Android Final Project COMP 10073

Overview

Design and implement a fully functional Android Application. The focus for the application should be the WebAPI materials introduced in the last part of the course. Most projects provide an application that includes several elements:

- Configuration
- User Level Help
- Web access to some service
- User refinement of search / detailed data access / data manipulation

You can diverge from the above general outline, however, if you want to do something that we haven't talked about in the course (an interactive game for example) please e-mail me to discuss by March 21st. I may wish to set up a meeting and ask you about it.

Project Proposal

A short 2 page project propsal should be submitted by March 24th explaining what your project will do. The proposal should look something like what you might see on the Android Google Play Store. It should include an overview of functionality and some images showing your idea for the layout. The proposal will be graded as pass / fail and will represent 10% of the project grade. Your proposal should include 3 sections:

1) Overview of your apps purpose and function (about 250 words). Written descriptions should generally be in the third person. Avoid a first person narrative style ("I"-will-do-this or "My"-project-is). Sell the proposal. For example: *"Proposal for the Mohawk Course Reviewer App (MCRA). MCRA provides a novel way for students to access the Mohawk calendar. It sorts, organizes, presents, and categorizes courses while including an option for student reviews, course ratings, and snapshots of the campus and class rooms. MCRA is your helper on campus. This 'positive-review-only' app automatically replaces rude comments with the word *GREAT* to keep everything friendly. Benefit from the experience of senior students, plan your studies while having fun, ..."*

Your overview should answer basic questions like: "Who cares about this?" and "Why is this application interesting, useful or worthwhile?"

2) API breadth requirement plan. See evaluation section of document.

3) Two activity layouts:

- ○ Several screen shots of Android Studio's layout tool showing widget placement

  or

- ○ Images made with a professional Android design tool, see for example:
  - https://www.mockplus.com/blog/post/android-mockup

Suggested Topics

Your project proposal should be based on one of these topics, or represent a variation on the theme of a tool that retrieves data from a web service, organizes it, stores it and presents it to the user. Your proposal should flesh out the details. Don't just copy and paste one of these items. Explain with diagrams, text, screen shots of Android Studio interface layouts, or other means what your final project will look like.

Mohawk Course Browser

- Retrieve Program Names and Courses from a cloud-based web service.
- Store retrieved information in a local SQLLite database. Allow "refreshing" data.
- Allow users to select a program, and select a semester.
- Show a scrolling list of Courses (Code, Title) available in that semester.
- When the user selects a course, show the course description.

Science Fair Project Browser

- Retrieve BASEF project details and pictures from a cloud-based web service.
- Allow users to select a year, and optionally a student name, level, school, school board, or project title search string.
- Display a list of projects by title that match the search.
- When a project is selected from the list, display a detail screen that includes picture, project title, project description, level, school, school board, etc.

Open Movie Database Browser

- Retrieve Movie information and thumbnails from a cloud-base web service.
- Users specify a partial movie title and a year of release to search the database.
- Display a list of movies that match the search; handle “too many results” error.
- When a movie is selected from the list, query the API using the specific ID/Title and retrieve detailed information including cast and plot. Display this in a details screen that includes the movie thumbnail image.

Books API Browser

- Retrieve Book information from a Google web API.
- Allow the user to enter a title search and/or an author name.
- Retrieve a list of books matching the search query.
- Allow the user to select a book from the list, and then show a details screen that includes the title, publisher, date of publication, description, and shows a cover thumbnail image.

Novel Design using a Web Based API Service of your Choice

- Should maintain the essential theme of accessing a Web API and storing / presenting information.
- Subject to instructor approval, plan must be submitted on time.
- Can be similar to one of the above, or an entirely novel variation.

General Project Requirements

- Create a new Android Application. Select **API 23 (Android 6.0)** as the Minimum SDK. Use a domain naming pattern of **ca.mohawkcollege.yourlastname** when you create your project. You must also include the following statement of authorship in a comment at the top of your MainActivity java code:
  I, *John Doe, 000123456* certify that this material is my original work. No other person's work has been used without due acknowledgement.
  (Replace with your own name and student number).

- The Application should be named "Project". The displayed name ("**app_name**") should be changed to the following, with your name and student number:

  Project - Your Name 000123456

Marking (out of 100)

1. Project planning document. Due Sun March 24th 10
   - Pass = reasonable plan
   - Fail = submission is missing any of
     - at least 2 activity layout plans
     - breadth requirement planning
     - 250 word overview
2. Stability, Error Checking, Exception Handling 10
   - Should build okay
   - Should not crash, but handle exceptions
   - Should include debugging logs for developers
3. Coding Style / Documentation 10
   - JavaDoc should be used in class and method headers
   - Minimal inline documentation
   - Code is appropriately distributed across classes
   - Methods use reasonable dimensions (< 50 lines < 80 cols)
   - No dead code, avoids obvious cut-and-paste duplication
   - Good use of constants, and Android resources
4. Appropriate use of Activities, Fragments, Lists and Layouts 10
   - Some Activities are subdivided into fragments
   - Activities have meaningful functionality
5. Functionality judged by Project Proposal and Breadth Requirement 25
   - Does not need to exactly follow proposal, should be similar
   - Breadth requirement is spelled out on the last page
6. Video demo (about 2-4 mins) 15
   - Shows all aspect of running application
   - explains any special code / design techniques
7. Visual Quality (clean layout, good visual elements) 10
   - App looks nice, colors balanced, custom themes used
   - Good use of images, icons, borders, and hints
8. Usability, Novelty, interesting design choices 10
   - App should make sense, be something someone would use
   - API elements well integrated, not just there to satisfy breath req.
   - Use of Android features not covered in the course, good
   - Interesting Web APIs beyond suggested ones, good

Breadth and API Requirements

All projects must include several of the Android widgets taught in class. You can expand this list if you want to, but consider the following as a minimum :

- Use a web based service to access content
  - Must discuss with me well in advance if you don’t want to do a Web API

- Contain all the following components from Labs 1-4:
  - At least 2 separate activities
  - One activity must utilize multiple fragments, or custom layouts
  - Several widget types (at least 4)
    - Output: TextView, ImageView
    - Input: Buttons, switches, spinner, edit text
    - Other: your choice

- Contain at least 4 of the following features presented in Weeks 6-13:
  - SQLite Database (required for Mohawk Course Browser)
  - Downloading image from web
  - SMS Messaging
  - GPS location
  - Navigation Drawer
  - Dialog Fragment or Custom Toast
  - ListView
  - Contacts

Submission

March 24th - Two page project proposal due. Must use PDF.

April 6th - Export Android Application project code to a .zip file. Include a link to your video demo in a comment in the main activity.

If you are unable to meet either of these deadlines you must communicate with me in advance. In order to have the final due date modified you must provide some sort of documentation explaining your circumstances, a doctors note for example. Also be aware that near the end of the course scheduling flexibility is quite limited. Please do your best to get started early and complete the work on time.