Department of Electrical, Computer, and Software Engineering

Faculty of Engineering and Applied Science

SOFE 3650-Fall 2025

Software Design and Architectures

Project Description and Deliverables

This document presents a description of the project expectation, deliverables, and grading scheme.

Prepared by: Dr. Hani Sami
Date: October 2025

Objectives

The objective of this project is to demonstrate a methodological set of steps in the design of a software architecture for a set of requirements listed in Appendix A. The expected design approach to take is the Attribute Driven Design (ADD) presented in some detail in the text book [1] and briefly overviewed here.

Deliverables

The expectation is to submit a set of artifacts in a GitHub classroom repository that your instructor will create for you that demonstrates the ADD steps as applied to the design and implementation of a software application.

All deliverables should be uploaded to a GitHub repository that the instructor will set up. Individual contributions in the project will be graded the based-on GitHub commitments of the deliverables and use of the KanBan project board available in GitHub. A good way to manage your project is by leveraging the Issues tagging available through GitHub and the automated KanBan project board.

Deliverable 1 – Requirements Analysis (Due Oct 28)

Prior to commencing the ADD process the use cases, quality attributes and constraint requirements of the project need to be defined based on the requirements that were provided to you. The format of these submissions should follow closely the FCAPS case example in the text book [1].

Expected artifacts are:

- Use Case models
- Quality Attributes for the application
- System Constraints for the application
- Architectural Concerns
- Business Case

Grading Scheme (worth 25% of total project grade)

- Proper identification of the Use Cases
- Proper identification of the quality attributes
- Proper identification of the constraints and architectural concerns
- Clear description of the business case.

Deliverable 2 – ADD Iteration 1&2 (Due Nov. 16)

At this point in the project you should have gone through the drivers of the project and ready to start he design.

Expected artifacts are:

- All relevant artifacts from ADD Iterations 1&2

Grading Scheme (worth 50% of total project grade)

- Organization and navigation of the GitHub repository with use of README files.
- Quality of the 1st and 2nd iteration of the ADD steps.
    - Choice of reference architecture / framework along with justifications
    - Deployment diagrams
    - Major components of the architecture
    - Domain specific components
    - Interface specification (both internal and external)
    - Domain specific models

Deliverable 3 – Design of a Use Case (Due Dec. 6 – end of the term)

At this point the project submission should include all 3 iterations of the ADD. The implementation should follow the architectural model and it should be clear which code files are associated with the design components. The grading of this deliverable is based on how well your team has reviewed the architectural design using the ATAM review process. You should produce a scenario analysis table similar to that presented in Lab 5.

Grading Scheme (worth 20% of total project grade)

- Organization and navigation of the GitHub repository with use of README files.
- The quality of the ATAM scenario analysis

Team Assessment (worth 5% of project grade)

As a requirement of the course there is a team assessment that should be completed. The purpose of this team evaluation is for students to understand the dynamics and roles of a team. This will be assessed using the ITP metrics tool and instructions will be posted in Canvas.

References

[1] “Designing Software Architectures: A Practical Approach” by Humberto Cervantes and Rick Kazman and covered in the course.

Appendix A: Project Requirements

The project you are to design is a AI-Powered Digital Assistant Platform (AIDAP). Below is a list of the requirements for the project.

The project you are to design is an **AI-Powered Digital Assistant Platform (AIDAP)**. The system provides a conversational interface for students, faculty, and administrators to interact with institutional data such as course schedules, deadlines, announcements, and academic analytics. The assistant integrates with external university systems (LMS, registration, calendars, and mail) and uses AI to deliver contextual answers.

Stakeholders

Symbol	Stakeholder	Description
S	Students	End users who query academic and campus information.
L	Lecturers	Provide course-related content and respond to academic queries.
A	Administrators	Maintain institutional data, integrations, and policies.
M	System Maintainer	Responsible for deployment, monitoring, and upgrades.
D	Data Source Systems	External systems such as LMS, Registration, Calendar, and Email servers.

General Requirements

ID	Requirement
R1	The system shall provide conversational access to institutional data and services.
R2	The system shall store historical interactions for personalization.
R3	The system shall integrate with existing data sources (registration, LMS, calendars).
R4	The system shall support both text and voice interaction modes.
R5	The system shall use AI models to interpret natural-language queries.
R6	The system shall generate responses using both stored knowledge and live data.
R7	The system shall be deployable as a cloud-native, scalable service.
R8	The system shall protect user data and comply with institutional privacy policies.

Requirements of Students

ID	Requirement
RS1	The system shall allow students to ask academic or administrative questions (e.g., “When is my next exam?”).
RS2	The system shall notify students of deadlines, schedule changes, and announcements.

RS3	The system shall allow students to access personalized dashboards summarizing upcoming events and performance indicators.
RS4	The system shall support multi-language queries and responses.
RS5	The system shall learn from previous conversations to improve response relevance.
RS6	The system shall allow students to change preferences for notifications and language.
RS7	The system shall provide secure authentication through the institution's single sign-on (SSO).
RS8	The system shall ensure that student-specific data are visible only to the authenticated user.
RS9	The system shall be accessible on mobile, web, and voice-assistant devices.
RS10	The system shall respond to queries within 2 seconds on average under normal load.
RS11	The system shall remain available 99.5% of the time per month.
RS12	The system shall have an intuitive UI consistent with conversational design best practices.
RS13	The system shall allow export of calendar events to personal calendars.
RS14	The system shall support offline cache of recent responses for limited connectivity.

Requirements of Lecturers

ID	Requirement
RL1	The system shall allow lecturers to publish or update course materials accessible to students through the assistant.
RL2	The system shall enable lecturers to post announcements via conversational commands.
RL3	The system shall allow lecturers to view summarized class analytics (grades, attendance, engagement).
RL4	The system shall enable lecturers to schedule automated reminders (e.g., assignment deadlines).
RL5	The system shall allow lecturers to manage access rights for teaching assistants.
RL6	The system shall allow lecturers to query the assistant for aggregated statistics (e.g., average GPA by course).
RL7	The system shall notify lecturers of system-detected anomalies (e.g., sudden drop in participation).
RL8	The system shall ensure that only authorized lecturers can modify course data.

Requirements of Administration

ID	Requirement
RA1	The system shall allow administrators to manage institutional integrations (LMS, registration, calendars).
RA2	The system shall allow administrators to define global policies (data retention, response logging).
RA3	The system shall allow administrators to broadcast campus-wide announcements via the assistant.
RA4	The system shall allow administrators to monitor system usage and generate analytics reports.
RA5	The system shall ensure compliance with institutional security and privacy regulations.
RA6	The system shall provide high availability with automatic fail-over and backup recovery.
RA7	The system shall support scalability to handle up to 5,000 concurrent users.

Requirements of System Maintainer

ID	Requirement
RM1	The system shall allow maintainers to deploy updates with zero downtime using continuous deployment pipelines.
RM2	The system shall provide monitoring dashboards (health, latency, errors).
RM3	The system shall support configuration of AI model versions and API keys.
RM4	The system shall log performance metrics for model accuracy and latency.
RM5	The system shall be easily extensible to integrate new AI services or external data sources.
RM6	The system shall allow secure backup and restore of user and configuration data.
RM7	The system shall support role-based access for maintenance operations.

Requirements of Data Source Systems

ID	Requirement
RD1	The system shall synchronize data with connected university systems at configurable intervals.
RD2	The system shall use standard APIs (REST or GraphQL) for interoperability.
RD3	The system shall handle failures in data source availability gracefully (retry and recovery).
RD4	The system shall maintain data integrity and consistency across systems.