Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate 'your' section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

To be completed by the student(s) prior to final submission:

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

Student ID or IDs for group work	**e.g. 1234567**

To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:

Date set	15/10/21
Submission date (excluding extensions)	25/5/22, 12 noon, report submitted to Tabula.
Submission guidance	The Requirements Specification, Logical Design and Test Plans are to be submitted as a single PDF document.
Marks return date (excluding extensions)	22/6/22
Late submission policy	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.
Resubmission policy	If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.

To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:

Module title & code	WM240-24 Cyber Context of Software Engineering
Module owner	Tony Green
Module tutor	see above
Assessment type	essay and group work
Weighting of mark	100%

Assessment brief
Refer to the "Assignment Brief" below:

Word count	The entire report should be in the region of 8000-10,000 words excluding tables, figures, and references.
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.
Learning outcomes assessed in this assessment (numbered)	1 & 2
Marking guidelines	See below
Academic guidance resources	Guidance for successfully completing the assessment will be provided throughout the module.

Assessment Brief

Throughout module WM240, you encounter a range of important software engineering practices while designing and developing a simulation tool that models a complex and safety critical system.

You will undertake this assessment as a member of a team.

This assessment examines your team's ability to thoroughly investigate a highly complex scenario and derive a set of requirements that will drive design, build, and test activities for a simulator program (described below).

Submission Details

There are two main elements to the submission; an individual report and a team presentation.

Individual Component:

You will submit a single PDF report comprising the sections described in the following table.

Table 1. Report Sections

Section	Description
Software Requirements Specification (SRS)	This section covers the functional and non-functional requirements obtained directly from the scenario (below). This section should also highlight any potential significant omissions or conflicting requirements that may be hidden in the scenario as well as any other requirements you deem important.
Logical Design (LD)	This section describes the logical design of the team's proposed solution that addresses the requirements identified in the SRS. Along with appropriate UML diagrams and threat models, the LD should also highlight, essential data structures and event handling as well as how messaging and interfacing with external systems is achieved.
Test Plan	The section identifies the key testing goals for the simulator tool and how the tests combine to satisfy all requirements in the SRS and use cases in the LD.

The entire report should be in the region of 8000-10,000 words excluding tables, figures, and references. The report shall be in your own words. However, in order to control the word count, you may refer directly to documentation in the source code that describes in-depth explanations of complex design features or code structures. While such documentation will not directly contribute to the mark for the report, it may indirectly help with the style and presentation of the report which does contribute to the final mark.

Team Component:

You and your team will demonstrate a functioning software simulation tool that has been designed, constructed, and tested throughout the module. As part of the demonstration, you and your team will jointly cover:

- A brief introduction to the project along with evidence of the contributions made by all members of the team throughout the process.

- Any compilation, linking, unit testing, building and deployment of code to your test environment.
- The execution of test scenarios that validate the safe and secure operation of the solution under a wide variety of conditions.

Further details of the simulator program are described in the section "Your Organisation's Role" in the scenario details below.

Supporting Material:

In addition to the report and the team demonstration, you will provide a link to a private repository in which your code and any other documentation shall be stored. A clone of the repository will be taken immediately after the submission deadline and may be used to support the marking process.

Marking Scheme

Guidelines for how marks are evaluated are summarised in the table below.

Component	Component Element	Mark
Group	Assessment of the delivered software	36%
Group	Presentation Quality & Q&A	4%
Individual	Report (selected section only)	54%
Individual	Style and presentation	6%
TOTAL MARK		**100%**

It is important to understand that of the three sections that comprise the Individual Report, **only one of the three sections (Requirements, Design, Test Plan) shall be selected for marking**. You will be made aware of the selected section in your assessment feedback. To ensure fairness, the selected section will apply to all submissions. Other sections may contribute to the final mark in borderline cases, but it is unsafe for you to "guess" which section will be selected for the marking. There is a high degree of cohesion between the 3 sections meaning that whatever section is selected for marking, close attention to the other sections would be pre-requisite for a good mark.

The report component carries 60% of your final assessment mark of which 54% will be awarded for the substance of the selected section and 6% shall be awarded for style and presentation.

The group component carries 40% of the final assessment and shall be determined from the group presentation that all team members are required to make some contribution. After the presentation, there will be a Questions and Answer (Q&A) section where each team member should expect to respond to one or more questions on ***any aspect*** of the project.

The group presentation of the solution shall last between 45 and 60 minutes. The Q&A should last no more than 30 minutes meaning the entire session should not exceed 90 minutes. This will probably take place over MS Teams and be recorded. All team members must be capable of functional audio and screenshare during the team presentation.

While most of the marks allocated will be allocated according to the capabilities of the team's delivered solution, 4% of the total module mark will be awarded for the general presentation skills and performance in the Q&A session.

More details of the allocation of marks are provided at the end of the Assessment Overview section.

Achieving Success In this Module

This is a 24-CAT module. Be prepared to expend more effort in this module in comparison with 18-CAT modules.

There are several factors you need to be aware of in order to gain a good mark for this module:

1. At all times, respect yourself and others. Avoid situations where the work is blocked. Be honest about your own capabilities. Own your problems but reach out for help from your team or your instructor when you need it. Always try to help others if you can.
2. Time management and careful planning is critical for all assessments but especially so when working in teams. Break the assignment into small manageable chunks with the aim of delivering an incremental piece of meaningful work each week.
3. Carefully choosing a development strategy that is inclusive and empowers ALL team members to contribute to the development effort is critical. However, it is essential that you and your team set to work immediately on the problem and endeavour to work together consistently throughout the module with a common vision. Get involved from the start.
4. Avoid complication where possible. Solve the problem that is set before you - not the problem you think you would like to solve. Trying to over complicate things is a common reason for failure in projects.
5. Avoid situations where you are blocking the work of others and in particular, avoid leaving work to the last moment. Problems will appear to escalate rapidly close to the submission deadline with significant consequences that can impact the entire team.
6. Careful analysis of the scenario presented is crucial to elicit the full set of functional and non-functional requirements. A good appreciation of the problem space will also help you identify any implicit requirements that have been omitted from the scenario.
7. Make sure you regularly refer to your requirements to ensure your project is on track. Changes can be made to the requirements at any time but make sure the impacts on any part of the design, section of code or test script are appropriately assessed.
8. A solid test plan is of paramount importance for this safety critical project. The test plan will provide assurances that conceivable safety and security requirements are met. This is crucial for the demonstration of your simulation tool.

9. Careful examination of the marking scheme is highly recommended. This will suggest additional guidance on how to deliver a strong submission.
10. Listen to advice and seek to act on any recommendations from your instructor at the earliest opportunity.

Marking Scheme

Criteria	**Component**	**70+**	**69-55**	**40-54**	**<~~>~~40**
Software Simulation Demonstration NOTES: All teams must be present and prepared to start the demonstration at the allotted time (timings may be delayed in some circumstances). In addition to the marking scheme described here 10% of the group component shall be awarded based on the general quality and preparedness of the presentation and the ability to answer questions. There is no requirement (and therefore no extra marks allocated) for a hardware implementation.	TEAM	Fully operational software simulation tool that reports on a complete set of tests that cover all important use cases and fault conditions. The simulation provides a simple, correct, and complete representation of the state of the facility as the test progresses. All information needed to validate operation is available throughout. The simulation accounts for timings and delays introduced by the opening / closing or gates and the responsiveness of environmental controls and finally, the operation of the system under extreme conditions that require safe evacuation of parts of (or the entirety of) the facility. The process by which new test cases can be supported is simple and well thought through and robust.	Fully operational software simulation tool that reports on a reasonably complete set of tests that cover the most important use cases and fault conditions (a few important tests are deemed to be missing). The simulation provides a simple representation of the state of the facility as the test progresses. Most of the state changes are represented. The simulation accounts for timings and delays introduced by the opening / closing or gates and the responsiveness of environmental controls and finally, the operation of the system under extreme conditions. The process by which new test cases can be supported is reasonable.	The software simulation tool can be demonstrated to create and execute a limited set of test sequences. The simulation could demonstrate in part the safe and secure operation of the facility however, the limited set of test scripts means that full assurance would be impossible OR The software fails to work in a way that provides assurance of the safe and secure operation of the system, but a comprehensive set of tests has been developed. The process by which new test cases can be supported is cumbersome.	The software failed to work during the demonstration and there were several important tests missing from the test plan.

Criteria	Component	70+	69-55	40-54	<~~>~~40
Report - Requirements Engineering	INDIVIDUAL	A thorough understanding of the scenario is evidenced by all conceivable appropriate functional, non-functional and subconscious requirements. All requirements are cross referenced with all appropriate tests. Requirements are clearly articulated and organised at the right levels of abstraction. Clear organisation of requirements into features and team user stories is provided. Ownership and traceability of requirements and attribution to individual team members the assessment. Conflicts, omissions, inconsistencies with the scenario are discussed and resolved. The individual report is fully consistent with the team's demonstration	An understanding of the scenario is evidenced by a comprehensive set of functional, non-functional and subconscious requirements. All requirements are cross referenced with all appropriate tests. Requirements are clearly articulated and organised at the right levels of abstraction. Clear organisation of requirements into features and team user stories is provided. Ownership and traceability of requirements and attribution to individual team members the assessment. Conflicts, omissions, inconsistencies with the scenario are partially dealt with. The individual report shows minor inconsistencies with the team's demonstration	A good understanding of the scenario has resulted in the derivation of most appropriate functional and non-functional requirements. All requirements are cross referenced with all appropriate tests. Requirements are reasonably well described but are generally considered difficult to translate to tasks or designs. Ownership and traceability of requirements and attribution to individual team members the assessment. Few if any conflicts, omissions, or inconsistencies have been identified. The individual report shows significant inconsistencies with the team's demonstration	There is little to evidence any requirements engineering strategy. OR The report is predominantly inconsistent with the team's submission.

Criteria	Component	70+	69-55	40-54	<~~>~~40
Report - Software Design Section NOTES: To avoid including implementation specific details in the designs, the use of pseudo-code is permitted in the Logical Design section. There is no requirement to document the design of the software development pipeline. Open source libraries or third party components must be identified. It is not necessary to include lists of sub dependencies that exist in a package.	INDIVIDUAL	Logical design of the prototype solution contains meaningful, legible and original diagrams including UML component / use case / state transition / sequence diagrams along with threat models. There is a complete mapping between the requirements and the design elements. An excellent analysis of all significant threats is provided. Logical, well considered designs that fully conform to the background information are provided. Apart from references to all open source libraries, software packages used for the simulation, the design avoids implementation specific details. Developers would encounter few no significant problems when implementing the designs. The individual report is fully consistent with the team's demonstration.	Logical design of the prototype solution contains meaningful, legible and original diagrams including UML component / use case / state transition / sequence diagrams along with threat models. Most aspects mentioned in the background information are covered. There is a complete mapping between the requirements and the design elements. An good analysis of most significant threats is provided Designs conform well most of the significant background information provided Apart from references to all open source libraries, software packages used for the simulation, the design avoids implementation specific details. Some open source libraries or dependencies may be missing from the document. Developers would encounter a few significant problems when implementing the designs. The individual report shows minor inconsistencies with the team's demonstration.	Logical design of the prototype solution along with threat models is provided. Several aspects covered in the background information are missing. The absence of diagrams somewhat detracts from the design. There is a somewhat tenuous mapping between the requirements and the design elements. A good analysis of some threats is provided. Designs align reasonably well with the background information provided although several important omissions may exist. The design avoids implementation specific details. There is an incomplete list of libraries or dependencies provided. Developers would encounter significant problems when implementing the designs. The individual report shows significant inconsistencies with the team's demonstration.	Designs are weak or incomplete. Architectural and implementation considerations are poorly articulated or non-existent. There is insufficient mapping between the requirements and the design elements. There is insufficient discussion on security concerns in the designs. A development team would be unable to create a prototype the meets the system objectives that could be reasonably expected. The report is predominantly inconsistent with the team's submission.

Criteria	**Component**	**70+**	**69-55**	**40-54**	**<~~>~~40**
Test Plan and Test Architecture NOTES: Use cases should cater for normal operation of the system. Tests should not only cover these use cases but also evaluate the behaviour of the system under a full range of hazardous conditions/events. Test scripts should not be articulated in full but referenced by a test script identifier and a short description of its purpose. Any referenced script is expected to fully operational and available from the team's code repository.	INDIVIDUAL	Clear descriptions backed with credible discussions of a complete set of i. test scripts that assess the system under normal operation ii. test scripts that assess the system under a wide range of fault conditions. All test scenarios are uniquely identified and associated with requirements or use cases. Failed tests would succinctly and unambiguously report their cause. There is a convincing argument that the test platform offers robust verification of the safe and secure system operation for the set of use cases that might be reasonably expected. There is compelling evidence that all tests were successfully executed. The individual report is fully consistent with the team's demonstration.	Clear description of the design of a good number of i. test scripts that assess the system under normal operation ii. test scripts that assess the system under a reasonable range of fault conditions (a few important omissions exist). Test scenarios are uniquely identified and associated with most requirements or use cases. Cause of failed tests might be reported in an ambiguous or over-verbose manner. There is a strong argument that the test platform offers robust verification of the safe and secure system operation for the set of use cases that might be reasonably expected. Clear evidence that most tests were successfully executed. The individual report shows minor inconsistencies with the team's demonstration.	Descriptions of a minimal number of i. test scripts that assess the system under normal operation ii. test scripts that assess the system under a small number of fault conditions (several important omissions exist). Test scenarios are identified and associated with many requirements or use cases. Some import requirements or use cases are missing. There is a partially convincing argument that the test platform offers verification of the safe and secure system operation for the set of use cases that might be reasonably expected. The individual report shows significant inconsistencies with the team's demonstration.	The test plan is weak, lacks coverage and objectives are unclear. Few of the expected tests are provided or otherwise do not work as expected. The report is predominantly inconsistent with the team's submission.

ADDITIONAL FACTORS

Style And Presentation (10% of the marks allocated to the report)

Style and presentation are important for the assessment. The following elements (in no particular order) contribute to this mark.

Logical structure organised into appropriate sections and subsections
Clear, concise paragraphs - avoidance of overly convoluted narrative.
Complex themes are accompanied with appropriate, captioned and diagrams.
Tables and diagrams shall be accompanied by narrative that explains their purpose.
Grammar and spelling (English) throughout
Page numbering.

NOTES:
Where appropriate, the Harvard Referencing scheme should be adopted.

TEAM ETHICS (0% of the marks allocated for the presentation and the report)

Despite the fact that no marks are directly awarded for good team working ethics, it should be noted that meeting the following criteria will almost indirectly lead to a better mark.

Clear evidence of consistent progress throughout the assessment provided through clear management summaries that account for key tasks, risks, project status and some measurement of progress.

Separate project management records are maintained that clearly and succinctly show progress between MONTHLY reporting periods.

The care taken in planning the project as a team is evidenced by a well-considered development strategy from the outside along with evidence of continual progress.

Assessment Scenario

Background

The UK government has decided to enter the race to extract valuable minerals from planetary bodies and plans to setup the new Facility for Lunar and Space Exploration ("FALSE"). This decision was made following the breakdown in the talks held at the recent UN summit on the Allocation of Lunar and Asteroid Rights for Mining ("ALARM").

It is apparent that the commercialisation and with it, the militarisation of the moon has the unfortunate consequence of having a negative impact on the scientific community's disposition to share information.

The primary objectives of FALSE programme are:

1. Safety: Essential support systems must take all measures to guarantee that safe operation of the facility. The facility should aim to provide emergency life support for the crew members for as long as it takes to mount a full evacuation of the facility.
2. Security: The success of the research and mining facility is crucial for the UK economy and as such, the facility shall be protected at all times (there is an underlying assumption that other nation states may have similar facilities nearby).

Further developments following the breakdown of ALARM summit have raised fears of interference with the FALSE programme during development, and also when it is eventually deployed on the moon.

The "FALSE" Overview

Figure 1 shows the layout of the completed facility comprising a collection of interconnecting "pods". The layout shows the initial configuration of the facility, but future expansion is anticipated.

The Communications and Control Centre pod is at the heart of the facility and supports the primary communications link with Mission Control in Royston Vasey, Derbyshire. A backup communications link is installed in the Emergency Quarters pod.

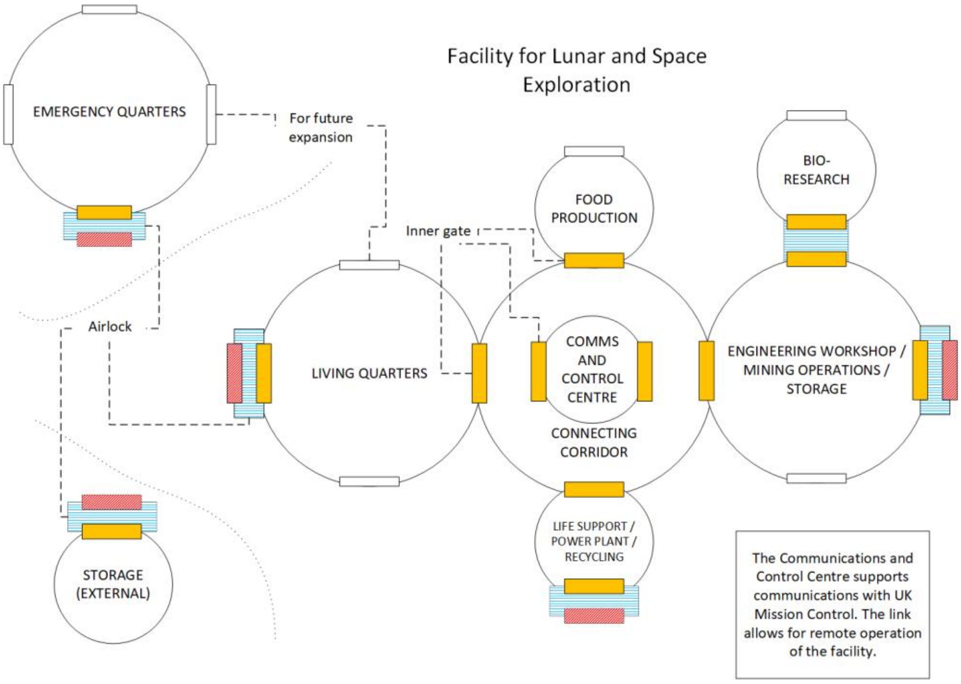
Facility for Lunar and Space Exploration
EMERGENCY QUARTERS
For future expansion
Inner gate
FOOD PRODUCTION
BIO-RESEARCH
Airlock
LIVING QUARTERS
COMMS AND CONTROL CENTRE
CONNECTING CORRIDOR
ENGINEERING WORKSHOP / MINING OPERATIONS / STORAGE
LIFE SUPPORT / POWER PLANT / RECYCLING
STORAGE (EXTERNAL)
The Communications and Control Centre supports communications with UK Mission Control. The link allows for remote operation of the facility.

Figure 1 Overview of the planned FALSE facility.

Pod Design and Life Support

Pods are made from a flexible but highly resilient material. Pods come in two sizes (figure 2).

(i) "A-Pods" - these are the larger of the two sizes and can connect to a maximum of 4 other pods.

(ii) "B-Pods" - these smaller pods can connect to a maximum of 2 other pods.

Pods are connected by air-tight gates. For pods that lead to the external lunar landscape (or the Bio-Research pod) airlocks are used in place of single gates. An airlock is a discrete unit that comprises a short tunnel with an air-tight gate at each end. All gates (and consequently airlocks) are controlled by the Airlock Control Management system described below.

Under normal circumstances, life support is provided by multiple systems installed in a Type-B pod (Life Support / Power Plant / Recycling pod, figure 1).

Each A-Type pod has its own independent life support system that can support 3 astronauts for a period of 3 days. Each B-Type pod also has its own independent life support system that can support 1 astronaut for a period of 3 days. Three days is the estimated time to evacuate all astronauts from the facility in the event of a catastrophic failure.

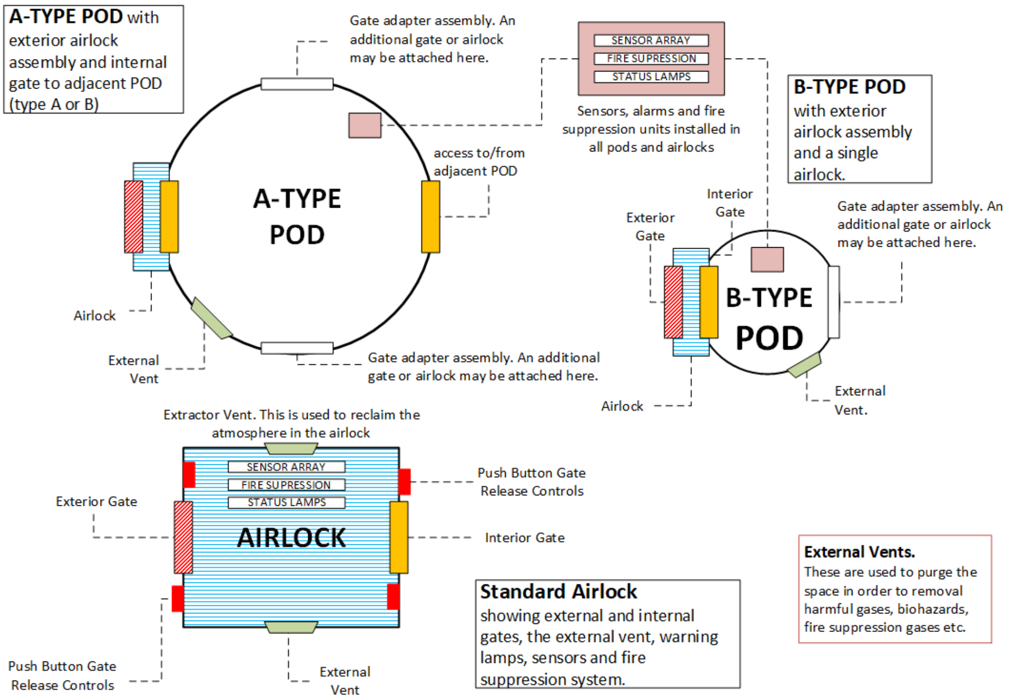
A-TYPE POD with exterior airlock assembly and internal gate to adjacent POD (type A or B)
Gate adapter assembly. An additional gate or airlock may be attached here.
SENSOR ARRAY
FIRE SUPRESSION
STATUS LAMPS
Sensors, alarms and fire suppression units installed in all pods and airlocks
B-TYPE POD with exterior airlock assembly and a single airlock.
access to/from adjacent POD
A-TYPE POD
Airlock
External Vent
Gate adapter assembly. An additional gate or airlock may be attached here.
Interior Gate
Exterior Gate
Gate adapter assembly. An additional gate or airlock may be attached here.
B-TYPE POD
External Vent.
Airlock
Extractor Vent. This is used to reclaim the atmosphere in the airlock
SENSOR ARRAY
FIRE SUPRESSION
STATUS LAMPS
AIRLOCK
Exterior Gate
Push Button Gate Release Controls
Interior Gate
Standard Airlock showing external and internal gates, the external vent, warning lamps, sensors and fire suppression system.
External Vents. These are used to purge the space in order to removal harmful gases, biohazards, fire suppression gases etc.
Push Button Gate Release Controls
External Vent

Figure 2. Pod and Airlock Design

To prevent unnecessary loss of the air to the lunar environment, prior to opening an external door of an airlock, extraction units that link back to the Life Support pod, reclaim the air within the airlock. When the external door is closed, the airlock is flooded with air until normal atmospheric conditions are restored. Emptying (or refilling) an airlock takes 10 seconds.

Gate Operation and the Gate Control Unit (GCU)

There is functionally no difference between an exterior gate and an interior gate.

Figure 3 shows the front and plan views of the gate. The gate is electronically operated in most circumstances. As the gate opens and closes, encoders determine the angle of the gate with respect to the frame (0^{O} fully shut and 135^{O} when fully open). A gate takes 5 seconds to fully open and 5 seconds to close (from a fully open state).

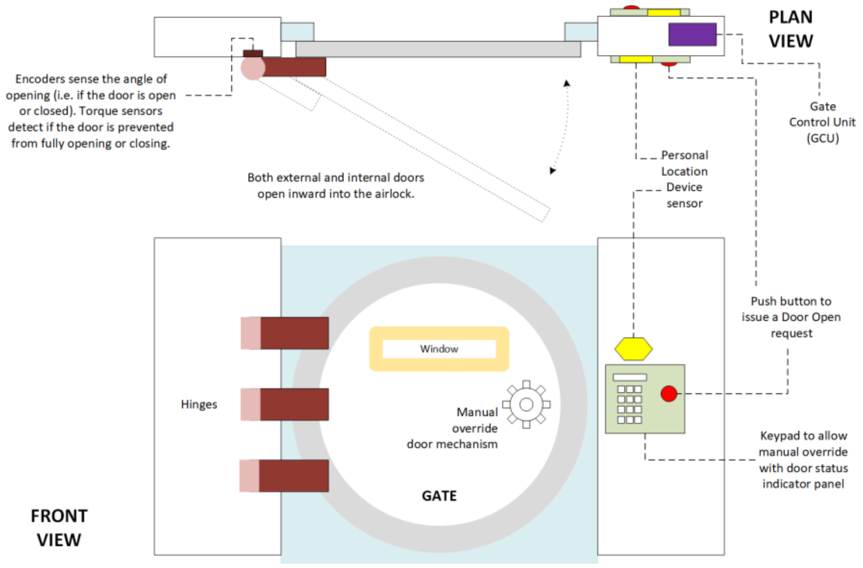
PLAN VIEW
Encoders sense the angle of opening (i.e. if the door is open or closed). Torque sensors detect if the door is prevented from fully opening or closing.
Both external and internal doors open inward into the airlock.
Gate Control Unit (GCU)
Personal Location Device sensor
Push button to issue a Door Open request
Window
Hinges
Manual override door mechanism
GATE
Keypad to allow manual override with door status indicator panel
FRONT VIEW

Figure 3. Gate Design

On each side of the gate, there is a door open/close button, a keypad and Personal Location Device sensor. These three components connect to the Gate Control Unit (GCU) located within the frame of the gate. The GCU must therefore differentiate between the buttons and sensors on the "inside" (furthest from the lunar landscape) from those located on the "outside" (nearest to the lunar landscape)

Each GCU connects to a centralised, facility-wide Airlock Control Management System (ACMS) that (with the assistance of other environmental control systems) safely and securely controls the opening and closing of the gate. This means that ACMS must ensure that throughout the facility, ***no two adjacent gates are open at the same time***. This is to ensure safe operation of the airlock, to preserve the atmosphere throughout the facility and to limit the spread of contaminated air should an accident arise.

The ACMS is located in the Communications and Central Control room.

Proposed Rules Governing Gate Operation - ACMS Integration with Other Environmental Safety Systems

The ACMS is one part of a suite of systems that monitor the environment for the safety and security of the crew and the facility. In order to ensure the safe operation of all gates including those that combine to form the airlocks distributed around the facility, the ACMS processes information from the range of systems as shown in figure 4.

With this information the ACMS can ensure safe and secure operation of all gates in the facility when a gate is activated.

The following rules have been proposed:

1. No two adjacent gates can ever be open at the same time. (unless the emergency manual override feature has been set from the ACMS).

2. Following a short period of time after the initial detection of a hazardous event within a pod or airlock, gates connecting to the affected pod to other pods in the facility will be locked.
3. By applying locks to one or more gates, the ACMS prevents an astronaut from traversing a gate into a pod that unduly threatens either the astronaut or the facility. For example, an astronaut is prevented from entering a pod that has a fire, radiation leak or biohazard leak.
4. Door operation requires the astronaut to wear a Personal Locator Device (PLD) to activate the door control button.
5. Any gate can be administratively locked from the ACMS console.
6. A manual override capability enables locked gates to be opened manually.

As mentioned previously, access into and through the facility is restricted to authorised astronauts (which would typically exclude any astronaut from any other space agency). The facility Commander has a special PLD that ensures access checks are not performed traversing gates.

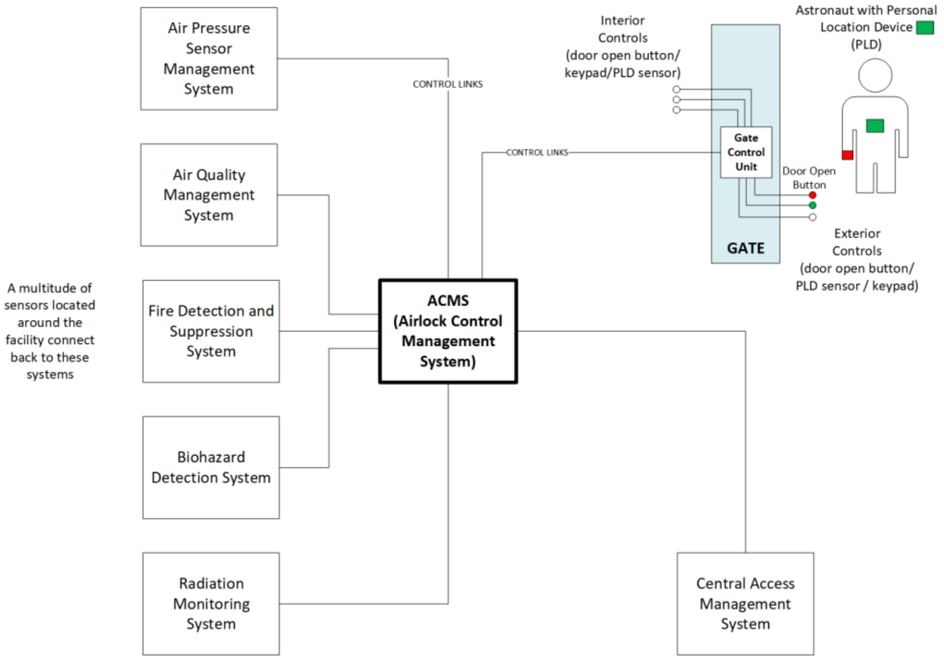
Air Pressure Sensor Management System
Air Quality Management System
Fire Detection and Suppression System
Biohazard Detection System
Radiation Monitoring System
A multitude of sensors located around the facility connect back to these systems
CONTROL LINKS
ACMS (Airlock Control Management System)
Interior Controls (door open button/ keypad/PLD sensor)
CONTROL LINKS
Gate Control Unit
GATE
Astronaut with Personal Location Device (PLD)
Door Open Button
Exterior Controls (door open button/ PLD sensor / keypad)
Central Access Management System

Figure 4. ACMS and Environmental Management Systems

Table 3. describes in more detail the other environmental management systems and how these interoperate with the ACMS.

Table 2. Interaction between ACMS and other Environmental Management Systems

Management System	**System Overview**	**Suggested ACMS Response to Signals from Management Systems.**
Central Access Management System.	Each astronaut has a Personal Location Device (PLD) that sends precise information about the location and biometric information of the individual to sensors placed at gates and throughout the facility. Short range sensors deployed at both sides of a gate detect the PLD and send the astronaut's details to the ACMS.	The door control button (entry to or exit from a pod) is activated in the presence of an astronaut's Personal Locator Device that has been registered in the system. The astronaut is identified to the ACMS which checks the Access Control System for authorisation.
Ambient Pressure Management System	Detects the air pressure in a pod or an airlock. When the air pressure fails, a warning klaxon is activated, and an alarm signal is sent to the ACMS requesting a 20 second delay before initiating local lockdown procedure for that pod or airlock. An "ALL_CLEAR" signal is sent to the ACMS when normal atmospheric conditions have been restored in the pod. In the case of an airlock however, no alarm is generated if the loss of pressure follows an external door open request. In this case, extraction of air from the airlock for the purpose of recycling is normal and this does not result in a local lockdown. In this case, a warning light is activated until the air pressure is restored to normal levels. The external door enters the GATE_OPENING state only after the air has been evacuated from the airlock.	On receiving an alarm signal, the ACMS waits the requested delay period before locking down all gates to connecting pods. Handling alarm systems must be carefully handled as these take priority over any on-going event. On receiving the "all clear" signal, the gates are unlocked.

Management System	System Overview	Suggested ACMS Response to Signals from Management Systems.
Air Quality Management System	Detects concentrations of O2 as well as harmful gases such as carbon dioxide, carbon monoxide and fire suppressant gas. In the event that levels of O2 fall below a certain level or concentrations of harmful gases exceed certain limits, an alarm is raised, a signal is sent to the ACMS requesting a 20 second delay prior to initiating local lockdown procedures. When air-quality levels have returned to normal, air quality alarms are de-activated and an "ALL_CLEAR" signal is sent to the ACMS. In the case of an airlock, if the external door control button is depressed, no alert is created as the air is extracted from the airlock.	On receiving a warning signal, the ACMS waits the requested delay period before locking down all gates in the affected pod that connect to other pods. On receiving the "all clear" signal, the gates are unlocked.
Radiation Monitoring System	In the event that radiation levels exceed a preset safety limit in a given pod or airlock, an alarm is raised and a signal is sent to the ACMS requesting a 20 second delay before initiating local lockdown procedure. When radiation levels have returned to safe levels (perhaps after purging the environment or some other clean up activity has taken place) alarms are de-activated and an "ALL CLEAR" signal is sent to the ACMS.	On receiving the radiation alert signal, the ACMS waits the requested delay period before locking down all gates in the affected pod to other pods. On receiving the "all clear" signal, normal gate operation is restored.
Biohazard Management System	In the event that biohazard levels exceed a preset limit in a given pod or airlock, an alarm is raised and a signal is sent to the ACMS requesting immediate local lockdown.	On receiving the biohazard alarm, the ACMS immediately locks all gates in the affected pod to other pods.

Management System	System Overview	Suggested ACMS Response to Signals from Management Systems.
	When biohazard indicators show safe levels (perhaps after purging the environment or some other clean up activity has taken place) an "ALL_CLEAR" signal is sent to the ACMS.	On receiving the "all clear" signal, normal gate operation is restored.
Fire Suppression System.	In the event of a fire in a given pod or airlock, an alarm is raised and a signal is sent to the ACMS requesting a 20 second delay before initiating local lockdown procedure. Fire suppressant gas is pumped into the chamber. When sensors indicate that fire has been extinguished, the pod is purged of the suppressant (via the external vent) before O2 is pumped back into the pod. Fire alarms are then deactivated but no ALL_CLEAR signal is sent. This is the responsibility of the Air Quality Management System.	On receiving the fire alarm signal, the ACMS waits the requested delay period before locking down all gates in the affected pod to other pods. A signal from the Air Quality Management system indicates that it is safe to restore normal gate operation.

ACMS and GCU Internals

Internally, the ACMS and GCU make use of a set of internal registers to manage gate operation. This section describes the registers and events in use by the GCU.

The GCU maintains the state of the gate using 4 registers (A,B,C and D). These registers may be accessed directly but the recommended approach for reading and writing these registers is via a set of system calls that provide a layer of abstraction. In addition to accessing registers, other system calls are available for reading/writing any associated data.

Register A

Register A supports the following binary flags that represent the state of the gate at any one time.

- GATE_CLOSED
- GATE_OPENING
- GATE_OPEN
- GATE_CLOSING
- GATE_STATE_UNKNOWN

Table 3. Bit flags for the various states

Gate Status	GATE_CLOSED	GATE_OPENING	GATE_OPEN	GATE_CLOSING
Fully open	0	0	1	0
Fully closed	1	0	0	0
Opening	0	1	1	0
Closing	0	0	1	1

The GATE_STATE_UNKNOWN flag is normally set to 0 but may be set to 1 if there is damage to the GCU unit or the door itself. In this state, any request to open the door would fail (although closing the door may still be possible). Clearing the flag requires manual intervention from an astronaut via the keypad or at the ACMS console.

Register B

Register B contains additional information via the following binary flags:

- GATE_OK
- GATE_ERROR
- GATE_ACCESS_RESTRICTED
- GATE_LOCKED
- GATE_OPERATION_PENDING
- GATE_MANUAL_OVERRIDE_ACTIVATED
- GATE_EMERGENCY_OVERRIDE

Register C

Register C supports binary flags that are set when sensors connected to the GCU raise various events.

The flags are set when

(i) the button detects a button press,
(ii) the Personal Locator Device or PLD sensor detects the presence of a PLD device (see below),
(iii) a key is entered on the keypad or
(iv) the status of the gate is changed (GATE_OPEN, GATE_CLOSED etc).

The full list of events is:

- GATE_INSIDE_PLD_DETECTED_EVENT (raised when a PLD moves to within range of the gate's PLD sensor)
- GATE_OUTSIDE_PLD_DETECTED_EVENT
- GATE_INSIDE_PLD_CLEAR_EVENT (raised when the last PLD within range of the gate's sensor moves away from the gate)
- GATE_OUTSIDE_PLD_CLEAR_EVENT

- GATE_INSIDE_BUTTON_PRESS_EVENT
- GATE_OUTSIDE_BUTTON_PRESS_EVENT
- GATE_INSIDE_BUTTON_CONTROL_EVENT
- GATE_OUTSIDE_BUTTON_CONTROL_EVENT

- GATE_INSIDE_KEYPAD_EVENT (raised when a key on the keypad is depressed)
- GATE_OUTSIDE_KEYPAD_EVENT
- GATE_STATUS_CHANGE_EVENT (raised when the Register A flags change)

Register D

The GCU must continually listen for control or acknowledgement messages from the ACMS. These are associated with the following flags in Register D and are set when a message is received from the ACMS.

- ACMS_CONTROL_MESSAGE_EVENT
- ACMS_ACKNOWLEDGE_MESSAGE_EVENT

ACMS_CONTROL_MESSAGE_EVENT flag is set when a command is issued from the ACMS. Such commands are usually designed to set the Register B flags. The ACMS_ACKNOWLEDGE_MESSAGE_EVENT is set when ACMS responds to a message sent from the GCU.

Your Organisation's Task

Your organisation has been selected to compete for the contract for supplying a simulation tool that

i. showcases the safe and secure operation of the ACMS and GCU for the FALSE programme.

ii. identifies any potential situations where adverse events threat the safety and security of the crew or the facility.

iii. identifies potential weaknesses inherent in the intended operation of ACMS and GCU given the scenario details above.

You are required to demonstrate to senior programme management the operation of the ACMS and its ability to safely and securely control multiple GCUs under a variety of conditions. To achieve this, you are required to create a simulation tool that mimicks:

1. a single instance of the ACMS that can support multiple gates/GCUs,
2. multiple instances of GCUs for the facility layout shown in figure 1,

The simulation tool will parse and launch test scenarios and create a report that shows the result of various conditions and events on the system. Ideally, a comprehensive set of tests should result in no harm to either astronauts or the facility as a result of the electronic operation of gates.

A suggested architecture for the simulation tool is showing in figure 5.

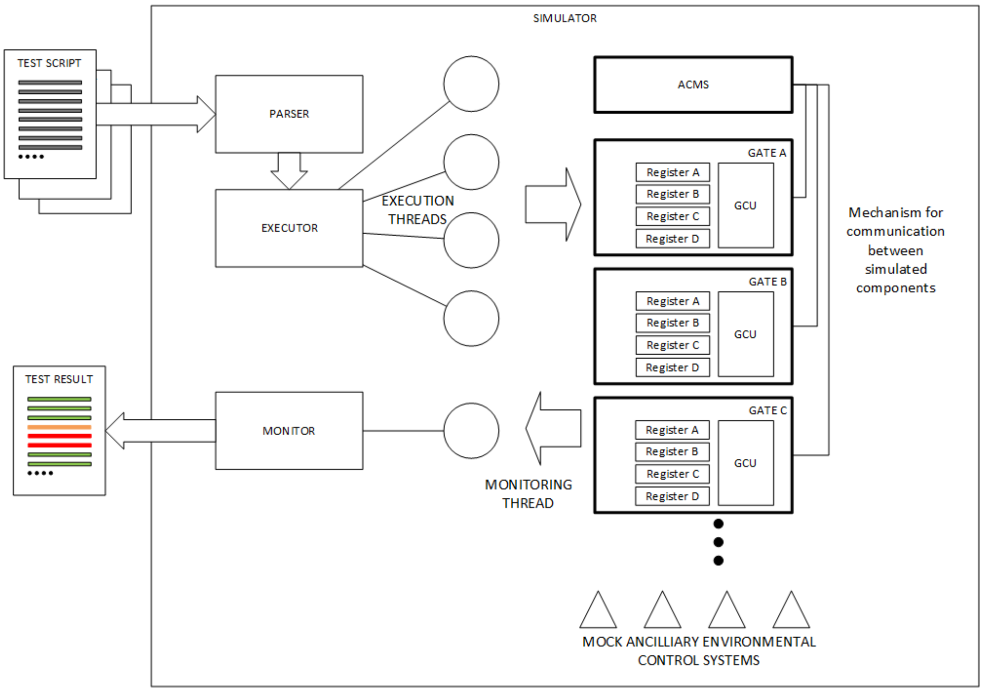

Figure 5. Possible architecture for the simulator.

Solution Constraints

The simulation that must meet the following design goals:

1. The simulation can be written in any coding language. Trustworthy open-source tools may be used for parts of the required but in general, the number of external

libraries used for the construction of the simulation tool should be kept to a minimum.

2. The simulation must run on a Ubuntu Linux (x86-64) platform.
3. The ACMS UI will comprise a simple text-based ncurses interface or SPA (single page web application) that provides a simple representation of the facility indicating pods, connecting gates and airlocks. The UI will also indicate the locations of 10 astronauts, the status of each gate and of each pod (with respect to the environmental control systems).
4. The simulation will read from test scripts that describe a wide range of conditions and events in order to simulate the safe and secure operation of the ACMS, in particular:
    a. gates are opened in a way that does not compromise the security or environmental integrity of the facility.
    b. the system operates in a manner that preserves life when the facility is threatened by one or more hazardous events (for example, an astronaut cannot be trapped in a pod or prevented from escaping).
5. The method of communication between the ACMS, GCUs and simulate environmental control systems is undefined - you are free to choose any appropriate communications, signalling, messaging layer.
6. The simulation will show how safety and security is preserved with the addition of new pods and gates.

While the details of intended operation of the FALSE facility have been provided with the best intentions, it is acknowledged that one or more conflicts or omissions may exist. You are required to identify and resolve these anomalies as part of your work.

Furthermore, it is acknowledged that you will may need to make several assumptions. For example, you have been provided with no information about the system calls you might use to interact with the GCU registers or handle events. You are free to implement these as you see fit. Also, you have been provided with no information about the interfaces of external environmental control systems interfaces. You are free to design and implement your own "mocked" interfaces for the simulation, but they must adhere to the functionality described in table 3. There is no requirement to build these external systems. There is also no suggestion that the simulation software will find its way into the production environment.

Your organisation has been asked to provide documentation relating to the simulator tool. This documentation includes the Software Requirements Specification, Logical Designs and Test Plan (see the Assessment Overview for more details).

Testing Considerations

The test plan will provide a description of the structure of the test scripts along with clear descriptions of the various test scenarios used by the simulation tool.

The sections of a test script define the following:

(i) the initial status of all gates in the facility,

(ii) the initial status of simulated environmental monitoring systems,
(iii) the intended movement of one or more astronauts through one or more gates (starting location and direction of travel),
(iv) the expected outcome.

The script engine will monitor the following

(v) the changes in state of all gates resulting from the movement of astronauts throughout the facility,
(vi) the changes in state of all gates as a result of a change in the environmental conditions (including extreme events that might require the controlled evacuation from a pod or the facility),
(vii) changes in the threat level to any astronaut or the wider facility resulting from the movement through gates or through changes in the environmental conditions.

The simulation must account for timings and delays introduced by the opening / closing or gates and the responsiveness of environmental controls as per the descriptions above.