LAP1 — MOBILE APPLICATION DEVELOPMENT USING C#

MOBILE APPLICATION DEVELOPMENT USING C# — C971

PRFA — LAP1

COMPETENCIES

4026.02.1: : Introduction to Mobile Development
The graduate develops a simple mobile application using an integrated development environment (IDE).

4026.02.2: : User Interfaces
The graduate develops a user interface for a mobile application.

4026.02.3 : Managing Data
The graduate creates solutions to store, retrieve, and transmit mobile application data.

4026.02.4 : Supporting Devices and Platforms
The graduate develops an application that accounts for different platforms and device conditions.

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.

For this task, you will develop a multiple-screen mobile application for WGU students to track their academic terms, courses associated with each term, and assessments 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, provide notifications for upcoming performance and objective assessments, and notify students of deadlines and tracking milestones even after the application is closed. You will create a wireframe as a visual guide, representing the skeletal framework of the application. Your application will use an SQLite database.

This task will allow you to demonstrate your ability to apply the skills learned in the course and help you to apply these skills in a familiar, real-world scenario.

REQUIREMENTS

Note: *The assessment must be submitted using the Xamarin.Forms framework.*
For this assessment, you are welcome to use these plugins:

> https://docs.microsoft.com/en-us/xamarin/essentials/
> https://github.com/edsnider/LocalNotificationsPlugin
> https://docs.microsoft.com/en-us/dotnet/api/Xamarin.Forms.Picker?view=xamarin-forms
> https://docs.microsoft.com/en-us/dotnet/api/Xamarin.Forms.DatePicker?view=xamarin-forms

All other external plugins and libraries are not allowed in the project.

The submission needs to include a zip file with the project file and folder structure intact for the Visual Studio IDE.

Your submission must be your original work. No more than a combined total of 30% of a submission can be directly quoted or closely paraphrased from sources, even if cited correctly. Use the report provided when submitting your task as a guide.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

A. Draw a low-fidelity wireframe for your mobile application, that includes *all* of the following requirements:

> Note: This assessment requires you to submit pictures, graphics, and/or diagrams. Each file must be an attachment no larger than 30 MB in size. Diagrams must be original and may be hand-drawn or drawn using a graphics program. Do not use CAD programs because attachments will be too large.

1. As many academic terms as needed; *each* term should include the addition, editing, deletion, and storage of *all* the following:
    - the term title
    - start and anticipated end dates (use a DatePicker)
2. Add **six** courses for *each* term. For *each* course, include the addition, editing, deletion, and storage of *all* the following course information:
    - course name
    - start and anticipated end dates (use a DatePicker)

- course status (use a Picker)
- course instructor's information: name, phone, email
- add, share, and display optional notes
- set notifications for the start and end dates of *each* course
- display of a detailed view of *each* course, including the due date
- editing of the detailed course view screen

3. Add **two** assessments for *each* course; *each* assessment should include the addition, editing, deletion, and storage of *all* the following assessment information:
- **one** objective assessment
- **one** performance assessment
- name of *both* assessments
- set notifications for the start and end dates of *each* assessment

B. Create within the Xamarin.Forms framework, a mobile application aligned to the wireframe drawn in part A, and include *all* of these features:
1. Provide an interface for *all* the following information for as many academic terms as needed:
- academic term title (e.g., Term 1, Term 2, Spring term)
- start and end dates (use a DatePicker)
  You are welcome to use the following plugin for this step:
  https://docs.microsoft.com/en-us/dotnet/api/Xamarin.Forms.DatePicker?view=xamarin-forms
2. Provide an interface that allows the user to access *all* the following features for *each* academic term:
- add and display a list of **six** courses for *each* term
- display a detailed view of *each* term, including *all* the information from part B1
3. Provide the interface that allows the user to access and edit *all* the following details for *each* course:
- course title
- start and anticipated end dates (use a DatePicker)
  You are welcome to use the following plugin for this step:
  https://docs.microsoft.com/en-us/dotnet/api/Xamarin.Forms.DatePicker?view=xamarin-forms
- course status (e.g., in progress, completed, dropped, plan to take) (use a Picker)
  You are welcome to use the following plugin for this step:
  https://docs.microsoft.com/en-us/dotnet/api/Xamarin.Forms.Picker?view=xamarin-forms
- the course instructor's name, phone number, and email address
  - include validation to prevent the user from saving a null value (e.g., an invalid email address)
4. Create features that allow the user to do *all* the following for *each* course:
- add **two** assessments: **one** performance assessment and **one** objective assessment
- add and display optional notes
- enter, edit, and delete course information
- display an editable detailed view of the course, including the due date
- set alerts (e.g., notifications) for the start and end date of the course
  You are welcome to use the following plugin for this step:
  https://github.com/edsnider/LocalNotificationsPlugin
- share notes via a sharing feature (e.g., email, SMS)
  You are welcome to use the following plugins for this step:
  https://docs.microsoft.com/en-us/xamarin/essentials/
5. Provide an interface for the user to do *all* the following for *each* assessment:
- include the names and due dates
- enter, edit, and delete assessment information
- set notifications for anticipated due dates
  You are welcome to use the following plugin for this step:
  https://github.com/edsnider/LocalNotificationsPlugin
6. Write code to create a set of data for evaluation purposes, including **one** term and **one** course from part B3, and include the **two** assessments from part B4 for that course. Include your own name, phone number, and email address as the course instructor for the course.

C. Examine the wireframe from part A, to determine any changes that you made during the development of the mobile application.
1. Explain the reasons for any changes made during the development of the mobile application, by adding comments within the wireframe. If no changes were made, include that comment.
2. Confirm that, after the inclusion of the changes made during development, the wireframe aligns with the mobile application.

D. Provide a complete and operational source file for your mobile application.

E. Acknowledge sources, using APA-formatted in-text citations and references, for content that is quoted, paraphrased, or summarized.

F. Demonstrate professional communication in the content and presentation of your submission.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 200 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A1:ACADEMIC TERMS IN WIREFRAME

NOT EVIDENT

The low-fidelity wireframe does not allow for the inclusion of academic terms.

APPROACHING COMPETENCE

The low-fidelity wireframe includes as many academic terms as needed, but it does not allow for the addition, editing, deletion, and storage of term titles, or the start and anticipated end dates associated with each term. Or dates are not entered using a DatePicker, or it is not verified that all start dates for terms are set before the anticipated end dates.

COMPETENT

The low-fidelity wireframe includes as many academic terms as needed, the addition, editing, deletion, and storage of term titles, and the start and anticipated end dates associated with each term. Dates are entered using a DatePicker, and it is verified that all start dates for terms are set before the anticipated end dates.

A2:COURSES IN WIREFRAME

NOT EVIDENT

The low-fidelity wireframe does not allow for the inclusion of any courses for each term.

APPROACHING COMPETENCE

The low-fidelity wireframe allows for the inclusion of only 1–5 courses for each term, or it does not include the information associated with each course. Or the wireframe does not allow for the addition, editing, deletion, or storage of each additional given course information. Or it is not verified that all start dates for each course are set before the anticipated end dates.

COMPETENT

The low-fidelity wireframe allows for the inclusion of 6 courses for each term and the information associated with each course. The wireframe for each course allows for the addition, editing, deletion, and storage of all additional given course information. It is verified that all start dates for each course are set before the anticipated end dates.

A3:ASSESSMENTS IN WIREFRAME

NOT EVIDENT

The low-fidelity wireframe does not allow for the inclusion of any assessments for each course.

APPROACHING COMPETENCE

The low-fidelity wireframe allows for the inclusion of only 1 assessment for each course, or it does not include the addition, editing, deletion, or storage of all the given assessment information.

COMPETENT

The low-fidelity wireframe allows for the inclusion of 2 assessments for each course, 1 objective and 1 performance assessment, and includes the addition, editing, deletion, and storage of all the given assessment information.

B1:INTERFACE FOR TERMS

NOT EVIDENT

The mobile application does not allow the user to enter any information for any academic terms.

APPROACHING COMPETENCE

The mobile application does not allow the user to enter the title of each academic term or all the start and end dates for each term. Or dates are not entered using a DatePicker, nor is it verified that all start dates for terms are set before the anticipated end dates.

COMPETENT

The mobile application allows the user to enter the title of each academic term and all start and end dates for each term. Dates are to be entered using a DatePicker, and it is verified that all start dates for terms are set before the anticipated end dates.

B2:TERM INTERFACE FOR USER

NOT EVIDENT

The mobile application does not provide an interface that allows the user access to any features for the courses.

APPROACHING COMPETENCE

The mobile application provides an interface for the user, but it does not allow the user to add or display all 6 courses for each term, or it displays only a limited view of the data from part B1.

COMPETENT

The mobile application provides an interface for the user to add and display 6 courses for each term and to display a detailed view of each term that includes all the information from part B1.

B3:COURSE INTERFACE

NOT EVIDENT

The mobile application interface does not allow the user to access or edit any of the given details for each course.

APPROACHING COMPETENCE

The mobile application interface allows the user to access, but not to edit the given details for each course. Or the application does not use a DatePicker for all start and anticipated end dates, or does not use a Picker to enter course status. Or it is not verified that all start dates for each course

COMPETENT

The mobile application interface allows the user to access and edit all of the given details for each course. The application uses a DatePicker for all start, and anticipated end dates and uses a Picker to enter course status. It is verified that all start dates for each course are set before the anticipated end

	are set before the anticipated end dates. Or the interface does not include validation to prevent the user from saving a null value.	dates. The interface includes validation to prevent the user from saving a null value.

B4:COURSE FEATURES FOR USER

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The mobile application does not allow the user to enter any information for the courses.	The mobile application allows the user to enter only 1 assessment, or allows for 2 assessments for each course, but with no distinction between the objective and performance assessment. Or the created features do not allow the user to do all additional given features.	The mobile application allows the user to enter 2 assessments, 1 performance and 1 objective assessment for each course, and allows the user to do all additional given features.

B5:ASSESSMENT INTERFACE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The interface provided does not allow the user to enter any information for any assessments.	The mobile application provides an interface but the interface does not allow the user to include names or dates, or enter, edit, or delete assessment information, or set notifications for due dates.	The mobile application provides an interface that allows the user to include names and dates, and enter, edit, and delete assessment information, and set notifications for due dates.

B6:DATA SET

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The code to create a set of data is not provided.	The code written creates a set of data for evaluation purposes, but does not include either 1 term or 1 course from part B3, or 1 or more assessments from part B4. Or the candidate's own name, phone number, or email address is not provided as the course instructor.	The code written creates a set of data for evaluation purposes, including 1 term and 1 course from part B3, and the 2 assessments from part B4. The candidate's own name, phone number, and email address are provided as the course instructor.

C1:CHANGES TO WIREFRAME

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The wireframe from part A contains no comment as to the changes made during the development of the mobile application.	The wireframe from part A includes comments that identify changes made during the development, but do not explain the reasons for the changes.	The wireframe from part A includes comments that explain the reasons for any changes made to the design, during the development of the mobile application. Or there is a comment stating no changes were made.

C2:CONFIRM WIREFRAME MATCHES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The response does not address the alignment of the wireframe and the mobile application.	The wireframe aligns with the completed mobile application from part B, but only before the inclusion of the comments from C1.	The wireframe aligns with the completed mobile application from part B which includes the comments from C1.

D:SOURCE FILE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A source file for the mobile application is not provided.	The source file for the mobile application is incomplete or cannot run on any platform as a Xamarin.Forms application.	The source file for the mobile application is complete and runs on any platform as a Xamarin.Forms application.

NOT EVIDENT

The submission does not include in-text citations and references according to APA style for content that is quoted, paraphrased, or summarized.

APPROACHING COMPETENCE

The submission includes in-text citations and references for content that is quoted, paraphrased, or summarized but does not demonstrate a consistent application of APA style.

COMPETENT

The submission includes in-text citations and references for content that is quoted, paraphrased, or summarized and demonstrates a consistent application of APA style.

NOT EVIDENT

Content is unstructured, is disjointed, or contains pervasive errors in mechanics, usage, or grammar. Vocabulary or tone is unprofessional or distracts from the topic.

APPROACHING COMPETENCE

Content is poorly organized, is difficult to follow, or contains errors in mechanics, usage, or grammar that cause confusion. Terminology is misused or ineffective.

COMPETENT

Content reflects attention to detail, is organized, and focuses on the main ideas as prescribed in the task or chosen by the candidate. Terminology is pertinent, is used correctly, and effectively conveys the intended meaning. Mechanics, usage, and grammar promote accurate interpretation and understanding.