Singapore Institute of Technology - University of Glasgow
Joint Degree in Computing Science Degree Programme

CSC3101 Capstone Project

Please complete the following form and attach it to the Capstone Report submitted.

Capstone Period: 01 SEP 2025 to 10 APR 2026
Assessment Trimester: Start Trimester
Project Type: Industry

Work Supervisor Details & Declaration:

Name: **FRANCIS TEO EE CHEOK**
Designation: **CHIEF EXECUTIVE OFFICER**
Department: **MANAGEMENT**
Email Address: **FRANCIS.TEO@WIZVISION.COM**
Contact Number: **+65 9815 3140**

Academic Supervisor Details:

Name: **JACOB ABRAHAM**
Designation: **SENIOR PROFESSIONAL OFFICER**
Email Address: **JACOB.ABRAHAM@SINGAPORETECH.EDU.SG**
Contact Number: **+65 6592 1448**

Student Particulars & Declaration:

Name of Student: **ONG JIA EN, DARRYL**
Student ID: **2301402**

I hereby acknowledge that I have engaged and discussed with my **Academic Supervisor** and **Work Supervisor** on the contents of this Capstone Report (Problem Definition) and have sought approval to release the report to the Singapore Institute of Technology and the University of Glasgow.

Signature

Date: 25 SEP 2025

END OF FORM

Singapore Institute of Technology - University of Glasgow
Joint Degree in Computing Science Degree Programme

Capstone Report (Problem Definition) "Lifecycle Monitoring and Governance of Enterprise Batch Jobs"

For **Start Trimester** from 01 SEP 2025 to 10 APR 2026

ONG JIA EN, DARRYL
Student ID: 2301402

Work Supervisor: *FRANCIS TEO EE CHEOK, CHIEF EXECUTIVE OFFICER, WIZVISION*

Academic Supervisor: *JACOB ABRAHAM, SENIOR PROFESSIONAL OFFICER, SINGAPORE INSTITUTE OF TECHNOLOGY*

Submitted as part of the requirement for CSC3101 Capstone Project

1. Introduction

Batch jobs remain a critical but often overlooked component of enterprise systems. They orchestrate repetitive tasks such as data synchronization, financial report generation, and routine system maintenance. Despite their ubiquity, batch jobs frequently operate in the background with little direct oversight. Once deployed, many jobs run for years without review. As the number of scheduled jobs grows into the hundreds or even thousands, administrators face challenges in identifying which jobs are still relevant, which have silently failed, and which may be consuming resources unnecessarily. Forgotten or redundant jobs increase operational risk, hinder maintainability, and reduce trust in enterprise automation.

1.1 Motivation

The motivation for this project arises from the gap between the importance of batch jobs and the limited visibility organisations have into their lifecycle. Currently, administrators typically monitor jobs using logs or simple dashboards that show success or failure after execution. These tools are inadequate for detecting silent degradation, redundant workloads, or governance drift, leaving many jobs running in the background unchecked.

The idea for this project stems directly from my experience working with enterprise batch jobs during my Integrated Work Study Programme. In practice, I observed that when hundreds of jobs were deployed and allowed to run for long periods of time without regular review, there is potentially outdated or redundant jobs that remain active in production. I also encountered a situation where a job appeared to be in progress but had in fact failed silently. Due to the fact that there was no alert or feedback mechanism in place, I only discovered the error later on when I returned to the terminal console and saw the error message. Experiences like this highlighted to me how important it is for batch systems to provide clear feedback and governance mechanisms and inspired the ideation of this project.

The purpose of this capstone project is therefore to design and implement a Batch Job Lifecycle Monitor, a system that provides transparent oversight of jobs from creation to retirement. By introducing features such as lifecycle tracking and governance checks, the project aims to help enterprises reduce risks, improve maintainability, and ensure their scheduled jobs remain relevant and efficient over time.

1.2 Problem Definition

1.2.1 Insufficient Visibility into Batch Job Lifecycles

Existing enterprise monitoring tools are often limited to binary outcomes such as “success” or “failure”, providing little information about the intermediate states a job undergoes such as queued, waiting, or partially completed. This lack of granularity hinders administrators’ ability to diagnose issues or understand long-term job behaviour. For example, a job may repeatedly retry due to transient conditions yet still be recorded as “succeeded”, concealing underlying inefficiencies. Without end-to-end job tracking and reporting, administrators are forced into reactive troubleshooting rather than proactive oversight.

1.2.2 Silent Drift and Hidden Inefficiencies

While not necessarily failing outright, batch jobs may degrade subtly over time. They might execute longer than expected, process smaller data sets, or run at times that are no longer align with business needs. As most dashboards focus on just success or failure, these inefficiencies remain unnoticed. Over months or years, such hidden drifts can lead to resource waste, missed dependencies, or misaligned reporting cycles. The lack of built-in governance tools to flag such behaviour means administrators often lack the awareness to take corrective action.

1.2.3 Accumulation of Outdated or Redundant Jobs

Over the years, organisations accumulate jobs that are no longer required, duplicated across teams, or left behind after business changes. Without structured governance, these outdated or redundant jobs continue to consume infrastructure resources and complicate maintenance. The presence of unused jobs makes it harder for teams to audit the environment, increase cognitive load during troubleshooting, and heightens the risk of human error. The absence of lifecycle review mechanisms results in a cluttered and fragile automation ecosystem.

1.3 Literature Review

Several strands of research have addressed monitoring and scheduling in batch and workflow systems, but they leave gaps in lifecycle governance and redundancy management. A visualization tool, BatchLens, was proposed to help administrators analyse batch jobs by exploring runtime patterns and resource usage [1]. While useful for diagnosing job behaviour, the focus remains on visualization rather than providing mechanisms for long-term governance or job retirement. A containerized batch monitoring system was developed to capture CPU, memory, and Input/Output metrics for jobs in distributed environments [2]. This demonstrates the feasibility of detailed runtime monitoring, yet it does not address organisational issues such as outdated or redundant job persisting in production.

In parallel, efforts to standardize communication between workflow managers and schedulers have emerged. The Common Workflow Scheduler Interface (CWSI) defines a uniform interface for exchanging scheduling and execution information [3]. While this advances interoperability and portability, it remains focused on system integration rather than governance of large and evolving job ecosystems. Workflow scheduling strategies have also been compared in simulated distributed environments, showing how different heuristics perform under uncertainty [4]. These results highlight the complexity of scheduling decisions but concentrate on execution efficiency instead of maintainability of long-lived job fleets.

Together, these works advance the state of monitoring, visualization, interoperability, and scheduling efficiency. However, they largely overlook the “set-and-forget” problem common in enterprise environments, where forgotten jobs, hidden drift, and redundancy accumulate over time. The proposed Batch Job Lifecycle monitor aims to fill this gap by focusing not only on execution monitoring but also on visibility, governance, and maintainability across the full lifetime of scheduled jobs.

1.4 Summary

Taken together, these challenges reveal a persistent gap between what research and tools currently provide and what enterprises require. Existing work has advanced visualization, monitoring, standardization, and scheduling efficiency, yet the practical problem of forgotten jobs, hidden inefficiencies, and redundant workloads remain unresolved. Current solutions

focus heavily on execution metrics and interoperability, leaving long-term governance unaddressed. The proposed Batch Job Lifecycle Monitor fills this gap by combining visibility, tracking, and governance features to ensure that jobs remain relevant, efficient, and manageable throughout their entire lifecycle.

2. Project Objectives

The objectives of this project are designed to address the operational challenges of managing batch jobs in enterprise environments and to improve overall maintainability.

- **Enhance Visibility into Batch Job Executions:** Develop an end-to-end tracking system that captures detailed job states, runtimes, and outcomes, providing administrators with actionable insights beyond basic success or failure reporting.

- **Identify Inefficiencies Through Rule-Based Governance:** Implement a configurable rules engine that applies defined conditions such as maximum runtime thresholds, inactivity periods to identify jobs exhibiting inefficiencies or drift.

- **Introduce Governance Mechanisms for Redundancy and Obsolescence:** Provide tools to flag and review outdated, redundant, or inactive jobs, enabling structured lifecycle governance and reducing the clutter of unmanaged workloads.

- **Enable Automated Alerting for Job Anomalies:** Design and implement an alerting mechanism that notifies administrators when jobs breach configured rules or thresholds, ensuring timely intervention and reducing reliance on manual log reviews.

2.1 High-Level Target

- **Rule-Based Detection:** Ensure that all jobs violating configured rules are correctly flagged, with minimal false positives when validated against test cases.

- **Governance:** Enable administrators to identify and categorize at least 10% of jobs as redundant, obsolete, or inactive during lifecycle reviews, thereby reducing clutter in the job ecosystem.

- **Alerting:** Ensure that notification for breached rules or anomalies is delivered within one minute of detection to support timely intervention.

2.2 Deliverables

1. A fully functional Batch Job Lifecycle Monitor application with the following features:
    - Job execution tracking and history logging
    - Configurable rule-based governance engine
    - Job redundancy and obsolescence flagging
    - Automated alerting system
    - User dashboard for monitoring and governance

2. Capstone Interim Report due on 7th December 2025, with the following content:
    - Introduction
    - Literature Review

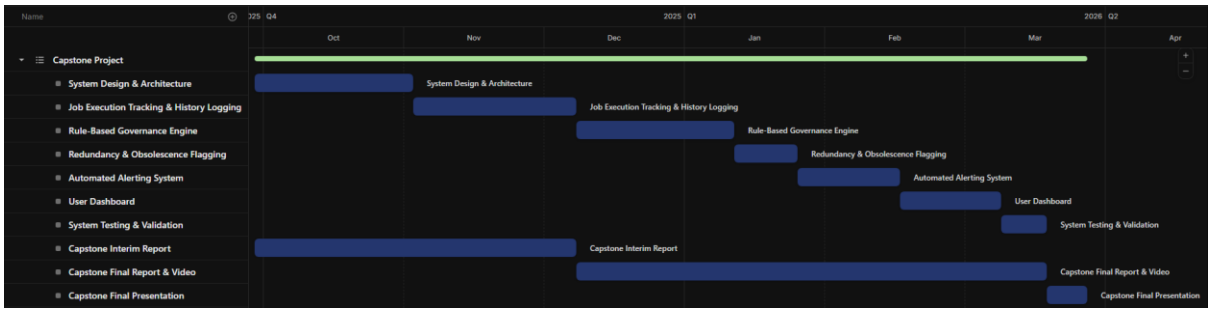
- Methodology / Proposed Design
- References
- Knowledge and Training Requirements

3. Capstone Final Report due on 18th March 2026, with the following content:
    - Introduction
    - Literature Review
    - Methodology / Proposed Design
    - Results and Analysis
    - Project Management
    - Conclusion
    - References
    - Knowledge and Training Requirements

4. Capstone Video Submission due on 18th March 2026

5. Capstone Final Presentation the week after Final Report and Video Submission

2.3 Timeline



- **System Design & Architecture**
  29th September 2025 – 2nd November 2025
  Define architecture, technology stack, and integration approach.

- **Job Execution Tracking & History Logging**
  3rd November 2025 – 7th December 2025
  Develop core tracking features to capture job states, runtimes, and history.

- **Rule-Based Governance Engine**
  8th December 2025 – 11th January 2026
  Implement configurable rules such as runtime thresholds and inactivity detection.

- **Redundancy & Obsolescence Flagging**
  12th January 2026 – 25th January 2026
  Build features to identify outdated, redundant, or inactive jobs.

- **Automated Alerting System**
  26th January 2026 – 15th February 2026
  Implement notifications for jobs breaching configured rules.

- **User Dashboard**
  16th February 2026 – 8th March 2026
  Develop an interactive dashboard to visualize job states, and governance insights.

- **System Testing & Validation**
  9th March 2026 – 18th March 2026
  Conduct functional testing, rules validation, and administrative review

- **Capstone Interim Report**
  29th September 2025 – 7th December 2025
  Prepare interim report deliverables

- **Capstone Final Report & Video**
  8th December 2025 – 18th March 2026
  Prepare final report deliverables and video submission

- **Capstone Final Presentation**
  19th March 2026 – 27th March 2026
  Present project outcomes and demonstrate the Batch Job Lifecycle Monitor

3. References

[1] S. Ruan, Y. Wang, H. Jiang, W. Xu, and Q. Guan, "BatchLens: A visualization approach for analyzing batch jobs in cloud systems," arXiv preprint arXiv:2112.15300, 2021. [Online]. Available: https://arxiv.org/abs/2112.15300

[2] A. Gellrich, F. Gosewinkel, J. Heupel, and J. Letts, "Containerized Batch System Monitoring," EPJ Web of Conferences, vol. 214, 07008, 2019. [Online]. Available: https://www.epj-conferences.org/articles/epjconf/pdf/2019/19/epjconf_chep2018_07008.pdf

[3] F. Lehmann, M. Geimer, B. Bergmann, and J. Krüger, "The Common Workflow Scheduler Interface (CWSI)," arXiv preprint arXiv:2311.15929, 2023. [Online]. Available: https://arxiv.org/abs/2311.15929

[4] J. Beránek, S. Böhm, and V. Cima, "Analysis of Workflow Schedulers in Simulated Distributed Environments," arXiv preprint arXiv:2204.07211, 2022. [Online]. Available: https://arxiv.org/abs/2204.07211

END OF REPORT