5.4 CODE SNIPPET

5.4.1MAP ACTIVITY

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_map);

stationBeanList = ((ApplicationContext) this.getApplicationContext()).getStationBeanList();

try {

MapsInitializer.initialize(this);

} catch (GooglePlayServicesNotAvailableException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

seekerMapFragment = (MapFragment) getFragmentManager()

.findFragmentById(R.id.mapView);

seekerMap = seekerMapFragment.getMap();

LatLng latlong = new LatLng(18.4, 73.8);

CameraPosition cameraPosition = new CameraPosition.Builder().target(latlong).zoom(6).build();

seekerMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));

5.4.2WebServiceParser

public ArrayList<AnnouncementBean> getAnnouncementInfo(Map<String, String> params) {

String response = wSCommunictor.invokeMethod("getAnnouncementInfo",params);

ArrayList<AnnouncementBean> announcementBeanList = new ArrayList<AnnouncementBean>();

try {

if (response != null) {

System.out.println(response);

Log.d("getAnnouncementInfo\_webServiceParser", response);

JSONArray jsonArray = new JSONArray(response);

for (int i = 0; i < jsonArray.length(); i++) {

AnnouncementBean announcementBean = null;

JSONObject jsonObject = new JSONObject(jsonArray.getString(i));

if (jsonObject != null) {

announcementBean = new AnnouncementBean(); announcementBean.setAnnouncementId(jsonObject.getInt("AnnouncementId"));

announcementBean.setTrainId(jsonObject.getInt("TrainId"));

announcementBean.setStationId(jsonObject.getInt("StationId"));

announcementBean.setAnnouncementName(jsonObject.getString("AnnouncementName"));

announcementBean.setAnnouncementDescription(jsonObject.getString("AnnouncementDescription")); announcementBean.setAnnouncementDate(jsonObject.getString("AnnouncementDate")); announcementBeanList.add(announcementBean);

}

}

}

} catch (Exception e) {

e.printStackTrace();

5.4.3 DBConnection

public Connection connect() {

{

try {

Class.forName(dbDriver);

connection = DriverManager.getConnection(dbConnectionString,

dbUsername, dbPassword);

if (connection == null) {

System.out.println("connection is not established");

}

return connection;

} catch (ClassNotFoundException e) {

System.out.println(e);

} catch (SQLException e) {

System.out.println(e);

}

}

return null;

}

5.4.4 UserBean

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getUserAddress() {

return userAddress;

}

public void setUserAddress(String userAddress) {

this.userAddress = userAddress;

}

public String getUserPhoneNumber() {

return userPhoneNumber;

}

public void setUserPhoneNumber(String userPhoneNumber) {

this.userPhoneNumber = userPhoneNumber;

}

public String getUserMailId() {

return userMailId;

}

public void setUserMailId(String userMailId) {

this.userMailId = userMailId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}