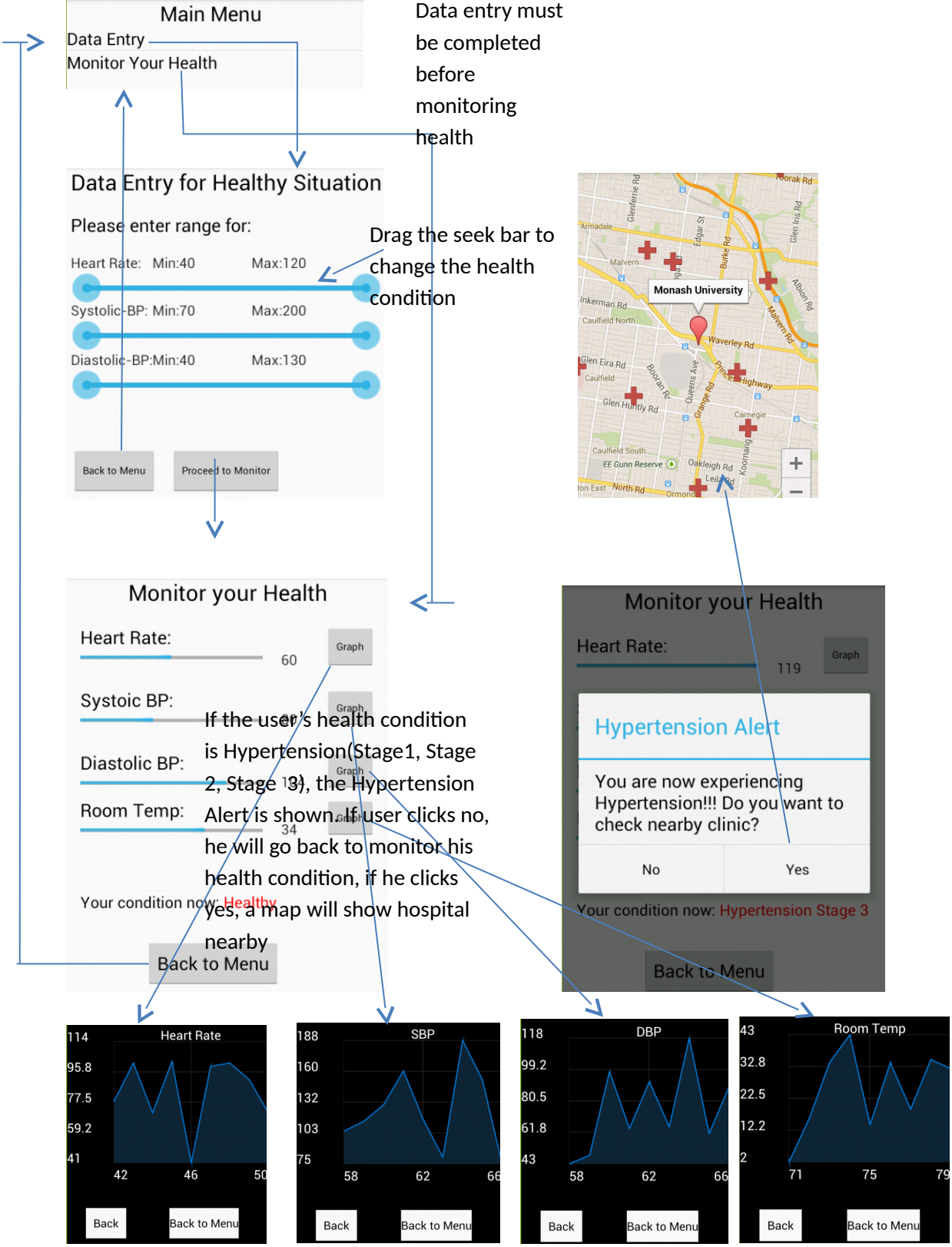
3. Test Plan

3.1 Test Environment

- Platform Details: Eclipse Android Developer Tools
- Android Version: Android 4.1.2 with google play service installed
- Toolkit and packages: google-play-services_lib

3.2 How Operations Performed

- Just run the project



4. Database

Data is inserted into the database in the monitor health screen.

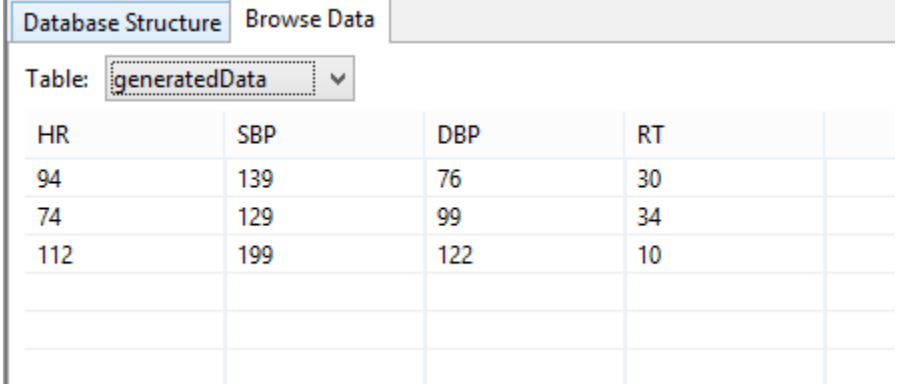
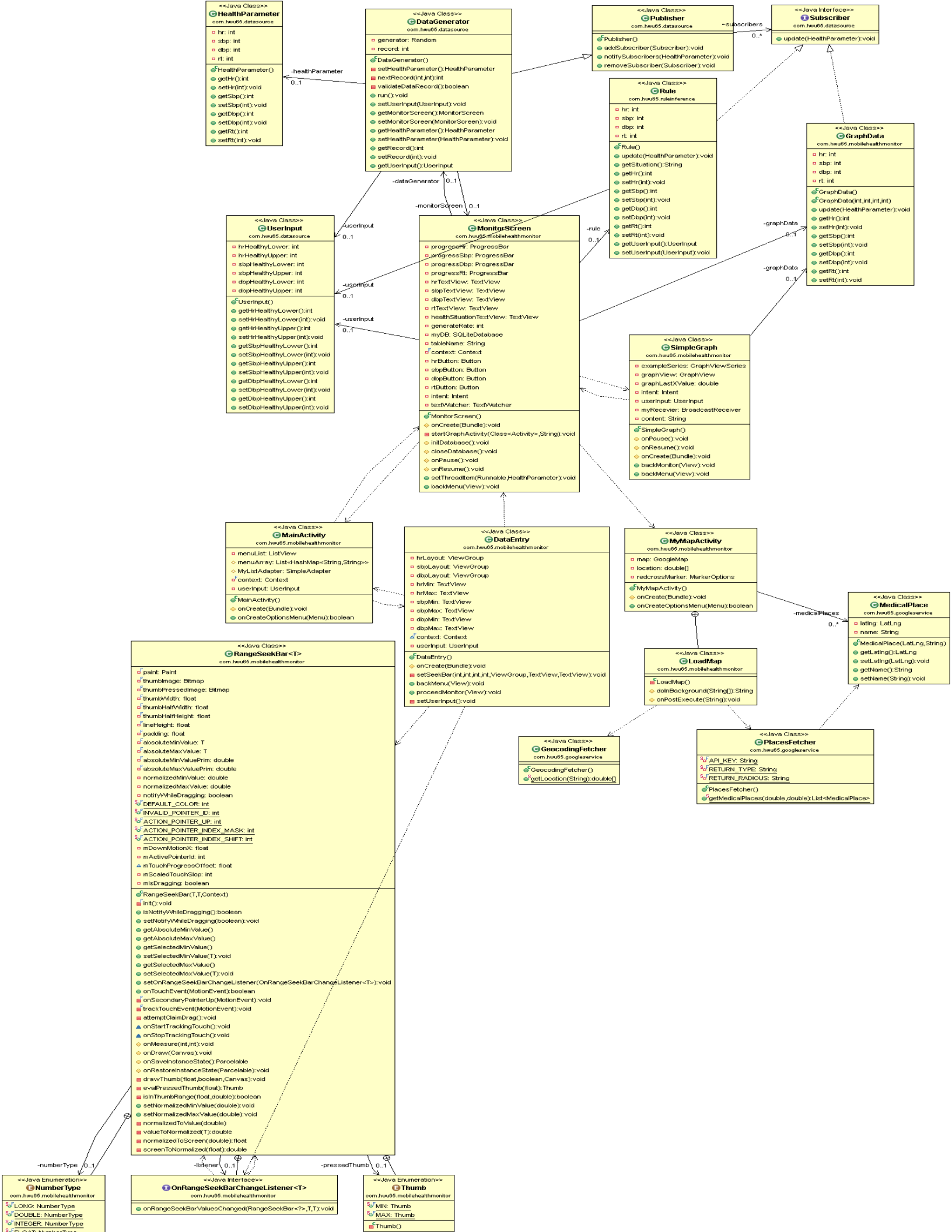

Database Structure Browse Data

Table: generatedData

HR	SBP	DBP	RT
94	139	76	30
74	129	99	34
112	199	122	10

5. Class Diagram (with image file availabe)


«Java Class»
HealthParameter
com.hwu65.datasource
hr: int
sbp: int
dbp: int
rt: int
HealthParameter()
getHr():int
setHr(int):void
getSbp():int
setSbp(int):void
getDbp():int
setDbp(int):void
getRt():int
setRt(int):void
-healthParameter
0..1
«Java Class»
DataGenerator
com.hwu65.datasource
generator: Random
record: int
DataGenerator()
setHealthParameter():HealthParameter
nextRecord(int,int):int
validateDataRecord():boolean
run():void
setUserInput(UserInput):void
getMonitorScreen():MonitorScreen
setMonitorScreen(MonitorScreen):void
getHealthParameter():HealthParameter
setHealthParameter(HealthParameter):void
getRecord():int
setRecord(int):void
getUserInput():UserInput
«Java Class»
Publisher
com.hwu65.datasource
Publisher()
addSubscriber(Subscriber):void
notifySubscribers(HealthParameter):void
removeSubscriber(Subscriber):void
~subscribers
0..*
«Java Interface»
Subscriber
com.hwu65.datasource
update(HealthParameter):void
«Java Class»
Rule
com.hwu65.ruleinference
hr: int
sbp: int
dbp: int
rt: int
Rule()
update(HealthParameter):void
getSituation():String
getHr():int
setHr(int):void
getSbp():int
setSbp(int):void
getDbp():int
setDbp(int):void
getRt():int
setRt(int):void
getUserInput():UserInput
setUserInput(UserInput):void
«Java Class»
GraphData
com.hwu65.mobilehealthmonitor
hr: int
sbp: int
dbp: int
rt: int
GraphData()
GraphData(int,int,int,int)
update(HealthParameter):void
getHr():int
setHr(int):void
getSbp():int
setSbp(int):void
getDbp():int
setDbp(int):void
getRt():int
setRt(int):void
-dataGenerator
0..1
-monitorScreen
0..1
«Java Class»
UserInput
com.hwu65.datasource
hrHealthyLower: int
hrHealthyUpper: int
sbpHealthyLower: int
sbpHealthyUpper: int
dbpHealthyLower: int
dbpHealthyUpper: int
UserInput()
getHrHealthyLower():int
setHrHealthyLower(int):void
getHrHealthyUpper():int
setHrHealthyUpper(int):void
getSbpHealthyLower():int
setSbpHealthyLower(int):void
getSbpHealthyUpper():int
setSbpHealthyUpper(int):void
getDbpHealthyLower():int
setDbpHealthyLower(int):void
getDbpHealthyUpper():int
setDbpHealthyUpper(int):void
-userInput
0..1
-userInput
0..1
-userInput
0..1
«Java Class»
MonitorScreen
com.hwu65.mobilehealthmonitor
progressHr: ProgressBar
progressSbp: ProgressBar
progressDbp: ProgressBar
progressRt: ProgressBar
hrTextView: TextView
sbpTextView: TextView
dbpTextView: TextView
rtTextView: TextView
healthSituationTextView: TextView
generateRate: int
myDB: SQLiteDatabase
tableName: String
context: Context
hrButton: Button
sbpButton: Button
dbpButton: Button
rtButton: Button
intent: Intent
textWatcher: TextWatcher
MonitorScreen()
onCreate(Bundle):void
startGraphActivity(Class<Activity>,String):void
initDatabase():void
closeDatabase():void
onPause():void
onResume():void
setThreadItem(Runnable,HealthParameter):void
backMenu(View):void
-rule
0..1
-graphData
0..1
-graphData
0..1
«Java Class»
SimpleGraph
com.hwu65.mobilehealthmonitor
exampleSeries: GraphViewSeries
graphView: GraphView
graphLastXValue: double
intent: Intent
userInput: UserInput
myRecevier: BroadcastReceiver
content: String
SimpleGraph()
onPause():void
onResume():void
onCreate(Bundle):void
backMonitor(View):void
backMenu(View):void
«Java Class»
MainActivity
com.hwu65.mobilehealthmonitor
menuList: ListView
menuArray: List<HashMap<String,String>>
MyListAdapter: SimpleAdapter
context: Context
userInput: UserInput
MainActivity()
onCreate(Bundle):void
onCreateOptionsMenu(Menu):boolean
«Java Class»
DataEntry
com.hwu65.mobilehealthmonitor
hrLayout: ViewGroup
sbpLayout: ViewGroup
dbpLayout: ViewGroup
hrMin: TextView
hrMax: TextView
sbpMin: TextView
sbpMax: TextView
dbpMin: TextView
dbpMax: TextView
context: Context
userInput: UserInput
DataEntry()
onCreate(Bundle):void
setSeekBar(int,int,int,int,ViewGroup,TextView,TextView):void
backMenu(View):void
proceedMonitor(View):void
setUserInput():void
«Java Class»
MyMapActivity
com.hwu65.mobilehealthmonitor
map: GoogleMap
location: double[]
redcrossMarker: MarkerOptions
MyMapActivity()
onCreate(Bundle):void
onCreateOptionsMenu(Menu):boolean
-medicalPlaces
0..*
«Java Class»
MedicalPlace
com.hwu65.googleservice
latlng: LatLng
name: String
MedicalPlace(LatLng,String)
getLatlng():LatLng
setLatlng(LatLng):void
getName():String
setName(String):void
«Java Class»
RangeSeekBar<T>
com.hwu65.mobilehealthmonitor
paint: Paint
thumbImage: Bitmap
thumbPressedImage: Bitmap
thumbWidth: float
thumbHalfWidth: float
thumbHalfHeight: float
lineHeight: float
padding: float
absoluteMinValue: T
absoluteMaxValue: T
absoluteMinValuePrim: double
absoluteMaxValuePrim: double
normalizedMinValue: double
normalizedMaxValue: double
notifyWhileDragging: boolean
DEFAULT_COLOR: int
INVALID_POINTER_ID: int
ACTION_POINTER_UP: int
ACTION_POINTER_INDEX_MASK: int
ACTION_POINTER_INDEX_SHIFT: int
mDownMotionX: float
mActivePointerId: int
mTouchProgressOffset: float
mScaledTouchSlop: int
mIsDragging: boolean
RangeSeekBar(T,T,Context)
init():void
isNotifyWhileDragging():boolean
setNotifyWhileDragging(boolean):void
getAbsoluteMinValue()
getAbsoluteMaxValue()
getSelectedMinValue()
setSelectedMinValue(T):void
getSelectedMaxValue()
setSelectedMaxValue(T):void
setOnRangeSeekBarChangeListener(OnRangeSeekBarChangeListener<T>):void
onTouchEvent(MotionEvent):boolean
onSecondaryPointerUp(MotionEvent):void
trackTouchEvent(MotionEvent):void
attemptClaimDrag():void
onStartTrackingTouch():void
onStopTrackingTouch():void
onMeasure(int,int):void
onDraw(Canvas):void
onSaveInstanceState():Parcelable
onRestoreInstanceState(Parcelable):void
drawThumb(float,boolean,Canvas):void
evalPressedThumb(float):Thumb
isInThumbRange(float,double):boolean
setNormalizedMinValue(double):void
setNormalizedMaxValue(double):void
normalizedToValue(double)
valueToNormalized(T):double
normalizedToScreen(double):float
screenToNormalized(float):double
«Java Class»
LoadMap
com.hwu65.mobilehealthmonitor
LoadMap()
doInBackground(String[]):String
onPostExecute(String):void
«Java Class»
GeocodingFetcher
com.hwu65.googleservice
GeocodingFetcher()
getLocation(String):double[]
«Java Class»
PlacesFetcher
com.hwu65.googleservice
API_KEY: String
RETURN_TYPE: String
RETURN_RADIOUS: String
PlacesFetcher()
getMedicalPlaces(double,double):List<MedicalPlace>
-numberType
0..1
-listener
0..1
-pressedThumb
0..1
«Java Enumeration»
NumberType
com.hwu65.mobilehealthmonitor
LONG: NumberType
DOUBLE: NumberType
INTEGER: NumberType
«Java Interface»
OnRangeSeekBarChangeListener<T>
com.hwu65.mobilehealthmonitor
onRangeSeekBarValuesChanged(RangeSeekBar<?>,T,T):void
«Java Enumeration»
Thumb
com.hwu65.mobilehealthmonitor
MIN: Thumb
MAX: Thumb
Thumb()