University Academic Performance Management System (UAPMS)

Njugi Elijah Muiru

146412

Karogo Joe Ndungu

167141

ICS 3 Group: B

Supervisor Name: Dr. Kennedy Ronoh

An Informatics Project Proposal Submitted to the School of Computing and Engineering Sciences (SCES) in partial fulfilment of the requirements for the award of a Degree in Informatics and Computer Sciences

School of Computing and Engineering Science

Strathmore University

Nairobi, Kenya

15 May 2025

Declaration and Approval

We declare that this work has not been previously submitted and approved for the award of a degree by this university or any other university. To the best of our knowledge and belief, the research proposal contains no material previously published or written by another person except where due reference is made in the research proposal itself.

Student Name: Njugi Elijah Muiru

Admission Number: 146412

Student Signature: ________________ Date: 5/15/2025

Student Name: Karogo Joe Ndungu

Admission Number: 167141

Student Signature: ________________ Date: 5/15/2025

Supervisor Name: Dr. Kennedy Ronoh, who will provide guidance and oversight throughout the project.

Supervisor Signature: ________________ Date: 16/05/2025

Kenned
y Ronoh

Digitally signed by Kennedy Ronoh
DN: OU=SCES, O=Strathmore Univuersity, CN=Kennedy Ronoh, E=kronoh@strathmore.edu
Reason: I am approving this document
Location: Nairobi
Date: 2025.05.16 11:44:38+03'00'
Foxit PDF Reader Version: 2025.1.0

Abstract

University students are struggling to effectively track the academic progress. This is due to the lack of deeper analysis of their progress by the already existing systems. The University Academic Performance Management System (UAPMS) is a web-based application that revolutionizes how university students monitor, evaluate, and manage their academic journey. It provides an interactive interface where students can log units, grades, and credit hours while receiving real-time feedback on GPA trends, unit status (pass/fail), and degree classification projections. Unlike other Academic Management Systems (AMS) such as Moodle and Strathmore AMS that offer minimal interaction and feedback, UAPMS enhances student decision-making through visual charts, intelligent alerts for retake requirements, and tailored academic guidance. It also generates academic performance summary reports for the Dean and Examination Office and confirms a student's final degree classification based on institutional grading policies. The system will be built using React, PHP, and PostgreSQL, follows an Agile development methodology with sprint-based feedback cycles to ensure adaptability and user-centered improvement. The potential impact of UAPMS on students' academic journey is promising, offering a more insightful and actionable approach to educational management.

Table of Contents

Declaration and Approval ii

Abstract iii

List of Abbreviations vii

Chapter 1: Introduction 1

1.1 Background Information 1

1.2 Problem Statement 1

1.3 Aim/ Specific Objectives 2

1.3.1 Specific Objectives 2

1.3.2 Research Questions 3

1.4 Justification 3

1.5 Scope and Limitations 4

1.5.1 Scope 4

1.5.2 Limitations 5

1.5.3 Delimitations 5

Chapter 2: Literature Review 6

2.1 Introduction 6

2.2 Exploration of Current Academic Performance Tracking Practices 6

2.3 Challenges in Academic Tracking Tools 7

2.3.1 Challenge 1: Delays in Report Generation and Distribution 7

2.3.2 Challenge 2: Difficulty in Identifying and Addressing Missing Grades 7

2.3.3 Challenge 3: Shallow academic report insights 7

2.3.4 Challenge 4: Lack of Visual and Analytical Feedback 7

2.4 Review of Existing Systems 8

2.4.1 Ellucian Banner Student 8

2.4.2 Moodle LMS with Gradebook 9

2.4.3 Strathmore University Academic Management System (AMS) 10

2.5 Gaps in Current Approaches and Related Applications 11

2.6 Conceptual Framework 11

Chapter 3: METHODOLOGY 14

3.1 Introduction 14

3.2 Methodology and Paradigm 14

3.2.1 Requirements Gathering and Analysis (Sprint 1) 15

3.2.2 System Design (Sprint 2) 16

3.2.3 System Development (Sprint 3 - Ongoing) 16

3.2.4 Testing and Feedback (Upcoming Sprints) 16

3.2.5 Documentation and Final Review 16

3.3 System Analysis 17

3.3.1 Use Case Diagram 17

3.3.2 Sequence Diagram 17

3.3.3 Entity Relationship Diagram (ERD) 17

3.3.4 Class Diagram 17

3.3.5 Activity Diagram 17

3.4 System Design 17

3.4.1 Database Schema 18

3.4.2 Wireframes 18

3.4.3 System Architecture 18

3.5 System Development Tools and Techniques 18

3.5.1 NetBeans IDE 18

3.5.2 Programming Languages 18

3.5.3 Database – Postgresql 19

3.5.4 Version Control – Git and GitHub 19

3.5.5 Visualisation Tool – Chart.js or Google Charts 19

3.6 Deliverables 19

3.6.1 Proposal Document 19

3.6.2 Design Diagrams 19

3.6.3 Working Web-Based System 20

3.6.4 Test Reports 20

3.6.5 Documentation 20

3.6.6 Feedback Logs 20

3.6.7 Final Report 20

References 21

Appendix 24

List of Abbreviations

Abbreviation	Full Meaning
AMS	Academic Management System
API	Application Programming Interface
CI/CD	Continuous Integration / Continuous Deployment
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DBMS	Database Management System
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GPA	Grade Point Average
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JS	JavaScript
JWT	JSON Web Token (if token-based auth is used)
LMS	Learning Management System
PDF	Portable Document Format
PHP	Hypertext Preprocessor
QA	Quality Assurance
SCES	School of Computing and Engineering Sciences
SQL	Structured Query Language
UAPMS	University Academic Performance Management System

Abbreviation	Full Meaning
UI	User Interface
UX	User Experience
UX/UI	User Experience/User Interface

List of Figures

Figure 2.1 Example of the Ellucian Banner Student system 9

Figure 2.2 Example of Moodle System 10

Figure 2.3 Example of Strathmore AMS 11

Figure 2.4 Conceptual Framework diagram 13

Figure 5.1 Gantt chart showcasing our sprints.. 24

Chapter 1: Introduction

1.1 Background Information

In higher education, effectively managing academic performance is essential to ensuring that students achieve graduation requirements and develop intellectually and professionally. However, many university students find it challenging to monitor their academic journey due to a lack of accessible, meaningful, and actionable academic data. Typically, students receive raw grades at the end of a semester, but these are often shared without sufficient explanation or deeper analysis. (Ronel B. Dayanghirang, Alexander A. Hernandez, 2022). As a result, learners are left to interpret their academic status on their own, without system-generated insights, performance forecasts, or timely alerts. (John P. Campbell, Peter B. DeBlois, Diana G. Oblinger, 2007).

Although educational technology has seen rapid advancement, most university platforms continue to serve primarily administrative purposes, such as registering students and posting grades. These platforms cannot often deliver personalised academic insights, track performance trends over time, or provide predictive feedback—all crucial features for students making strategic academic choices. (Full Fabric, 2025).This becomes especially concerning in institutions where students must meet minimum GPA requirements or pass core courses to continue in their academic programs.

According to studies on academic advising (George D. Kuh, Jillian Kinzie, Jennifer A. Buckley, Brian K. Bridges, and John C. Hayek, 2006), timely and meaningful feedback is essential for student success and retention. Without integrated systems to help interpret academic data, many students mismanage course selections, miss retake deadlines, or remain unaware of looming academic probation. In Kenya, where institutions like Strathmore University aim to provide holistic and tech-enabled education, this gap highlights a missed opportunity for leveraging data-driven technologies to enhance student outcomes.

1.2 Problem Statement

Although educational institutions have increasingly adopted technology, platforms like the Academic Management System (AMS) and the eLearning portal at Strathmore University—and others like it—still lack smart, tailored academic management features. These systems mostly serve as static hubs for posting grades, tracking attendance, and listing courses. They fall short of offering real-time analysis or actionable insights. Core tools such as GPA trend tracking, early alerts for struggling students, and predictive degree classification are notably absent.

The absence of dynamic and insightful features in these systems creates real challenges for both students and staff. For example, students may only discover they missed a compulsory unit when it affects their progress. Similarly, they often cannot project their GPA trajectory or estimate their likelihood of achieving a First Class or Second Upper degree classification. (MRCC EdTech, 2024).Additionally, many of these systems have poorly designed user interfaces and inconsistent experiences across devices, further complicating their usability.

As a result, students are often left without the necessary tools to plan their academic path effectively. This can lead to delayed graduations, unexpected retakes, and academic penalties—all of which could be avoided with real-time, data-informed feedback. (SEAtS Software, 2023). Academic personnel, such as advisors and deans, also lack efficient reporting tools that could support timely interventions. Implementing a smart academic system could significantly enhance academic planning, decision-making, and institutional effectiveness.

1.3 Aim/ Specific Objectives

To develop a responsive, web-based University Academic Performance Management System (UAPMS) that enables students to track their academic performance, receive real-time feedback and projections, and obtain academic guidance and degree classification confirmation, thereby enhancing decision-making and reducing risks of educational failure.

1.3.1 Specific Objectives

The following objectives will guide the development and implementation of the University Academic Performance Management System (UAPMS):

I. To investigate current practices and challenges in academic performance tracking among university students and staff.

II. To analyse existing academic tracking systems and identify gaps in functionality, usability, and feedback.

III. To design a user-focused system architecture that enables real-time academic tracking, feedback, and guidance.

IV. To develop a working prototype of the UAPMS using selected frontend, backend, and database technologies.

V. To conduct system testing and evaluation to assess functionality, usability, and overall effectiveness.

1.3.2 Research Questions

The following research questions align with the project's specific objectives and will guide the investigation:

1. What are the current practices and challenges in tracking academic performance in universities?
2. What are the limitations of existing academic tracking systems in terms of functionality and user experience?
3. How can a user-centred system be designed to offer real-time feedback and academic guidance?
4. What technologies are suitable for developing a reliable and interactive academic performance tracking system?
5. How effective, usable, and reliable is the developed system in supporting students and academic staff?

1.4 Justification

Tracking academic performance effectively is essential for helping students manage their learning paths, make sound decisions, and reach their academic objectives. However, in many universities, students only have access to scattered data like grades, GPAS, and comments, without any analytical insight or forecasts about their academic standing. As a result, they may remain unaware of potential academic issues, miss the chance to plan for resits in time, or misunderstand how close they are to graduating. Institutions also face challenges in offering timely academic support because they lack a unified view of student performance.

The University Academic Performance Management System (UAPMS) aims to bridge this gap by introducing a smart, centralised, and user-friendly platform. This system will allow students to track and manage their academic records while receiving immediate feedback, alerts for retakes, predicted degree classifications, and visual GPA trends over time. These tools will encourage students to take active control of their academic progress (Evelyn Montgomery, 2024).

In addition to helping students, UAPMS is intended to assist lecturers, academic advisors, and administrators. Staff will be able to access summarised reports and quickly identify students who might need extra academic attention. Offices such as the Dean's and Exams Office can also rely on these reports for validating degree classifications, checking academic progression, and maintaining academic standards. (Evelyn Montgomery, 2024).

Furthermore, this system will help universities enhance their academic quality, minimise failure and repetition rates, and promote timely graduations. It also aligns with Kenya's vision for higher education modernisation and supports digital transformation goals set by the Commission for University Education. (Commission for University Education (CUE), 2025). Over time, the insights generated by the system could contribute to shaping national education policies and guiding curriculum development.

By using modern tools like React, Postgresql, PHP, and Chart.js, and focusing on a user-first design approach, UAPMS will deliver an engaging, responsive platform that brings value to students, educators, and administrators alike.

1.5 Scope and Limitations

1.5.1 Scope

This project will involve the creation of a web-based academic performance tracking system designed specifically for university settings. It will serve different user groups, including students, lecturers, academic deans, and examination officers. The system's core features will include recording academic data such as courses, grades, and semesters, calculating GPA in real-time, generating visual performance analytics, tracking academic progress, issuing alerts for retakes, and projecting final degree classifications. Additionally, it will provide reporting tools for administrators to support strategic academic planning and informed decision-making.

The scope includes:

- Frontend development using **React.js** for interactivity and responsiveness.

- Backend logic using **PHP** to manage server-side processes.
- Academic data storage and management using **Postgresql**.
- Data visualisation using **Chart.js** to generate intuitive, interactive charts.
- Authentication and role-based access for different users (students, lecturers, admins).
- Reporting tools to support academic offices.

The project will not cover:

- Integration with live university portals or databases (e.g., real AMS systems).
- Mobile application development (though the system will be mobile-responsive).
- Broader eLearning functionalities such as assignment uploads or discussion forums.
- Financial or fee tracking.

1.5.2 Limitations

While the proposed system offers several innovations, it faces some inherent limitations. First, it relies heavily on the accuracy of user-entered data. If there are incorrect grades or course information that is input, the system's projections and reports may be misleading. Second, the GPA and classification projections will be based on fixed institutional grading rules subject to periodic change. Adapting the system to policy changes may require manual updates. Lastly, the system will not integrate with live institutional databases due to privacy concerns and restricted API access, which limits real-time syncing with official records.

1.5.3 Delimitations

To manage these limitations, the project will include validation rules during data entry to reduce user errors and ensure that entries follow correct academic formats. The system will be designed to allow administrators to easily update grading rules, ensuring that institutional changes can be accommodated with minimal technical intervention. Moreover, users will be trained on how to use the system accurately through a built-in help guide, and the platform will be designed with scalability in mind for future integrations or extensions, such as mobile apps or API-based syncing.

Chapter 2: Literature Review

2.1 Introduction

This chapter discusses how academic performance tracking was previously done in universities before modern systems were introduced. It also highlights current academic tracking systems created to improve grade management and report generation. In addition, the gaps in these systems are identified, and a conceptual framework of the proposed improved system is presented.

2.2 Exploration of Current Academic Performance Tracking Practices

Universities need a simple way to record and manage student grades from start to finish. Instructors enter marks into a central system, which calculates each student's average grade. The system creates reports of their performance and enables students to access them online. These functions of grade entry, average grade calculation, report creation, and student access to grades provide the necessary data for decisions on student progression and graduation. (DiplomaSafe, 2023)

So, let's focus on how academic reports are created. After lecturers enter grades and the system calculates average marks, the platform turns this data into performance reports. These usually include course names, credit hours, and overall averages, and are available as PDFS or online pages. Students and school staff can view them through the student portal, with options to check by semester or academic year. (Ellucian, What is a Student Information System in Higher Education?, 2025)

Report creation must follow university grading rules, e.g., A=70, B=60, C=50, D=40, and ensure reports are available and correct. (Wikipedia, 2024) Ideally, the system would automatically flag missing grades, create reports once grades are entered, and notify students immediately. In reality, many systems wait for manual checks by the exams office before releasing reports, causing delays. Students often aren't notified when their exam results are ready, which can affect their ability to plan for their studies.

2.3 Challenges in Academic Tracking Tools

2.3.1 Challenge 1: Delays in Report Generation and Distribution

One major challenge in academic tracking is the delay in generating and sharing student performance reports. In many institutions, exam results must be manually reviewed and approved by the exams office before being released to students. (Collegis Education, 2021) This process slows things down and prevents students from accessing their results on time. As a result, critical academic decisions, like applying for graduation, planning for supplementary exams, or tracking progress, can be delayed. Although some systems have introduced partial automation to speed things up, manual checks are still standard and continue to cause bottlenecks.

2.3.2 Challenge 2: Difficulty in Identifying and Addressing Missing Grades

Another major challenge is difficulty identifying and addressing missing or incorrect grades on time. In many cases, students are unaware that a grade is missing until they try to access their final report. This is often because the system does not automatically detect incomplete records or notify users of something wrong. As a result, students may experience delays in graduation applications, scholarship processing, or enrollment for the next academic period. While some platforms have started to include basic alerts, most still rely on students or staff noticing these issues manually, which is unreliable and inefficient.

2.3.3 Challenge 3: Shallow academic report insights

Finally, there is a problem with how academic progress is reported. Some systems only provide basic summaries, like final grades, without offering deeper insights such as performance trends over time or personalised feedback (Full Fabric, 2025). This makes it harder for students to understand where they're doing well or falling behind. Although a few solutions have started to include more advanced reporting features, many platforms still present data in a way that lacks depth and makes it hard to support meaningful academic improvement.

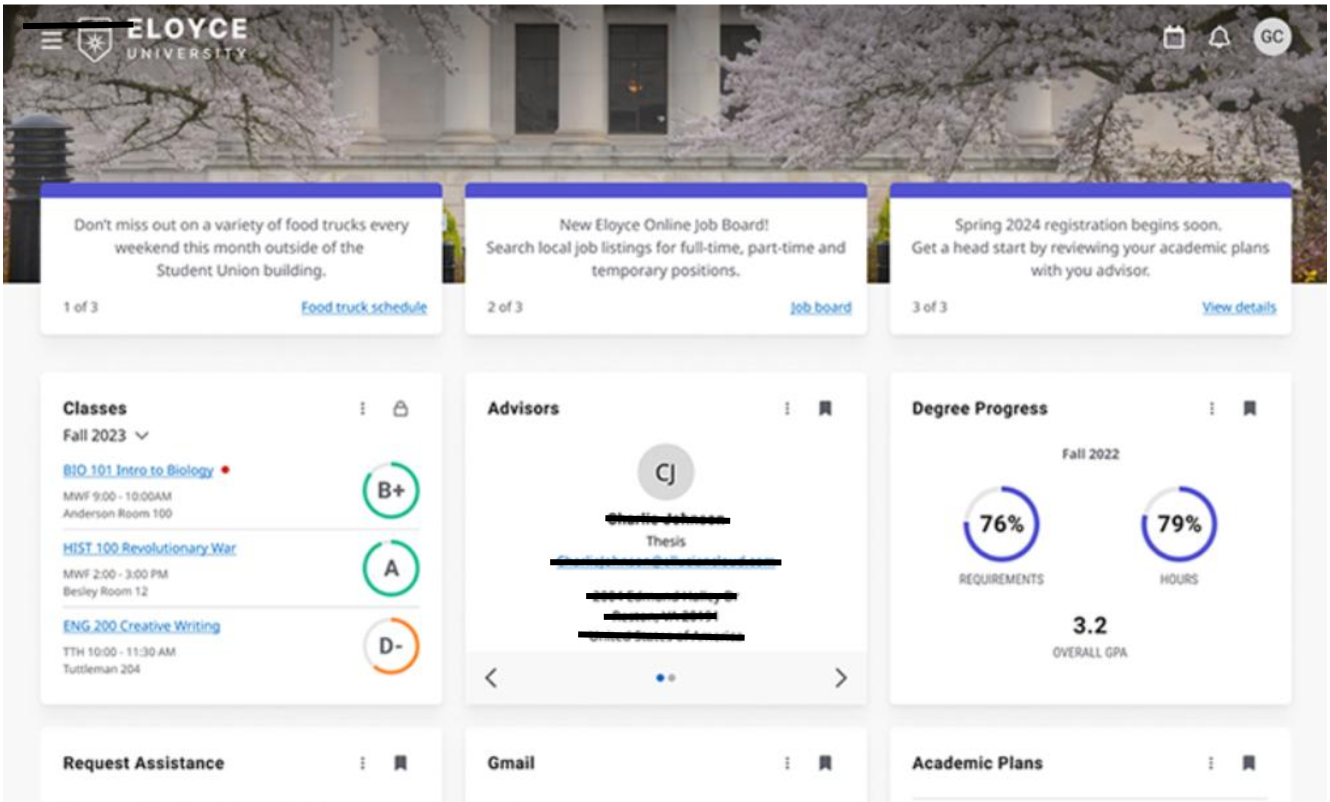
2.3.4 Challenge 4: Lack of Visual and Analytical Feedback

Many academic platforms, such as charts or trend graphs, do not provide visual representations of student performance. Instead, data is displayed using plain text or static tables, making it harder for users to interpret results over time. Without tools like GPA trend graphs, course performance comparisons, or semester-over-semester analytics, students and administrators miss opportunities to spot patterns, identify strengths or weaknesses, and make informed academic decisions (Ronel B. Dayanghirang, Alexander A. Hernandez, 2022). This absence of analytical support can lead to delayed interventions, reduced student engagement, and limited insight into overall academic progress. While a few modern platforms have started integrating dashboards and visual analytics, these features are still missing in most systems.

2.4 Review of Existing Systems

2.4.1 Ellucian Banner Student

Ellucian Banner Student is a student information system developed for use in higher education, aimed at reducing manual handling of grades and improving administrative workflows. It provides self-service features that allow students to access their grades, enrol in courses, and monitor their academic progress. Built on Oracle infrastructure and supported by cloud integration, it ensures real-time availability of academic data. However, the system primarily displays information in static formats like tables and lists, making it difficult for students to analyse their academic performance over time. The proposed system addresses this limitation by incorporating visual elements such as GPA trend charts, semester performance comparisons, and personalised dashboards, which offer clearer insights for better academic planning. (Ellucian, nd).



ELOYCE UNIVERSITY

GC

Don't miss out on a variety of food trucks every weekend this month outside of the Student Union building.

1 of 3 Food truck schedule

New Eloyce Online Job Board! Search local job listings for full-time, part-time and temporary positions.

2 of 3 Job board

Spring 2024 registration begins soon. Get a head start by reviewing your academic plans with you advisor.

3 of 3 View details

Classes

Fall 2023

BIO 101 Intro to Biology
MWF 9:00 - 10:00AM
Anderson Room 100 — B+

HIST 100 Revolutionary War
MWF 2:00 - 3:00 PM
Besley Room 12 — A

ENG 200 Creative Writing
TTH 10:00 - 11:30 AM
Tuttleman 204 — D-

Advisors

CJ

[redacted]

Thesis

[redacted]

Degree Progress

Fall 2022

76% REQUIREMENTS

79% HOURS

3.2 OVERALL GPA

Request Assistance

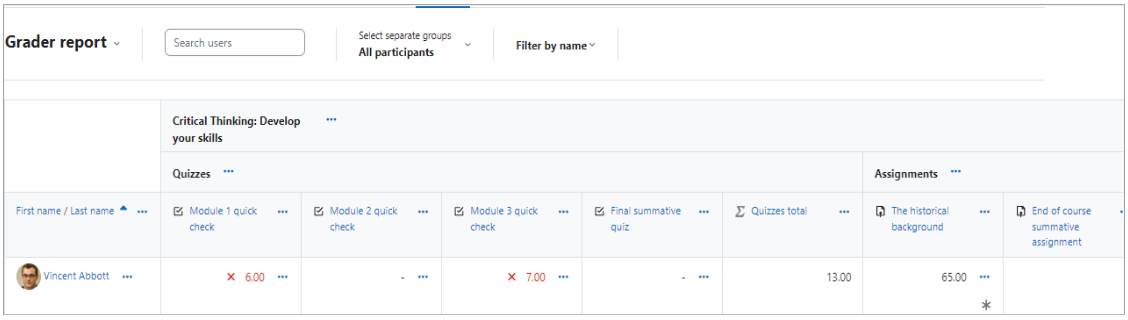
Gmail

Academic Plans

Figure 2.1 Example of the Ellucian Banner Student system

2.4.2 Moodle LMS with Gradebook

Moodle is a popular open-source Learning Management System (LMS) many universities use to organise course materials and assessments. Its Gradebook function enables instructors to input scores, compute final grades, and present course results to students, thereby solving the issue of fragmented academic tracking through centralised grade access. Developed with PHP and backed by SQL databases, Moodle is highly adaptable and can be integrated with other platforms. However, although it allows students to check their grades for each course, it does not provide automated tools like GPA calculations, semester analytics, or alerts for poor performance. The proposed system bridges this gap by introducing intelligent automation, performance visualisations, and personalised feedback to promote tailored academic development. (Moodle, nd).

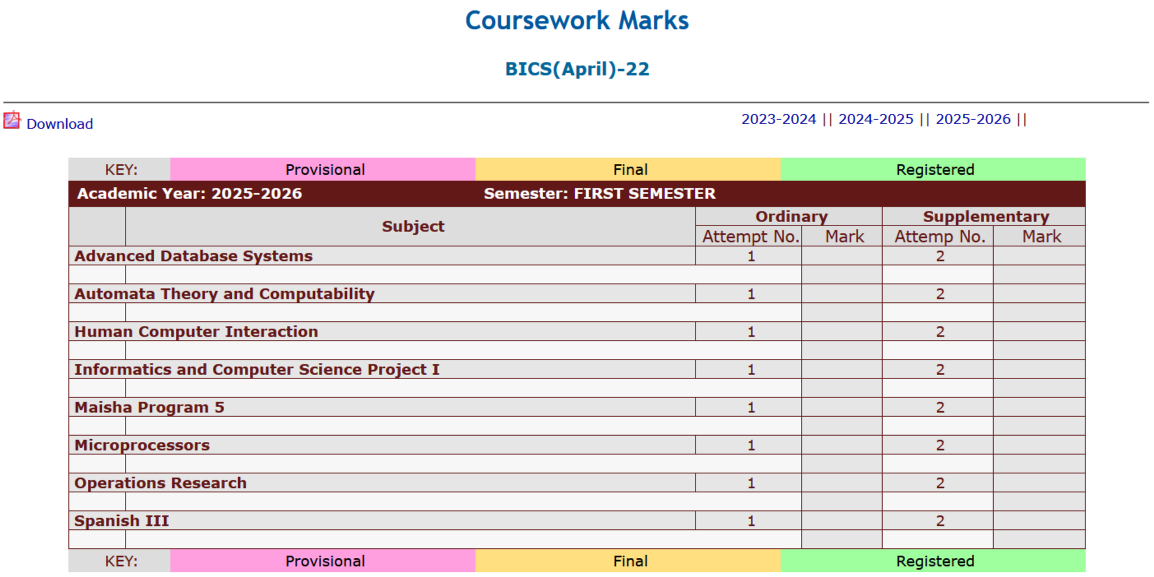


	Critical Thinking: Develop your skills						
	Quizzes					Assignments	
First name / Last name	Module 1 quick check	Module 2 quick check	Module 3 quick check	Final summative quiz	Quizzes total	The historical background	End of course summative assignment
Vincent Abbott	✕ 6.00	-	✕ 7.00	-	13.00	65.00 *	

Figure 2.2 Example of Moodle System

2.4.3 Strathmore University Academic Management System (AMS)

Strathmore University's Academic Management System (AMS) is a homegrown platform that facilitates the management of academic records for students and staff. It addresses the challenge of academic monitoring by enabling students to register for courses, check results, and access academic reports digitally, reducing reliance on paperwork and manual result dissemination. Developed using custom web-based tools, AMS offers secure logins and role-specific access. However, its presentation of performance data is limited to basic tables and downloadable PDFS, which makes it difficult for students to analyse trends or pinpoint areas of weakness. Moreover, the system lacks features like real-time alerts or intelligent academic guidance. The proposed system aims to overcome these shortcomings by incorporating interactive dashboards, live performance analytics, and personalised academic insights, enhancing the usefulness of academic data for students and staff. (Strathmore University, nd).



Coursework Marks

BICS(April)-22

Download

2023-2024 || 2024-2025 || 2025-2026 ||

KEY: Provisional Final Registered

Academic Year: 2025-2026 Semester: FIRST SEMESTER

Subject	Ordinary		Supplementary	
	Attempt No.	Mark	Attemp No.	Mark
Advanced Database Systems	1		2	
Automata Theory and Computability	1		2	
Human Computer Interaction	1		2	
Informatics and Computer Science Project I	1		2	
Maisha Program 5	1		2	
Microprocessors	1		2	
Operations Research	1		2	
Spanish III	1		2	

KEY: Provisional Final Registered

Figure 2.3 Example of Strathmore AMS

2.5 Gaps in Current Approaches and Related Applications

Many existing academic tracking systems, such as Ellucian Banner, Moodle, and Strathmore AMS, have helped institutions manage academic records but still face several key inadequacies. These include delays caused by manual verification of grades, which slow down the release of results and hinder timely decision-making for students. (Collegis Education, 2021). In addition, most systems lack advanced features like trend analysis or personalised feedback, offering only basic grade summaries that do not help students understand their academic strengths and weaknesses. (Full Fabric, 2025). The user interfaces in many platforms also present information in rigid tables without visual tools like graphs, making it harder for users to track progress or spot performance patterns. These inefficiencies show that while current systems serve essential functions, they fail to deliver fast, insightful, and user-friendly academic tracking experiences.

2.6 Conceptual Framework

The system will support three main types of users: Students, Lecturers, and Administrators. Each of them will interact with the system based on their role. Students will log in to request academic reports such as GPA summaries or performance overviews for specific semesters or academic years. They can also view alerts or receive messages related to their educational progress.

Lecturers can submit student grades and corrections directly into the system. Their inputs undergo a validation process that checks for missing or incorrect data. Administrators, on the other hand, have access to complete system control. They will set academic rules such as grading scales and progression thresholds, and monitor student performance trends using admin reports.

The processing unit is at the core of the system, handling validation, GPA calculation, and rule-based alerts. It converts raw marks into GPA points, checks for failed core units, and identifies students who may be at academic risk. All this data is stored in a central academic records database.

From this database, the system can generate outputs like student dashboards, in-portal notifications, and performance reports for both students and administrators. By centralising data and automating analytics, the system ensures accuracy, transparency, and faster decision-making, replacing traditional manual academic tracking methods.

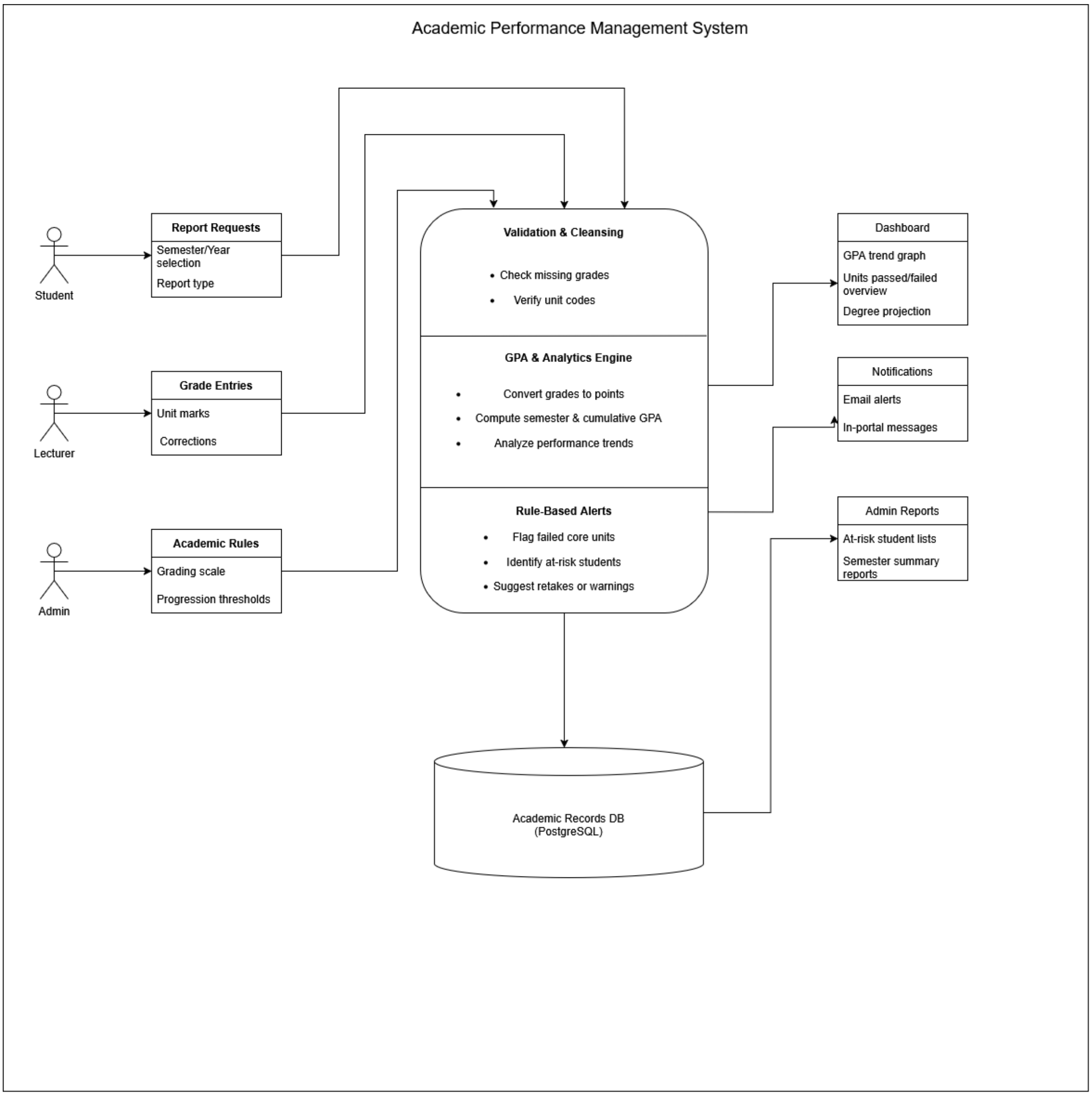


Figure 2.4 Conceptual Framework diagram

Chapter 3: METHODOLOGY

3.1 Introduction

This chapter outlines the methodology that will be followed by this proposed project. It presents the framework, tools, and structured activities used throughout the system's design and development. The methodology supports iterative and incremental growth to align with user requirements and system goals. It also details the tools and design strategies that will be used to ensure the final product is robust, user-centred, and adaptable.

3.2 Methodology and Paradigm

This project adopts the **Object-Oriented Analysis and Design (OOAD)** paradigm, combined with the **Agile Software Development** methodology, specifically the **Scrum framework**. OOAD is well-suited for modular systems like UAPMS, where objects such as *Student*, *Course*, and *GPA Record* can encapsulate data and behaviour (GeeksforGeeks, 2024). Its reusability and scalability make it a preferred choice for interactive, evolving systems.

Agile with Scrum was chosen because of its iterative and feedback-driven nature. (Project Management Institute (PMI), 2025). Given the academic setting and weekly reviews with our supervisor, it aligns well with our need for constant improvement and evolving project requirements. (Wikipedia Contributors, 2025). Each sprint results in a tangible deliverable, from documentation to functional modules, reviewed and refined based on feedback.

1. Choice of OOAD

OOAD was selected because it aligns with user-centric, modular systems and needs to evolve. Unlike SSAD, which focuses heavily on data flow and fixed processes, OOAD structures the system around **interacting objects** such as Student, Course, GPA Calculator, and Administrator, which match the real-world entities in our university environment (GeeksforGeeks, 2024). OOAD allows for high **reusability**, **extensibility**, and **precise mapping between requirements and implementation**.

2. Choice of Agile Methodology – Scrum

We use Scrum, an Agile methodology supporting **adaptive planning**, **early delivery**, and **continuous improvement**. Scrum was chosen because:

- The project allows for **incremental development**, with feedback from our supervisor at every stage.
- It accommodates **changing requirements** (e.g., adding performance graphs or feedback messages).
- It promotes **team collaboration** and **short sprints**, ideal for academic timelines.

3. **Scrum Practical Implementation**

The project started around **April 21** with an initial sprint to present the idea and proposal to the supervisor. The following sprints included work on Chapter 1, Chapter 2, and now Chapter 3. Each sprint lasts about one week and includes writing, revising, developing features, and testing tasks.

4. **Scrum Phases (Aligned to UAPMS Project)**

The UAPMS development is broken into **weekly sprints**, allowing consistent progress while incorporating feedback from the supervisor. Our team of two students acts as both the development team and product owners, collaboratively managing the **Product Backlog**, which contains features like login, GPA calculation, academic feedback, and reporting.

Each sprint begins with planning (task breakdown and assignment), then implementation, and ends with a review and retrospective discussion. Communication is streamlined via GitHub and WhatsApp. So far, the following sprints have been carried out:

- **Sprint 1:** Project concept was submitted and approved.
- **Sprint 2:** Drafted and submitted Chapter 1. Feedback was received with improvement suggestions.
- **Sprint 3:** Submitted corrected Chapter 1 and completed Chapter 2. Both were approved.
- **Sprint 4 (Current):** Drafting Chapter 3 (Methodology), to be submitted for supervisor review.

3.2.1 Requirements Gathering and Analysis (Sprint 1)

In this phase, we consulted with our supervisor to define the system's scope. The key problems identified included a lack of automated GPA tracking, no system-generated performance feedback, and an absence of visual academic progression indicators. Moreover, he suggested adding a feedback or summary generation report for lectures and students to help keep track of progress. Functional and non-functional requirements were documented, leading to the initial project proposal and refined problem statement.

3.2.2 System Design (Sprint 2)

With the problem and requirements well-defined, the second sprint involved designing the structural and behavioural components of the UAPMS. Key OOAD diagrams were drafted, such as the Use Case Diagram, Class Diagram, and Entity Relationship Diagram (ERD). These provide the blueprint for the system architecture, guiding development for modules like user authentication, GPA calculation, and degree tracking.

3.2.3 System Development (Sprint 3 - Ongoing)

Development began after the design was approved. This sprint-wise breakdown ensures modular implementation:

- **Sprint 3:** Developed and tested the **Login and Authentication** module.
- **Sprint 4:** Currently implementing the **GPA Calculation and Grade Entry** feature.
- **Sprint 5:** Planned work on **Feedback and Academic Progression Graphs**.
- **Sprint 6+:** Finalisation of **Admin Functionalities**, **User Roles**, and **Report Generation**.

Each feature is incrementally built and tested before moving to the next, ensuring continuous integration and user feedback adaptation.

3.2.4 Testing and Feedback (Upcoming Sprints)

After development, each module will undergo unit and integration testing. Test results will be analysed to fix bugs and improve performance. Feedback from the supervisor and users will refine the system further.

3.2.5 Documentation and Final Review

System documentation, user manuals, and final testing reports will be compiled and submitted. This includes the final project report and demo preparation.

3.3 System Analysis

System analysis involves identifying the functional and non-functional components of UAPMS and modelling them through diagrams that explain how users interact with the system and how data flows internally. These models help in planning system architecture and guiding developers in implementation.

3.3.1 Use Case Diagram

This diagram will illustrate the interactions between actors (students, admins) and the system features (e.g., login, view GPA, generate report, get feedback). It helps define the scope and major functionalities.

3.3.2 Sequence Diagram

Sequence diagrams will show the flow of interactions over time between system components (e.g., student submits grades → system calculates GPA, → feedback is displayed). This will clarify the logic behind each feature.

3.3.3 Entity Relationship Diagram (ERD)

The ERD will define how data entities like Student, Course, Grades, and User relate. It will be used to design the database schema with proper foreign and primary key relationships.

3.3.4 Class Diagram

The class diagram will represent the objects in the system and their attributes, methods, and relationships. Key classes include Student, Admin, Course, GPAEngine, and Feedback System.

3.3.5 Activity Diagram

The activity diagram will show the workflow of key system operations such as student login, grade entry, GPA calculation, and report generation. This will ensure logical process flow.

3.4 System Design

System design is the bridge between analysis and implementation. It refines the models created in the analysis phase and prepares for actual development. The diagrams here will inform how interfaces are structured, how data is managed, and how components interact.

3.4.1 Database Schema

The schema will organise all system data, including student info, grades, login credentials, degree requirements, and feedback. It defines relationships and ensures data integrity.

3.4.2 Wireframes

Wireframes will be developed to design the user interface for:

- Student login
- Dashboard with GPA and feedback
- Graphs for performance
- Admin panel

These will help plan intuitive navigation and layout.

3.4.3 System Architecture

UAPMS will follow a layered architecture:

5. **Presentation Layer** – Built using React/HTML/CSS
6. **Application Layer** – PHP or Javascript for logic and interaction
7. **Data Layer** – database storing system records

This architecture separates concerns and ensures easier maintenance.

3.5 System Development Tools and Techniques

These tools and technologies were selected for their effectiveness, ease of use, and compatibility with the UAPMS project goals.

3.5.1 NetBeans IDE

NetBeans will be used to write and manage Javascript and PHP code. It offers excellent support for back-end development, syntax checking, and version control integration.

3.5.2 Programming Languages

- **Javascript**: Used for complex logic, such as GPA calculation and backend logic
- **PHP**: For server-side operations such as login and data submission

- **React/HTML/CSS**: To create interactive and responsive front-end interfaces

This mix allows flexible development of both UI and back-end services.

3.5.3 Database – Postgresql

Postgresql will manage relational data such as users, courses, grades, and feedback. It's reliable, scalable, and works well with PHP and Javascript.

3.5.4 Version Control – Git and GitHub

Git is used to manage source code and documents. GitHub is the central repo where all sprint deliverables and chapters are uploaded and reviewed collaboratively.

3.5.5 Visualisation Tool – Chart.js or Google Charts

These tools will render graphs of GPA over time and other performance trends, providing students with visual insights into their academic progression.

3.6 Deliverables

Below are the major deliverables for the UAPMS project:

3.6.1 Proposal Document

Includes background, problem statement, objectives, scope, literature review, and methodology.

3.6.2 Design Diagrams

Includes all analysis and design models such as:

- Use Case Diagram
- Class Diagram
- ERD
- Activity Diagram
- Sequence Diagram
- Wireframes
- Database Schema

3.6.3 Working Web-Based System

The deployed system will include:

- Student login and registration
- Grade entry and GPA calculation
- Graphical performance feedback
- Degree requirement checker
- Admin dashboard

3.6.4 Test Reports

Document test cases, outcomes, and bug fixes for all modules. It will confirm that the system meets the defined requirements.

3.6.5 Documentation

The user manual, developer notes, system admin instructions, and troubleshooting guide are included.

3.6.6 Feedback Logs

It contains supervisor feedback per sprint and describes how it was addressed in the subsequent development stage.

3.6.7 Final Report

The final integrated project report combines all chapters, analysis, design, implementation, testing, and conclusions.

References

Evelyn Montgomery. (2024). *5 Key Benefits of Academic Management Systems*. Retrieved from Class Connect Pro: https://classconnectpro.com/blog/key-benefits-of-academic-management-systems

Collegis Education. (2021). *Level up your student information system (SIS): Moving processes from manual to automated*. Retrieved from Collegis Education: https://collegiseducation.com/insights/technology-management/level-up-student-information-system/

Commission for University Education (CUE). (2025). Retrieved from Commission for University Education (CUE): https://www.cue.or.ke/index.php?Itemid=140&id=10&layout=blog&option=com_content&view=category

contributors, Wikipedia. (2024). *Academic grading in Kenya*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Academic_grading_in_Kenya

DiplomaSafe. (2023, n.d.). *Student Information Systems (SIS) Explained*. Retrieved from DiplomaSafe: https://diplomasafe.com/student-information-system/

Education, C. (2021, nd). *Level Up Your Student Information System (SIS): Moving Processes From Manual To Automated* . Retrieved from Collegis Education: https://collegiseducation.com/insights/level-up-student-information-system/

Education, C. (2025). *Level Up Your Student Information System (SIS): Moving Processes From Manual To Automated* . Retrieved from Collegis Education: https://collegiseducation.com/insights/level-up-student-information-system/

Ellucian. (2025, nd). *What is a Student Information System in Higher Education?* Retrieved from Ellucian: https://www.ellucian.com/blog/what-student-information-system-higher-ed

Ellucian. (nd). Retrieved from https://www.ellucian.com/solutions/ellucian-student

Full Fabric. (2025). *The Role of Data Analytics in Higher Education*. Retrieved from Full Fabric: https://www.fullfabric.com/articles/how-predictive-analytics-can-boost-student-success-rates

GeeksforGeeks. (2024). *Object-Oriented Analysis and Design (OOAD)*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/object-oriented-analysis-and-design/

George D. Kuh, Jillian Kinzie, Jennifer A. Buckley, Brian K. Bridges, and John C. Hayek. (2006). *What Matters to Student Success: A Review of the Literature.* National Postsecondary Education Cooperative.

John P. Campbell, Peter B. DeBlois, Diana G. Oblinger. (2007). Retrieved from EDUCAUSE Review: https://er.educause.edu/articles/2007/7/academic-analytics-a-new-tool-for-a-new-era

Moodle. (nd). Retrieved from https://docs.moodle.org/403/en/Grader_report

MRCC EdTech. (2024). Retrieved from MRCC EdTech: https://www.mrccedtech.com/how-predictive-analytics-can-help-improve-student-outcomes/

Project Management Institute (PMI). (2025). *Agile Project Management with Scrum*. Retrieved from Project Management Institute (PMI): https://www.pmi.org/learning/library/agile-project-management-scrum-6269

Ronel B. Dayanghirang, Alexander A. Hernandez. (2022). *Mobile Based Gradebook with Student Outcomes Analytics.* arXiv.

SEAtS Software. (2023). Retrieved from SEAtS Software: https://www.seatssoftware.com/why-student-retention-strategies-must-be-data-driven/

Strathmore University. (nd). Retrieved from AMS.

Wikipedia. (2024). *Academic grading in Kenya*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Academic_grading_in_Kenya

Wikipedia Contributors. (2025). *Agile Software Development*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Agile_software_development

Appendix

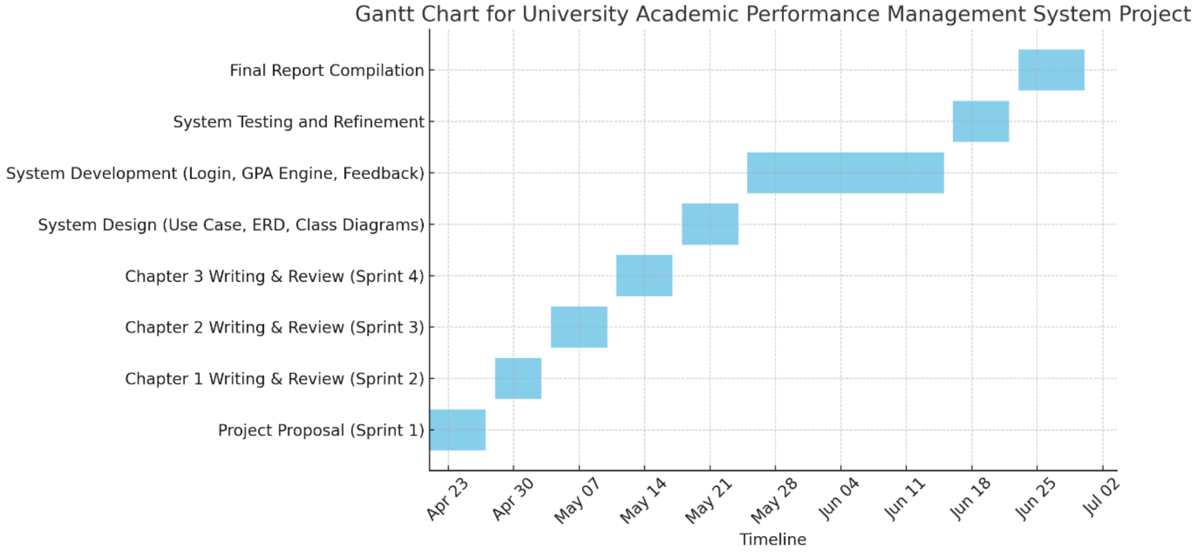


Figure 5.1 Gantt chart showcasing our sprints