



ASSESSMENT BRIEF	
Module Title:	Software Architecture
Module Code:	KV5035
Academic Year / Semester:	2024-25 / Semester 2
Module Tutor / Email (all queries):	Dr John Rooksby (john.rooksby@northumbria.ac.uk)
% Weighting (to overall module):	50%
Assessment Title:	Technical documentation of system architecture
Date of Handout to Students:	13 th March 2026
Mechanism for Handout:	Module Blackboard Site
Deadline for Attempt Submission by Students:	Monday 18 th May 2026, 4pm.
Mechanism for Submission:	Blackboard.
Submission Format / Word Count	Blackboard (Turnitin) / 1500 words + diagrams
Date by which Work, Feedback and Marks will be returned:	15 th June 2026
Mechanism for return of Feedback and Marks:	Marks and individual written feedback will be uploaded to the Module Site on Blackboard.

MODULE LEARNING OUTCOMES

The module learning outcomes (MLOs) for this module are:

MLO1 – Demonstrate knowledge of approaches to modelling, representing and reasoning about software architectures, and knowledge of major and emerging architectural styles.

MLO2 – Demonstrate knowledge of components (such as packages, libraries, services, layers and resources), their interfaces, and associated quality issues including security.

MLO3 – Demonstrate knowledge of the tools and approaches used to build, maintain and test component-based software systems and for managing quality and risks across the project lifecycle.

MLO4 – Use architectural patterns and concepts to build a component-based system.

MLO5 – Effectively communicate information, arguments and analyses about software using appropriate technical documentation

This assessment addresses MLO1, MLO4 and MLO5



ASSESSMENT OVERVIEW

This is an individual assessment worth 50% of the overall module mark. This assessment is marked out of 50. This is part 2 of the module assessment.

For this assessment you will create technical documentation of a system architecture. This documentation should be created to inform the development of a system (as opposed to describing the architecture of a finished system).

The documentation should describe an improved version of the academic conference system originally outlined in part 1. Your documentation should address the whole, overall system (not just the REST API you focused on in part 1). The system you describe will *replace* the prototype you worked on for part 1 and therefore is not subject to the 'essential requirements' and 'quality requirements' stated in part 1.

SCENARIO

You are a software architect working with a small team of developers to create software applications to present information to the attendees of academic conferences.

An academic conference is an event at which research is presented and discussed. Conferences can have thousands of attendees and will take place over several days. There may be hundreds of presentations. A conference will typically take place at a conference centre or university. Different presentations can take place at the same time using different rooms.

Conference attendees would benefit from a web app that they can use before and during the conference to read the conference schedule, find out the names and affiliations of the authors who will be presenting research, and find out about the research papers that will be presented.

Conference attendees would benefit from a mobile app in same way they would a web app. Additional functionality making use of location and other sensors could be added in the future.

The web and mobile apps should be able to show information about different conferences. This could be different occurrences of the same conference (e.g. CHI 2026, CHI 2027, etc.) or completely different conferences (e.g. ICRA 2026, SfN 2026, etc.). The data for each conference should be held in a separate database.

Users of the apps should be able create a user account before accessing data in the app. A user should have a single account that they can use for accessing information about any conference. For example, they should use the same account for the mobile app as they do the web app. It must be possible for users to view information about current and upcoming conferences. It must also be possible to access information about conferences after they have happened.

Users should be able to store their own, personal "notes" about items of content. Each note should be associated with specific item of content. For example, a user might make notes during a conference presentation about the content being presented. Users should be able to create and access their notes in both the web and mobile app.

An administrator should be able to add a new conference and to update information about existing conferences.



ASSESSMENT TASK

You should produce a software architecture description document for the conference application outlined in the scenario. Your description does not need to specify a full solution but should use good quality diagrams and text to explain important aspects of the system.

Your document must contain the following sections:

Cover page (pass/fail). This should state your name, student id, and the word-count of your document.

AI Declaration (pass/fail). This should use the template provided below.

Introduction (optional). The report should contain an introduction giving a brief description of the system and introduce the modelling approach taken for documenting the system.

Proposed architecture (40 Marks). This should contain your diagrams and documentation that represent the software architecture.

- **System requirements** should be represented by use case diagrams and scenarios (if you produce use case tests and a test plan, these should be put in an appendix).
- **Logical structure** should be represented by class diagrams and interaction sequence diagrams.
- **System behaviour** should be represented with activity diagrams.
- **Implementation** should be represented using component diagrams

Diagrams should be included where appropriate and embedded into the report itself rather than submitted separately. The diagrams you use do not need to cover every aspect of the system in depth but should focus on key parts and/or give an overview. You can use as many diagrams as you wish, but these must be appropriate and relevant for your architecture. All diagrams included in the document must be discussed in the text. You may lose marks for including diagrams that are not explained in the document and/or do not make a clear contribution to the description of your architecture.

Conclusion (5 Marks). This should give a short summary, describe any strengths and limitations of your architecture, and describe any important questions that would still need to be addressed before the architecture can be finalised.

References (optional). If your work draws on any sources of information, including academic papers and reports, and any online sources, they should be referenced in a consistent and appropriate format (such as Harvard).

Appendix (optional). If you produce use case tests and a use case test plan, these should be in the appendix.



Marks are not awarded for the Cover Page and AI Declaration but reports that fail to include these may be given zero marks overall. Including an introduction, references and an appendix is optional, but may contribute to the marks for overall report quality.

Report quality (5 marks). The report should be well structured and presented. Diagrams should be used in an appropriate way. Each diagram in the report should have a figure number and caption. Any sources of information should be appropriately referenced. Academic references are not essential but if included they must be in Harvard format.

DELIVERABLE

You must submit a single Architecture Design Document using Turnitin on Blackboard.

Format: You should create a single document in Word or PDF format.

Word count: The document should be no longer than **1500 words**. The word count should not include the cover page, AI declaration, references or appendix. Diagrams and images do not contribute to the word count. The word count is a limit, not a target and reports containing unnecessary text may be marked lower than shorter, more focused reports.

Diagrams: Your diagrams must be embedded into the document.

Where to submit: Turnitin (via Blackboard)



MODULE SPECIFIC ASSESSMENT CRITERIA

Grade descriptors	
Outstanding work. An insightful system architecture is presented with almost no errors, omissions or ambiguities. There is outstanding use of the range of techniques taught on the module, demonstrating advanced insight into key approaches and concepts. The work goes above and beyond by using additional approaches and techniques in fully appropriate ways. A software team would be excited to use the architecture description.	40-50 marks
Excellent work. A highly plausible system architecture is presented with few errors, omissions or ambiguities. There is excellent use of the techniques taught on the module, demonstrating a thorough understanding of key approaches and concepts. A software team would be able to use the architecture description.	35-39 marks
Very good work. A very plausible system architecture is presented but with some errors, omissions or ambiguities. There is very good use of multiple techniques taught on the module, demonstrating a sound understanding of the key approaches and concepts. Some further detail would be necessary if a software team were to use the architecture description.	30-34 marks
Good work. A plausible system architecture is presented but with errors, omissions or ambiguities. There is some use of the techniques taught on the module, demonstrating a basic understanding of several key approaches and concepts. Further detail would be necessary if a software team were to use the architecture description.	25-29 marks
Acceptable work. A somewhat plausible system architecture is presented but there are multiple errors, omissions or serious ambiguities. Techniques taught on the module for documenting architecture are used in a limited or sometimes inappropriate way. Substantial additional detail or corrections would be necessary if a software team were to use the architecture description.	20-24 marks
Fail. There is an attempt at creating architecture documentation but this does not coherently present a system that would meet requirements. The architecture may be too simple, for example simply representing the prototype built for part 1, or it might be significantly incomplete. Many of the approaches taught on the module for documenting architecture are not used.	15-19 marks
Clear fail. There is little or no attempt at architecture documentation. The documentation does not use an appropriate format and/or does not use appropriate concepts.	0-14 marks



ASSESSMENT REGULATIONS

You are advised to read the guidance for students regarding assessment policies. They are available online [here](#).

Late submission of work

Where coursework is submitted without approval, after the published hand-in deadline, the following penalties will apply.

For coursework submitted up to 1 working day (24 hours) after the published hand-in deadline without approval, **10% of the total marks available for the assessment** (i.e. 100%) **shall be deducted** from the assessment mark.

Coursework submitted more than 1 day (24 hours) after the published hand-in deadline without approval will be marked as zero but will be eligible for referral.

The full policy can be found [here](#).

Academic Misconduct

In all assessed work you should take care to ensure that the work you submit is your own. The University takes academic dishonesty and cheating very seriously and it is your responsibility to ensure that you don't attempt to cheat or become victim to cheating.

There are many different forms of academic misconduct or 'cheating'. Plagiarism is the most common and both the University library and your academic tutors are able to provide further guidance on proper citation and referencing in your assessed work.

The full Academic Misconduct Policy is available [here](#).

Useful guidance for avoiding academic misconduct can be found [here](#).

Module specific guidance on Generative AI

For this part of the coursework (assessment part 2) the following uses of generative AI **are not** acceptable:

- Using AI generated text anywhere within your report (for example copying and pasting text from a generative AI tool such as ChatGPT into the report).
- Using images and diagrams that are wholly or mostly generated by AI.

The following uses of generative AI **are** acceptable (only if declared on the cover sheet):

- Using generative AI tools for researching the area and generating ideas.
- Using generative AI for planning the structure of your submission.
- Using generative AI tools to gain feedback and suggestions on your work as you progress.
- Using spelling and grammar checkers

All students must make a declaration about their uses of AI on their submission coversheet. If a student claims they have not used generative AI but we later find they have, we will consider any use of generative AI as academic misconduct (including any of the 'acceptable' uses listed above).



AI Declaration

	Yes	No
I have read the section “module specific guidance on generative AI” in the assessment brief		
AI has been used for preparing this submission (if yes, please explain below)		

If you have used generative AI for this submission, explain which tools you have used and how.