CMSC436: Fall 2013 – Week 14 Lab

Objectives:

Familiarize yourself with Android ContentProviders. Create an application that reads information from a ContentProvider.

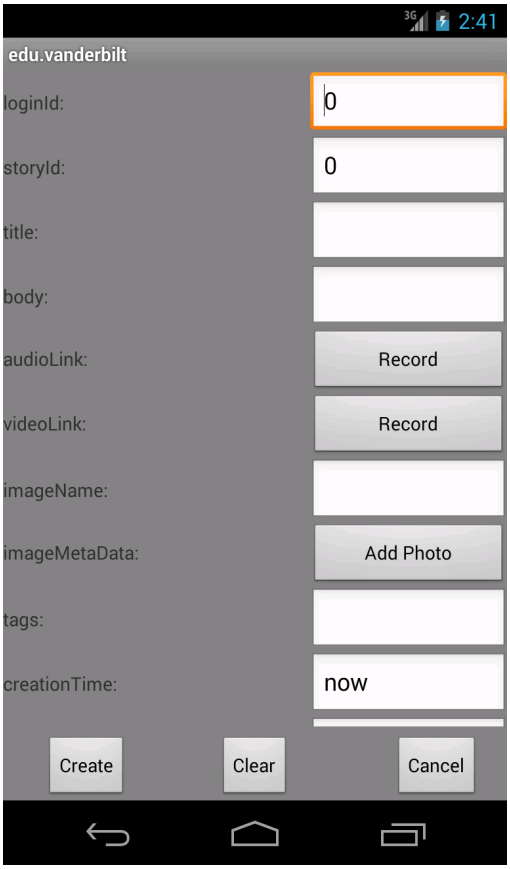
Once you've completed this lab you should have a better understanding of how to access data managed by a ContentProvider.

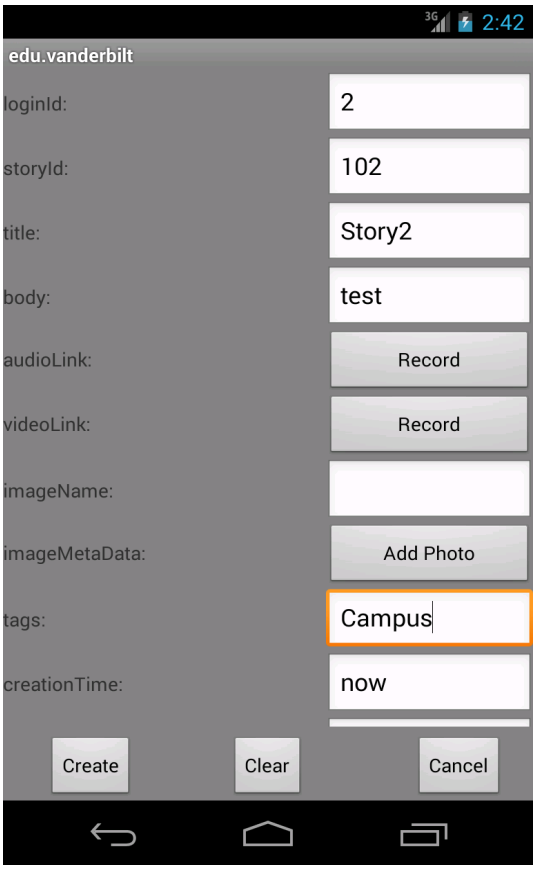
Overview:

This lab has one part.

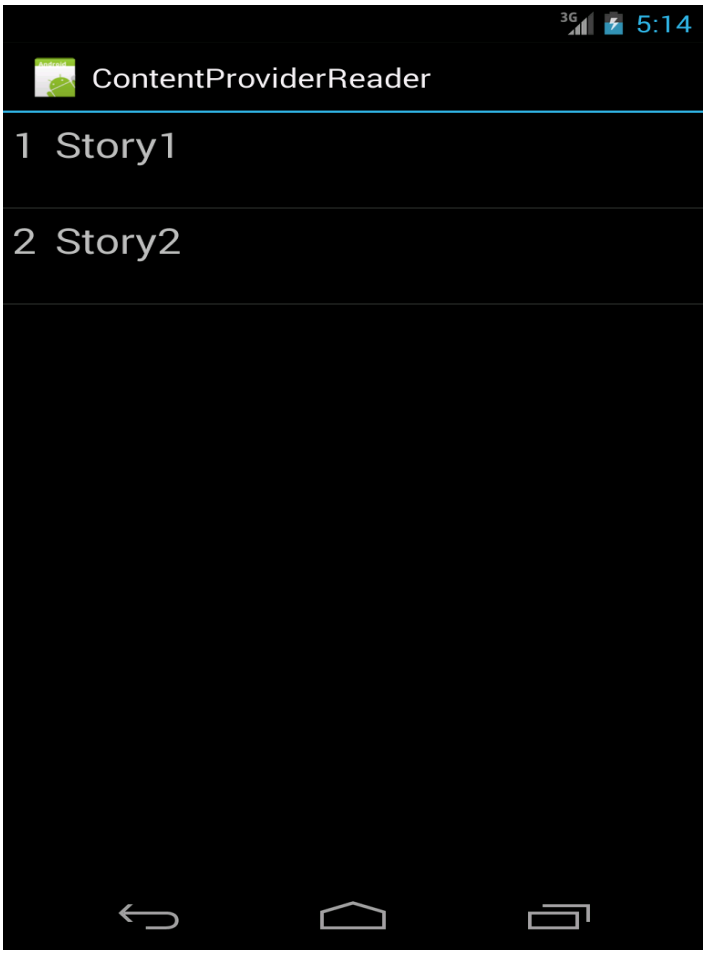
Part 1: ContentProvider

The ContentProviderExample application records information, called stories, associated with a particular location. When users enter a new story, the application stores it in a ContentProvider. The user interface for this application is very simple. Users enter parts of a story, such as a story ID, a title, some tags, and more, as shown below. **Note**: some buttons may not work on the emulator, so focus on the text entry fields for now.





As a user enters new stories, the application stores them in a ContentProvider. This data is accessible to other applications, for example, a story reader application. In this application, you will complete a separate app called ContentProviderReader, that can read story information for the ContentProvider used by the ContentProviderExample application, described above. To simplify this assignment, the ContentProviderReader, will only retrieve the IDs and Titles, associated with stories. After acquiring the data, the application will display them in a list view.



Implementation Notes:

1. Download the application skeleton files from the Lectures & Labs web page and import them into your IDE. The ContentProviderExample project is a complete application[1]. You do not need to do anything with it other than running it and using it to create story data. The ContentProviderReader project is the skeleton you need to complete.

[1] To capture video and images in the emulator, enable the Webcam() option in the AVD manager. (Devices -> [Device Name] -> Edit -> Front/Rear facing camera).

2. In class StoryInfoListAdapter.java:

    a. Implement public View newView(Context context, Cursor cursor, ViewGroup parent) to inflate the list_view every time a new view is needed.

    b. Implement public void bindView(View view, Context context, Cursor cursor) to display a story's ID and Title on corresponding TextViews “id” and “title”.

3. In class StoryListExample.java

    a. Declare the column data you want to retrieve from the ContentProvider. In this case, we only need 2 columns: ID and Title.

    String[] STORY_COLUMNS = new String[] { };

    b. Implement public Loader<Cursor> onCreateLoader(int id, Bundle args) to return a new CursorLoader when a new loader is created.

    c. Implement public void onLoadFinished(Loader<Cursor> loader, Cursor data) to swap the new cursor into the list adapter. This method is called when the loader has finished loading its data

    d. Implement public void onLoaderReset(Loader<Cursor> loader) to set list adapter's cursor to null. This method is called when the last cursor provided to onLoadFinished() is about to be closed.

Deliverables: Your source code project