Assignment Guidance and Front Sheet

This sheet is to be populated by the Module Tutor, checked by the Programme Team, and uploaded to Moodle for students to fill in their ID and submit with their assessment.

Student ID or IDs for group work	Student fill in own ID and attach document for submission

Module Title & Code	*WM240 – Cyber Context of Software Engineering.*
Module Owner	*Dr. Tony Green*
Module Tutor	*Dr. Tony Green*
Module Marker	*Dr. Tony Green*
Assesment type	*Group work with strong individual component*
Date Set	*Term 1*
Submission Date (excluding extensions)	*10th May, noon 2021;* *Team presentations w/c: 24th May (tbc)*
Marks return date (excluding extensions)	*7th June 2021*
Weighting of mark	*100*

Assessment Detail	This assessment relates to the case study on the security of sensor data acquisition and sensor control that has been the focus of this module throughout the year. Early in the module, students were provided with a scenario along with a conceptual architecture for a prototype environmental monitoring and control system. The architecture identifies 6 components in support of the acquisition, aggregation, transmission, storage, processing, and display of environmental data such as temperature, humidity, light etc. Additionally, there is an option to develop a component that is intended to deliver some physical security monitoring capability. Full details of the scenario and along with guidance relating to the expected deliverables are available on Moodle. Working in groups of 6, each member is required to work on the full software engineering lifecycle of their chosen component from requirements management, design, construction through to testing and finally, delivery of a working prototype to the module tutor for whole-system testing. Each student is fully responsible for building the prototype component and documenting all stages of the engineering process. The progress of individual students (as well as that of the group as a whole) will be tracked throughout the duration of the module. Tracking requires students to submit drafts for the project requirements and design stages at pre-determined times. These early submissions are intended to serve as the initial development plan and

	while the content of these submissions does not contribute to the final mark, it is expected that there is no significant and unaccountable divergence from the sections that will appear in the final submission. Tracking will also be carried out throughout the construction and testing process. All students have been told that they must demonstrate progress by recording clearly defined and appropriately sized tasks on a weekly or bi-weekly basis. For the final submission, students are required to deliver a complete written specification for the component for which they are responsible for. A template for the submission has been available on Moodle since the end of Term 1 and is also provided below in a more usable form. In addition to the written submission, students must work within their group to ensure that all source code, installation scripts, wiring diagrams and other material needed for the automated reconstruction of the entire system is made available for independent assessment. This will involve each team providing (to the module tutor) access to the code repository on or before the submission deadline. It is the responsibility of the group to ensure the tutor has sufficient access to be able to clone the repository. Following the final individual submission of the component specification document, a group presentation will be held. Each student is required to attend (date and time to be confirmed) and will be expected to offer answers to detailed questions about their own component as well as general questions about the overall group solution. As part of the project design and planning activities, each team must develop mitigation strategies in the case where the failure or absence of one or more components in the submitted solution does not impact the final presentation or the overall project.
Additional details	*The Prototype design document SHOULD aim to be around 6000-8000 words but MUST be less than 12,000 words excluding contents section, tables and appendices. A word count fewer than 6000 words may be acceptable providing all of the requirements of the document are met. A detailed break down of the document sections is provided below supercedes any previous versions that may exist.*
Module learning outcomes (numbered)	1.Apply cyber security good practice to various phases of the software engineering lifecycle. 2.Participate in a team engaged in a project at some phase of the software engineering lifecycle.

<table>
<tr><td>Learning outcomes assessed in this assessment (numbered)</td><td>1.Apply cyber security good practice to various phases of the software engineering lifecycle.

2.Participate in a team engaged in a project at some phase of the software engineering lifecycle.</td></tr>
<tr><td>Marking guidelines</td><td>The assessment accounts for 100% of the module mark. Marks are allocated for the group contribution (30%) and the individual component (70%).

The group component is assessed during group vivas to be arranged following the submission of the individual report. The duration of a vivas will be approximately 1 hour but no longer than 90 minutes.

<table>
<tr><td>Criteria – Group Component</td><td>Mark</td></tr>
<tr><td>Group presentation and overall performance.</td><td>10</td></tr>
<tr><td>Ability to demonstrate effectiveness of group activities and communication throughout the project.</td><td>10</td></tr>
<tr><td>Ease of deployment of fully operational and security validated solution that meets the general project requirements set out at the start of the module.</td><td>10</td></tr>
<tr><td>TOTAL</td><td>30%</td></tr>
</table>

<table>
<tr><td>Criteria – Individual Component</td><td>Mark</td></tr>
<tr><td>Individual contribution to group activities throughout the module.</td><td>10</td></tr>
<tr><td>Presentation, interpretation, and prioritisation of requirements.</td><td>5</td></tr>
<tr><td>Component design.</td><td>15</td></tr>
<tr><td>Effectiveness of delivered component in meeting the requirements as defined or otherwise implied in the project brief.</td><td>15</td></tr>
<tr><td>Effectiveness of Test Plans and installation instructions.</td><td>15</td></tr>
<tr><td>Style, presentation, and completeness of the report.</td><td>10</td></tr>
<tr><td>TOTAL</td><td>70%</td></tr>
</table>
</td></tr>
<tr><td>Submission guidance</td><td>A single PDF report submitted on time to Tabula. The report will comprise all sections as described below.

The overall solution should be made available to the tutor via a code repository - the details of which are agreed and setup in advance of the project submission deadline. Any commits made to the repository after the official deadline has passed shall be ignored.</td></tr>
</table>

Academic Guidance	All details required for successful completion of the module have been discussed in lectures and tutorials. The material is available on Moodle.
Resubmission details	The University policy is that students should be given the opportunity to remedy any failure at the earliest opportunity. What that "earliest opportunity" means in terms of timing and other arrangements is different depending on Programme (i.e. Undergraduate, Full Time Masters, Part Time Postgraduate, or Overseas). Students are advised to consult your Programme Team or intranet for clarity.
Late submission details	If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). "Late" means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late.

Submission Guidance:

Individual Component Prototype

Each team member is required to make available via a code repository, all code, test scripts, installation scripts, configuration files needed to be able to launch a fully working instance of the component under automation. Students might find it prudent to submit a single, zip file to Tabula, so than in the unlikely event the code repository is unavailable, the code may still be assessed.

It is the responsibility of each team to ensure their solution can be installed and tested independently on a set of networked, vanilla Ubuntu 20.04 (default Desktop Installation) and Raspberry PI 3B+ target machines. Consequently, each team is responsible for ensuring that details of the Raspberry PI operating system version, network settings, key installation scripts (and execution sequence) are provided to the module tutor. Should the solution require the installation of third-party components, tools, or libraries (Ubuntu or Raspberry PI), details must be made available to the module tutor - all in advance of the submission date.

Individual Component Specification Document

Each team member is required to submit to Tabula their component specification document. In the table below, Column 1 shows the section headers that must be included in the report. Columns 2 and 3 are provided for guidance only. Note that Column 3 is intended to provide guidance on the relative size of one or a group of sections and does not necessarily relate to how marks are allocated.

In the following table, the terms "production ready", "fully deployed" or "fully deployable" refer to the component as it might exist in at an actual (hypothetical) site as opposed to the

prototype version that you will deliver for this module. A gap between the capabilities of your prototype component and the expected capabilities of a production ready is expected.

While there is no requirement to create a production ready component, there is, however a requirement to understand and articulate in your report any shortcomings of your prototype compared with what might reasonably be envisaged as a production-ready version of the component (including but not limited to functionality and the limitations of the testing environment).

Sections (Mandatory)	**Suggested Content**	**Indicative Section Weighting (Word Count)**
Title and document control information	Student ID, team name, word count, table of contents, table of figures.	This section is excluded from word count.
Executive Summary	Brief overview and also, the final state of your deliverable and the gap between your deliverable and the target, production ready state (e.g. does it meet none/some/all objectives?).	15%
Background, Introduction and Aims	Write about your component only. Exclude other components that comprise the fully deployed (non-prototype) solution.	
Project Requirements	Summary description of the top critical requirements (either functional or non-functional) for your component. Choose between 3 and 5 requirements that would apply in a fully deployable version of the system.	15%
Conceptual Design	Very high-level conceptual view of a production ready, fully deployable version of your component with key modules and summary of operation.	35%
Project Constraints and Threat Model	Details should relate to a production-ready version of your component only.	
Options analysis	Comparison of the pros and cons of chosen solution with **TWO** alternative options.	
Logical Design (Prototype)	More detailed description of your prototype component and its interfaces including UML/ERD diagrams as required.	
Test Plan (Prototype)	Description of unit / functional / system tests – these must cover the key requirements.	25%
Further Development	Description of the gaps between the features your prototype and what might reasonably be conceived to be the features of a production-ready version of your component.	10%
Appendices	Key team meetings (date/time/meeting title/summary outcome). Summary logs of Git commits (date/time/branch/code file(s)/message). References (if any).	This section is excluded from word count.