Android Fundamentals Project Description

CaeSer

Problem:
The University of Pretoria's Computer Science department runs their entire system separately from the university's official clickUp system, where lecturers can post announcements. practical specifications, assignments. etc. for each module the student is registered for.
This forces students who are enrolled for modules in the Computer Science department to check the website regularly for announcements and assignments for fear of missing important information. There are countless students who have missed crucial information regarding announcement requirement changes, lecture venue or time changes, etc. and paid dearly for it.

Solution:
Create an app that will send the user notifications as soon as something is posted on the CS site, so that they won't miss anything important.
Currently the CS site hosts 30+ modules per semester, so to avoid providing and downloading unnecessary amounts of data, the user will be able to select modules he/she wish to be updated on.

Features:
-Notifications on:
- announcements
- assignments
- discussion posts
- module content

-User can subscribe to modules
-Download module content
-Add prac and assignment due dates to calendar

Description:
An app that will allow the students of the University of Pretoria stay up to date on any changes made to the Computer Science website, focussing specifically on updates on announcements. Additional functionality will follow in future additions, such as: updates on news, contacting the lecturer, updates on new active assignments, updates on discussions, logging in to the website, viewing submissions, marks, uploading assignments, etc.

Android Fundamentals Project Self-Evaluation

Instructions: Once you've completed your Final Project, please respond to the questions below. This is a chance for you to briefly explain to the grader your thought-process during development. Once you are done, include this with the source code and accompanying files you are submitting. Then, give yourself a pat on the back for making a great app!

Questions about Required Components

Permissions

Please elaborate on why you chose the permissions in your app.

I chose to use the INTERNET: as I'm using HTTP request to retrieve content from cs.up.ac.za
and WRITE_EXTERNAL_STORAGE: as I store the content downloaded by the user through the download manager at the default location of the device.

Content Provider

What is the name of your Content Provider, and how is it backed? (For example, Sunshine's Content Provider is named `WeatherProvider` backed by an SQLite database, with two tables: `weather` and `location`.)

DateProvider is the name of CaeSr's content provider.
It is backed by an SQLite database with 5 tables: modules, announcements, assignments, discussions and content

What backend does it talk to? (For example, Sunshine talks to the OpenWeatherMap API.)

I wrote my own utility that retrieves data from my University's Computer Science website, cs.up.ac.za.
The utility parses the retrieved data into usable and useful data that I store in my content provider and display in my app.

If your app uses a SyncAdapter, what is it called? What mechanism is used to actually talk over the network? (For example, Sunshine uses `HttpURLConnection` to talk to the network, but your app may use a third-party library to do the talking.)

What loaders/adapters are used?

CursorLoader
CursorAdapter

User/App State

Please elaborate on how/where your app correctly preserves and restores user or app state. (See rubric for examples on this question)

The same activity stays on rotation by using a savedInstanceState to store the state of the activity before it is paused, stopped or resumed.
To keep the selected item in a list visible on rotation, the item's position is saved in the savedInstanceState bundle and restored by the loader.

Questions about Optional Components

Answer the questions that are applicable to your final project

Notifications

Please elaborate on how/where you implemented Notifications in your app:

ShareActionProvider

Please elaborate on how/where you implemented ShareActionProvider:

I thought it would be a great idea to enable users to share announcements with each other to ensure their friends don't miss any important information.
In much the same way as was done in Sunshine, I implemented it in the Detail view of my announcements, which only makes sense.
Instead of having an options/settings menu, I implemented the share action as an action

item so that it would be displayed on the action bar. The share action provider uses a implicit share intent to send the date through any app. that can send text.

Broadcast Events

Please elaborate on how/where you implemented Broadcast Events:

Custom Views

Please elaborate on how/where you implemented Custom Views:

*Unfortunately I have run out of time to implement all the features that I would like to have submitted with my app. Such as the use of a SyncAdapter, notifications and general improvements on my code and design.
*Please note that circumstances forced me to work on a laptop without hardware acceleration, thus rendering me incapable of testing whether or not my Master-Detail flow works.
*This app has major potential and I would love to receive feedback on it and suggestions on how I can improve it.