Integration of Software Architecture and Process Models

Project: Temporary Staff Coordination System (TSWMS)

Organization: Department of Industrial Management, University of Kelaniya

Student ID: IM/2022/055

Student Name : Chamika Denuwan

1. Introduction

The effectiveness of the **Temporary Staff Coordination System (TSWMS)** depends on the careful integration of software process methodologies and architectural decisions. This project aims to replace manual, document-based operations—characterized by paper-based marking sheets and manual salary calculations—with a centralized digital solution.

The project adopts an **Agile methodology** to accommodate evolving departmental needs through iterative development. While a **Microservice architecture** is planned for the final implementation to ensure high scalability and modularity, this report justifies the integration of **SOLID principles** and the **C4 architectural model** to ensure long-term stability[55555555].

2. Software Process Methodology

The development of TSWMS follows an **Agile** methodology. This addresses the dynamic reality that departmental requirements (such as specific evaluation criteria or leave policies) evolve over time.

- **User Stories:** The project utilizes detailed user stories (e.g., US1 for HOD uploading candidate lists, US16 for staff applying for leave) to break down complex coordination tasks into deliverable increments.
- **Stakeholder Feedback:** Continuous feedback from the Head of Department (HOD) and Staff Coordinator ensures the system remains aligned with university policies.

3. System and Design Thinking

System Thinking (The Big Picture)

Instead of viewing "manual salary errors" as isolated incidents, System Thinking identifies the root cause: the lack of communication between attendance books and the finance spreadsheet.

- **Traditional View:** "The salary calculation is wrong; let's recount the attendance books.".
- **System Thinking View:** "The calculation is wrong because data is fragmented across@

@physical books and disconnected Google Forms.

Design Thinking (Human Needs)

By applying **Empathic** mindsets, the project focuses on the users' pain points, such as mentors' difficulty in tracking mentee job descriptions (JDs).

- **Divergent Thinking:** Brainstorming various features like biometric integration, mobile apps, and automated service letters.
- **Convergent Thinking:** Selecting essential features for the MVP, such as the automated marking scheme and role-based login, to solve the department's immediate efficiency gaps.

---

4. Software Architecture: Microservices for TSWMS

1. Architectural Overview

Unlike a monolithic system, the TSWMS utilizes a **Microservice architecture**. Functional modules—such as Registration, Leave Management, and Salary Reporting—are built as independent services[17].

2. Justification for Microservices

- **Modularity and Independence:** Each service (e.g., Interview Evaluation) can be built, tested, and deployed independently, which is vital for a complex departmental system.
- **Scalability:** The system can scale to accommodate more departments or an increasing number of temporary staff in future semesters without affecting core stability.
- **Technology Flexibility:** Different modules can use the most suitable tools (Spring Boot for backend, React for frontend) while sharing a centralized MySQL database.

---

5. Integration of SOLID Principles

Implementing SOLID principles ensures the system is resilient to change.

- **Single Responsibility:** The SalaryController handles only salary-related logic, ensuring changes to pay rates don't affect the InterviewController.

- **Open/Closed:** The system allows adding new types of staff evaluations by extending existing classes rather than rewriting the core interview logic.

- **Interface Segregation:** Mentors only see the "Assigned Mentee" interface, while the HOD sees the "Salary Approval" interface.

---

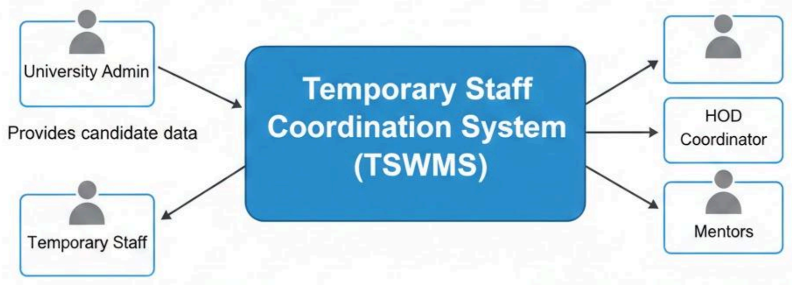
6. Integration of the C4 Model

Level 1: Context Diagram

- **System:** Temporary Staff Coordination System.
- **External Actors:** University Admin (provides candidate data).

- **Internal Actors:** HOD, Coordinator, Mentors, and Temporary Staff.

Figure 1. C4 Level 1: Context Diagram for Temporary Staff Coordination Systerr



A high-level view showing the system and its interactions with primary external and internal users.

Level 2: Container Diagram

- **Web Application (React):** The responsive user interface for all roles.

- **API Application (Spring Boot):** The logic engine handling leave, registration, and task assignment.

- **Database (MySQL):** Stores user credentials, JDs, and attendance records.

Level 3: Component Diagram

- **Registration Component:** Manages interview marks and shortlisting.
- **Leave Component:** Handles substitute filtering and HOD approval.
- **Notification Component:** Sends reminders for contract expiry and task deadlines.

---

7. Project Feasibility and Costing

The project is economically viable through **Azure Cloud** hosting.

- **Estimated Monthly Cost:** Approximately **$49.05**.
- **Efficiency Gains:** Automation of salary reports and service letters significantly reduces manual administrative hours.

Would you like me to create the **Level 1 Context Diagram** image tag for your system to include in this report?