COMM2020 Team Project
Project Specification (Project 3)
Social Study Teammates for Collaborative Learning

Usage of AI Tools: AI-Minimal

These tasks have been set to assess your problem-solving abilities, and so you may not use AI for generating code or autocomplete while completing the project.

The University of Exeter is committed to the ethical and responsible use of Generative AI (GenAI) tools in teaching and learning, in line with our academic integrity policies where the direct copying of AI-generated content is included under plagiarism, misrepresentation and contract cheating under definitions and offences in TQA Manual Chapter 12.3.

This assessment falls under the category of AI-Minimal in the University's Guidance on use of Gen AI in Assessment.

This means: You may use AI tools for checking spelling and grammar mistakes only, with no other impact on the structure or content of the assessment. This is because using GenAI tools outside of these uses prevents fair assessment of your ability to achieve module learning outcomes.

When writing your assessment, you must never use AI tools:

- For uses other than checking your spelling and grammar.
- To translate more than a word or short phrase into English.
- To upload sensitive or identifying material to an AI tool.
- To present material that has been generated by AI as your own work or the work of someone else.

When submitting your assessment, you must:

- Check the box during the submission process that confirms you have adhered to the university's academic conduct policy and the expectations on use of GenAI in your assessment brief.

NOTICE: Your coding activity must all be completed on the GitHub repository. This logs your activity as you code, and the logs will be checked for submissions that have made use of AI-generated code.

Guidelines

- This assessment is AI-Minimal: spelling/grammar checks only; no AI code generation or autocomplete.
- Your application will be demonstrated online. Provide a deployed URL for the live demo and a clear local run guide for markers.
- All study teammate 'agents' must be rule-based and explainable. Do not use GenAI tools or external LLM APIs to generate agent messages.
- If you use third-party APIs or services (e.g., authentication or hosting), handle failures gracefully and avoid embedding secrets (use environment variables).

1 Requirements

Build a web-based collaborative learning environment in which groups of students work through short learning activities (e.g., problem sets, reading discussions, design critiques). The environment includes 'Social Study Teammates'—software agents that support collaboration through structured prompts, turn-taking support, fairness nudges, and reflective checkpoints.

1.1 The practical problem

Online group study often suffers from uneven participation, unclear task structure, low-quality discussion, and a lack of accountability. Students may hesitate to contribute, dominant speakers may crowd out others, and groups may drift off-task.

Your system addresses this by providing:

- Structured activities with clear phases (understand → propose → critique → decide).
- Rule-based study teammate agents that support discussion quality and inclusive participation.
- Transparent feedback and lightweight analytics to help groups self-regulate.
- A clear audit trail of contributions (useful for reflection and accountability).

1.2 Multi-agent concept (required)

You must implement at least 3 distinct study teammate agents, each with a different role. Agents must be deterministic or rule-based (no GenAI), and must provide a brief explanation of why they intervened. Suggested roles include:

- Facilitator agent: keeps the group on the current phase, prompts quieter members, and summarises decisions using templates.
- Socratic agent: asks clarification questions when reasoning is incomplete or unsupported (template-driven prompts).

- Equity agent: monitors participation balance and triggers fair-turn prompts (e.g., 'We haven't heard from X yet').
- Evidence agent: requests citations or evidence when claims are made (e.g., 'What in the reading supports this?').

2 Users and user profiles

Your application must support three user types:

2.1 Learners (students)

- Join a study room, choose an activity, and collaborate in real time or in short turns.
- Contribute messages, answers, and votes; respond to teammate prompts.
- View a summary of the group outcome and personal contribution indicators.

2.2 Facilitators (tutors/mentors)

- Create activities using a simple authoring interface (phases, prompts, assessment criteria).
- Configure which agents are active and set thresholds (e.g., participation imbalance).
- Review session summaries and engagement analytics (without exposing sensitive content publicly).

2.3 Maintainers (developers)

- Maintain activity templates, agent rules, and deployments.
- Rely on clear documentation and reproducible setup and testing.

3 Features

3.1 Core collaboration workflow (required)

1. Study rooms: create/join a room with a unique code; assign roles (optional).
2. Activity selection: choose from seeded activities, each with phases and time/turn limits.
3. Phase control: the system guides groups through phases; facilitator can pause/advance.
4. Contributions: participants submit messages/answers; system records timestamps and author IDs.
5. Decision step: group must submit a final outcome (e.g., chosen solution, consensus summary, ranked options).
6. Session summary: auto-generated using templates (not AI) including key decisions, action items, and unanswered questions.

3.2 Agent interventions (required)

- Agents must observe events (new message, inactivity, phase change, vote result) and trigger interventions based on explicit rules.

- Each intervention must include: (a) the prompt/message, (b) the triggering condition (human-readable), and (c) the recommended next action.
- At least 12 rules across all agents (e.g., inactivity rules, imbalance rules, off-topic heuristics, missing-evidence rules).
- Users must be able to view a short 'Why am I seeing this?' explanation for any agent prompt.

3.3 Analytics and feedback (required)

- Participation view: per-user contribution counts and turn balance (aggregated at session level).
- Process view: time spent in phases, number of interventions, and common triggers.
- Quality checks: simple rubric-style flags (e.g., 'claims without evidence', 'no critique phase used').
- Export: facilitators can export a session summary and analytics as a PDF.

3.4 Technical requirements (required)

- Authentication: role-based access control (learner/facilitator).
- Security: secrets via environment variables; no secrets in repository or ELE ZIP; protect private rooms from unauthorised access.
- Accessibility: keyboard navigation for core screens, readable contrast, and mobile-friendly layouts.
- Professional practice evidence: use GitHub commits/issues/PRs; the repository URL must be included in 0_admin/submission.txt.
- Testing (mandatory): you must submit BOTH (A) an automated test suite that can be executed by the markers, AND (B) a manual end-to-end test plan with results evidence.
- Automated tests (A) must include at least 15 automated tests covering: authentication/authorisation, room creation/join, activity progression, message posting, agent triggers, and summary/export.
- Manual tests (B) must include at least 8 end-to-end scenarios (happy path + failure cases). Include expected results and screenshots/logs of completed runs in 4_technical/testing_evidence.pdf.
- Your deployment_guide.pdf must include a 'How to run tests' section with the exact commands/steps.

4 Data and seeded content

You must include seeded content so the system works during marking without requiring external data. You may add optional enrichment, but the core demo must function using your seeded content.

4.1 Minimum seeded content (required)

- At least 12 activities across at least 3 activity types (e.g., problem-solving, discussion, design critique).
- At least 6 room/session examples in your seeded dataset (including at least 2 sessions demonstrating agent interventions).
- At least 50 ‘message events’ and 20 ‘agent intervention events’ included as sample logs for analytics testing.
- At least 3 facilitator configurations (different agent thresholds and enabled/disabled settings).

4.2 Suggested data entities (example)

Entity	Key fields (minimum)
User	user_id, role, display_name (non-identifying), settings
Room	room_id/code, created_by, members, privacy settings
Activity	activity_id, title, type, phases, prompts, success criteria
Phase	phase_id, activity_id, name, rules (time/turn limits), prompts
MessageEvent	event_id, room_id, user_id, timestamp, phase_id, content, tags(optional)
Agent	agent_id, role_name, rule_set_id, explanation_templates
AgentRule	rule_id, agent_id, trigger_condition, action_template, threshold
InterventionEvent	event_id, room_id, agent_id, timestamp, trigger_rule_id, explanation_text
SessionSummary	summary_id, room_id, outcome, decisions, action_items, unanswered_questions

5 Measures of success (evaluation)

You must define and report success measures with evidence. Suitable measures include:

- Participation equity: reduction in participation imbalance compared to a baseline session without agents (using your seeded sessions or a small pilot).
- Task completion: proportion of groups reaching a final outcome within the allotted phases.
- Intervention usefulness: user ratings of prompts (e.g., quick 1–5 usefulness click) and qualitative comments.
- Process quality: increased use of critique and evidence prompts; fewer off-topic flags; clearer session summaries.

6 Process and deliverables

You must submit all group deliverables on ELE as a single ZIP. GitHub is used to evidence professional practice and must be referenced in submission.txt.

Intended Learning Outcomes assessed by the coursework

Coursework 1 and Coursework 2 assess all module Intended Learning Outcomes (ILOs):

- ILO1 – Function effectively as a member of a team.
- ILO2 – Apply an integrated or systems approach to the solution of complex problems.
- ILO3 – Apply knowledge of domain context, project and change management, and relevant legal matters including intellectual property rights.
- ILO4 – Select and apply appropriate materials, technologies, and processes, and recognise their limitations.
- ILO5 – Plan self-learning and development to support the activity of the wider team.
- ILO6 – Support an inclusive approach to teamwork and problem solving, recognising the responsibilities, benefits and importance of supporting equality, diversity, and inclusion.
- ILO7 – Evaluate the environmental and societal impact of solutions to complex problems and minimise adverse impacts.

Evidence is expected across reports, implementation, evaluation, ethical/legal materials, and teamwork/process artefacts in both sprints.

6.1 Sprint 1 (Coursework 1) – Prototype v1, report and demonstration (week 5)

Submission to ELE is a single ZIP file named GroupX_CW1.zip.

Sprint 1 deliverable expectations

- Prototype v0.1.0 (vertical slice): create/join a room → run one activity through at least two phases → agent interventions occur and explain themselves → group submits a final outcome → session summary is produced.
- At least 2 agents active with at least 6 rules implemented in Sprint 1, and evidence of triggers in a demo session.
- Testing evidence (CW1): include at least 5 automated tests running successfully and document how to run them in deployment_guide.pdf; include one manual end-to-end test run with evidence in testing_evidence.pdf.
- Prototype report must include: executive summary, prioritised requirements, architecture v1, agent rule design (rules + rationale), initial evaluation evidence, and a sprint plan for CW2.
- Ethical/legal considerations must cover privacy/safeguarding, data retention, informed participation if piloting, accessibility, and IP/licensing implications.
- Software/data inventory must list all dependencies and any datasets/templates (licence, provenance, cost model, versions).

Live demo and presentation (10 minutes) – what to cover (CW1)

7. 30 seconds: what problem you solve in group study and what the ‘study teammates’ do.
8. 6–7 minutes: demonstrate room + activity + at least two agent interventions with ‘why’ explanations.

9. 1–2 minutes: show the session summary and analytics view.
10. Final minute: what will be completed in Sprint 2 and your top risks.

6.2 Sprint 2 (Coursework 2) – Final prototype, client handover and presentation (week 11)

Submission to ELE is a single ZIP file named GroupX_CW2.zip.

Individual reflection (submitted separately by each student on ELE): reflection.pdf (800–1,000 words; includes evidence links to commits/issues/PRs; and an AI-Minimal compliance statement).

Sprint 2 deliverable expectations

- Final prototype v1.0.0: stable end-to-end study workflow across multiple activity types; at least 3 agents implemented with 12+ rules in total; 'why' explanations on all interventions.
- Facilitator authoring: facilitators can create or edit activities and configure agent thresholds; include at least one new activity created via the authoring interface.
- Client handover pack: clear deployment, operations, and maintenance guidance so another team could run and extend the system.
- Testing evidence (CW2): meet the full testing requirement (15+ automated tests + 8+ manual scenarios) and include clear pass/fail evidence in testing_evidence.pdf. Marks will be reduced if tests cannot be run by markers.
- Final evaluation: report success measures with method and limitations; include an explicit discussion of inclusion, fairness, and potential harms/mitigations.
- Updated ethical/legal and licensing materials consistent with the final system and all dependencies.

Live demo and presentation (10 minutes) – what to cover (CW2)

11. 1 minute: recap problem and what is now delivered.
12. 6–7 minutes: demo final system including authoring/configuration and multiple agent interventions.
13. 1–2 minutes: show handover pack structure and how a maintainer would add a new agent rule or new activity.
14. Final minute: evaluation highlights, limitations, and next steps.

Individual reflection (Coursework 2 – individual deliverable)

Each student must submit an individual reflection on ELE (not in the group repository). Suggested length: 800–1,000 words. This reflection is used to evidence individual learning and contribution and may be used to resolve contribution disputes.

15. Your role and contributions: describe what you owned (features, testing, documentation, deployment). Reference concrete evidence (PR links, issue IDs, commits).
16. What you learned: at least three specific technical or professional learning points linked to module outcomes (e.g., requirements negotiation, risk management, testing, deployment).
17. Challenges and how you addressed them: one technical challenge and one teamwork/process challenge; what changed as a result.
18. Responsible computing: what ethical/legal risk you personally focused on (privacy, accessibility, safety) and how you mitigated it.
19. AI-Minimal compliance statement: confirm you adhered to the brief and did not use GenAI for code generation or content generation beyond spelling/grammar checks.

7 Ethical, legal, and safeguarding boundaries

- Do not collect or display sensitive personal data. Use non-identifying display names and anonymise any pilot data.
- If you conduct a pilot with peers, obtain informed consent and include a data minimisation and retention plan.
- Agents must not provide medical, legal, or high-stakes advice; keep prompts limited to study facilitation.
- Ensure accessibility: avoid content that excludes participants; support keyboard navigation and readable layouts.

8 Marking Rubric

The same COMM2020 marking rubric applies across all project options. The rubric will be provided by the module team on ELE and used consistently for CW1 and CW2 (including the individual reflection in CW2).

[END OF SPECIFICATION]

Document owner: Module team (COMM2020)

Date: 23 December 2025