

## COMP.6215 – GUI Programming

### Assessment 02

Part 1 Due Date: October 14<sup>th</sup> 2pm (in class)

Part 2 Due Date: October 31<sup>th</sup> 5pm

Assessment	Type of Assessment	Due Date & Time	Marks	Weighting
Assign #2 (TGA)	Coding Assignment	31 <sup>st</sup> October 2019 5pm	64 Marks	50%
<b>Learning outcomes assessed:</b>				
<ol style="list-style-type: none"><li>1. Demonstrate competency in the use of a GUI programming language and environment.</li><li>2. Demonstrate the use of a GUI language for the creation of an effective front end.</li><li>3. Demonstrate the testing, debugging and documentation of GUI programs.</li></ol>				

## Reminder

The assignment must be a product of your own work, except for the use of resources supplied with the course, discussions conducted with the lecturers, and other assistance shown as acceptable in the section *Assistance to Other Students* below.

## Assistance to other students

Students themselves can be excellent resources to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are acceptable in an assessment.

### Beneficial Assistance

- Study Groups
- Discussion
- Sharing reading material

### Unacceptable Assistance

- Working together on one copy of the assessment and submitting it as your own work
- Giving another student your work
- Copying someone else's work
- Changing or correcting another student's work
- Copying from books, the Internet etc. and submitting it as your own work

## Helpful hints

Although you don't need to submit the diagram designs, it is strongly recommended that you make use of flow diagrams and UML diagrams to design your application before you start coding. This will save you a lot of time.

## Assessment outcome

Upon successful completion of this assessment you will have a good understanding on how to separate UI logic from the business logic of the application. You will also have a good understanding of the solid principles.

## Individual Assessment

This assessment is an individual assessment. You are to create a product of your own, but are able to work with other students as per the guidelines above.

## Submission Type

You will receive a link to GitHub Classroom where you can submit your assignment. You will need to put all your apps inside this single repository.

## Deliverables

You are to create 2 applications that make use of pulling data from multiple web API's and display its content in a well designed GUI application.

For each of the following applications you are to hand in the following:

- 1 x a small report (no more than 500 words) explaining the choice of your framework and back it up with some references to media and or blog articles that support your choice.
- 1 x a readme file with the details of the API's – you will need to supply any API keys so that the teacher does not have to create accounts for the services.
- 1 x an application that displays the result of the API in a good looking application.

# Task 1

You are to create an application of your choosing made in any platform owned by Microsoft. Some of these have been discussed in this class – others have not.

## Part 1

You are to rationalise the choice of the framework and back it up with media articles as to why you think it is a good choice. These articles will need to be referenced.

You are to create a good layout of your application that matches the API of your choosing – you will need to wireframe your idea, but only for one size (desktop app @ 800 x 600)

## Part 2

Your application must make use of at least 2 API's

Your app is only to consume an existing API – do not create your own API

You will need to think about good user experience – show things like indicators and alerts to make sure the user knows something is happening instead of the application freezing.

Your application must use clean coding standards, so think about the solid principles and code organisation.

Your application should be based on one (or 2) of the ideas below

# Task 2

Task 2 is the same as task 1 as far as criteria go, but you have the choice to make a completely different application or the same application using a different framework.

Since it is a different application, you will need to redesign it from scratch (even if it uses the same resources, because different framework allow for different options and looks)

# Topics

## Location and Weather Application

For this option you are to create an application that displays the weather of a selected place by the user and some photo's of the location that they type in.

### Relevant API's

- <https://openweathermap.org>
- <https://unsplash.com/developers>

## User profile with Map

For this application you are to create a list of 10 users and display their details – your application needs to have an address book type and also display a map to show their location.

### Relevant API's

- <https://randomuser.me>
- <https://github.com/judero01col/GMap.NET>

## Photo Gallery

For this application you are to create a photo search application that gives the user the option to search from the 3 API's below and show the results to the user

### Relevant API's

- <https://apidocs.imgur.com/?version=latest>
- <https://pixabay.com>
- <https://unsplash.com/developers>

## Marking Guide

Criteria	NA	M	G	VG
Acceptable Rational for Framework choice (no less than 200 words) (no more than 500 words) <b>(due 14<sup>th</sup> of October during class)</b>	0	1	2	4
Wireframe the application design <b>(due 14<sup>th</sup> of October during class)</b>	0	1	2	4
Application uses the API's listed	0	1	2	4
Good user experience for the app (easy to understand and use and clear UX indicators)	0	1	2	4
Clean coding – use of SOLID principles	0	1	2	4
Frequent git commits and use of branches	0	1	2	4
Application runs and does not crash	0	1	2	4
Clear folder structure – is your app setup well, documentation easy to read and find, all the information that we need to run the app (i.e. API keys and information)	0	1	2	4

Criteria	NA	M	G	VG
Acceptable Rational for Framework choice (no less than 200 words) (no more than 500 words) <b>(due 14<sup>th</sup> of October during class)</b>	0	1	2	4
Wireframe the application design <b>(due 14<sup>th</sup> of October during class)</b>	0	1	2	4
Application uses the API's listed	0	1	2	4
Good user experience for the app (easy to understand and use and clear UX indicators)	0	1	2	4
Clean coding – use of SOLID principles	0	1	2	4
Frequent git commits and use of branches	0	1	2	4
Application runs and does not crash	0	1	2	4
Clear folder structure – is your app setup well, documentation easy to read and find, all the information that we need to run the app (i.e. API keys and information)	0	1	2	4