**D424 – Software Engineering**

**Task 3**

|  |  |
| --- | --- |
| Capstone Proposal Project Name: | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Term Planner mobile application |
| Student Name: | Sif Eddine Brahimi |

Table of contents

[Capstone Proposal Project Name: 1](#_Toc174494524)

[Student Name: 1](#_Toc174494525)

[Application Design and Testing 4](#_Toc174494526)

[Class Design 4](#_Toc174494527)

[UI Design 5](#_Toc174494528)

[Unit Test Plan 5](#_Toc174494529)

[Introduction 5](#_Toc174494530)

[Purpose 5](#_Toc174494531)

[Overview 6](#_Toc174494532)

[Test Plan 6](#_Toc174494533)

[Items 6](#_Toc174494534)

[Features 6](#_Toc174494535)

[Deliverables 7](#_Toc174494536)

[Tasks 7](#_Toc174494537)

[Needs 7](#_Toc174494538)

[Pass/Fail Criteria 8](#_Toc174494539)

[Specifications 8](#_Toc174494540)

[Procedures 10](#_Toc174494541)

[Results 11](#_Toc174494542)

[User guide for maintenance purposes 12](#_Toc174494543)

[1. Introduction. 12](#_Toc174494544)

[2. System Requirements 12](#_Toc174494545)

[3. Setting Up the Development Environment 12](#_Toc174494546)

[3.1 Install Visual Studio 2022 12](#_Toc174494547)

[3.6 Run the application. 14](#_Toc174494548)

[4. Running the Application 14](#_Toc174494549)

[4.1 First Launch 14](#_Toc174494550)

[4.2 Navigating the Application 14](#_Toc174494551)

[4.3 Generating Reports 14](#_Toc174494552)

[5. Maintenance and Updates 15](#_Toc174494553)

[5.1 Updating the Application 15](#_Toc174494554)

[5.2 Database Maintenance 15](#_Toc174494555)

[5.3 Monitoring and Logs 15](#_Toc174494556)

[5.4 Error Handling 15](#_Toc174494557)

[6. Troubleshooting 16](#_Toc174494558)

[6.1 Common Issues 16](#_Toc174494559)

[6.2 Getting Help 16](#_Toc174494560)

[User Guide 16](#_Toc174494561)

[Introduction: 16](#_Toc174494562)

[Installation and Using the Application 17](#_Toc174494563)

[System Requirements 17](#_Toc174494564)

[Installation Steps 17](#_Toc174494565)

[Launch the Application: 17](#_Toc174494566)

[Login and Signup 18](#_Toc174494567)

[Signup Process 18](#_Toc174494568)

[Using the Application 18](#_Toc174494569)

[Panopto Video Link 25](#_Toc174494570)

Task 3 Design Document

# Application Design and Testing

## Class Design

Using concepts from object-oriented programming, such as inheritance, encapsulation, and polymorphism, the application's class design represents the different entities, pages, and functionalities within the system. This architecture guarantees that the code is reusable, scalable, and maintainable while also helping to organize the application's structure.

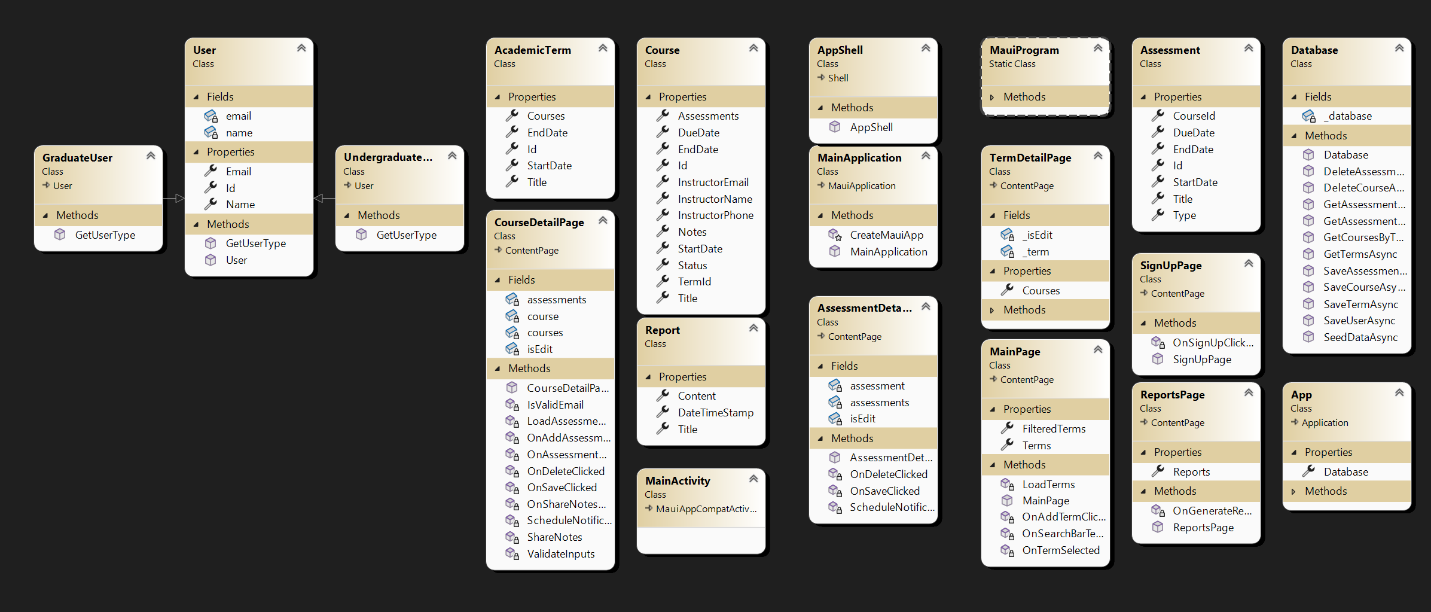


Figure 1: Class diagram

## UI Design

This application's UI design prioritizes simplicity and navigation, allowing users to easily manage courses, terms, and evaluations. With a simple appearance, the app offers a user-friendly experience designed for effective academic management.

A diagram of a wireframe

Description automatically generated

## Unit Test Plan

### Introduction

### Purpose

The purpose of this unit test plan is to outline the testing methodologies used to validate the application's functionality and reliability. The tests were conducted to ensure that each unit of code performs as expected, with any identified issues documented and remediated.

### Overview

The unit tests were designed to cover critical components of the application, particularly focusing on key functions such as managing academic terms, courses, assessments, and user interactions. Each function was tested in isolation to ensure that it behaved as expected under various conditions. In some cases, similar testing methods were applied across different parts of the application to maintain consistency, while unique tests were designed for specific, more complex features. Errors detected during testing were addressed promptly, with adjustments made to the code and retested to ensure resolution.

## Test Plan

### Items

Framework**:** NUnit framework is required for writing and running unit tests.

Development Environment**:** Visual Studio 2022 with .NET MAUI installed.

Database**:** SQLite database for storing academic terms, courses, assessments, and user details.

**Classes:**

AcademicTerm, Course, Assessment, and User classes.

Database class to manage database operations.

**Test Data:**

A test database file path.

Sample data for academic terms and user emails.

### Features

Save Academic Term: This test checks whether an academic term can be successfully saved to the database.

Email Validation: This test verifies that email addresses are properly validated before being stored.

### Deliverables

Test Scripts: NUnit-written unit tests that focus on email address validation and saving academic terms.

Test Results: A report with screenshots of tests that were successful as well as a record of any errors encountered.

### Tasks

Setting Up the Environment: Install and set up NUnit, SQLite, and.NET MAUI.  
Write Tests: Create unit tests for important features like email validation and academic term saving.  
Prepare Data: Make sure that sample data is added to the SQLite database so that it can be tested.  
Run Tests: Use NUnit in Visual Studio to run the tests, and monitor results.  
Analyze Results: Document the results, take screenshots of tests that pass, and identify any issues and their fixes.

### Needs

Software: .NET 8.0, .NET MAUI, SQLite, and NUnit, all properly installed.

Testing Tools: NUnit, configured in Visual Studio.

Source Code: The code and test scripts for the application are available for inspection and modification.  
Test Data: For realistic testing scenarios, use prefilled SQLite data.

### Pass/Fail Criteria

Academic Term Saving Test:

Pass: If the academic term is accurately allocated to the term ID and properly stored to the database, the test passes.

Fail: If the term is not stored or if the term ID is not created upon saving, the test is not successful.

Email Validation Test:

Pass: If the email validation accurately identifies valid email formats and throws an exception for invalid formats, the test is considered successful.  
Fail: If the email validation fails to identify invalid email formats or mistakenly throws an exception for legitimate emails, the test is considered a failure.

## Specifications

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Procedures

Step 1: If it's not already installed, install NUnit and the NUnit3TestAdapter in Visual Studio.

Navigate to Tools > NuGet Package Manager > Manage NuGet Packages for Solution.

Search for NUnit and NUnit3TestAdapter and install them for your test project.

Step 2: Using NUnit, write the unit tests as the example code shows.

Step 3: Use the Test Explorer in Visual Studio to run the tests.

Go to Test > Test Explorer.

Select "Run All" to run all the tests

Step 4: Check the Test Explorer for the test results.

Tests will have green checkmarks next to them if they pass.

The tests that do not pass will have red crosses next to them.

Step 5: If a test fails, revise the code and re-run the tests until all tests pass.

Step 6: Document the results

## Results

Both unit tests succeeded, indicating that the academic term was correctly saved in the database and that the email validation logic works as anticipated. The results can be found in the screenshots below from the Test Explorer.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# User guide for maintenance purposes

## 1. Introduction.

This guide contains complete instructions for installing, running, and maintaining the application. The application is built with .NET MAUI and SQLite, and it includes functionality for managing academic terms, courses, assessments, and creating reports.

## 2. System Requirements

Operating System: Windows 10 or later, macOS 10.15 or later

Development Environment: Visual Studio 2022 (with .NET MAUI workload installed)

.NET SDK: .NET 8.0 or later

Database: SQLite

**Additional Libraries/Packages:**

Microsoft.Maui.Controls

Microsoft.Maui.Controls.Compatibility

Plugin.LocalNotification

sqlite-net-pcl

## 3. Setting Up the Development Environment

### 3.1 Install Visual Studio 2022

Download and install Visual Studio 2022 from the official website.

During installation, ensure the following workloads are selected:

.NET Desktop Development

Mobile Development with .NET

.NET MAUI (Make sure the .NET MAUI workload is installed)

**3.2 Clone the Repository**

Clone the application repository from Gitlab using the command:

git clone https://gitlab.com/sbrahi2/capstone.git

Open the cloned project in Visual Studio.

**3.3 Restore NuGet packages.**

In Visual Studio, launch the Solution Explorer.

Right-click the solution and choose Restore NuGet Packages. This will install all of the project's dependencies.

The application uses SQLite as the database. Ensure that the SQLite database file is correctly configured and accessible.

The database file is typically located in the application's local data folder. You can customize the database path in the Database.cs file.

**3.5 Create the application.**

To build the application, go to Visual Studio's top toolbar and choose the proper target (for example, Android, iOS, or Windows).

Click the Build menu and choose Build Solution, or press Ctrl+Shift+B.

### 3.6 Run the application.

Once the build is complete, you can launch the application by pressing F5 or clicking the Start button.

The application will launch on the emulator of your choice (Android/iOS) or on your native desktop (Windows/macOS).

## 4. Running the Application

### 4.1 First Launch

The program will seed initial data (such as specified words, courses, and exams) into the SQLite database upon first run.

### 4.2 Navigating the Application

Main Page: Displays a list of academic terms. You can add new terms, view existing terms, and search for specific terms.

Term Details Page: Provides comprehensive details about the chosen term and any related courses. Here, you can add, modify, or remove courses.

Course Details Page: Displays comprehensive details about the chosen course, including evaluation results. Notifications can be set, assessments can be added, edited, and deleted.

You can create logs and search through created log files on the reports page.

### 4.3 Generating Reports

Navigate to the Reports Page.

Click on the Generate Logs button to create multiple log files.

Use the search bar to filter and view specific log entries.

## 5. Maintenance and Updates

### 5.1 Updating the Application

To update the application, pull the latest changes from the repository using Git:

git pull origin main

Rebuild the application to apply the updates.

### 5.2 Database Maintenance

Backups: To avoid losing data, make regular backups of the SQLite database file.

Migrations: Make sure the required migrations are implemented if the database schema has changed.

### 5.3 Monitoring and Logs

Logs are created by the application and stored in the local data folder for significant occurrences. Check these logs frequently for any mistakes or problems.

### 5.4 Error Handling

Errors encountered by the application will be recorded. Examine the log files for specifics and address any problems as they arise.

## 6. Troubleshooting

### 6.1 Common Issues

Build Errors: Make sure the project is configured correctly and that all necessary packages have been recovered.

Database Errors: Verify the accessibility of the SQLite database file by checking its path.

### 6.2 Getting Help

If you have any questions about topics not addressed in this book, you can visit the.NET MAUI and SQLite community forums or contact the development team.

# User Guide

### Introduction:

This user guide offers detailed instructions on how to install the application, log in, sign up, and use all of its features. This tutorial guarantees that users can easily navigate the application, providing clear and simple instructions to prevent confusion. To verify correctness and functionality, the instructions have been put through testing.

### Installation and Using the Application

### System Requirements

Before installing the application, ensure that your system meets the following requirements:

Operating System: Windows 10 or later, macOS 10.15 or later

.NET SDK: .NET 8.0 or later

Development Environment: Visual Studio 2022 (with .NET MAUI workload installed)

### Installation Steps

Download the Application:

Download the program installation package from the specified repository or download URL.

Extract the Files:

Extract the downloaded ZIP file to a preferred location on your system.

Install the Application:

Navigate to the extracted folder.

Open Visual Studio 2022 and select the solution file (.sln).

Choose Windows, Android, or iOS as the target platform in Visual Studio, then choose Build > Build Solution or hit Ctrl + Shift + B to build the solution.

### Launch the Application:

Once the build has completed successfully, press F5 to launch the program, or click the Start button in Visual Studio.

After the application launches, the login or sign-up screen will appear.

### Login and Signup

A screenshot of a sign up form

Description automatically generated

### Signup Process

Create a New Account:

Provide a distinct username and a strong password (no fewer than six characters).

Verify the password you have.

Select if you are an undergraduate or graduate student.

In order to create an account, click "Sign Up".

### Using the Application

**Managing Terms and Courses**

**Accessing the Terms Page:**

A listing of academic terms will appear on the home page after you log in.

To narrow down the terms, use the search bar at the top.

A screenshot of a phone

Description automatically generated

#### Adding a New Term:

At the bottom of the terms list, click the "Add Term" button.

In the designated fields, enter the start and end dates of the term.

To include the new term in the list, click "Save Changes".

A screenshot of a cell phone

Description automatically generated

#### Editing a Term:

Click on an existing term to view its details.

Update any information as needed (e.g., term title, dates).

Click "Save Changes" to apply your edits.

#### Adding a Course to a Term:

Click "Add Course" while browsing the details of a term.

Add the name of the course, the start and end dates, the instructor's information, and any notes.

In order to add the course to the term, click "Save Changes".

A screenshot of a cell phone

Description automatically generated

#### Editing or Deleting a Course:

From the term details page, pick a course.

Click "Save Changes" after making any necessary edits to the course details.

Click the "Delete Course" button to remove a course.

**Managing Assessments**

**Adding an Assessment:**

On the page containing the course details, select "Add Assessment".

Enter the title, type (Performance or Objective), and deadline for the evaluation in here.

In order to add the assessment, click "Save Changes".

**A screenshot of a cell phone

Description automatically generated**

#### Editing or Deleting an Assessment:

From the page with the course details, choose an assessment.

Click "Save Changes" after updating the assessment's details.

Click "Delete Assessment" to remove an assessment.

A screenshot of a phone

Description automatically generated

#### Generating Reports

**Accessing the Reports Page:**

A screen shot of a computer

Description automatically generated

From the main page, navigate to the "Reports" section by selecting the "Reports" tab.

#### Generating Logs:

Click on the "Generate Logs" button.

The application will create multiple log files with detailed information about terms, courses, and assessments.

The logs will be displayed on the screen.

A screenshot of a phone

Description automatically generated

#### Searching Logs:

Use the search bar at the top of the reports page to filter logs by keywords.

The search results will be displayed below the search bar.

A screenshot of a phone

Description automatically generated

# Panopto Video Link

Panopto Video Link Name: D424\_sif\_eddine\_brahimi

Link:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=10eb1812-3272-48fc-a877-b1cc006556b0