**Application Class and Function Explanation**

1. Data Structure

One Direction consists of the Queue List of NavigationMarkers class.

1. LocQueue – Queue Structure
2. SearchMarkers – Now Location & Destination Marker
3. String addr
4. String lat
5. String lng
6. RankMarkers – Destination Marker for History
7. Extends SearchMarkers
8. Int callCount
9. NavigationMarkers – Step Marker
10. Boolean walkingToNext
11. String announce
12. String dist
13. String time
14. String to\_lat, to\_lng
15. String from\_lat, from\_lng
16. LocQueue<LatLng> smoothLoc
17. FastSearch – Data Structure for History
18. SearchMarkers myHome
19. SearchMarkers myWork
20. LocQueue<SearchMarkers> myRecent
21. List<RankMarkers> myRank
22. Google API

One Direction Search needs three API, Google Map API, Search API, and Directions API. This chapter deals with Search API and Directions API.

1. FindDest – Search API
2. Variable
3. String dest

Destination Name that users want to search

1. LatLng loc

Users’ Location (Help for searching near destination)

1. List<SearchMarkers> positions

Result of Destinations

1. ResponseHandler<String> responseSearchHandler

Global Variable for response of Search API

1. Function, Inner Class
2. **public** List<SearchMarkers> reqDest()

Call Search API and get the result after JSON parsing.

It use SearchDest Inner Class

1. **private** **class** SearchDest **extends** Thread (*Inner Class*)

For Search API, there is HTTP GET Connection

1. **private** String changePar(String bef)
2. **public** **void** run()
3. **public** URI makeParameter(LatLng loc)
4. **new** ResponseHandler<String>(){…} (*Variable that needs functions*)

For response from HTTP GET Connection

1. **public** String handleResponse(HttpResponse arg0)
2. **public** List<SearchMarkers> parseJSON(String json)
3. Navigation – Directions API
4. Variable
5. LatLng src\_loc
6. LatLng dest\_loc
7. HttpClient httpclient
8. List<LatLng> smoothMk
9. LocQueue<NavigationMarkers> lnm
10. Function, Inner Class
11. **public** LocQueue<NavigationMarkers> reqNavigation()
12. **public** List<LatLng> reqSmooth()
13. **public** **class** searchNavigation **extends** Thread (*Inner Class*)
14. **public** **void** run()
15. **public** URI makeParameter(LatLng src\_loc, LatLng dest\_loc)
16. **new** ResponseHandler<String>(){…} (*Variable that needs functions*)
17. **public** String handleResponse(HttpResponse arg0)
18. **public** LocQueue<NavigationMarkers> parseJSON(String json)
19. **private** LocQueue<LatLng> getSmoothLoc(String encoded)
20. **private** LocQueue<LatLng> decodePoly(String encoded)
21. **public** String mod(String anc)
22. Store and Load for History

History needs the External Storage in Smart devices. There are two functions about Load and Store in the Code.

1. HelpFastSearch – Load and Store, etc.
2. Function
3. **public** HelpFastSearch(String fileName) (*Constructor*)
4. **public** **void** testFS(FastSearch fs)
5. **public** **boolean** deleteFS()
6. **public** FastSearch loadFS()
7. **public** **void** saveFS(FastSearch fs)
8. FastSearch – Recent and Rank System
9. Variable
10. SearchMarkers myHome
11. SearchMarkers myWork
12. LocQueue<SearchMarkers> myRecent
13. List<RankMarkers> myRank
14. Function
15. **public** FastSearch() (*Constructor*)
16. **private** **void** writeObject(ObjectOutputStream stream)
17. **private** **void** readObject(ObjectInputStream stream)
18. **public** **void** searchExec(SearchMarkers sm)
19. **private** **void** setMyRank(SearchMarkers sm)
20. **private** **void** changeRank(**int** ind, **int** len, **int** cnt)
21. **private** **void** setMyRecent(SearchMarkers sm)
22. **private** **boolean** findSame(**int** len, SearchMarkers nowsm)
23. getter and setter of myHome and myWork
24. getter of myRecent and myRank
25. Activity & UI

The application has 2 screens and 1 Expandable List. And there is a class for considering error of the compass.

1. MainActivity – First Screen with Text for Searching Destination and History
2. Variable
3. TextView destText
4. ExpandableListView elv, ExpandableListAdapter listAdapter
5. List<String> listHeader
6. HashMap<String, List<String>> listChild
7. **private** **final** **int** NEWSEARCH
8. Function
9. **protected** **void** onCreate(Bundle savedInstanceState) (*@Override*)
10. **public boolean** onChildClick(ExpandableListView parent, View v, int groupPosition, int childPosition, long id) (*Child Click Listener of ExpandableListView*)
11. **public** **void** onClick(View v) (*Click Listener of TextView*)
12. **public** **void** setAllList()
13. ShowMap – Second Screen (Google Map, Markers, Lines, Announcement, etc.)
14. Variable
15. GoogleMap tempMap
16. LocationManager lm
17. LocationListener gpsll, netll

………….

1. Function
2. **protected** **void** onCreate(Bundle savedInstanceState) (*@Override*)
3. **public** **void** requestLocAndMap()
4. **public** **void** setGPS()
5. **protected** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data)
6. **public** **void** setTempMap()
7. **public** **void** requestLoc()
8. **public** **void** onLocationChanged(Location arg0) (*Called Function when location changed of LocationListener*)
9. **protected** **void** onResume() (*@Override*)
10. **protected** **void** onPause() (*@Override*)
11. **public** **void** realOnLoctionChanged(Location arg0, **boolean** isGPS)
12. **public** **void** setBearingSt(Double lat, Double lng)
13. **public void** setBearingSm(Double lat, Double lng, NavigationMarkers nm)
14. **public** SearchMarkers returnFastLoc(FastSearch fs, **int** group\_opt, **int** child\_opt)
15. **public** **void** applyLog(FastSearch fs, SearchMarkers dest)
16. **public** **void** findRoad(SearchMarkers dest)
17. **public** **void** showMarker(SearchMarkers sm) **……… Processing**
18. **public** **void** showMarkers(List<SearchMarkers> mk) **……… Processing**
19. **public** **void** showNMMarkers(LatLng src, LatLng dst, LocQueue<NavigationMarkers> mylocqueue)
20. **public** **void** compareLocation(Location now\_arg0, LocQueue<NavigationMarkers> mylocqueue)
21. **public** **boolean** checkNear(Location my, LatLng smooth, NavigationMarkers step)
22. **public** **float** returnDist(Location my, LatLng smooth, NavigationMarkers step)
23. **public** **void** speak(String tmp)
24. **public** **boolean** onTouch(View arg0, MotionEvent e)
25. **public** **void** onSensorChanged(SensorEvent arg0) (*@Override*)
26. **public** **void** onInit(**int** status) (*@Override*)
27. HelpCompass – In consideration of Error in Marker’s Location
28. Function
29. **public** HelpCompass(Double from\_lat, Double from\_lng, Double to\_lat, Double to\_lng) (*Constructor*)
30. **public** Location getBearingDest(Location my)