# Group Project Report

Start Assignment

- Due Sunday by 23:59
- Points 14
- Submitting a file upload
- Available 19 Oct at 9:00 - 3 Dec at 23:59

Please read the project description here first - (https://myuni.adelaide.edu.au/courses/101188/files/18485167?wrap=1) Group Project_Secure Software Engineering.pdf (https://myuni.adelaide.edu.au/courses/101188/files/18485167?wrap=1) (https://myuni.adelaide.edu.au/courses/101188/files/18485167/download?download_frd=1) (https://myuni.adelaide.edu.au/courses/74205/files/11822609/download?download_frd=1).

Your team will summarize your completed course project in a report (at most 10 A4 pages, excluding references and appendix).

This report should summarize the key requirements and functionalities of your product and the team contributions of the project and the following structure is for your **reference. You may customize the structure, this is only a hint to help you organize your report.**

- **[Title of Your Project]**
- **Group Project Report - Group <group number>**
  - your names and your IDs
- **Introduction/Overview**
  - Introduction of the background and current status
  - Describe the issue and the motivation of the project
  - Goals and objectives of the project
  - Key functionalities of the final product (summarized with bullet points)
- **Key Requirements**
  - You are expected to summarize and report what requirements you defined during **milestone 1**.
  - Provide references and evidence to support your arguments.
  - The number of requirements is not restricted, at least 5 security requirements should be described.
- **System Design**
  - Please specify the design of your product according to each requirement (e.g., link each design to one or more requirements).
  - This part could be a summary of your group **milestone 2** task
  - It would be better to use some diagrams to illustrate your design in an easy-to-understand way
  - Please present how your design can meet the related requirement(s)
- **Implementation**
  - Describe how the modules in the system are implemented (in a high-level view
  - Describe how the product is tested or verified

- **Conclusion**
  - Conclude the project. Summarize your contributions (as a team).
- **Future work**
  - What could be extended in the future
- **Appendix**
  - The list of group tasks(Check 'Group Project Secure Software Engineering 2025.pdf' for more information)
  - User interface screenshots of key functionalities/modules
  - The output of group milestone 1 & 2
  - An instruction to run the code
  - Anything else

**Group Project Report**

| Criteria | Ratings | | | Pts |
|---|---|---|---|---|
| Introduction/Overview & Conclusion & Future Work & Appendix | **3 to >2.0 Pts**<br>**Excellent**<br>Background, goals, and objectives are well-articulated. The conclusion effectively summarizes the project, and future work is creative and insightful. Appendix is complete and relevant. | **2 to >1.0 Pts**<br>**Good**<br>Covers most parts effectively but lacks minor details or depth in some sections. The appendix is mostly complete but may miss a few minor elements | **1 to >0 Pts**<br>**Fair/Fail**<br>Some parts are incomplete, unclear, or missing key details. The appendix is partially complete with significant gaps. | 3 pts |
| Key Requirements | **3 to >2.0 Pts**<br>**Excellent**<br>Requirements are well-defined, thoroughly explained, and supported by strong evidence or references. Includes at least five well-documented security requirements. | **2 to >1.0 Pts**<br>**Good**<br>Requirements are mostly clear and supported, with minor gaps in explanation or fewer security requirements. Evidence and references are mostly accurate. | **1 to >0 Pts**<br>**Fair/Fail**<br>Requirements are vague or incomplete, lacking clarity or strong support. Few or no security requirements are mentioned. | 3 pts |
| System Design | **3 to >2.0 Pts**<br>**Excellent**<br>System design is well-documented and linked clearly to requirements. Includes detailed diagrams that are easy to understand. Demonstrates how the design meets the requirements. | **2 to >1.0 Pts**<br>**Good**<br>System design is mostly well-documented, with clear links to requirements. Diagrams are included but may lack some detail or clarity. | **1 to >0 Pts**<br>**Fair/Fail**<br>System design is vague or lacks connection to the requirements. Diagrams are incomplete or difficult to interpret. | 3 pts |
| Implementation | **3 to >2.0 Pts**<br>**Excellent**<br>Implementation is well-documented at a high level, with clear descriptions of all modules. Includes | **2 to >1.0 Pts**<br>**Good**<br>Implementation is mostly clear, but minor details or testing steps may be missing. Modules are described adequately | **1 to >0 Pts**<br>**Fair/Fail**<br>Implementation lacks clarity or is incomplete. Testing or verification steps are insufficient or missing. | 3 pts |

| Criteria | Ratings | | | Pts |
|---|---|---|---|---|
| | thorough testing and verification steps. | but not comprehensively. | | |
| Presentation | **2 to >1.5 Pts**<br>**Excellent**<br>Report is well-structured, professional, and highly readable. All arguments are supported by accurate evidence and references. | **1.5 to >0.5 Pts**<br>**Good**<br>Report is mostly well-structured and easy to follow, with minor grammatical or structural issues. Most arguments are supported by evidence and references. | **0.5 to >0 Pts**<br>**Fair/Fail**<br>Report is poorly structured or difficult to follow, with various grammatical or spelling mistakes. Arguments lack proper evidence or references. | 2 pts |
| | | | | Total points: 14 |