G53MDP Coursework 1-3 Recipe Book

Summary

In this exercise you are required to build an Android Recipe Book application. This is an assessed exercise and will account for **10% of your final module mark**. This is an individual coursework, and your submission must be entirely your own work – please pay particular attention to the section of this document regarding plagiarism. This document sets out general requirements and broad instructions for developing the application.

Your application should be submitted no later than:

- **5pm on Monday, 26th November 2018**

Submissions should be made electronically via Moodle. Standard penalties of 5% per working day will be applied to late submissions.

Your application should be submitted as a .zip or .tar.gz file containing all relevant source code, configuration and related files, and a compiled .apk file – i.e. the contents of the directory containing your Android Studio project. Do not submit RAR files.

Specification

You should create an application with the functionality of a simple recipe book management system for Android, which allows the user to enter and store simple cooking recipes, and to browse and view recipes that have been previously stored.

At minimum your application should support:

- An *Activity* presenting an interface for the user that allows the user to enter a new recipe, consisting of
  - A large text field to enter recipe instructions
  - A text field to enter the title of the recipe
- An *Activity* that allows the user to browse all currently stored recipes by title
- An *Activity* that displays the title and instructions for an existing recipe
- A *ContentProvider* that provides storage for the recipes

You must implement a **ContentProvider** component to handle the storage requirements of the application. You should make sure to use the ContentProvider when retrieving and adding data, rather than directly accessing the underlying storage from other components. There is no requirement that your ContentProvider is used remotely, however you should implement an appropriate contract class so that it could be at some point in the future.

A simple database schema and associated SQL for instantiating it is provided as a starting point, however you can add any additional columns as you see fit.

Your application must be written in Java and make use of the Android SDK. There are no requirements to target a specific Android API version, however you should assume that your application would be tested on an emulated Nexus 6 device (1440x2560 560dpi) running Android API version 25 (Android 7.1.1 Nougat).

You should consider the following when implementing your application:

- Decomposition of the task into logical, discrete Activity components.
- Appropriate use of Activities, Intents and appreciation of the Activity life-cycle.
- Appropriate use of Widgets and ViewGroups for layouts that support devices of differing screen sizes and resolutions.
- Appropriate implementation of a ContentProvider component and Contract class, and appreciation of the ContentProvider life-cycle.
- Your application should have appropriate comments and variable / class names, so that a reader can easily understand how it works at the code level

Adding further additional functionality to the application is encouraged, but please note that this coursework will be primarily assessed on the above specification, and further additions must not detract from the required functionality given above.

Extensions of an appropriate scope might be to add a simple search function, to add the ability to edit or delete a recipe, or to associate an image with a recipe.

Plagiarism

N.B. Use of third party assets (tutorials, images, example code, libraries etc.) MUST be credited and referenced, and you MUST be able to demonstrate that they are available under a license that allows their reuse.

Making significant use of tutorial code while referencing it is poor academic practice, and will result in a lower mark that reflects the significance of your own original contribution.

Copying code from other students, from previous students, from any other source, or soliciting code from online sources and submitting it as your own is plagiarism and will be penalized as such. FAILING TO ATTRIBUTE a source will result in a mark of zero – and can potentially result in failure of coursework, module or degree.

All submissions are checked using both plagiarism detection software and manually for signs of cheating. If you have any doubts, then please ask.

Assessment Criteria

	Marks Available
Basic Application Functionality	
The application allows to store and view recipes	40
Application Structure and Implementation	
Activities appropriately support the task	5
Appropriate use of Views and Layouts	5
Appropriate implementation of a ContentProvider	15
Appropriate implementation of URIs and contract classes	5
Appropriate use of database storage	10
Programming style	
The application is easy to understand, with comments explaining each part of the code, correct formatting, and meaningful variable names	10
Extension	
Appropriate additional functionality not in the specification	10
Total	100

Each element of your coursework will be assessed against the standard criteria.

The following areas will be taken into account for each part of the assessment:

- Demonstrating knowledge of the area
- Quality of the concept, including appropriateness and novelty
- Quality of the technological design, including appropriate use of software design concepts, and appropriate good coding practice (abstraction, commenting, naming)
- Quality of the realization, including how well it works and elaborations over and above the basic requirements
- Including all of the above aspects, clarity of structure, quality of argument / evidence, and insight / novelty

Hints

Note that this coursework aims to serve as an assessment of the database storage and ContentProvider material covered in lectures and used in lab exercises 5 and 6 – as such try to think about how to make use of the concepts covered in those.

Database and Content Provider

Begin by creating a new application in the usual way.

As in previous exercises create a new database and database helper where, when the database is instantiated, the following SQL is executed to create a table to store recipes:

create table test (_id INTEGER PRIMARY KEY AUTOINCREMENT, recipetitle TEXT, recipeinstructions TEXT);

Next wrap the database in a ContentProvider object, creating an appropriate URI schema with an appropriate authority, and implementing the required ContentProvider interface methods to allow inserting of new recipes, and returning existing recipes in response to a query.

Register the new ContentProvider in the manifest as usual.

Finally, build appropriate Activities to make use of the ContentProvider. A common mistake is to not check that the database creation code still works after you've made revisions, as it is only executed on the first execution after installation, or when the database version is upgraded.

References

http://developer.android.com/guide/topics/ui/declaring-layout.html

http://developer.android.com/guide/topics/ui/controls.html

http://developer.android.com/guide/components/activities.html

http://developer.android.com/guide/topics/providers/content-providers.html

http://developer.android.com/guide/topics/data/data-storage.html