Individual Assignment: In-Depth Study with Implementation

Mobile App Development (C1UM1B) Autumn 2025

Contents

1 Introduction

1.1 Purpose

The purpose of this assignment is to gain a deeper understanding in designing and implementing a .NET MAUI mobile application using .NET MAUI Platform Features, and optionally third-party NuGet packages.

The assignment assesses knowledge and understanding of the following learning outcomes:

2.1. design and implement mobile applications that are part of a larger distributed system

2.2. design and implement mobile applications with technologies intended to be executed on Android systems

2.3. design and implement mobile applications using technologies intended to be executed on iOS systems

2.4. apply graphical interface design constructions using the XAML markup language and the .NET MAUI framework

2.5. apply constructions for graphical interface design with the programming language C# and the .NET MAUI framework

2.6. apply constructions for database communication in C# and .NET through object-relational mapping

3.1. independently choose appropriate standards and techniques for application in the field of mobile application development

3.2. independently collect, compile, critically evaluate and present information in writing and orally about standards and technologies in the field of mobile application development

1.2 Preparation

To prepare for the assignment, you need to:

1. Identify a relevant problem and application area for your application.
2. Identify the appropriate .NET MAUI Platform Features (in namespace Microsoft.Maui), based on your needs.
    - See Lecture 3 - .NET MAUI Platform Features.
3. Find and evaluate NuGet packages for your project.
    - You can search for NuGet packages using the NuGet Gallery.

2 Task

Your task is to define, design, and implement a .NET MAUI mobile application of your own choice, which can be one of the following:

- A standalone .NET MAUI mobile application (Android or iOS) that leverages .NET MAUI Platform Features.
- A .NET MAUI mobile application (Android or iOS) that leverages .NET MAUI Platform Features AND integrates with services on the internet (e.g. map services).

The functionality of the application is completely up to you (within the constraints given above and in Section 3 Requirements).

- An example of a standalone application can be a game using the mobile device's sensors.
- An example of integrating with services on the internet, could be to automatically connect the application to Microsft Office 365.

The project task includes making a simple purpose description and requirement specification.

A user interface for the mobile app will be designed and implemented in Visual Studio.

To streamline the system development process, it is desirable to use ready-made software modules and to integrate these into the application whenever possible.

Visual Studio includes functionality to easily add modules from NuGet.

NuGet contains many different modules with varying functionality, such as:

- Scanning barcodes with the camera. See, for example, this open and free library: Barcode Scanner.

NOTE: An application that ONLY works on iOS will be a problem to run for the correcting teacher.

3 Requirements

The following requirements apply to the assignment:

- The chosen project must be within one of the techniques and frameworks covered in the course, as specified by the constraints given in Section 2 Task).
- The implemented .NET MAUI mobile application must be for Android or iOS.
- The mobile application must be implemented in C# and .NET MAUI.
- You must use the MVVM architecture for your UI.
  - Data should only ne in Models.
  - Presentation code should only be in Views.
  - Application logic should only be in ViewModels (and/or Services).
- It's permitted to include ready-made third-party software modules via NuGet.
  - This use must be clearly documented in the final report.

4 Hints

Some useful hints are provided below:

- Use the following NuGet packages:
  - CommunityToolkit.Mvvm (with samples) for a lot simpler MVVM code.
  - CommunityToolkit.Maui (with samples) for a lot simpler MAUI code.
- For .NET MAUI Platform Features:
  - See Lecture 3 - .NET MAUI Platform Features.
  - See Microsoft's documentation on .NET MAUI Platform Features.
  - Use the PlatformIntegrationDemo as a reference.
- Use ChatGPT (or Copilot) rather than Google to find quick answers (and examples).

5 Administrative Information

5.1 Individual Assignment

This is an individual assignment, so there are no groups in this assingnment.

5.2 Cooperation

It is not permitted to share solutions, or parts of solutions, between individuals.

5.3 Tutoring

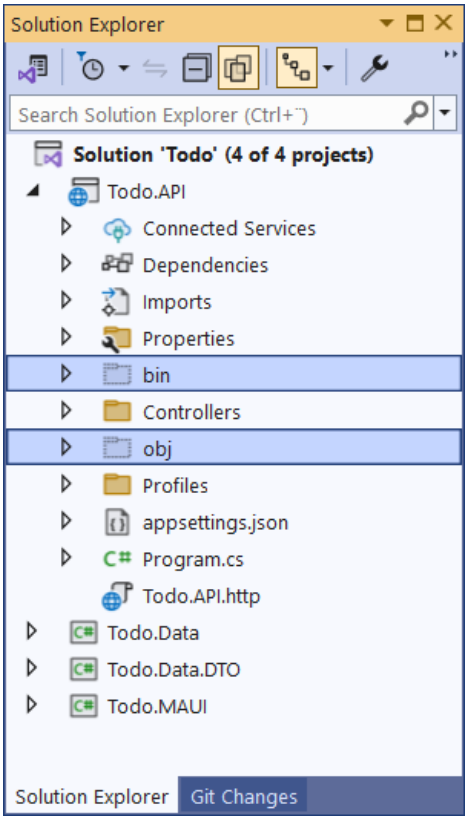
Tutoring sessions are available (see the information on Canvas and Kronox).

5.4 Submission

Submit your compressed (zipped) Visual Studio Solution, including your report, via Canvas (see the information on Canvas and Kronox.

Before submitting your solution:

- Right-click the Solution node in Visual Studio's Solution Explorer and choose **Open Folder in File Explorer**, which will take you to the Solution folder in your file system.
- Here you will see a sub-folder for each project in your solution. Enter each project folder and delete the sub-folders **bin** and **obj** (these are re-generated when compiling, take up a lot of space, and don't need to be submitted).
- Head back to the Solution folder, and place your report (PDF) in it.
- Enter the Solution folder's parent folder.
- Right-click the Solution folder (the one that contains the **.sln** file) and your report **.pdf**, then choose **Send to compressed (zipped) folder**, which will compress (zip) the solution folder, including its files and sub-folders.
- Upload the compressed (zipped) file to Canvas.



5.5 Presentation

The assignment will be presented at a seminar (see Canvas and Kronox for details).