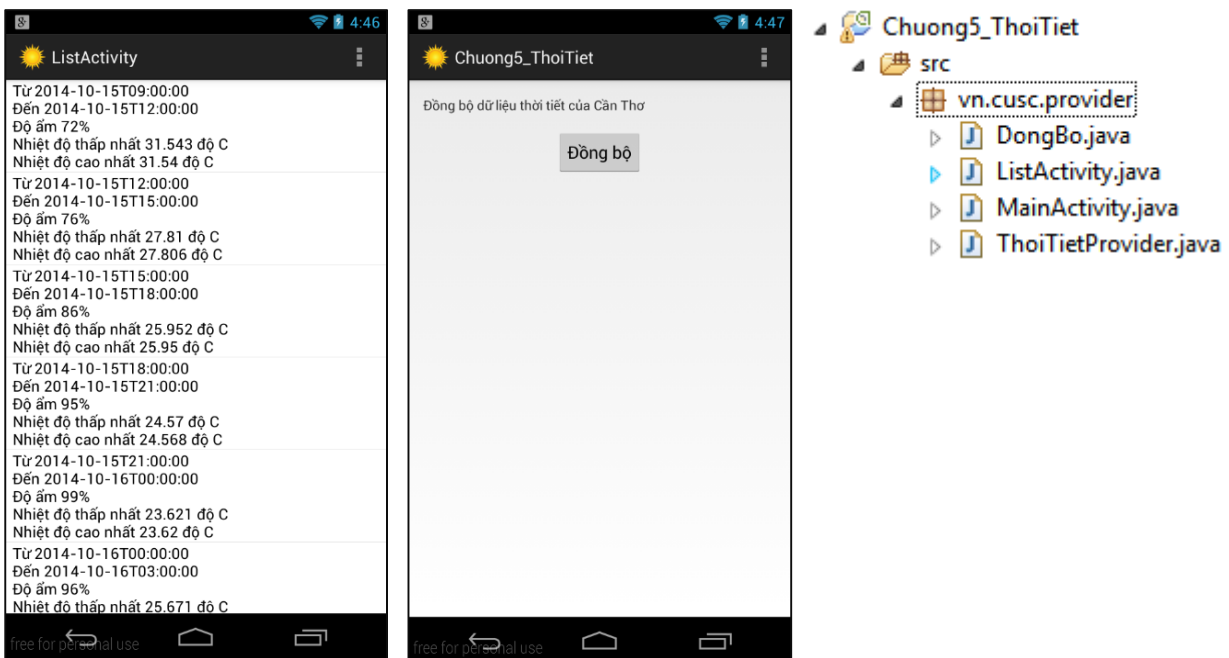


Content Provider

Yêu cầu

1. Tạo Content Provider chia sẻ dữ liệu thời tiết bên dưới
2. Thử các chức năng tìm, sửa, xóa, thêm mới
3. Đồng bộ dữ liệu thời tiết



Giao diện demo

Bước 1. Tạo dữ liệu SQLite

```
public class ThoiTiet extends SQLiteOpenHelper {

    public static final String ID ="id";
    public static final String DIA_DIEM ="diadiem";
    public static final String QUOC_GIA ="quocgia";
    public static final String LATITUDE ="latitude";
    public static final String LONGITUDE ="longitude";

    public static final String TU_GIO ="tugio";
    public static final String NHIET_DO_CAO_NHAT ="nhietdocaonhat";
    public static final String NHIET_DO_THAP_NHAT ="nhietdothapnhat";
    public static final String DO_AM ="doam";
    public static final String MAY ="may";
    public static final String DEN_GIO ="dengio";
```

```

public static final String TEN_BANG = "thoitiet";
public static final String TEN_CSDL = "thoitietdb";
public static final int PHIEN_BAN = 1;

private static final String TAO_BANG = "
    + "CREATE TABLE " + TEN_BANG + "("
    + ID + " integer primary key autoincrement, "
    + DIA_DIEM + " text, "
    + QUOC_GIA + " text, "
    + LATITUDE + " real, "
    + LONGITUDE + " real, "
    + TU_GIO + " integer, "
    + DEN_GIO + " integer, "
    + NHIET_DO_CAO_NHAT + " real, "
    + NHIET_DO_THAP_NHAT + " real, "
    + DO_AM + " real, "
    + MAY + " text)";

public ThoiTiet(Context context) {
    super(context, TEN_CSDL, null, PHIEN_BAN);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(TAO_BANG);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP IF EXISTS " + TEN_BANG);
    onCreate(db);
}
}

```

Bước 2. Tạo ContentProvider để chia sẻ dữ liệu thời tiết

```

public class ThoiTietProvider extends ContentProvider {
    SQLiteDatabase duLieuthoitiet;
    public static final String URI = "content://vn.cusc.provider/thoitiet";
    public static final String PROVIDER = "vn.cusc.provider";
    // Uri dữ liệu
    private static final int TATCA = 1;
    private static final int MOT = 2;
    // qui định URI
    static final UriMatcher taoURI;
    static {
        taoURI = new UriMatcher(UriMatcher.NO_MATCH);
        taoURI.addURI(PROVIDER, "thoitiet", TATCA);
        taoURI.addURI(PROVIDER, "thoitiet/#", MOT);
    }
    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        switch (taoURI.match(uri)) {
            case TATCA:

```

```

        break;
    case MOT:
        String id = uri.getPathSegments().get(1);
        selection = ThoiTiet.ID+ "=" + id + (!TextUtils.isEmpty(selection) ?
" AND (" + selection + ') ' : "");
        break;
    default:
        throw new IllegalArgumentException("Không hỗ trợ URI: " + uri);
    }
    int deleteCount = dulieuthoitiet.delete(ThoiTiet.TEN_BANG, selection,
selectionArgs);
    getContext().getContentResolver().notifyChange(uri, null);
    return deleteCount;
}
@Override
public String getType(Uri uri) {
    switch (taoURI.match(uri)) {
    case TATCA:
        // tra ve danh sach
        return "vnd.android.cursor.dir/" + PROVIDER + "/thoitiet";
    case MOT:
        // tra ve 1 phan tu
        return "vnd.android.cursor.item/" + PROVIDER + "/thoitiet";
    default:
        throw new IllegalArgumentException("Không hỗ trợ URI: " + uri);
    }
}
@Override
public Uri insert(Uri uri, ContentValues values) {
    if (taoURI.match(uri) == TATCA) {
        long id = dulieuthoitiet.insert(ThoiTiet.TEN_BANG, null, values);
        getContext().getContentResolver().notifyChange(uri, null);
        return Uri.parse(URI + "/" + id);
    } else {
        throw new IllegalArgumentException("Không hỗ trợ URI: " + uri);
    }
}
@Override
public boolean onCreate() {
    ThoiTiet db = new ThoiTiet(getContext());
    dulieuthoitiet = db.getWritableDatabase();
    return dulieuthoitiet != null ? true : false;
}
@Override
public Cursor query(Uri uri, String[] projection, String selection,
String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();
    queryBuilder.setTables(ThoiTiet.TEN_BANG);

    switch (taoURI.match(uri)) {
    case TATCA:
        break;
    case MOT:
        String id = uri.getPathSegments().get(1);

```

```

        queryBuilder.appendWhere(ThoiTiet.ID + "=" + id);
        break;
    default:
        throw new IllegalArgumentException("Không hỗ trợ URI: " + uri);
    }

    Cursor cursor = queryBuilder.query(dulieuthoitiet, projection,
        selection, selectionArgs, null, null, sortOrder);
    return cursor;
}

@Override
public int update(Uri uri, ContentValues values, String selection,
    String[] selectionArgs) {
    switch (taoURI.match(uri)) {
        case TATCA:
            // do nothing
            break;
        case MOT:
            String id = uri.getPathSegments().get(1);
            selection = ThoiTiet.ID + "=" + id +
                (!TextUtils.isEmpty(selection) ? " AND (" + selection + ')' : "");
            break;
        default:
            throw new IllegalArgumentException("Không hỗ trợ URI: " + uri);
    }
    int updateCount = dulieuthoitiet.update(ThoiTiet.TEN_BANG, values,
        selection, selectionArgs);
    getContext().getContentResolver().notifyChange(uri, null);
    return updateCount;
}
}

```

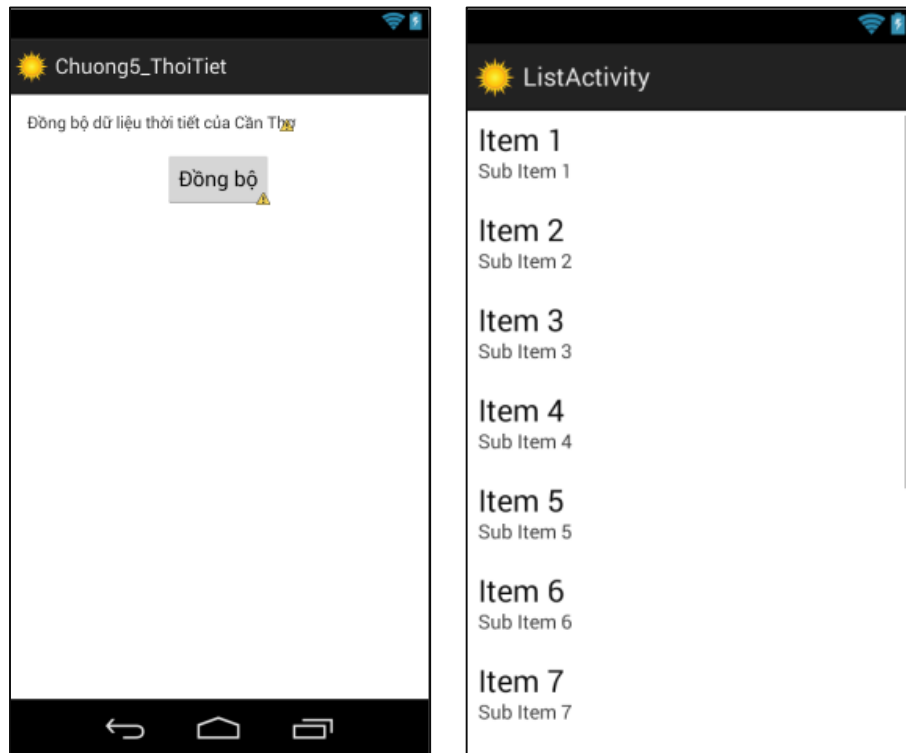
Bước 3. Đăng ký Provider

Bổ sung mã đăng ký sau vào file AndroidManifest.xml

```

<provider
    android:name=".ThoiTietProvider"
    android:authorities="vn.cusc.provider"
    android:exported="true" />

```

Bước 4. Thiết kế giao diện cho MainActivity và ListActivity như sau

Giao diện ứng dụng

Bước 5. Đưa dữ liệu SQLite thời tiết lên giao diện

Hiệu chỉnh lại nội dung của ListActivity như sau

```
public class ListActivity extends ActionBarActivity {
    ListView lst;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);

        Cursor c = getContentResolver().query(Uri.parse(ThoiTietProvider.URI),
null, null, null, null);
        String[] ds = new String[c.getCount()];
        int pos = 0;
        c.moveToFirst();
        do {
            ds[pos] = "Từ "
                + c.getString(c.getColumnIndex(ThoiTiet.TU_GIO))
                + "\nĐến "
                + c.getString(c.getColumnIndex(ThoiTiet.DEN_GIO))
                + "\nĐộ ẩm "
                + c.getString(c.getColumnIndex(ThoiTiet.DO_AM))
                + "%\nNhiệt độ thấp nhất "
                + c.getString(c.getColumnIndex(ThoiTiet.NHIET_DO_THAP_NHAT))
                + " độ C\nNhiệt độ cao nhất "
        } while (c.moveToNext());
    }
}
```

```

        + c.getString(c.getColumnIndex(ThoiTiet.NHIET_DO_CAO_NHAT))
        + " độ C";
        pos++;
    } while (c.moveToNext());
    c.close();
    lst = (ListView) findViewById(R.id.listView1);
    ArrayAdapter<String> adapter = new
        ArrayAdapter<String>(getApplicationContext(),
            android.R.layout.simple_list_item_1, ds) {
        @Override
        public View getView(int position, View convertView,
            ViewGroup parent) {
            View view = super.getView(position, convertView, parent);
            TextView textView = (TextView) view.findViewById(android.R.id.text1);
            textView.setTextSize(16);
            textView.setTextColor(Color.BLACK);
            return view;
        }
    };
    lst.setAdapter(adapter);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.list, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        Intent i = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(i);
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Chạy ứng dụng để quan sát kết quả.

Bước 6. Đồng bộ dữ liệu thời tiết từ internet về cơ sở dữ liệu SQLite

Tạo lớp có tên “**DongBo**” có nội dung như sau

```

public class DongBo {
    public DongBo() {
        try {
            try {
                URL url = new
URL("http://api.openweathermap.org/data/2.5/forecast?q=can+tho,vn&mode=xml");
                URLConnection conn = url.openConnection();
                InputStream input = conn.getInputStream();
            }
        }
    }
}

```

```

        BufferedReader read = new BufferedReader(new
InputStreamReader(input));
        File file = new
File(Environment.getExternalStorageDirectory().getAbsolutePath()+"/cantho.xml");
        FileWriter w = new FileWriter(file);
        String line = "";
        while ((line = read.readLine()) != null) {
            w.write(line);
        }
        w.flush();
        w.close();
        SAXParserFactory fac = SAXParserFactory.newInstance();
        SAXParser sax = fac.newSAXParser();
        handler h = new handler();
        sax.parse(new FileInputStream(file), h);
    } catch (Exception ex) {
        Log.d("Noi dung", ex.toString());
    }

    } catch (Exception ex) {
        Log.d("Noi dung loi: ", ex.toString());
    }
}

public ArrayList<ContentValues> list = new ArrayList<ContentValues>();

class handler extends DefaultHandler {

    boolean time = false;
    boolean temperature = false;

    @Override
    public void characters(char[] ch, int start, int length)
        throws SAXException {
        super.characters(ch, start, length);
    }

    @Override
    public void endDocument() throws SAXException {
        super.endDocument();
    }

    @Override
    public void endElement(String uri, String localName, String qName)
        throws SAXException {
        super.endElement(uri, localName, qName);
    }

    @Override
    public void startDocument() throws SAXException {
        super.startDocument();
    }

    ContentValues values;

```

```

@Override
public void startElement(String uri, String localName, String qName,
    Attributes attributes) throws SAXException {
    super.startElement(uri, localName, qName, attributes);
    if (qName.equalsIgnoreCase("time")) {
        values = new ContentValues();
        values.put(ThoiTietProvider.ThoiTiet.DIA_DIEM, "Can Tho");
        values.put(ThoiTietProvider.ThoiTiet.QUOC_GIA, "VN");
        values.put(ThoiTietProvider.ThoiTiet.LONGITUDE, 0);
        values.put(ThoiTietProvider.ThoiTiet.LATITUDE, 0);
        values.put(ThoiTietProvider.ThoiTiet.TU_GIO,
attributes.getValue(0));
        values.put(ThoiTietProvider.ThoiTiet.DEN_GIO,
attributes.getValue(1));
        Log.d("Nội dung", "Từ "
            + attributes.getValue(0).split("T")[1] + " ngày "
            + attributes.getValue(0).split("T")[0]
            + " đến "
            + attributes.getValue(1).split("T")[1] + " ngày "
            + attributes.getValue(1).split("T")[0]);
    }

    if (qName.equalsIgnoreCase("temperature")) {
        values.put(ThoiTietProvider.ThoiTiet.NHIET_DO_CAO_NHAT,
attributes.getValue(2));
        values.put(ThoiTietProvider.ThoiTiet.NHIET_DO_THAP_NHAT,
attributes.getValue(3));
        Log.d("Nội dung", "thấp nhất = " + attributes.getValue(2) + "
cao nhất = " + attributes.getValue(3));
    }

    if (qName.equalsIgnoreCase("humidity")) {
        values.put(ThoiTietProvider.ThoiTiet.DO_AM,
attributes.getValue(0));
        Log.d("Nội dung", "Độ ẩm = " + attributes.getValue(0) +
attributes.getValue(1));
    }

    if (qName.equalsIgnoreCase("clouds")) {
        Log.d("Nội dung", "Mây = " + attributes.getValue(0));
        values.put(ThoiTietProvider.ThoiTiet.MAY,
attributes.getValue(0));
        list.add(values);
    }
}
}
}

```

Hiệu chỉnh lại MainActivity như sau

```

public class MainActivity extends ActionBarActivity {
    DongBo load;
    Button btnDongBo;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnDongBo = (Button)findViewById(R.id.btnDongBo);
        btnDongBo.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                new abc().execute();
            }
        });
    }

    class abc extends AsyncTask{
        @Override
        protected Void doInBackground(Void... params) {
            load = new DongBo();
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            super.onPostExecute(result);
            File file = new
File(Environment.getExternalStorageDirectory().getAbsolutePath()+"/cantho.xml");
            file.delete();
            getContentResolver().delete(Uri.parse(ThoiTietProvider.URI), null, null);

            for (int i = 0; i < load.list.size(); i++) {
                getContentResolver().insert(Uri.parse(ThoiTietProvider.URI),
load.list.get(i));
            }
            Cursor c = getContentResolver().query(Uri.parse(ThoiTietProvider.URI),
null, null, null, null);
            Toast.makeText(getApplicationContext(), (c == null ? "Không có dữ liệu" :
"Đã có " + c.getCount()+" thông tin"), Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

Do có truy xuất ra ngoài Internet và đọc/ghi SDCard nên cần phải cấp các quyền sau

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Dữ liệu thời tiết tham khảo

<http://api.openweathermap.org/data/2.5/forecast?q=can+tho,vn&mode=xml>

```
- <location>
  <name>Can Tho</name>
  <type/>
  <country>VN</country>
  <timezone/>
  <location altitude="0" latitude="10.03333" longitude="105.783333" geobase="geonames" geobaseid="0"/>
</location>
<credit/>
- <meta>
  <lastupdate/>
  <calctime>0.363</calctime>
  <nextupdate/>
</meta>
<sun rise="2014-10-14T22:45:17" set="2014-10-15T10:39:55"/>
- <forecast>
  - <time from="2014-10-15T00:00:00" to="2014-10-15T03:00:00">
    <symbol number="501" name="moderate rain" var="10d"/>
    <precipitation value="4" unit="3h" type="rain"/>
    <windDirection deg="285.5" code="WNW" name="West-northwest"/>
    <windSpeed mps="3.11" name="Light breeze"/>
    <temperature unit="celsius" value="29.87" min="23.941" max="29.87"/>
    <pressure unit="hPa" value="1025.02"/>
    <humidity value="100" unit=""/>
    <clouds value="scattered clouds" all="48" unit=""/>
  </time>
```