

Programme: Changemakers: Creativity and Value Creation

Module Code: ACCA5033

Student Name: Vivien Tam

Student Number: 2505794

Q.1: Innovation

Innovation refers to the transformation of creative concepts into practical solutions that generate value for individuals, organisations, and society. It has consistently shaped human progress by addressing challenges, improving efficiency, and enhancing everyday life. Creativity fuels innovation by encouraging unconventional thinking, while innovation applies these ideas to produce meaningful and measurable change. In modern contexts, innovation is closely linked to technological advancement, economic growth, and organisational competitiveness, particularly as digital transformation accelerates across industries.

Historically, innovation has enabled major societal shifts. Foundational developments such as language, agriculture, the wheel, and the printing press reshaped communication, mobility, and knowledge sharing. More recent breakthroughs—including electricity, computing, the internet, and mobile technologies—have fundamentally changed how people work, communicate, and access information. These examples highlight that innovation encompasses both the creation of new technologies and the enhancement of existing systems to meet emerging needs [1].

Within the computing sector, innovation is especially critical due to rapid technological evolution. Organisations increasingly depend on automated and intelligent systems to reduce manual effort, minimise errors, and improve productivity. Technologies such as artificial intelligence, cloud computing, distributed systems, and automated workflows enable businesses to process large datasets efficiently and deliver reliable, responsive services. As data-driven decision-making becomes more prevalent, automated software solutions play a vital role in improving operational performance and user satisfaction.

From a developer's perspective, creativity and innovation are essential competencies. Designing effective software requires analytical thinking, problem-solving, and the ability to translate real-world challenges into functional solutions [2]. This project embodies these principles by addressing inefficiencies associated with manual file processing. Instead of requiring users to wait for synchronous operations, the proposed system introduces a background task queue that validates, processes, and monitors CSV files automatically. This demonstrates how innovative approaches can enhance user experience, increase processing efficiency, and improve system reliability.

Innovation also involves refining existing workflows. The proposed Task Queue Monitoring and Automation System incorporates automated processing, real-time monitoring, structured error handling, and retry mechanisms. These features reduce administrative workload and provide users with greater visibility of task progress. By applying software engineering principles to automate repetitive tasks, the project illustrates how creativity and innovation can be combined to deliver practical, value-driven solutions.

Q.2: Ideation

Several potential project ideas were explored before selecting the final concept. Each idea addressed a distinct real-world problem within the field of computing.

Project Idea	Area of Study
Student Attendance Management System	Web Development
Online Inventory Management System	Database Systems
Personal Finance Tracker	Software Engineering
Web-Based Task Queue Monitoring and Automation System	Software Engineering / Web Development
Appointment Booking System	Information Systems

After evaluating each option, the Web-Based Task Queue Monitoring and Automation System was selected as the most suitable project [3], [4]. This decision was based on its strong alignment with key software engineering concepts, including asynchronous processing, task queue management, database design, automation, and real-time monitoring. The project also provides opportunities to apply both theoretical knowledge and practical programming skills developed throughout the Computing programme.

Project Title

Web-Based Task Queue Monitoring and Automation System

Aim

To design and develop a web-based system that automates CSV file processing using background task queues while providing real-time monitoring, automated notifications, and robust error handling.

SMART Objectives

1. Investigate task queue architectures, background processing techniques, and suitable web technologies during Week 1.
2. Design the system architecture, database schema, and user interface by the end of Week 2.
3. Implement core functionality—including CSV upload, validation, queue management, and background processing—by Week 5.
4. Add real-time monitoring, notifications, retry mechanisms, and comprehensive error handling by Week 6.
5. Conduct testing, evaluation, and documentation before final submission at the end of Week 7.

Project Description

The system enables users to upload CSV files through a web interface. Each file is validated and then placed into a background queue for asynchronous processing. This prevents delays associated with synchronous operations and allows users to continue interacting with the system.

A central dashboard displays the status of each task—Queued, Processing, Completed, or Failed—along with real-time updates. When a task fails, detailed error information is provided, and users may retry processing without re-uploading the file. This improves usability and reduces redundant work.

Q.3: Research

Successful completion of the project requires knowledge from multiple areas of computing. The following courses and technologies contribute directly to the system's development:

Course / Resource	Contribution to the Project
Software Engineering	Supports system analysis, design, testing, and project management.
Web Development (ASP.NET Core, HTML, CSS, JavaScript)	Used to develop the web application and interactive dashboard.
Database Systems (SQL Server & Entity Framework Core)[7]	Stores user information, uploaded files, task status, and processing logs.
Distributed Systems and Background Processing (Hangfire)	Implements task queues, background job scheduling, and retry mechanisms.
Agile Software Development[5][6][7]	Provides an iterative development methodology that supports incremental delivery, continuous testing, and project management using Jira. This approach was selected during the project's planning stage because it allows individual features such as task submission, queue processing, notifications, and monitoring to be developed and tested in manageable sprints.

These resources form the technical foundation for developing a scalable and maintainable automation system capable of efficient background processing.

Project Management- Q1

Project scope defines the boundaries, objectives, deliverables, and resources required to complete a project. A clear scope ensures focus and reduces the risk of unnecessary features or delays. This project focuses on developing a web-based system that automates CSV file processing and provides real-time monitoring.

The system allows users to upload CSV files, which are validated and placed into a background queue. Asynchronous processing improves responsiveness and supports multiple concurrent tasks. A monitoring dashboard displays task statuses, and failed tasks can be retried without re-uploading files. To maintain feasibility within the seven-week timeframe, the project is limited to core functionality. Features such as authentication, role-based access control, cloud deployment, distributed processing, and mobile optimisation are excluded. This ensures the project remains aligned with its Minimum Viable Product (MVP) goals.

Technologies and Resources

The project will be developed using the Microsoft technology stack alongside modern software engineering tools that support asynchronous processing and Agile development.

Technology / Resource	Purpose
ASP.NET Core MVC	Develop the web application
C#	Backend programming language
SQL Server	Database management
Entity Framework Core	Database access and object-relational mapping
Hangfire	Background task queue processing
HTML5, CSS3, Bootstrap	User interface development
JavaScript / AJAX	Real-time dashboard updates
Visual Studio 2022	Software development environment
Git & GitHub	Version control and source code management
Jira	Sprint planning and Agile project management
Draw.io	UML diagrams and system design

These technologies provide the functionality required to develop a scalable and maintainable web application while supporting industry-standard software engineering practices.

Project Management-Q2

A Gantt chart was developed to visualise the project timeline, showing task durations, dependencies, and overlapping phases. This helps ensure each stage is completed within the allocated timeframe. The project begins with ideation and planning, followed by risk analysis and selection of the development methodology. System design activities—including database modelling and interface planning—are completed before implementation to reduce architectural changes later. Prototype development overlaps with design, reflecting Agile’s iterative nature [6], [7]. Core functionality is implemented incrementally, including background processing, validation, monitoring, notifications, and retry mechanisms. User testing occurs before final submission, allowing improvements to be made based on feedback.

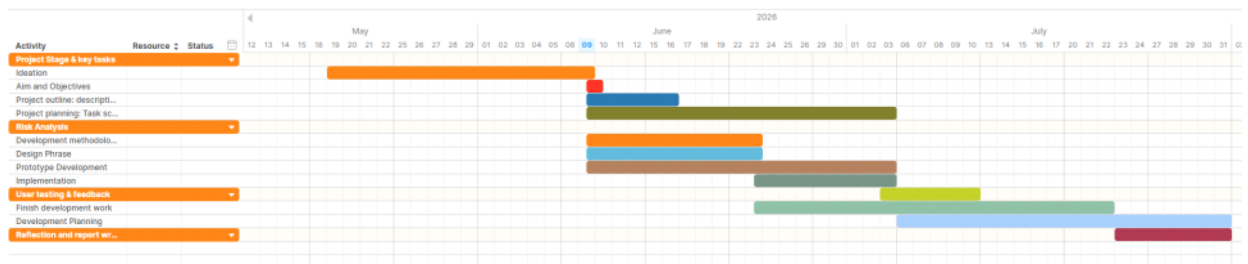


Figure 1: Gantt chart designed for this project, including planning and project lifespan

Project Management-Q3

Software projects often encounter technical, organisational, and academic risks. Identifying these early enables effective mitigation.

Technical Risks

Selecting overly complex or unfamiliar technologies may increase development time [10]. This project mitigates this by using well-documented tools such as ASP.NET Core, SQL Server, Entity Framework Core, and Hangfire.

Excessive reliance on generative AI may introduce insecure or inaccurate code [10], [11]. AI will therefore be used only as a supplementary tool, with all generated code manually reviewed.

Dependency on third-party libraries may lead to compatibility issues if APIs change [6]. This risk is reduced by relying primarily on stable Microsoft technologies.

Project Management Risks

Scope creep can delay completion and reduce quality [10]. The MVP approach ensures essential features are prioritised.

Time management is critical due to the seven-week schedule. Agile sprints and Jira tracking support realistic planning and continuous progress monitoring.

Version control issues may arise if changes are not committed regularly [6]. GitHub ensures consistent version tracking.

Academic and Resource Risks

Misuse of AI may lead to academic integrity concerns. All AI-assisted content will be critically evaluated and rewritten.

Hardware or software limitations may affect deployment [10], [11]. Testing across environments and maintaining backups will reduce this risk.

Project Development method & Design work - Q1

Agile Software Development, specifically the Scrum framework, was selected due to its suitability for iterative, time-boxed development. Agile supports continuous testing, flexibility, and incremental delivery, making it ideal for a seven-week project.

Sprints are used to deliver functional components progressively. Early sprints focus on requirements gathering and architecture design, followed by implementation of file upload, database integration, task queues, monitoring, notifications, and error handling. Jira supports sprint planning, backlog management, and progress tracking [1].

Project Development method & Design work - Q2

Before implementation begins, several design artefacts are produced to define the structure and behaviour of the proposed system[12]. These diagrams reduce ambiguity, improve communication, and provide a clear blueprint for development.

An Entity Relationship Diagram (ERD) has been designed to model how data will be stored within the application database.

The system contains four primary entities:

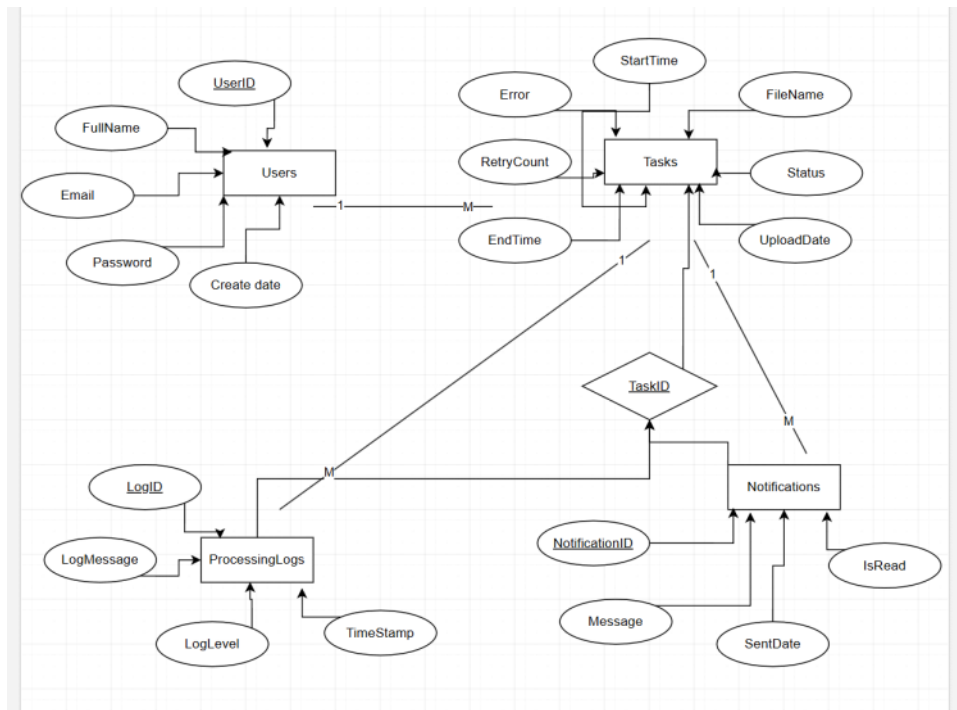


Figure 2: illustrate the ERD for this project

The ERD defines four main entities: Users, Tasks, ProcessingLogs, and Notifications. These entities support user management, task tracking, processing history, and user alerts. Relationships between entities ensure data integrity and efficient retrieval.

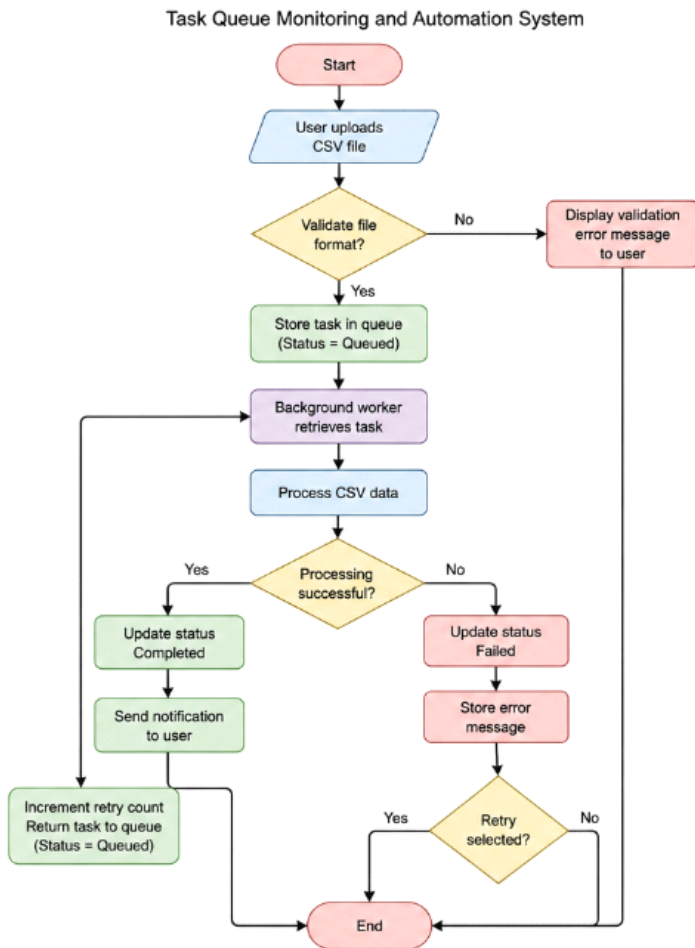


Figure 3: illustrates the flowchart

The flowchart outlines the lifecycle of a task—from upload and validation to background processing, error handling, and retry logic. This visual representation supports clear understanding of system behaviour.

Project Development method & Design work - Q3

Although the software implementation is still in its early stages, several practical activities have already been completed to establish a structured foundation for the project. At this stage, the primary focus has been on organising the development process using Agile principles rather than immediately beginning software implementation. This approach ensures that project requirements, timelines, and deliverables are clearly defined before coding begins.

Jira Project Setup

A Jira workspace was created to manage tasks, sprints, and progress tracking [14]. The backlog includes tasks related to research, design, prototype development, implementation, testing, and documentation..

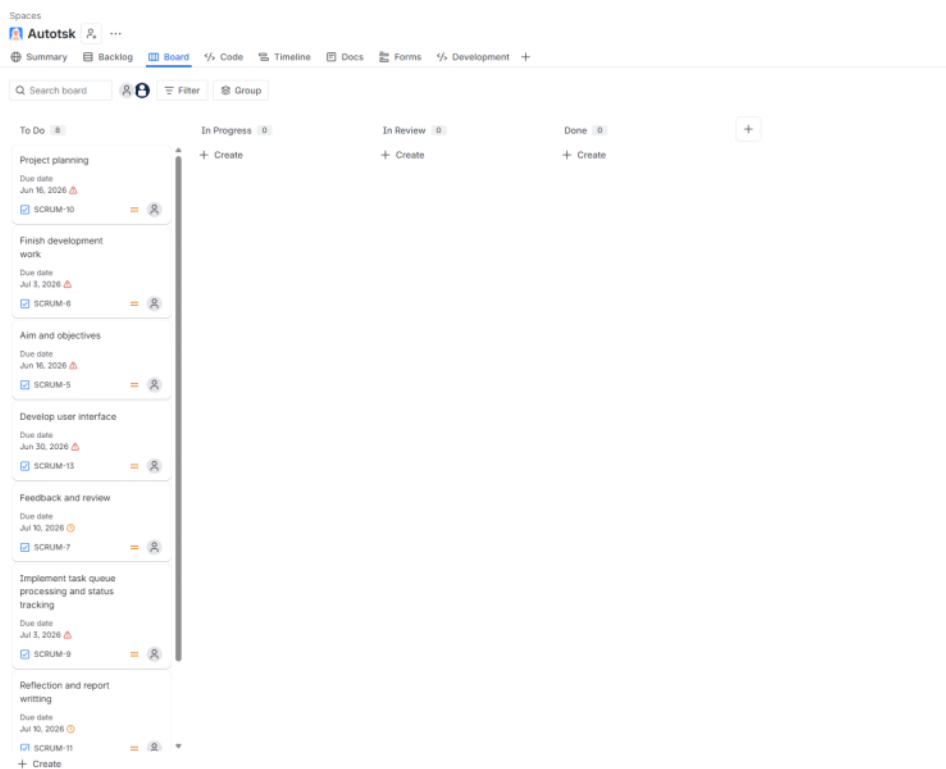


Figure 4: Jira project workspace showing the product backlog and project tasks.

Sprint Planning and Task Management

Tasks were organised into weekly sprints aligned with the project timeline. Jira's Agile board provides visibility of task status and supports incremental development.

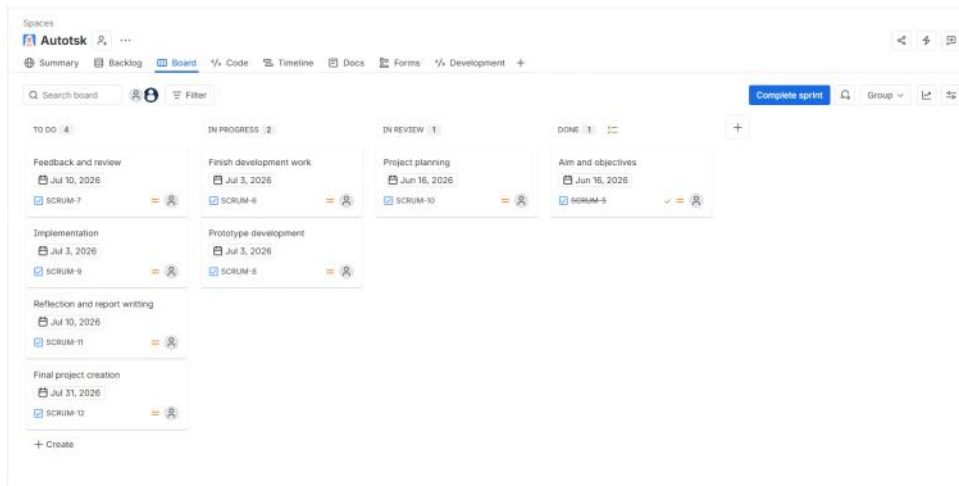


Figure 5: Sprint planning.

Project Documentation

Alongside project management activities, the initial project documentation has been prepared. This includes defining the project aim, SMART objectives, project scope, risk assessment, Gantt chart, and system design artefacts. Maintaining documentation throughout development will improve traceability and support the final evaluation of the completed system.

The next phase of implementation will involve creating the ASP.NET Core project within Visual Studio, designing the SQL Server database, and implementing the file upload functionality before integrating Hangfire for background task processing.

References

09 July 2026 21:49

- [1] *The 71 Most AMAZING Innovations of All Time*, YouTube transcript, Unit 1 Topic 1 Learning Materials, 2026.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Harlow, England: Pearson Education, 2016.
- [3] Microsoft, "Background tasks with hosted services in ASP.NET Core." [Online]. Available: <https://learn.microsoft.com/aspnet/core/fundamentals/host/hosted-services>. Accessed: Jul. 9, 2026.
- [4] Hangfire, "Hangfire Documentation." [Online]. Available: <https://docs.hangfire.io/>. Accessed: Jul. 9, 2026.
- [5] K. Beck *et al.*, "Manifesto for Agile Software Development," 2001. [Online]. Available: <https://agilemanifesto.org/>. Accessed: Jul. 9, 2026.
- [6] Atlassian, "What is Agile?" [Online]. Available: <https://www.atlassian.com/agile>. Accessed: Jul. 9, 2026.
- [7] K. Schwaber and J. Sutherland, *The Scrum Guide: The Definitive Guide to Scrum*, Nov. 2020. [Online]. Available: <https://scrumguides.org/scrum-guide.html>. Accessed: Jul. 9, 2026.
- [8] OpenAI, *GPT-4.1 System Card*, 2025. [Online]. Available: <https://openai.com/index/gpt-4-1/>. Accessed: Jul. 9, 2026.
- [9] Quality Assurance Agency for Higher Education (QAA), *Generative Artificial Intelligence in Assessment*, 2024. [Online]. Available: <https://www.qaa.ac.uk/>. Accessed: Jul. 9, 2026.
- [10] Microsoft, "Entity Framework Core documentation." [Online]. Available: <https://learn.microsoft.com/ef/core>. Accessed: Jul. 9, 2026.
- [11] Microsoft, "SQL Server documentation." [Online]. Available: <https://learn.microsoft.com/sql>. Accessed: Jul. 9, 2026.
- [12] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2002.
- [13] Atlassian, "Jira Software Documentation." [Online]. Available: <https://support.atlassian.com/jira-software-cloud/>. Accessed: Jul. 9, 2026.