UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Faculty of Engineering and Information Sciences
School of Computing & Information Technology

CSIT321: Project

Capstone Project - Spring 2025

Project Specification Presentation

Team Members

Noor Ahmed - 7295297
Nishad Gyawali - 8124553
Chelsea Okan - 8438675
Azwad Ahnaf Zihan - 7663055

University of Wollongong

Liverpool Campus

Subject Coordinator

Dr. Khin Win

8th September, 2025

Task Allocation (Presentation)

At the start of the project, responsibilities were divided among team members. The table below records the initial allocation, with notes indicating where sections were later shortened, revised, or re-written to maintain consistency.

Section	Primary Author (Assigned)	Notes (Revisions / Secondary)
1. Team Roles and Responsibilities	Chelsea	
2. Problems and Goals	Chelsea	
3. Requirements and Scope	Chelsea	
4. Market and Analysis	Chelsea	Originally assigned to Azwad but no deliverable provided
5. Design Principles	Noor	Re-written and edited by Chelsea
6. User Flow & Mockups	Noor	Rewritten and edited by Chelsea
7. Features	Chelsea	Originally assigned to Nishad but no deliverable provided
8a. Technical Requirements (Backend)	Chelsea	Originally assigned to Nishad but no deliverable provided
8b. Technical Requirements (Infrastructure)	Chelsea	Originally assigned to Azwad but no deliverable provided
9. Supervisor/Client Interaction	Chelsea	Originally assigned to Nishad but no deliverable provided
10a. Risk Overview	Chelsea	
10b. Risk (High Priority)	Chelsea	
10c. Risk (Project Management)	Chelsea	
10d. Risk (Technical)	Chelsea	
10e. Usability and Organisational	Chelsea	

11a. Timeline	Chelsea	Originally assigned to Nishad but no deliverable provided
11b. Gantt Chart	Chelsea	Originally assigned to Nishad but no deliverable provided
12. Conclusion	Chelsea	
Speaker's Notes	Chelsea	Noor wrote the original speaker's notes for his section before they were re-edited by Chelsea
Presenting	Nishad	Nishad presented on presentation day

Colour Coding Key

- Chelsea
- Noor
- Azwad
- Nishad

Note: All document editing, integration, and quality assurance were completed by Chelsea to maintain consistency of style, tone, and structure across sections. No additional quality control was performed outside of individual section drafting.

Slide 1A - Team Members & Roles

Notes:

- "This is the SmartResearch team, supervised by Dr. Jack Yang.
- The team's divided into clear roles to make delivery smoother.
    - Chelsea is the Project Manager and QA lead, also working as a swing developer. She manages the GitHub repository, project documentation, the risk register, and consistency checks across the system.
    - Noor is our Frontend Lead, focusing on building the React-based interface and making sure user flows are simple and intuitive."

Slide 1B - Team Members & Roles

Notes:

    - "Nishad is the Technical Lead. He's responsible for the NLP pipeline and clustering models, which form the core intelligence of the system.
    - Azwad is the Infrastructure Lead, handling integration and deployment to keep the system stable.
- This structure gives each person ownership of their area while letting the team progress in parallel without bottlenecks."

Slide 2A - Problems & Goals

Notes:

- "The problem SmartResearch addresses is the overwhelming volume of academic papers. With millions published every year, research teams lose days just screening for relevance.
- The current process is slow - searching, skimming, and reading - and most existing tools only solve one part of that process. For example, Scholarcy summarises, ResearchRabbit maps citations, and Elicit extracts claims. But none deliver a full, end-to-end workflow."

Slide 2B - Problems & Goals

Notes:

- "The goal is to deliver a single web application that turns days of screening into minutes.
- Researchers will be able to upload batches of PDFs, generate consistent four-field summaries under 120 words, cluster papers by theme with keyword labels, and export results to CSV or JSON.

- The vision is a workflow that's fast, structured, and interpretable - something researchers can trust in practice."

Slide 3A - Requirements & Scopes / Functional Requirements
Notes:

- "The system requirements are split into functional, non-functional, and constraints.
- Functionally, it needs to handle batch uploads of at least ten PDFs, generate structured summaries in four fields - objective, method, findings, and limitations - and cluster papers with automatic keyword labels.
- These requirements translate directly to researcher needs: faster screening, consistent summaries, and thematic grouping without manual effort.

Slide 3B - Requirements & Scopes / User Interface
Notes:

- "The interface follows the same workflow researchers already use, just compressed into four views.
- Upload View for adding PDFs, Browse View to search and filter summaries, Cluster View to explore grouped themes, and Export View for downloading structured results.
- By mapping one screen to each stage, the design stays simple and keeps the process intuitive: upload, review, cluster, and export."

Slide 3C - Requirements & Scopes / Non-Functional Requirements
Notes:

- "Non-functional requirements make the system usable at scale.
- It must process a 10-paper batch in under 60 seconds, keep every paper accessible in three clicks or fewer, remain stable when encountering malformed PDFs, and preserve privacy through local-only processing.
- These requirements ensure the system is not just functional, but reliable, fast, and trustworthy."

Slide 3D - Requirements & Scopes / Non-Functional Requirements
Notes:

- "Our constraints are clear. The system will only support English, text-selectable PDFs. It must use open-source components, and everything must be developed within the semester timeframe.

- These boundaries keep the project realistic, prevent scope creep, and keep us aligned with research needs instead of chasing features we can't deliver."

Slide 4: Market and Analysis
Notes:

- "Existing tools solve only fragments of the workflow.
    - Scholarcy summarises individual papers but doesn't cluster them.
    - ResearchRabbit maps citations but doesn't summarise.
    - Elicit extracts claims but has no bulk upload or clustering.
- SmartResearch is unique because it integrates the entire pipeline - upload, summarise, cluster, and export.
- That end-to-end process reduces screening time from days to minutes, which is a real competitive advantage."

Slide 5 - Design Principles (UI/UX)
Notes:

- "The interface is guided by four design principles.
    - Simplicity: only the essential workflow is included -> upload, review, cluster, export.
    - Clarity: all summaries follow the same four-field structure, so results are easy to scan and compare.
    - Accessibility: navigation stays intuitive, with any document reachable in three clicks.
    - Consistency: the same layout patterns are used across screens, so researchers always know where to look.
- These principles were chosen because researchers under time pressure need predictable, low-friction interfaces. For example, summaries in four fixed fields let them scan findings instantly, while consistent layouts across screens reduce cognitive load."

Slide 6 - User flow & Mockups
Notes:

- "Here's how a researcher moves through SmartResearch.
- First, they upload multiple PDFs through drag-and-drop with a progress bar.
- Then they move to the All Papers Screen, where summaries, clusters, and filters can be browsed in a table view.
- From there, they can switch to the Cluster Screen to see papers grouped by keywords.

- Finally, they use the Export Screen to download results into CSV or JSON for use in other tools.
- This flow keeps the process structured and easy to follow: Upload → Browse → Cluster → Export."

Slide 7 - Features
Notes:

- "These features directly deliver on the requirements we set.
    - Fast Uploads: batch uploads of at least ten PDFs with drag-and-drop and a progress bar, reducing friction for researchers.
    - Smart Summaries: every paper gets a structured summary in four fields, making results easy to scan side by side.
    - Thematic Clusters: papers are auto-grouped with keyword labels, so themes are visible without manual coding.
    - Easy Export: CSV and JSON outputs allow results to be reused in citation managers or collaborative projects.
- Together, these features transform the specification into a usable workflow: upload, summarise, cluster, and export."

Slide 8A - Technical Requirements / Backend
Notes:

- "The backend is designed to meet the functional requirements directly.
    - Summaries are produced in four fields - objective, method, findings, and limitations - giving researchers consistent, comparable outputs.
    - Clustering uses scikit-learn with keyword extraction, so groups are meaningful, not just raw clusters.
    - FastAPI provides the API layer, chosen for speed and lightweight performance.
    - SQLite serves as the database, storing papers, summaries, and cluster data reliably while keeping development realistic for the semester.
- Each choice was made to balance functionality, reliability, and scope."

Slide 8B - Technical Requirements / Infrastructure
Notes:

- "The infrastructure ensures stability and compliance with non-functional requirements.
    - Docker is used to containerise the application so it runs consistently across environments.

    - CI/CD pipelines through GitHub Actions automate testing and integration, reducing bugs and saving time.
    - Privacy is maintained through local-only processing - no data leaves the system.
    - Reliability is supported with caching, error handling, and recovery routines, so one failure doesn't break the workflow.
- This setup keeps the system stable, private, and deployable."

Slide 9 - Supervisor/Client Interaction
Notes:

- "The team meets fortnightly with our supervisor, Dr. Jack Yang, to validate scope and check progress.
- His feedback has directly shaped development -> for example, after his suggestion we are now adopting GitHub Projects as a Kanban board for sprint tracking.
- We also formalised escalation processes so uncompleted tasks don't block delivery.
- Dr. Yang has confirmed that our requirements scope and interface direction align with research workflows, ensuring the system meets user expectations."

Slide 10A - Risks Overview
Notes:

- "The team maintains a live risk register, updated weekly in GitHub.
- Risks are grouped into four categories: High Priority, Project Management, Technical, and Usability & Organisational.
- This structured approach makes risks visible and ensures they are actively managed rather than ignored."

Slide 10B - Risks / High Priority
Notes:

- "High-priority risks include summary fidelity, cluster coherence, and overall consistency.
- We mitigate these through hybrid models, rubric checks, parameter tuning, keyword labelling, and benchmarking against samples.
- This reduces the risk of inaccurate outputs undermining researcher trust."

Slide 10C - Risks / Project Management
Notes:

- "Project management risks include schedule slippage, scope creep, and uneven team workloads.
- Mitigations include milestone gates, fortnightly sprints, GitHub issue tracking, clear acceptance criteria, and a RACI structure for rotating tasks.
- These measures keep delivery on track and balanced across the team."

Slide 10D - Risks / Technical
Notes:

- “Technical risks cover performance, compatibility, and reliability.
- To address these, we use caching, batch jobs, asynchronous processing, dependency locks, stress testing, and recovery routines.
- This ensures the system remains stable under real-world use.”

Slide 10E - Risks / Usability & Organisational
Notes:

- “Usability and organisational risks include confusing interfaces, low adoption, and coordination challenges.
- These are managed with UX testing, simplified workflows, quick-start guides, regular feedback loops, and shared agendas.
- We review risks weekly, close them once exposure drops below three, and keep governance tight.”

Slide 11A - Timeline and Sprint Plan
Notes:

- “The sprint plan runs in fortnightly blocks aligned with milestones.
    - Sprint 1, weeks 4–5, completed the specification draft, GitHub repo setup, and risk logging.
    - Sprint 2, weeks 6–7, delivered wireframes, workflows, and this presentation.
    - Sprints 3 and 4, weeks 8–11, focus on frontend build, backend integration, and prototype preparation, with the mid-semester break as a buffer.
    - Sprint 5, week 12, is polish and integration into a working prototype for the demo.
- That completes Phase 1. Next semester, Phase 2 extends this into a full system for the Design Progress milestone and final delivery.”

Slide 11B - Gantt Chart
Notes:

- “And here is the Gantt chart. It expands the sprint plan into concrete tasks and deadlines, showing exactly how the project stays on schedule.”

Slide 12 - Conclusion
Notes:

- “SmartResearch addresses the challenge of overwhelming academic literature by delivering a single streamlined workflow: upload, summarise, cluster, and export. Existing tools only solve fragments of this process; SmartResearch integrates the full pipeline, cutting screening time from days to minutes.

- The design is guided by simplicity, clarity, accessibility, and consistency, making it practical for researchers under pressure.
- Risks are identified, governance is in place, and delivery is structured through a sprint plan.
- Functional and non-functional requirements are satisfied, the interface is justified for its target users, and stakeholder validation confirms alignment with research needs. That concludes our presentation. We welcome any questions.”