| **Section 1 – Assessment Task Overview and Description** |
| --- |

# Student / Class Details

**Full name:** Joey Sutton Preece

**Student ID:** 102568393

**Teacher:** Lachlan Van Der Velden

**Date / Time started:** 5/11/2019 12:00 AM

Please read the whole assessment before starting, including the Summary of Evidence and Items / Criteria. Any missing files / an incomplete submission or a submission submitted after the due date **will not be accepted**.

By checking the box below, you agree that penalties exist for plagiarized work, that all work submitted is your own and that you have read the above highlighted statement. Please refer to the bottom of the document for more information on plagiarism.

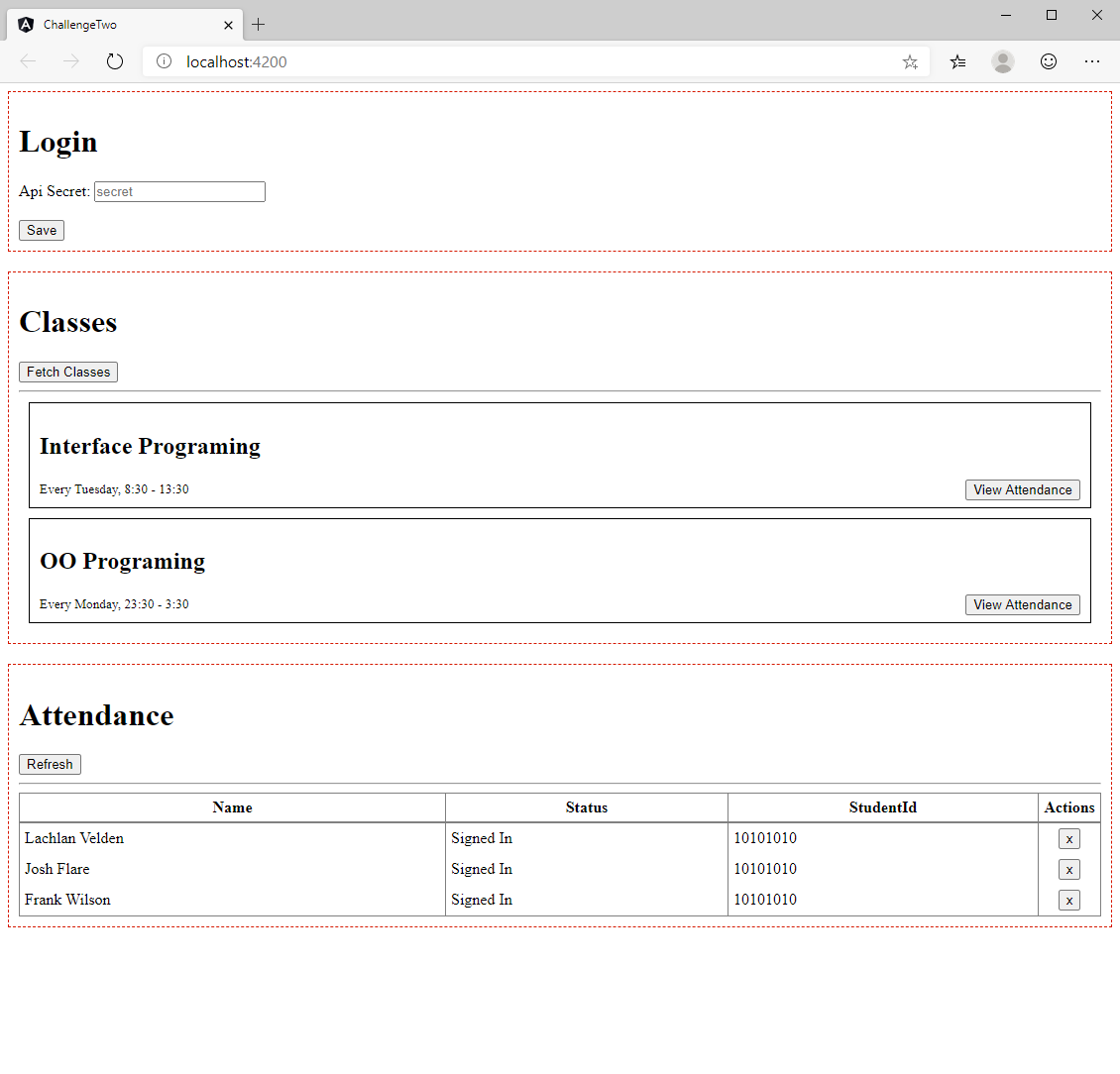
I confirm that all work is my own and have read the entire assessment.

Please complete the following tasks **in class and in your own time**. Remember that this assessment is **Individual**, however you will be using you groups API for attendance and class data. Please ensure that all changes made at home and in class are **committed directly after** you have finished for that period so that we can keep track of the progress made on your assessments.

# Tasks

1. Create a new **private** Github Repository called ‘InterfaceProgramming\_Challenge2’ and add ‘LachlanVelden\_Swin’ as a collaborator.  
   Click or tap here to enter text.
2. Ensure to commit regularly during your development of this interface
3. **Keep track of the time** that you have worked on this assessment. Please log this in a file called ‘WorkLog.MD’ and commit this with your code. Eg: ‘worked 8:30-12:30 17/11/2019’ on a line.
4. **Keep track of online resources** you use during the next task. Please save a link to these to a file called ‘References.MD’ and ensure that this is being committed with your code.
5. Complete all non-optional Functional Requirements (provided on the next page)

# Functional Requirements

[](https://i.imgur.com/1HOi84C.png)

1. Implementation of the API endpoints defined at <https://dev.azure.com/swinburneswd/_git/Sign-In-App-Public?path=%2Fendpoints.md&version=GBmaster> in the “for Angular Web Interface”. These should already be implemented on your own API from the OOP class by the end of this assessment
2. At least one service that stores all requested Information
3. At least 3 components connected to the above service.
   1. Login
      1. Enter your API Secret Here
   2. Classes
      1. Shows all classes from the server
      2. Allows you to select a class for attendance
   3. Attendance
      1. Shows all recorded attendance for the selected class
4. Some more attractive styling than the wireframe provided above. This is **optional** however will make your application look loads better!
   1. Flat color buttons?
   2. Card style layout? (where the red boxes would have a drop shadow)
   3. Better laid out login?
   4. Delete button is instead a greyed out x (no box)
5. Routing between sections. Eg Login, Classes and Attendance with route parameters. **Optional**

| **Section 2 – Assessment Task Submission Information** | |
| --- | --- |
| **Submission Details** | **Due date: 4 Hours from Start Date** |
| 1. The assessment task must be submitted via Canvas or directly to the teacher with an assessment cover sheet. 2. Ensure to fill out:    * your name    * student ID    * your teacher’s name    * the time the assessment was started    * plagiarism and read conformation 3. Submissions received after the submission date must be approved by your teacher. 4. Incomplete submissions will not be accepted. |

| **Summary of Evidence to be Submitted / Included** |
| --- |
| * This document will all forms filled out |
| * A Github Repository with your teacher added as a collaborator. |
| * References provided within Github Repository |
| * A Functional Angular Application Provided with Github Repository |
| * WorkLogs provided within the Github Repository |
| The task will be assessed as satisfactory when all of the required evidence listed has been satisfactorily demonstrated.  \* If applicable, for graded units, the task must be satisfactorily completed before marks will be allocated. Refer to your unit outline for more information. |

| Section 3 – Assessment Task Criteria and Outcome | |
| --- | --- |
| *All items/criteria must be demonstrated satisfactorily to achieve this task. The items/criteria for this activity will be assessed as S – Satisfactory or US – Unsatisfactory.* | |
| Items/criteria | |
| 1. | At least 3 Angular Components developed |
| 2. | At least 1 Angular Service developed and implemented into components |
| 3. | A building and functional Angular Application |
| 4. | Clearly defined references |
| 5. | HTTP GET requests being made to an API |
| 6. | All GET requests containing an API Secret |

|  |  |
| --- | --- |
| **Section 4 – General Assessment Information** | |
| **Decision Making Rules** | Each activity in the assessment task must be satisfactorily completed for the task to be assessed as satisfactory.  Every task must be satisfactorily completed to be assessed as competent in the unit.  *\* For graded units, competence must be demonstrated before a mark can be given.* |
| **Plagiarism** | There are serious penalties for plagiarism that may include repeating a new assessment task or being withdrawn for the unit / course.  Students must ensure that all assessments are their own work (or group work and clearly noted as such).  Please refer to [www.swinburne.edu.au/corporate/registrar/plagiarism/index.html](http://www.swinburne.edu.au/corporate/registrar/plagiarism/index.html) |
| **Reasonable Adjustment** | Students may request reasonable adjustment for assessment tasks.  Reasonable adjustment usually involves varying:   * the processes for conducting the assessment (eg: allowing additional time, varying the venue) * the evidence gathering techniques (eg: oral rather than written questioning, use of a scribe, modifications to equipment)   However, the evidence collected must allow the student to demonstrate all requirements of the unit.  If you have any other issue that may impact your ability to undertake the assessment, please discuss with your teacher. |
| **Re-submission** *(where tasks are not satisfactorily completed)* | Assessment tasks that are not satisfactory can be resubmitted up until the end of the unit as scheduled on the Unit Outline. The timing on this may depend on the equipment required for this assessment task.  Resubmissions received after the scheduled unit end date may not be accepted unless approved by the teacher prior to the end date.  Note: Assessment tasks submitted for the first time after the unit end date as scheduled in the Unit Outline will not be assessed and the student should re-enrol into the unit. |
| **Special consideration** | Students may apply for Special Consideration where personal circumstances have adversely affected their task result or ability to undertake an assessment. A Special Consideration form can be completed prior to, but no later than 3 days after, the date of assessment and submitted to the relevant Manager. |
| **Work Health & Safety** | Activities may require the use of equipment or participation in group exercises. If the teacher identifies any unsafe activity or potentially dangerous situations, the teacher can stop the assessment at any time. |