Programmazione di sistemi mobile e tablet

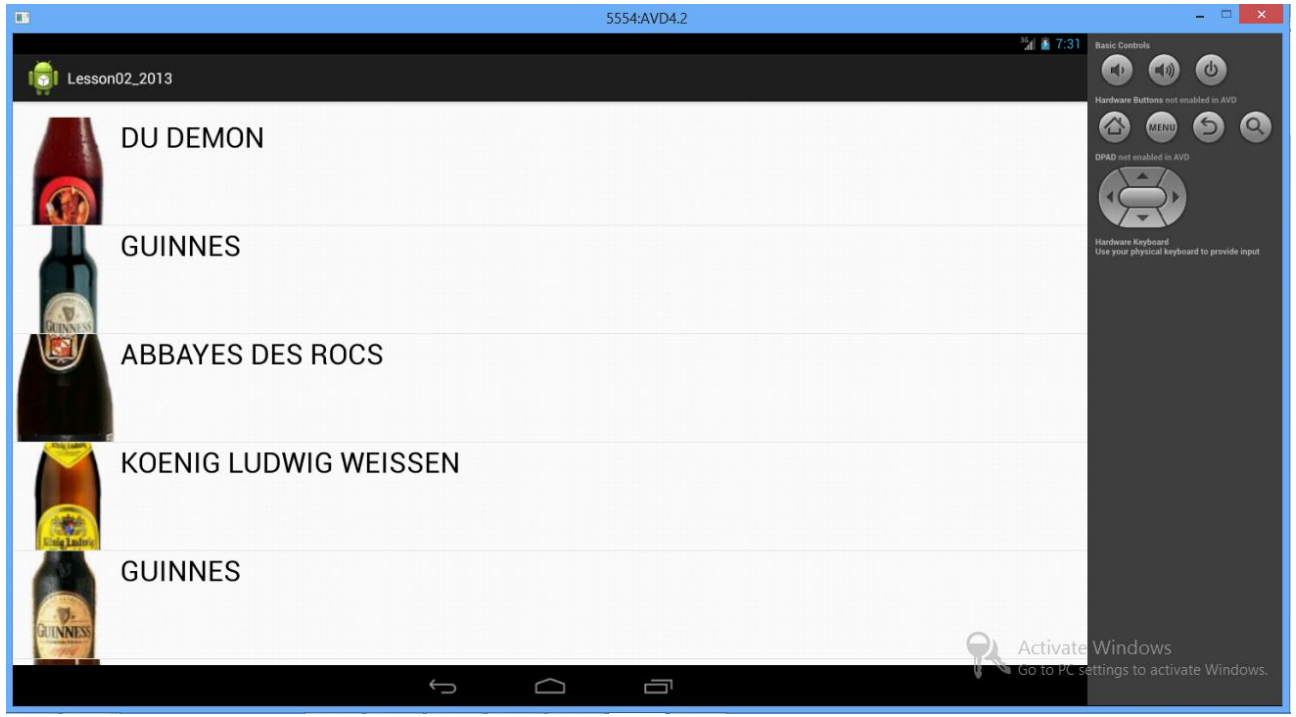
Android development

Carlo Menapace

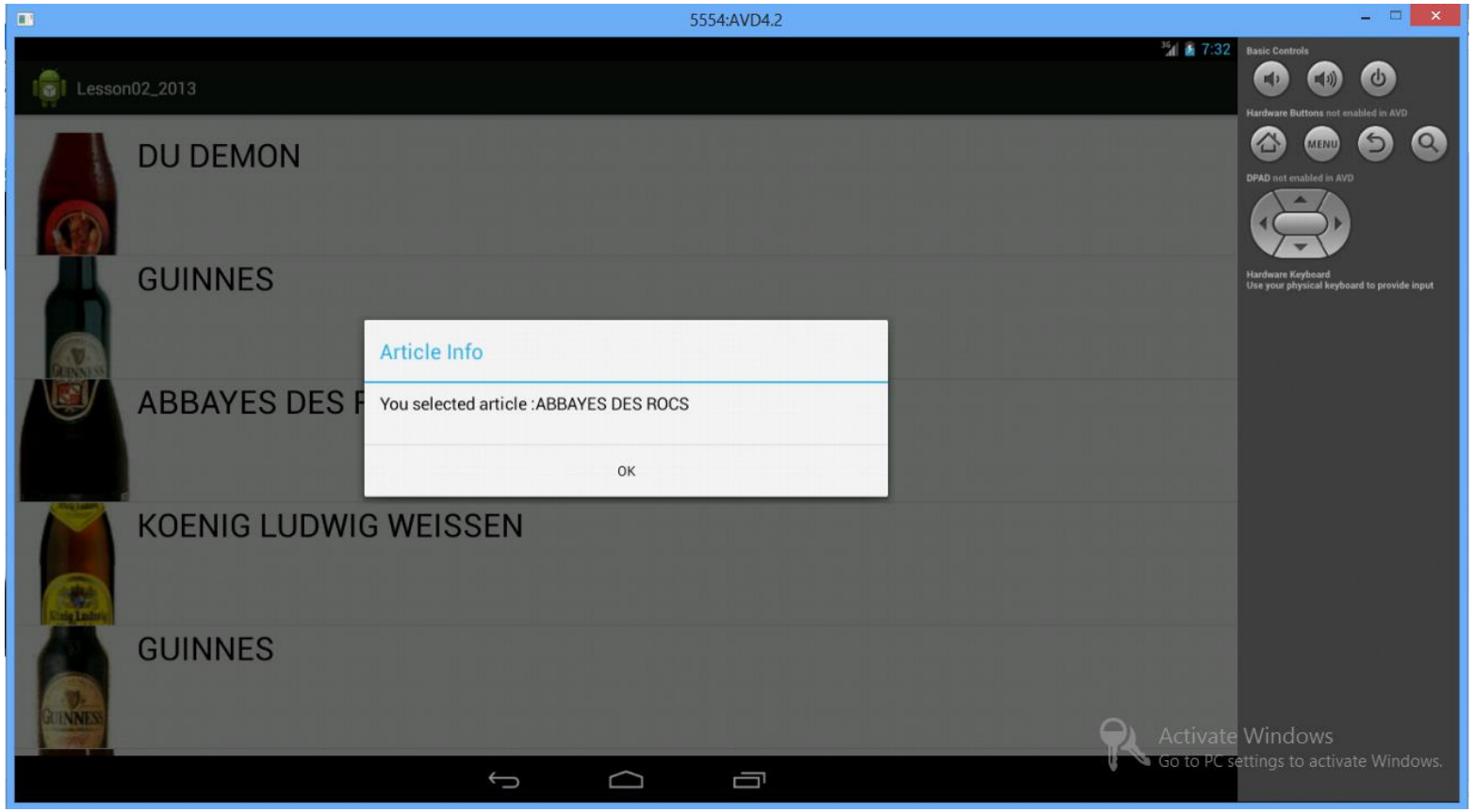
carlo87@gmail.com

TODAY'S ROADMAP

Using the last lesson exercise, let's start using a database as a data storage.
We want to retrieve and store data in a local database.
The following picture shows the expected result for today's exercise.



ALERT DIALOG



Clicking on a row, we want to view some data in a Alert dialog!

Sample code

```
public class DBHelper extends SQLiteOpenHelper {

    public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    private static final String ARTICLE_TABLE_CREATE = "CREATE TABLE IF NOT EXISTS "
            + DBMetadata.ARTICLE_TABLE
            + " ("
            + DBMetadata.ARTICLE_ID + " integer primary key autoincrement,"
            + DBMetadata.ARTICLE_NAME + " text not null, "
            + DBMetadata.ARTICLE_DESCRIPTION + " text not null, "
            + DBMetadata.ARTICLE_IMAGE + " blob not null"
            + ");";

    @Override
    public void onCreate(SQLiteDatabase _db) { _db.execSQL(ARTICLE_TABLE_CREATE); }

    @Override
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
        throw new UnsupportedOperationException("You have to implement this in order to upgrade database");

    }
}
```

Just some lines of code, not all the source code!

```
public class DBOperations {

    private SQLiteDatabase mDb;
    private DBHelper dbHelper;
    private static final String DB_NAME = "lesson03DB.db";
    private static final int DB_VERSION = 1;

    private static DBOperations instance = null;

    private DBOperations(Context ctx) { dbHelper = new DBHelper(ctx, DB_NAME, factory: null, DB_VERSION); }

    public static DBOperations getInstance(Context ctx) {
        if (instance == null) {
            instance = new DBOperations(ctx);
        }
        return instance;
    }

    public void eraseDB() { mDb.execSQL("DROP TABLE IF EXISTS " + DBMetadata.ARTICLE_TABLE); }

    public void populateDB(Context ctx) {
        // Insert articles
        insertArticles(ctx);
    }

    private void insertArticle(Article article) {
        ContentValues cv = new ContentValues();
        cv.put(DBMetadata.ARTICLE_NAME, article.getArticleName());
        cv.put(DBMetadata.ARTICLE_DESCRIPTION, article.getArticleDescription());
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        article.getImage().compress(Bitmap.CompressFormat.PNG, quality: 100, baos);
        byte[] imageBytes = baos.toByteArray();
        cv.put(DBMetadata.ARTICLE_IMAGE, imageBytes);

        mDb.insert(DBMetadata.ARTICLE_TABLE, nullColumnHack: null, cv);
    }

    public ArrayList<Article> fetchArticle() {
        ArrayList<Article> res = new ArrayList<~>();

        Cursor c = mDb.query(DBMetadata.ARTICLE_TABLE, columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
        int idCol = c.getColumnIndex(DBMetadata.ARTICLE_ID);
        int nameCol = c.getColumnIndex(DBMetadata.ARTICLE_NAME);
        int descCol = c.getColumnIndex(DBMetadata.ARTICLE_DESCRIPTION);
        int imageCol = c.getColumnIndex(DBMetadata.ARTICLE_IMAGE);

        if (c.moveToFirst()) {
            do {
                Article article = new Article();
                article.setId(c.getInt(idCol));
```

EXTRA SLIDE

If you want something more exciting, try to create a contextual Menu and open a Dialog by clicking on one of its button!
Feel free to use your favorite MENU

- Ready to use
- Customized by you (for example by inflating your **layout**)

Note

In order to create a menu layout, you have to put your xml layout file under the res menu folder.