

MOBILE APPLICATION DEVELOPMENT

Competencies:

4026.1.1: Introduction to Mobile Application Development - The graduate explains mobile development, develops a simple mobile application using IDE, documents debugging the mobile application, and describes how to use an emulator.

4026.1.2: Activity Lifecycle - The graduate describes the Activity lifecycle in the mobile application, and creates and links an activity.

4026.1.3: User Interfaces and Handling User Input - The graduate creates a user interface and describes how to handle user input.

4026.1.4: Saving Data - The graduate explains ways to save data in a mobile application, and creates a database in a mobile application.

4026.1.5: Sharing Information - The graduate explains how to share information in mobile applications and creates a user-defined content provider.

4026.1.6: Supporting Different Devices - The graduate describes how to utilize the available hardware and services available in different devices.

4026.1.7: Deploying Mobile Application - The graduate describes mobile application deployment and prepares an application for deployment.

Task 1: Mobile Application Development

Introduction:

As a competent mobile application developer, your understanding of mobile application structure and design will help you to develop applications to meet customer requirements. The following project to develop a student scheduler/student progress tracking application, will help you to apply these skills in a familiar, real-world scenario. This task will allow you to demonstrate your ability to apply the skills learned in the course.

You will develop a multiple-screen mobile application for WGU students to track their terms, courses associated with each term, and assessment(s) associated with each course. The application will allow students to enter, edit, and delete term, course, and assessment data. It should provide summary and detailed views of courses for each term and provide alerts for upcoming performance and objective assessments. This application will use a SQLite database.

Input

Each of the following needs to be input into the application:

- terms, including the following:
 - the term title (e.g., Term 1, Term 2)
 - the start date
 - the end date
- courses, including the following:
 - the course title
 - the start date
 - the anticipated end date
 - the status (e.g., in progress, completed, dropped, plan to take)
 - the course mentor name(s), phone number(s), and email address(es)
- objective and performance assessments associated with each course
- the ability to add optional notes for each course
- the ability to set alerts or notifications for the start and end date of each course
- the ability to set alerts for goal dates for each objective and performance assessment

Output

Each of the following needs to be displayed by the application on a separate screen:

- the navigation panel
- a list of terms
- a detailed view of each term, including the following:
 - the title (e.g., Term 1, Term 2)

- the start date
- the end date
- a list of courses for each term
- a detailed view of each course, including the following:
 - the course title
 - the start date
 - the anticipated end date
 - assessments
- a list of performance and objective assessments for a selected course
- a detailed view of each objective and performance assessment, including the following:
 - the due date for a selected course
 - notes for the selected course
 - sharing features (e.g., email, SMS)

Requirements:

Note: Submit your performance assessment by including all Android project files, APK, and signed apk files in one zipped.zip file.

Note: For your convenience an optional checklist is attached to help guide you through this performance assessment.

A. Create an Android (version 4.0 or higher) mobile application with the following functional requirements:

1. Include the following information for *each* term:
 - the term title (e.g., Term 1, Term 2)
 - the start date
 - the end date
2. Include features that allow the user to add as many terms as needed.
3. Implement validation so that a term cannot be deleted if courses are assigned to it.
4. Include features that allow the user to do the following for *each* term:
 - a. add as many courses as needed
 - b. display a list of courses associated with *each* term
 - c. display a detailed view of *each* term, which includes the following information:
 - the term title (e.g., Term 1, Term 2)
 - the start date
 - the end date
5. Include the following details for *each* course:
 - the course title
 - the start date
 - the anticipated end date
 - the status (e.g., in progress, completed, dropped, plan to take)
 - the course mentor name(s), phone number(s), and e-mail address(es)
6. Include features that allow the user to do the following for *each* course:
 - a. add as many assessments as needed
 - b. add optional notes
 - c. enter, edit, and delete course information
 - d. display optional notes
 - e. display a detailed view of the course, including the due date
 - f. set alerts for the start and end date
 - g. share notes via a sharing feature (e.g., e-mail, SMS)
7. Include features that allow the user to do the following for *each* assessment:
 - a. add performance and/or objective assessments for *each* course, including the titles and due dates of the assessments
 - b. enter, edit, and delete assessment information
 - c. set alerts for goal dates

B. Design *at least* the following **seven** screen layouts, including appropriate GUI elements for *each* screen:

- a home screen

- a list of terms
- a list of courses
- a list of assessments
- a detailed course view
- a detailed term view
- a detailed assessment view

Note: You are required to design the seven screens listed in part B but are not limited to only seven.

C. Create a scheduler and progress tracking application in your mobile application from part A.

Note: This can be the implementation of part A.

1. Include the following implementation requirements:
 - an arrayList
 - an intent class
 - navigation capability between multiple screens using activity
 - **at least three** activities
 - events (e.g., a click event)
 - the ability to work in portrait and landscape layout
 - interactive capability (e.g., the ability to accept and act upon user input)
 - a database to store and retrieve application data
 - an application title and an icon
 - the use of **at least two** different methods to save data (e.g., SharedPreferences, SQLite)
 - notifications or alerts
 - the use of **both** declarative and programmatic methods to create a user interface

Note: Your application should work at least on Android 4.0 (API level 14).

2. Include **at least** the following interface requirements:
 - **seven** screens
 - the ability to scroll vertically
 - an action bar
 - a menu
 - an imageview
 - **two** layouts
 - input controls
 - buttons

D. Create a storyboard that includes *each* of the menus and screens from part B to demonstrate application flow.

E. Provide screenshots to demonstrate that you have created a deployment package.

Note: Verify that all the required functions of your application are working by executing the apk file.

- F. Reflect on the creation of your mobile application by doing the following:
1. Explain mobile application development through the context of the architecture involved, including hardware and software capabilities and limitations.
 - a. Identify which version of the operating system your application was developed under and is compatible with.
 2. Describe (*suggested length of at least one paragraph*) the challenges you faced during the development of the mobile application.
 3. Describe (*suggested length of at least one paragraph*) how you overcame each challenge discussed in part F2.
 4. Describe (*suggested length of at least one paragraph*) what you would do differently if you did the project again.

5. Describe how emulators are used and the pros and cons of using an emulator vs. using a development device.

G. When you use sources, include all in-text citations and references in APA format.

Note: For definitions of terms commonly used in the rubric, see the Rubric Terms web link included in the Evaluation Procedures section.

Note: When using sources to support ideas and elements in an assessment, the submission MUST include APA formatted in-text citations with a corresponding reference list for any direct quotes or paraphrasing. It is not necessary to list sources that were consulted if they have not been quoted or paraphrased in the text of the assessment.

Note: No more than a combined total of 30% of a submission can be directly quoted or closely paraphrased from outside sources, even if cited correctly. For tips on using APA style, please refer to the APA Handout web link included in the APA Guidelines section.

Web Links:

1. [CLP Task 1 Rubric](#)
-