Assignment

3

BIT603 Programming 2

Assignment 3

Weighting: 40%

Due date: Please refer to the assessment tab for the due date.

Note: You must submit your work through the assignment section. Upload your assignment, then use the ‘Save’ and ‘Submit assessment’ buttons – or we will not receive your assignment. You will get an electronic receipt and your work is then logged and tracked through our system. Open Polytechnic will not accept assignments emailed directly to lecturers.

Overview

This assignment is worth 40% of your total course mark. The assignment consists of an application and report that contain tasks related to what you have learnt over this course.

This assignment focuses on the following learning outcomes:

LO1: Demonstrate the effective use of multiple contemporary programming languages and integrated development environments (IDEs)

LO2: Describe and apply the principles and techniques required for modular software development using object oriented concepts and core programming techniques

LO3: Demonstrate effective use of commonly used built-in data structures

LO4: Develop a comprehensive solution to a given problem, which includes designing, debugging, testing and maintenance techniques

LO5: Describe optimisation concepts and techniques and the application of source and version control as a critical function of project management

Instructions

The work that you submit should be the complete Android Studio project, including any code you have written yourself. This project must be able to be opened in an up-to-date version of Android Studio.

The code should be written in a simple and readable style, applying the principles you have learned throughout the modules. Good programming practices should be followed: variable names should be meaningful and descriptive, code should be neatly formatted and aligned.

The user interface should also follow good practices with meaningful widget names, appropriate position and sizing with proper constraints. The interface should look clean and uncluttered.

Comments should be added to the code as necessary to describe anything that may not appear obvious to the reader or should have further explanation.

Assignment guidelines

In this assignment, you will be creating a mobile application using Android Studio. Your application will meet the needs of a client in the provided case study.

This program will be written using Java and Android Studio. You will need to create the user interface along with the code to make this application meet the criteria specified within the assignment.

This assignment is an extension of the previous assignment (Assignment 2), which required you to build a simple app with a log-in system along with some basic functionality.

Submission

Refer to the marking criteria before submitting your assignment to ensure that it all areas have been addressed. Insert comments as necessary and check that the code is tidy. Remember to test your program to find any errors and address them.

You must submit a zip file with the entire project folder included. It is recommended that you extract this zip file in another folder and ensure that the project can be opened correctly and run successfully.

Name your zip file ‘BIT603_A2_[YourName].zip’. For example, BIT603_A2_JohnSmith.zip

Upload the zip file, the image for your wireframe, and your report.

Assignment

Case study

The store ‘Kiwi Cookies and Cakes’ has been using your previous app and have decided to increase the budget to have you build an updated and more thorough version. Although they would like to possibly expand it again in the future, they would like an upgrade of the previous app with some improvements.

Due to some limitations of your previous app, it has been decided that simply updating the previous project with some updates will cause problems. Instead, you will be creating a completely new project. The client has requested that you build this new app from scratch to prevent any unexpected problems. Fortunately, you have learned lessons from your previous app and will bring this new knowledge and experience to this project.

The previous app had a simple log-in system and inventory management. In this upgrade, the client would like a complete log-in system with the ability to create new users, storing these users in a database that is then used to perform the validation for logging in. For now, storing the database information insecurely is sufficient; a future update can handle security improvements.

Additionally, they would like the inventory management system to be updated to allow more information about the product to be stored, in addition to quantity they would like to include what type of product it is (Biscuit, Cookie, Cake, Ingredient, or Other). The user should be able to add new items into the inventory and query this information as a report.

Tasks

Creating the project

Although this project has similarities to your previous work for this business, this app is an entirely new project and must be started using a fresh project. You will be creating the app using a new Android Studio project. It is recommended not to use code from the previous project as the requirements have changed and there are some fundamental differences in the way the functions should be built.

Create a new Android project in Android Studio. Name it ‘BIT603_A3_[YourName]’ (for example, BIT603_A3_JohnSmith). The target Android devices are only phone and tablet using API 15 or higher.

When creating each activity be sure to uncheck ‘Backwards compatibility’.

Designing the app

Using the description that you have been provided with, you must create your own design and mock-up of the app. Use the description to come up with an idea of what the app will look and which screens it will require, keeping in mind there are additional screens from the previous app. Create a wireframe diagram of the app using your own choice of tool (pen and paper or digital, such as prototyping or wireframing software). This wireframe diagram should include descriptions of the screens, the layout, and any other information you consider to be useful. The target device is a Nexus 5 phone.

Logging in

The first screen the user should see is a screen with fields to enter a user name and password; there should be a button to attempt to log in (a submit button).

The user is expected to enter a user name and password, the app then checks the database to see if this matches the records stored. Although this could be done directly by using database queries, this can instead be done by reading data from a data structure with the table information stored.

If the user enters an incorrect login, they will receive the appropriate message and fail to log in. If they enter the correct log-in details, they are taken to the menu screen.

This log-in system will have a special hard-coded log-in account called ‘Admin’ with the temporary password ‘CookieManagement84’. This will bypass the standard log-in process (which checks the database) and move the user directly to the ‘Manage Users’ screen, where they can add or remove users from the database.

Managing users

These screens and features are accessible only by the ‘Admin’ account.

Use a menu screen to provide options to view, add, or remove users. Each of these options should take the user to a new screen that performs the appropriate actions.

The ‘View Users’ screen will show available details for each user. This information should be displayed clearly using an appropriate format.

The ‘Add Users’ screen will allow the admin to enter a username, password, date of birth, employee number, contact phone number, and address. These should use the appropriate fields. When the admin attempts to add a new user with any field blank or containing inappropriate data, a warning should be given (nothing should be inserted into the database if a problem was found). If all fields are correct, an appropriate message should be displayed (using toast) as the user, and the user data is entered into the database.

The ‘Remove Users’ screen will have a text field (username) and a button. Entering a username (in the text field) and clicking the button will attempt to remove the user from the database. If the user doesn’t exist, an appropriate message is displayed; if the user does exist then the user is removed from the database after confirmation from the admin (for example, a Yes/No message box).

Inventory menu

Inventory related screens are accessible by all users.

A simple menu showing the options: Inventory Status, Add Item, Clear Items, and Add Test Items. Viewing the status and adding an item will move the user to a new screen. Using the ‘Add Test items’ button will add 20 ‘test’ entries into the database (giving each a name, item type, and quantity); this will be used to test the inventory view easily.

Adding items to inventory

Display fields to the user with name, quantity, and item type (user can select from Biscuit, Cookie, Cake, Ingredient, Other). Name is a text string, quantity a number, and item type can be selected from a list or another method (such as radio buttons).

Item type should only be one of the predetermined options – the user should not be able to enter their own type.

The item should be added to the database when the appropriate button is clicked along with an appropriate message if the item was added or not. The user should not be able to add an item if one of the fields is not filled in, and in this case a warning message should be shown to the user.

Only one database table is required to store the inventory items: the rows required are name, quantity, and item type.

Add test items

A new screen is not required for this functionality, but confirmation is required before adding the test items. This feature is used to allow you to quickly test the app when it has many items added. This will create 20 items (with a variety of names, quantity, and types) and insert all 20 into the database. These items should act the same as an item added manually by the user.

Inventory status

This screen displays the available items from the database stored in the inventory, which shows the item name, type, and quantity in stock. It is recommended to read all data from the database (including test data if it was used), storing it into a data structure. You may choose to create objects to store individual item data, with the objects stored in a data structure such as a list.

Your screen will show only **five** items at a time. However, your app will need ‘next’ and ‘previous’ buttons to move the users through the list of items in order to view all items. Clicking next, for example, will show the next five items in the list, and clicking previous will show the previous five items. These buttons should only function when there are options to display – for example, you should not be able to use the ‘previous’ button if it is already displaying the first five items.

App quality

Ensure that all screens follow good guidelines, and consider any additional needs for these screens that will improve the quality of life of the user. For example, consider where the user would want to navigate to from each screen – they should not have to close the app to return to the log-in screen.

The app should be thoroughly tested for quality and debugged as necessary. Testing should cover the functionality to ensure accurate results and overall quality of the app.

Debugging and testing

Your app should be thoroughly tested to ensure that the functions work as expected. Each screen, feature, and function should be tested to ensure the quality of the application. Ensure that error checking is performed, and the appropriate error messages are shown to the user if necessary.

Report

In addition to your app, you are required to provide a short report (500–800 words). In this report, you should answer the questions listed below relating to optimisation and version control as they apply to the project management of your app. Your report should address the following:

- Explain the importance of version control when managing a project.
- Explain areas where you have optimised your app in any context.
- Discuss any areas you feel the app could be further optimised (regardless of whether you could perform these optimisations or not).
- From the perspective of a project manager, explain whether you would or wouldn’t delay the release of an app to perform optimisations and give your reasoning.

Marking schedule

Your tutor will use this marking schedule to provide you with a grade. They may also provide qualitative feedback (comments) about your work.

Criteria	Marks
Design	**/5**
Completed wireframe diagram showing screens, interface, and useful information	3
Wireframe looks professional (diagram is clean, uncluttered, and easy to read and understand)	2
Project	**/15**
User can move appropriately between the screens	5
Good programming practices have been followed with appropriate commenting	5
Evidence of debugging shown and the app is functioning correctly without unexpected errors, poor design, or unintended behaviour	5
Managing users and log-in process	**/25**
Appropriate user interface across all screens	5
Adding users functions correctly as described (including validation of data entered)	5
Removing users functions correctly as described	5
User and admin can successfully log in and move to the next screen	5
Login related screens demonstrate testing (evidenced by demonstrating quality and being free from bugs)	5
Inventory	**/45**
Appropriate user interface across all screens	5
Functionality of the ‘Add Test Items’ option	5
Functionality for adding and removing items to the database	10
Inventory is correctly displayed after being read from the database	10

Functionality for the ‘next’ and ‘previous’ buttons when viewing inventory	10
Inventory-related screens demonstrate testing (evidenced by demonstrating quality and being free from bugs)	5
Report	**/10**
Clearly answers given topics in a satisfactory manner	8
Punctuation, spelling, and grammar	2
	Total: 100