

## PROG2005 – Programming Mobile Systems

# Assignment 1 (T1 2022)

This assignment is due on **Monday, 21 November 2022, 11:59 pm**, and is worth **40%** of your overall mark. You should work on this assignment **individually** (This is NOT a group assignment). You must submit your source code as one ZIP archive containing both parts of the assignment to the Assignment 1 submission link on the unit site. If the file size is too large to upload to the submission link, please send the ZIP file via e-mail or a method specified by your tutor. Note that marks will be deducted for poorly structured or uncommented code. All source code files submitted must include title comments that at least identify the author and the assignment part. The separate parts of the assignment are to be submitted in separate subdirectories (e.g. Part1, Part2 etc.) – submissions that ignore this instruction and leave all assignment files in one directory will be penalised.

Please note that this assignment will be due after the weekly modules have covered the last of the required materials. Do not leave this assignment to the last minute – you can start on it while concepts are fresh in your mind. You may start Part 1 before you have studied all of the required materials. If you require an extension, you must apply to your tutor before the due date to be considered. Unless an extension is approved, there is a late submission penalty.

This assignment requires you to develop an application using TypeScript for Part 1 and Angular for Part 2 to demonstrate your knowledge of the TypeScript language and Angular framework. Your apps should be stand-alone apps without server contact for data. The app will not save data after the browser window is closed, but you will need to maintain a JavaScript structure (e.g. JavaScript object, Class, Interface) so that changes made remain while the browser window is open.

Part 2 can be an expansion of Part 1 though you can start again from scratch if you wish. It is recommended that you save your Part 1 before you start modifying it for Part 2. For lesser marks, you can divide your assignment into several smaller HTML pages to demonstrate the different aspects without having everything working together. This is not the preferred method.

### Part 1 – Basic TypeScript App (18 marks)

The aim is to develop a simple *TypeScript* app for a medical clinic to maintain a small database of patient information while the browser window is open. You can initialise your app with hard-coded data or start with an empty data structure. Types are important in your code; however, there is no need to type everything, and the general applicability of typing is as you have seen in Module 2 of the unit. For example, there is no need to define a function type in a function definition as TypeScript, and the reader can easily deduce the type from the function definition. You will have to type parameters and return values, though. Marks may be deducted for too few type definitions, i.e. writing JavaScript instead of TypeScript.

The Patient data has the following fields:

- Patient ID \*
- First name \*
- Last name \*
- Date of birth \*
- Gender (“Female”, “Male”, “Unspecified”) \*
- Primary insurance (“Medicare”, “Private Health Insurance”) \*
- Address \*
- Contact number \*
- Next of kin

The data has the following requirements (1 mark):

- Each record must have a value for all fields (\*). Only the ‘Next of kin’ field can be blank.
- The ‘Gender’ and ‘Primary insurance’ fields value must be one of the options shown.
- A Patient ID can only be entered and saved once (it must be unique for each patient).

Your database would probably be a TypeScript array of JavaScript objects; each object has properties corresponding to the above fields. You can use another data structure if you wish (Such as class or Interface), but types will be necessary to ensure the integrity of the data (1 mark).

Your app should implement the following:

- A way to edit and update patient data (2 marks). Remember, you do not have to permanently save the data between browser sessions, but you do have to save it while the app is active. HTML form widgets are the easiest here.
- A way to add new patient information (2 marks). Consider re-using the edit form above, and it will require careful thought so as not to confuse the user.
- A way to delete an individual patient with a “Do you really want to...” interaction to reduce accidental deletions (2 marks).
- A way to search the patient by the Patient ID field (2 marks). This will probably be done with an HTML list, but you may use other techniques (e.g. implement a search text field – harder).
- A way to display the list of all patients added to the data structure (2 marks).
- A way to mark patients as an emergency and display the list of all emergency patients (2 marks).

Your app should also:

- Provide an interface equally viewable by narrow and wider mobile screens (it will be tested). Note that this is just a request to keep your app simple and displayable on narrow screens as well as possibly wider screens. You can put links on a long page to navigate your app on narrow screens. This will be important when we get to mobile device apps later in the unit (1 mark).
- Use interactive features (e.g. use innerHTML assignments) to improve user experience (not alert() calls!). You are expected to do error messages where necessary (2 marks).
- Use of CSS styles (1 mark). Just use them; there are no marks for artistic design. We are looking for the ability to use styles.

## **Part 2 – Basic Angular App (22 marks)**

Modify the app from part 1 to become a multipage Angular app. For example, place the add records on a separate page to the search facility. You can use as many pages as you like but don't forget to use an error page (for malformed URLs), an Information page, e.g. about the app and have your name as an app developer. A Help page showing how to use the app would be nice as well.

Marks are allocated as follows:

1. Having all options implemented for adding, deleting, displaying (including favourites), searching and editing patients (4 mark).
2. The app should meet all data requirements stated in Part 1 (1 mark).
3. Having a working Angular app with appropriate Angular coding (3 marks). Minimally, the app will have one component and one module, but you will need more components to implement the following requirements.
4. Using Angular forms (3 marks). This is separate from the multipage facility. If you just add Angular code to the one-page TypeScript app from Part 1, you have not used Angular forms. There will be no marks for only HTML form elements.
5. The multipage facility using the Angular router (3 marks). This includes the navigation between pages, error indicators and no broken links. Your app should have at least four (4) pages.
6. Add at least one help button on each page that shows appropriate help messages. You should make interaction with the help facility as easy as possible. Also, add helpful information, including error messages, when data entry is incorrect (3 marks).

Use of CSS styles (2 marks). Just use them; there are no marks for artistic design. We are looking for the ability to use styles. Feel free to use styles you find in the Angular documentation or other online sources (a reference to the original designer should be made in your code if appropriate). Your app should be user-friendly and less

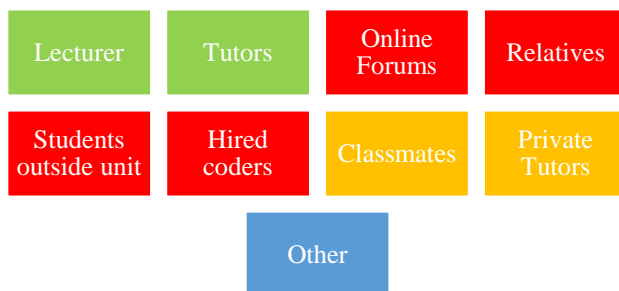
complex and needs to validate all inputs (e.g., input fields that need numbers only should not accept letters) (3 marks).

### Getting Help:

This assignment, which is to be completed individually, is your chance to gain an understanding of the fundamental concepts of *TypeScript* and *Angular* on which later learning will be based. It is important that you master these concepts yourself.

Since you are mastering fundamental skills, you are permitted to work from the examples in the MySCU site or study guide, but you must acknowledge assistance from other textbooks, classmates, or online resources. In particular, you must not use online material or help from others, as this would prevent you from mastering these concepts.

This diagram will help you understand where you can get help:



**Encouraged**  
**Not acceptable**

**Attribution Required**  
**Ask your tutor**

Be aware that if you do get help from one of the red sources, you will be reported for academic misconduct, which may have serious penalties. Please visit the following link for the guidelines: <https://bit.ly/scuAcadMisconduct>