Zambia University College of Technology

Advancing Knowledge and Innovation

REPORT BY:

MARTHA CHOOKA

DAVID CHISENGA

CHEELA MULILO.

FOR: MR CHIKWANDA

REPORT ON SOFTWARE DESIGN IN 2025

Executive Summary

The modern software design space has seen the balancing of classic engineering goals such as correctness and maintainability with newer demands such as rapid iteration, cloud-scale operation and AI native design patterns which are now shaping the current practice and tooling. In this report various emerging trends and long standing tools such as microservices and AI augmented tooling, are discussed in relation to the current Unified Student Experience Platform - hence forth will referred to as the project, under design consideration.

Introduction

Software design is the systematic process of conceptualizing and planning how a software system will work before it's implemented. It consists of both a process (activities and decisions that turn requirements into structure and behavior) and an artifact (the models, diagrams, and code organization that embody those decisions).

Discussion

A Process is a set of activities which include but are not limited to requirements analysis, conceptual modelling, architecture selection, pattern application, trade-off evaluation, prototyping and validation, and preparing implementation guidance. The software design process today is less about creating a perfect, static blueprint and more about facilitating a continuous, evidence-based conversation about the system's evolution, eliminating ambiguous needs and allowing implementation and evolution at low risk.

While an Artifact is the tangible outcomes of design work such as models, diagrams, patterns, and architectural documentation, that guide developers in building the system. They also serve as a blueprint for implementation, communication between stakeholders, and a living reference for maintenance and evolution.

One of the artifacts considered for this project is the Architectural Decision Record which will highlight and record important architecture and design decisions and the context in which they were made. Decisions are explicit and kept in an orderly manner that allows for future developers to be able to understand why certain choices were made over others.

UML diagrams are also an important part of the project design process as they will provide means to visualize, construct, and document the artifacts of the software system. UML’s provide improved communication and shared understanding using a common, unambiguous language that can easily be understood by developers, business analysts and stakeholders. Apart from that they also serve as comprehensive documentation that is directly linked to the system's design.

Software design and architecture trends have over the years evolved from centralized monoliths toward distributed, AI-enhanced systems. It has seen the continued use of timeless and actionable principles such modularity and separation of concerns now being coupled with modern architectural styles and patterns such as microservices and serverless architectures.

Microservices, over the years, gained favour due to its allowance of applications to be broken down into many independent services, each responsible for a distinct domain. With the enhanced robustness and fault isolation, as well as technological heterogeneity and flexibility boasted by microservice architecture, compared to the more traditional monolithic approach, it has proven to serve as a much more efficient and reliable solution to the world's software problems today.

Serverless architecture is also quickly becoming the norm, attributed mainly to its ability to accelerate innovation and reduce costs by removing infrastructure management, offering developers a pay-as-you-go model, and providing automatic, dynamic scaling for applications. This approach will also make it easier considering the project is meant to cater for a global community of students.

The introduction of AI assistance has also brought about a significant increase in service provision and elimination of manual handling of iterative tasks. The enhanced observability and intelligent monitoring and API and communication optimization are just but a few of the many benefits that are expected from a system with AI assistance. Taking into account the scale and expected audience for this project, incorporating AI assistance would greatly improve the systems efficiency and robustness.

Recommendations

We strongly recommend the use of microservice architecture alongside the serverless architecture, this will create a serverless microservice that will boast advantages such as granular scaling, optimal resource utilization and faster development cycles which will in turn ensure a robust, scalable and efficient system that will be able to handle a large student population and cater for the diverse requirements of the software.