**D424 – Software Engineering**

**Task 3**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  College Academic Planner |
| **Student Name:** | Anthony Stephenson |

Table of Contents

[Task 3 Design Document 4](#_Toc161518670)

[Application Design and Testing 4](#_Toc161518671)

[Class Design 4](#_Toc161518672)

[UI Design 5](#_Toc161518673)

[Unit Test Plan 6](#_Toc161518674)

[Introduction 6](#_Toc161518675)

[Purpose 6](#_Toc161518676)

[Overview 6](#_Toc161518677)

[Test Plan 7](#_Toc161518678)

[Items 7](#_Toc161518679)

[Features 7](#_Toc161518680)

[Deliverables 8](#_Toc161518681)

[Tasks 8](#_Toc161518682)

[Needs 8](#_Toc161518683)

[Pass/Fail Criteria 8](#_Toc161518684)

[Specifications 9](#_Toc161518685)

[Procedures 10](#_Toc161518686)

[Results 10](#_Toc161518687)

[User Guide 11](#_Toc161518688)

[Introduction 11](#_Toc161518689)

[User Guide for Developers 11](#_Toc161518690)

[User Guide as an End User 13](#_Toc161518691)

**Task 3 Design Document**

# Application Design and Testing

## Class Design

The following image is a class diagram for the College Academic Planner application. The app is mainly built around the term, course, and assessment classes and utilizes the LocalDBService class for interaction with the database and data validation. In the term class, six courses can be related to each term. Within each of these courses are two assessments, which are either performance or objective assessments.

A screenshot of a computer

Description automatically generated

## UI Design

The wireframe shown below was designed before the creation of the application. It provides a visualization of the required elements in each page’s interface. In addition, it facilitated development by providing guidelines for how navigation between pages should be handled. This is represented by the red arrows designating where each interaction with an element will take the user and what will be displayed alongside it.

During the development of the application, minor changes were made to better implement the functionality required of the customer as well as the presentation of information. However, the overall function stayed the same and the customer's needs were still met. The application allows users to navigate between pages that display terms, courses, and assessments. With each of these items, the user is then able to view, create, update, and delete each object.

A screenshot of a mobile application

Description automatically generated

# Unit Test Plan

## Introduction

### Purpose

Throughout the development of the application, there were several types of testing used to ensure the quality was meeting the expectations of the customer. These include functionality testing, acceptance testing, and automated testing. Each played a vital role in the successful completion of the project. Many changes were implemented during the functionality tests, which were ongoing during the project’s life. As for the automated testing, no changes were made because of the tests being performed.

### Overview

Functionality tests were performed during the creation of the application. They were conducted after each implementation of features, interfaces, and interactable elements. Performing these tests was done using an emulation of an Android mobile device and using the application to ensure expected outcomes were given. Occasionally, when debugging, the use of the command line was used to break down lines of code and view what results were given with each interaction of the application. Specific items of the application that were affected by this testing include navigation buttons, CRUD operation functions, and user interface elements.

Automated testing was introduced to the application in place of the functionality tests to save time and increase the accuracy of the tests. The tests created include the following:

* Valid Date Test
* Invalid Date Test
* Valid Email Test
* Invalid Email Test

These automated tests allowed the developers of the application to quickly and efficiently

ensure that the code is still functioning as expected.

Lastly were the acceptance tests. These tests are like that of the functionality test but differ in that they check for alignment between the application’s function and the customer’s requirements. These tests did not result in any changes to the application but provided confidence that all the needs of the customer were satisfied.

## Test Plan

### Items

* Visual Studio 2022 for the use of the .NET MAUI framework.
* SQLite NuGet package for interaction with the application’s local database.
* XUnit NuGet package is used to run the automated unit tests.
* Source code can be downloaded from the GitLab repository to interact with the application and use the unit test scripts that are included in the repository.

### Features

* Valid Date Test
  + When this test is run, it calls the IsValidDate method and passes a start and end date to simulate the creation of terms, courses, and assessments.
* Invalid Date Test
  + Like the valid date test, this test expects the IsValidDate method to return a false value when run.
* Valid Email Test
  + Ensures that when a valid email address is passed to the IsValidEmail method it returns true.
* Invalid Email Test
  + This unit test checks that any form of an invalid email will return a value of false.

### Deliverables

For each of the automated unit tests, the expected deliverable would be confirmation of every task in the test explorer showing a pass. Every test in the XUnit test scripts is expected to return either a true or a false boolean value.

### Tasks

* Retrieve the repository from Gitlab.
* Open the solution within Visual Studio.
* Ensure the XUnit NuGet package is installed, and all the tests are in the test explorer.
* Execute all the unit tests and ensure that each of the tests passes by checking the test explorer window.

### Needs

To run these automated tests, there are a few requirements needed of the environment. First, the XUnit package needs to be installed and integrated with the College Academic Planner project. It should be verified that XUnit is running version 2.4.2 or later and targets the same framework version of the project, which is .NET 8.0.

### Pass/Fail Criteria

* Valid Date Test
  + This test should pass when the start date is before or equal to the end date. Being returned a true value will result in a pass, and a false value will fail.
* Invalid Date Test
  + The test will provide parameters that should always return false and ensure users are unable to create terms, courses, and assessments that end before they start. Being returned a false value will result in a pass, and a true value will fail.
* Valid Email Test
  + This test is specific to the creation and updates of courses and will pass a valid email address string for its parameter. Being returned a true value will result in a pass, and a false value will fail.
* Invalid Email Test
  + It calls the IsValidEmail method multiple times each with different invalid values. Being returned a false value for any of the given method calls will result in a pass, and a true value will fail.

## Specifications

The following image provides a view of some code that can be found within the automated unit tests.

A computer screen shot of a program

Description automatically generated

## Procedures

1. Identify features and functionalities of the application that will require tests.
2. Recognize the expected input and outputs of the tests.
3. Implement unit tests that verify the previously mentioned test requirements.
4. Execute each test and record all findings from them.
5. Upon failures, identify the source of the failure and remediate that portion of code before beginning retests.

## Results

XUnit’s package allows users to view a test explorer before, during, and after running unit tests. The following screenshot is the expected outcome from the execution of the unit tests.

A screenshot of a computer

Description automatically generated

# User Guide

## Introduction

This user guide is meant to provide guidelines for both users and future maintainers of the application. These steps are laid out chronologically and supply the reader with accurate information on utilizing the College Academic Planner application.

## User Guide for Developers

*IDE Installation*

* Navigate to [Visual Studio’s](https://visualstudio.microsoft.com/downloads/) (2022) download page.
* Select the appropriate version for the use case.

A screenshot of a computer

Description automatically generated

* Continue with the installation instructions and select .NET framework and MAUI.

*Cloning a repository*

* After installation, open Visual Studio and select “Clone a repository.”
* Insert the following URL into the Repository location entry field.
  + <https://gitlab.com/wgu-gitlab-environment/student-repos/aste816/d424-software-engineering-capstone.git>
* Select the “Create” button and wait for the files to download.

A screenshot of a computer

Description automatically generated

* Open your preferred CLI and navigate to the location where the repository was cloned.
* Switch to the “working” branch. The commands should look like the following image:

A screenshot of a computer program

Description automatically generated

* All files should now be accessible within the solution explorer.

*Running Android Emulator*

* With the project open in Visual Studio, select the following options:
  + Tools 🡪 Android 🡪 Android Device Manager 🡪 New
* Create an emulator running Android 14.0 – API 34.

A screenshot of a computer

Description automatically generated

* To run the application in the emulator, select the following options:
  + Debug Dropdown 🡪 Android Emulators 🡪 The Created Device
* Now, when debugging, the application will run on the created emulator.

## User Guide as an End User

*Home Page*

* Upon launching the application, four buttons are displayed for navigation.

A screenshot of a college planner

Description automatically generated

* Terms contain up to six courses and courses contain two assessments. It is required to create your course load information in this order:
  + Assessments 🡪 Courses 🡪 Terms

*Terms*

* When the term button is selected, a list of all created terms will be displayed.

A white rectangular object with black border

Description automatically generated

* Selecting a term in the list will create a popup asking to view, edit, or delete it.

A white background with black text

Description automatically generated

* + Choosing to view will display a detailed page of the selected term.
  + Choosing to edit will display an editable page to change the details of the term.

A screenshot of a cell phone

Description automatically generated

* + Choosing to delete will remove the term from the list.
* A green add button is displayed at the bottom of the page. When choosing this, a term creation page will be displayed.
* To remove courses from the term, select the red X button next to the desired course.
* NOTE: When adding or editing a term, it is required that the term name is not empty, and the start date is before or equal to the end date. All other choices are optional.

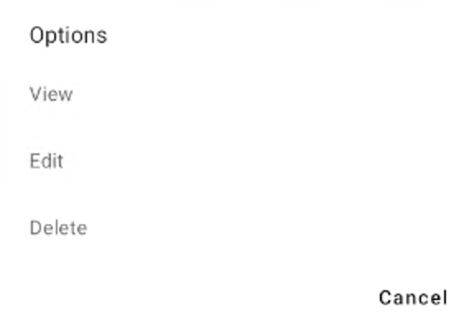
*Courses*

* When the course button is selected, a list of all created courses will be displayed.

A white background with black dots

Description automatically generated

* Selecting a course in the list will create a popup asking to view, edit, or delete it.



* + Choosing to view will display a detailed page of the selected course. At the bottom of this page, the share button can be pressed to export the course notes.
  + Choosing to edit will display an editable page to change the details of the course.

A screenshot of a cell phone application

Description automatically generated

* + Choosing to delete will remove the course from the list.
* Notifications can be enabled for courses. If chosen, a notification will be displayed when the app is launched, and the respective course either starts or ends on the current day.
* A green add button is displayed at the bottom of the page. When choosing this, a course creation page will be displayed.
* NOTE: When adding or editing a course, it is required that all entry fields except the notes are filled out, a status is selected, and the start date is before or equal to the end date.

*Assessments*

* When the assessment button is selected, a list of all created assessments will be displayed.



* Assessments are either a performance assessment, meaning the evaluation is a project, or an object assessment, which means the evaluation is an exam or test.
* Selecting an assessment in the list will create a popup asking to view, edit, or delete it.

A white background with black text

Description automatically generated

* + Choosing to view will display a detailed page of the selected assessment.
  + Choosing to edit will display an editable page to change the details of the assessment.

A screenshot of a test

Description automatically generated

* + Choosing to delete will remove the assessment from the list.
* Notifications can be enabled for assessments. If chosen, a notification will be displayed when the app is launched, and the respective assessment either starts or ends on the current day.
* A green add button is displayed at the bottom of the page. When choosing this, an assessment creation page will be displayed.
* NOTE: When adding or editing an assessment, it is required that the assessment name is not empty, and the start date is before or equal to the end date.

*Search*

* When the search button is selected, the search page will be displayed.

A green and white object

Description automatically generated with medium confidence

* The top of the page has a search bar to filter all terms, courses, and assessments by the entered text.
* A red X on the right-hand side will appear when typing and can be used to clear the text in the search box.
* Selecting the green button with an arrow will run the search and display a results page.
  + Leaving the search field blank and running the search will result in an error.

A screenshot of a cell phone

Description automatically generated

* The search results page divides all the results by a header stating what will be displayed below. If no results are found, that will be stated below the header.