

Mobile Application Development

Assignment 1

Due date: 25 September 2022 11:59 AWST

Total marks: 40

This is a group assignment. A maximum of two persons can be part of a group. It is okay if you want to submit it alone if you don't find a group partner.

There is no restriction on the Android API version and third-party libraries. Please justify your choices in the report, e.g., convenience, popularity, and better documentation.

The application you are going to develop is a Food Ordering Application. Here is the scenario: the restaurants in your area (at least 10 of them) have contacted you to build a mobile application for their customer. They want the customer to have a single place to scroll through all the restaurants, browse their menus and order foods. [Wait, is not it the same concept as Uber Eats or Menulog?]. Probably, they didn't want to be part of Uber Eats or Menulog due to financial reasons and decided to have their customised version. And you are to rescue as you are an awesome mobile application developer.

The clients provide you with the freedom to design your application's user interface (UI) as you want. They are only concerned about functionalities and maintenance.

The marks are assigned to each instruction. Note, if you cannot finish a functionality, you will be given partial marks based on your efforts or codes.

The **functionalities** your application should have:

1. Scrolling through a list of restaurants. We assume the restaurants have provided their images, e.g., KFC and Mcdonald's. Make sure you have enough restaurants, so it is scrollable. **[3 Marks]**
2. Scrolling through a list of food items for selected restaurants. We assume the restaurants have provided their food item pictures (e.g., burger, fries), descriptions of the items and the prices of one serving of the food. Make sure you have enough food items, so it is scrollable. **[3 Marks]**
3. The customer can select a food item and add it to their bucket. They should be able to specify how many servings they want to add to the bucket. **[3 marks]**
4. Once all the foods are added to the bucket, the customer can check out. If the customer wants to edit/remove their orders, they can do so until they press the checkout button. We assume that the financial part is handled automatically. **[5 marks]**

5. The checkout button asks for login/ register if the customer is not logged in. The registration/login requires a valid email and password. A successful checkout is stored in the ordered history. **[3 Marks]**
6. A logged customer can view their order history, e.g., time of orders and costs, which are scrollable. They can also check the order details, i.e., restaurant names, ordered food items and cost breakdown. Note that more than one customer may be registered in the application. So, we will only show the order history based on the logged customer. **[3 Marks]**
7. When the application starts, it will never ask for login or registration. The landing or home page of the app will include a random list of food items from different restaurants. This is similar to “Today’s Special Menu”. **[3 marks]**

Technical Constraints: If you don’t follow them, you will lose the corresponding marks.

1. All the information should be stored in the SQLite database. The images do not require to be stored in the database. However, the reference to the image should be stored in the database. For example, if the name of the KFC image file is ‘kfcimage.jpg’, you should store it in the database. It would help if you had a database schema and the entries for the database. When the application loads on the device for the first time, the database will be created with all the necessary information; make sure you have enough data to make the application scrollable. You can download the food, restaurant, or other images from the internet (e.g., <https://www.pexels.com> or others); make sure to refer to them in your report. **[5 marks]**
2. There should not be more than three (3) activities. **[5 marks]**

Documentation/report:

1. This is for maintenance and other developers who will look after your code. Let us assume that new restaurants may want to join your application and provide all the data, images, menus, food items, prices etc. Provide a step-by-step guide on updating the code to add a new restaurant. You will be asked to do the same in your demonstrations. **[5 marks]**
2. If you work in a group, it will be your report on individual contribution – overall contribution percentage (ideally 50%), which parts you have worked on, how long you have worked, how you communicate with your partner etc. **[2 marks]**

Functional Penalties: Your app will be tested with three devices, a Phone (your pick) and a Tablet (your pick size should be greater than 8 inches).

Mark Deductions 1: At least one screen/appearance should be different between the Phone and Tablet. The goal is to use the additional spaces in the tablet wisely. Failure to do so will lose marks **[-2 Marks]**.

Mark Deductions 2: When the device orientation changes or an exception is not caught or in any other situation, the app crashes or losses information on the screen **[-2 Marks]**.

Submission Guidelines:

Submit your assignment electronically, via Blackboard, before the deadline. **If you submit in a group, all the members must submit the same files plus the individual report.**

To submit, do the following:

1. Fill out and sign a declaration of originality. A photo, scan or electronically- filled out form is fine. Whatever you do, ensure the form is complete and readable! Place it (as a .pdf, .jpg or .png) inside your project directory.
2. Attach your Assignment report to your project directory.
3. Zip your entire project directory (as a .zip or .tar.gz file). Leave nothing out.
4. Submit your zip/tar.gz file to the assignment area on Blackboard.
5. Re-download, open, and run your submitted work to ensure it has been submitted correctly.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked. The late submission policy (see the Unit Outline) will be strictly enforced. Please note:

- DO NOT use WinRar.
- DO NOT have nested zip/tar.gz files. One is enough!
- DO NOT try to email your submission as an attachment. Curtin's email filters are configured to discard emails with potentially executable attachments silently. In an emergency, if you cannot upload your work to Blackboard, please instead upload it to Google Drive, a private GitHub repository, or another online service that preserves immutable timestamps and is not publicly accessible.

Marking Demonstration: You will be required to demonstrate and discuss your application with a marker in a one-to-one online/in-person session. Most of the marks for your assignment will be derived from this demonstration. The demonstrator will ask you to rebuild and run your application (provided by the demonstrator) and demonstrate its major features. They may ask you about any aspect of your submission.

The demonstration schedule and policy will be published later (Check blackboard announcements). We may also cancel the demonstration-based marking due to unavoidable circumstances. In that case, it will be a full inspection-based marking.

Academic Integrity: This is an assessable task. If you use someone else's work or obtain someone else's assistance to help complete part of the assignment that is intended for you to complete yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your original work.

Further, if you do not reference any external sources, you are committing plagiarism and collusion, and penalties for Academic Misconduct may apply. Please see Curtin's Academic Integrity website for information on academic misconduct (which includes plagiarism and collusion).

The unit coordinator may require you to provide an oral justification of or to answer questions about any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an Academic Misconduct inquiry.

End of Assignment