**Name:** Jacob Miller

**System Name:** Student Progress Tracker System

**Short project code-name:** STApp

**Requested Launch Date**: Late 2024-2025

**Department(s) Affected By Project:** Education, Software Development

**Project’s Customers:** Students, Teachers, Parents

**Date Request Submitted:** May 5th 2024



**Table of Contents**

Cover Page

Table of Contents

Executive Summary

1.0 Introduction and Overview

* Problem Statement
* Project Vision and Scope
* Requirements Summary
* Stakeholders and Their Interests
* Expected Costs and Benefits
* Constraints
* Recommendation
* Document Overview

2.0 System Initiation

* Project Initiation Request (PIR)
* Reply Letter

3.0 Feasibility Assessment

* Introduction
* Feasibility Analysis
* Technical Feasibility
* Resource Feasibility
* Schedule Feasibility
* Organizational Feasibility
* Legal and Contractual Feasibility
* Additional Comments
* Conclusion

4.0 Requirements Definition

* Introduction
* Functional Requirements
* Data Requirements
* Non-functional Requirements

5.0 Requirements Model

* Introduction
* Use-Case Diagram
* Use-Case Descriptions

6.0 System Evolution

7.0 Conclusions and Recommendations

Appendices

* Questionnaires and Surveys
* Additional Reference Materials

Glossary

Bibliography­­­­

**Executive Summary:**

The Student Progress Tracker app changes how students manage their academic progress. The project aims to tackle students’ common challenges with tracking grades, assignments, and overall performance.

Students often struggle to keep up with their academic responsibilities in today’s educational landscape. They miss deadlines, have poor time management, and lack clear insight into their progress, which can hurt their academic success. The Student Progress Tracker app addresses these issues by offering a comprehensive, easy-to-use solution tailored to students’ needs.

The app features a personalized dashboard that gives a quick overview of a student’s academic journey, including current grades, upcoming assignments, and GPA trends. With automatic GPA calculations, the grade tracker lets students easily input and monitor their grades for various assignments and exams. An assignment calendar provides an organized view of upcoming tasks and deadlines, with reminders and notifications. Students can benefit from the prioritization feature highlighting essential assignments by managing their workload more effectively. Students stay engaged in the app because of motivational elements, like achievement badges and quotes. The app prioritizes security and accessibility, ensuring data privacy and supporting disabled users.

This app offers significant benefits. This app’s intuitive design and motivational features are expected to improve student engagement and actively encourage students to participate in academic activities. The assignment calendar and prioritization tools will facilitate better time management, helping students stay organized and meet their deadlines. Using the app can help students do better in school by giving them detailed feedback and helping them identify areas where they can improve so they can take proactive steps toward reaching their goals.

The main stakeholders in this project include students who will benefit from better tracking and managing their academic progress. Teachers can provide feedback and monitor student performance, while parents will be interested in their children’s academic achievements. School administrators will ensure the app aligns with educational goals and policies. The project’s feasibility is high, with manageable resource allocation and scheduling risks. The technical requirements are within the development team’s capabilities, and the project has solid organizational support.

In summary, the Student Progress Tracker app represents a significant step forward in enhancing the academic experience for students. With its comprehensive features and user-friendly design, the app will empower students to take control of their academic success. Our collaboration with the client ensures the app meets the needs of all stakeholders and aligns with educational goals and policies.

**1.0 Introduction Overview**

**Problem Statement:**

We developed the Student Progress Tracking App in response to the clear challenge students face in effectively managing their academic responsibilities. Balancing assignments, grade monitoring, and maintaining motivation have become increasingly challenging in the modern educational landscape. Traditional methods, such as manual record-keeping or rudimentary planners, cannot support students adequately in navigating these complexities, resulting in disorientation and reduced engagement. Given the necessity for a more organized and supportive system, an interesting opportunity exists to create a sophisticated and user-friendly solution that promotes efficient academic management and encourages continuous student engagement and achievement.

**Project Vision and Scope:**

This section should capture the initial vision and scope of the proposed system. Take your client’s PIR information and craft a vision statement based on the Opportunity Statement, Initial Vision, and Expected Benefits sections. 1—3 sentences should be enough. Then, briefly describe what you plan to build in response to this vision, including scope, boundaries, and environment. An Expected Costs and Benefits section below will deal more with Costs and Benefits, focusing mainly on opportunity and vision.

**Vision statement:**

We aim to develop a Student Progress Tracking App that transforms how students navigate their academic path. By providing a user-friendly platform for tracking progress, managing assignments, and staying motivated, we aim to empower students to achieve their educational goals with confidence and success.

**Scope & Boundaries:**

Among the Student Progress Tracking App features are a personalized dashboard, functionality for tracking grades, tools for managing assignments, and motivational elements. The system’s scope involves the creation of a web-based platform that students can access using different devices. The project’s boundaries exclude advanced data analytics features and external learning management systems integration.

**Environment:**

The system’s development will employ modern web technologies to ensure compatibility with commonly used web browsers and mobile devices. It will leverage cloud-based infrastructure for scalability and reliability. The environment for deployment will include secure hosting services to safeguard student data and ensure compliance with privacy regulations.

**Requirements Summary:**

* **Efficient Academic Monitoring:**

Needs to optimize the academic monitoring process for students, facilitating seamless progress tracking, assignment management, and goal setting.

* **Comprehensive Insights for Decision-Making:**

Must deliver comprehensive insights into students’ academic performance, empowering educators and administrators to make informed decisions and interventions whenever necessary.

* **Effective Communication and Collaboration:**

Should enable smooth communication and collaboration among students, teachers, and parents/guardians, improving engagement and support networks within the educational community.

* **Data Security and Privacy Compliance:**

Must strictly comply with stringent data security and privacy regulations to protect sensitive student information and ensure trust and confidence among stakeholders.

* **Scalability and Accessibility:**

Needs scalability to accommodate future growth and accessibility across different devices and platforms to guarantee inclusivity and usability for all stakeholders.

* **Integration with Existing Systems:**

Integrate the system with current educational platforms and systems, minimizing disruptions and enhancing the efficiency of data management and workflow.

* **Training and Support:**

Provide adequate training and support resources to ensure all users can use the system’s features and functionalities effectively, promoting user adoption and satisfaction.

**Stakeholders and Their Interests:**

List the primary stakeholders and their interest(s) in the outcome of this development process. Again, the PIR is your starting point. But it is probably not a complete list of all interested parties. As the analyst, you may need to think beyond the usual stakeholders to create a full list. Do not forget that you and the developers will hold the stakeholders accountable for clarifying the product’s goals and requirements when needed.

* The system primarily caters to students and their preferences for a user-friendly interface, comprehensive academic insights, and effective tools for managing their academic journey.
* Educators and academic counselors seek resources that offer valuable insights into student performance, promote effective communication with students, and enable data-driven decision-making to improve academic support and interventions.
* Parents and guardians have a keen interest in remaining well-informed about their child’s academic progress, receiving prompt updates on assignments and grades, and actively collaborating with teachers to enhance their child’s learning experience.
* The objective of school administrators is to guarantee the efficient implementation and integration of the system within the educational institution, adherence to data security and privacy regulations, and alignment with institutional goals and priorities.
* Educational technology specialists prioritize the technical aspects of the system, ensuring its functionality, scalability, and compatibility with established educational platforms and systems.
* The IT department prioritizes the system’s technical infrastructure, security measures, and integration capabilities to guarantee seamless implementation and ongoing support.
* Curriculum developers may find value in utilizing the system to obtain insights on curriculum effectiveness, pinpoint areas for enhancement, and customize educational resources to suit students’ requirements.
* Regulatory bodies and privacy advocates promote the importance of adhering to data security and privacy regulations to safeguard student information and rights.
* Third-party vendors and service providers may be involved in providing technical support, hosting services, or integrations with external systems, aiming to ensure seamless operation and functionality of the system.

**Expected Costs and Benefits:**

**Business Benefits:**

* **Improved Academic Performance:**

The system aims to enhance students’ academic performance by equipping them with tools and insights for progress monitoring, goal establishment, and efficient workload management.

* **Enhanced Student Engagement:**

The system’s primary objective is to enhance student engagement and motivation in their academic endeavors by integrating gamified components and interactive functionalities, fostering a conducive learning environment.

* **Streamlined Communication and Collaboration:**

Through the facilitation of seamless communication and collaboration among students, teachers, and parents/guardians, the system aims to enhance support networks within the educational community and foster student achievement.

* **Data-Driven Decision Making:**

Educators and administrators can expect many advantages from having access to comprehensive academic insights, enabling them to make data-driven decisions and implement targeted interventions to support student learning and achievement.

**Cost Areas:**

* **Development and Implementation Costs:**

These costs cover system development, software licensing, infrastructure setup, and initial implementation initiatives.

* **Training and Support Costs:**

Investments in training resources and support services will be required to ensure the effective utilization of the system’s features and functionalities.

**Maintenance and Upkeep Expenses:**

The continuous maintenance and upkeep of the system, which includes software updates, security enhancements, and technical support, will cause recurring costs in the long run.

* **Integration Costs:**

We should consider the costs of integrating the system with current educational platforms and systems and ensure compatibility and smooth data exchange.

* **Regulatory Compliance Costs:**

Adherence to data security and privacy regulations may cause allocating resources and implementing measures to protect student information and ensure compliance with regulations.

**Constraints:**

* **Time Constraints:**

Limited timeframes may restrict the development scope or lead to rushed implementation, potentially compromising quality. Prioritizing tasks, employing agile development methodologies, and regularly communicating with stakeholders to manage expectations and adjust timelines as needed.

* **Resource Limitations:**

Limited resources such as budget, staff, or technology may impede development and restrict the system’s capabilities. Efficient resource allocation involves leveraging opensource technologies, outsourcing non-core tasks, and seeking alternative funding sources or partnerships to supplement resources.

* **Technical Constraints:**

Technical limitations, such as compatibility issues, platform restrictions, or legacy system dependencies, may pose challenges during system integration or deployment. We conduct thorough technical assessments, prioritize interoperability and scalability, explore workarounds or alternative solutions, and collaborate closely with IT specialists to address technical challenges.

* **Regulatory Compliance Requirements:**

Complying with data security and privacy regulations can complicate the development process, influencing system performance and user satisfaction. Complying with regulations causes adherence to industry rules, regular security issue assessments, utilization of solid encryption and access controls, and transparent data handling.

**User Adoption Challenges:**

Resistance to change or usability issues may impede user adoption and acceptance of the system, undermining its effectiveness and value. We conduct user testing and feedback sessions throughout the development lifecycle to ensure optimal results. We provide comprehensive training and support resources, implement design principles that prioritize the user, and cultivate a culture of continuous improvement and user engagement.

* **Budget Constraints:**

Having limited financial resources can restrict how much development can be done, lead to delays in project milestones, or compromise the quality of deliverables. Prioritizing essential features and functionalities, seeking cost-effective solutions and alternative funding sources, negotiating vendor contracts and payment terms, and implementing budget monitoring and control mechanisms are critical for optimizing resource utilization.

**Recommendation:**

* Carefully examine the details provided in this proposal to ensure all aspects align with Wanderer’s Tools’ goals and objectives. Provide feedback or request any clarifications needed.
* Establish a committee comprising key stakeholders, including representatives from the student body, teachers, school administrators, and IT staff. This committee will guide the project, provide input, and oversee progress.
* Identify and allocate the resources, including budget, personnel, and technology, to support the development and implementation phases of the project.
* Approve the project timeline and development approach outlined in the proposal. Build the Student Progress Tracker app, following the suggested methods, deadlines, and significant outcomes.
* Involve all relevant stakeholders in the development process. Ensure regular communication and feedback loops to address concerns and incorporate valuable insights into the project.
* Prepare a detailed testing and quality assurance plan to ensure the app meets all functional and non-functional requirements. Allocate resources for rigorous testing and evaluation before the full-scale launch.
* Begin creating comprehensive training materials and support resources to facilitate a smooth onboarding process for students, teachers, and administrators.
* Determine a realistic launch date for the app, considering the development and testing timelines. Plan a phased rollout if necessary to ensure a smooth transition and adoption.

**Document Overview:**

**Introduction and Overview:**

Provides a thorough rundown of our proposed system. Discusses the issues aimed to resolve, our ambitious aspirations, and the business requirements. Moreover, engages in a collaborative discussion to generate ideas for our next steps.

**Project Vision and Scope:**

Provides a comprehensive overview of the nature of our system. Drafts a plan for our grand visions, clearly defining the tasks the system will undertake and those it will not and establishing any necessary boundaries.

**Business Requirements:**

Delves into the system’s requirements to fulfill the business’s demands. It resembles a comprehensive checklist encompassing all the essential elements that the system must achieve in order to meet our business objectives.

**Stakeholders and Their Interests:**

Enumerates all the players involved and the aspirations they hold. Understanding the interests of all individuals is crucial in maintaining satisfaction and fostering collective progress.

**Expected Costs and Benefits:**

Delves into the financial aspect, specifically this project’s initial investment and anticipated returns. Examines both the monetary and intangible advantages.

**Constraints:**

Addresses any challenges that may arise and outline our approach to overcoming them. The focus is on ensuring readiness for unforeseen obstacles.

**Development Approach:**

Articulates our blueprint for developing the system. Our discussion encompasses our strategy and the projected completion date.

**Risk Management:**

Assesses potential challenges and develops approaches to promote forward movement.

**Testing and Quality Assurance:**

Outlines our strategy for ensuring optimal system functionality. The discussion encompasses everything, including testing approaches and the definition of success.

**Training and Support:**

Discusses strategies for familiarizing individuals with the system and ensuring their satisfaction during its usage.

**Implementation Plan:**

The strategic framework outlines the steps to establish and launch the system. Addresses all aspects of the process, including the setup and ensuring everyone’s participation.

**Maintenance and Upkeep:**

Discusses the strategies for ensuring the seamless operation of the system once it is deployed. This requires keeping up with updates, implementing necessary fixes, and ensuring the satisfaction of all parties involved.

**2.0 System Initiation**

# Project Initiation Request (PIR)

**General Project Information**

|  |  |
| --- | --- |
| **Project Name:** | *Student Tracker. Short project code-name: STApp* |
| **Two Sentence Request Description:** | *Develop a user-friendly app to help students track their academic progress, manage assignments, and set goals while providing insightful summaries and immediate feedback. The business result is to enhance student success and engagement by fostering accountability, organization, and motivation in their academic endeavors.* |
| **Requested Launch Date(s):** | *Late 2024-2025.* |
| **Department(s) Affected By Project:** | *Education*  *Software Development* |
| **Project’s Customers:** | *Students*  *Teachers*  *Parents* |
| **Date Request Submitted:** | *April 16, 2024* |

# Project Sponsor and Manager

**Project Sponsor Business Project Manager & Requestor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name:** | Andy Cameron |  | **Name:** | *Jacob Miller* |
| **Title:** | Professor | **Title:** | *Student/Developer of Product.* |
| **Department:** | Computer Science - SPU | **Department:** | *Computer Science – Software*  *Developer* |
| **eMail:** | acameron@spu.edu | **eMail:** | *millerj38@spu.edu* |

# Business Problem or Opportunity: The motivation for this request

*The request stems from recognizing a prevalent challenge. Students lack streamlined tools for monitoring academic progress, organizing assignments, and establishing goals. Students encounter difficulties maintaining organization, receiving prompt feedback, and sustaining motivation throughout their*

|  |
| --- |
| *educational journeys. Thus, the aim is to develop a user-friendly app that directly tackles these obstacles to support student achievement and engagement. Timely intervention is crucial to enhance academic outcomes and elevate student satisfaction and confidence in their educational pursuits.* |

# Justification, Impact, and Importance

**Assumptions**

* Positive Feedback from Users: Users should respond positively to the app’s features and functionality, which will generate increased adoption rates and referrals.
* Increased Student Engagement: Implementing the Student Progress Tracking App will lead students to engage more with their academic responsibilities, resulting in improved performance and outcomes.

**Competitive Landscape / Context**

* Market Demand for Educational Technology: The education technology market is experiencing rapid growth, driven by increased digitization in the education sector and a growing demand for innovative solutions to enhance learning outcomes.
* Competitor Offerings: Companies in the educational technology industry offer comparable solutions for tracking student progress and providing academic support. However, few apps offer the same features and easy-to-use interface as ours, giving us a competitive edge.

**One Time**

**On-Going**

**Tangible Return, Opportunity, or Value**

$ 0

$ 0

Increased Student Retention

Cost Savings from Improved Efficiency

$ 5,000

$ 2,000

$ 0

$ 1,000

**Intangible Benefits Impact or Value**

* Enhanced Reputation: Strengthening the company’s reputation as an innovative educational technology leader leads to increased brand loyalty and market differentiation.
* Improved Student Satisfaction: By enhancing the overall student experience and satisfaction, we can expect positive referrals and increased student enrollment.

# Product Requirements

**4.1. Must Haves**

|  |
| --- |
|  |
| Personalized Dashboard: A user-friendly dashboard that gives students an overview of their current grades, upcoming assignments, and academic progress trends.  Grade Tracking and GPA Calculation: Students can input and track their grades for various assignments and exams, automatically calculating GPA based on these grades. |

**4.2. Could Haves** (Nice to Haves)

|  |
| --- |
|  |
| Integration with Learning Management Systems: The ability to integrate with popular LMS platforms such as Canvas automatically sync assignments, grades, and deadlines.  Goal Setting and Progress Tracking: The feature allows students to set academic goals and track their progress toward achieving them, providing motivation and accountability. |

**4.3. Won’t Haves** (Don’t Do’s, aka Out of Scope)

|  |
| --- |
|  |
| Social Media Integration: Integrating social media features or sharing functionalities is out of this project’s scope, as the focus is solely on tracking academic progress and management. |

## 3.0 Feasibility Assessment

**Introduction:**

This section thoroughly examines the feasibility of the project, covering technical capabilities, available resources, scheduling constraints, organizational readiness, legal considerations, and contractual obligations. Each of these factors will be assessed using a rating system to gauge the level of feasibility or risk involved. The ratings will range from high, showing strong feasibility, to low, signifying notable challenges or concerns. The subsequent Feasibility Analysis will comprehensively evaluate the project’s viability by delving deeper into each area.

**Feasibility Analysis:**

**Technical Feasibility (Rating: Moderate):**

Our initial assessment suggests that the proposed system’s technical feasibility is moderate. While the system’s requirements align with existing technologies and platforms, some aspects may pose challenges, such as integrating with legacy systems or ensuring scalability to accommodate future growth. However, it is possible to overcome these technical hurdles with the right expertise and resources.

**Resource Feasibility (Rating: High):**

Given the availability of skilled personnel and adequate infrastructure required for system development and implementation, we deem resource feasibility to be high. Our organization has the human capital and technology resources to effectively support the project’s requirements.

**Schedule Feasibility (Rating: Moderate):**

Schedule feasibility is moderate, primarily because of the complexity of the project and potential dependencies on external factors. While we aim to adhere to a realistic timeline, unforeseen challenges or delays may affect the project’s schedule. We will promptly implement mitigation strategies to address any deviations from the planned timeline.

**Organizational Feasibility (Rating: High):**

Our organization rates the organizational feasibility high because the project aligns with our strategic goals and priorities. Key stakeholders have expressed support for the initiative, and there is a clear understanding of roles and responsibilities among team members. Regular communication and collaboration will ensure continued organizational buy-in throughout the project lifecycle.

**Legal and Contractual Feasibility (Rating: Moderate):**

Legal and contractual feasibility is assessed as moderate, considering the need to comply with relevant laws, regulations, and contractual obligations. While initial reviews show no significant legal barriers, we will continuously consult legal counsel to address any emerging issues and ensure compliance with applicable statutes and agreements.

**Additional Comments:**

Given the complexity of the proposed system and the various feasibility factors involved, ongoing monitoring and proactive risk management will be critical to the project’s success. Regular communication and collaboration among stakeholders and a flexible approach to addressing challenges will help navigate potential obstacles effectively. Continuous evaluation of feasibility factors throughout the project lifecycle will enable timely adjustments and optimizations to ensure the achievement of project goals.

**Conclusion:**

Overall, the feasibility of this project appears promising, with several key factors contributing to its potential success. Technical feasibility is moderate, considering potential challenges in integration and scalability, but it can be managed effectively with the right expertise. Resource feasibility is high, supported by the organization’s ample resources and skilled personnel. Schedule feasibility is moderate, acknowledging potential complexities, but mitigated by proactive risk management strategies.

Organizational feasibility is high, as the project aligns with strategic goals and enjoys strong support from stakeholders. Legal and contractual feasibility is moderate, requiring ongoing diligence to ensure compliance with regulations and agreements.

Reservations include the potential for unforeseen technical challenges and schedule delays, which may affect project timelines and resource allocation. Regular monitoring, proactive risk identification, and swift mitigation actions will be essential to managing these risks. Maintaining open communication channels and fostering a culture of collaboration will help address challenges effectively and ensure project success.

## 4.0 Requirements Definition

**Introduction:**

Within this section, we shall discuss our system’s required functionalities and desired performance levels. Functional requirements serve as an exhaustive catalog for the system, encompassing tasks and actions it must handle, ranging from user interactions to data processing. In contrast, non-functional requirements prioritize the system’s characteristics, including speed, reliability, and security. To fully comprehend the desired functionalities and operational procedures of our proposed system, we must delve deeper into these requirements.

**Functional Requirements:**

* **User-Friendly Interface:**
  + Description: The system must provide a user-friendly interface for students to easily track their academic progress, manage assignments, and set goals.
  + Priority: Must have
* **Grade Tracking and GPA Calculation:**
  + Description: The system must allow students to input and track their grades for various assignments and exams, automatically calculating GPA based on these grades.
  + Priority: Must have
* **Assignment Prioritization:**
  + Description: The system must prioritize assignments based on importance and due dates to help students manage their time and workload effectively.
  + Priority: Must have
* **Data Security and Privacy:**
  + Description: The system must ensure data security and privacy compliance to safeguard student information.
  + Priority: Must have
* **Comprehensive Insights:**
  + Description: The system should offer extensive insights into students’ academic performance through features like daily, weekly, and monthly summaries of completed tasks and GPA trends.
  + Priority: Should have
* **Assignment Calendar:**
  + Description: The system should include an assignment calendar feature that visually organizes upcoming tasks and deadlines, complete with reminders and notifications to help students stay on top of their responsibilities.
  + Priority: Should have
* **Cross-Platform Accessibility:**
  + Description: The system should be accessible across various devices and platforms to accommodate students’ diverse needs and preferences.
  + Priority: Should have
* **Gamification:**
  + Description: The system could incorporate gamified elements such as achievement badges and motivational quotes to enhance student engagement and motivation.
  + Priority: Could have
* **Social Media Integration:**
  + Description: The system will not have integration with social media platforms for sharing academic progress or achievements.
  + Priority: Won’t have
* **Third-Party Application Integration:**
  + Description: The system will not support integration with third-party applications or tools outside the academic tracking scope.
  + Priority: Won’t have

**Data Requirements:**

* **User Data:**
  + Student Profile Information:
    - Name:Full name of the student.
    - Student ID: Unique identifier for each student.
    - Email: Contact email for notifications and password recovery.
    - Password: Encrypted user password.
    - Profile Picture: A profile picture for user personalization.
  + Preferences and Settings:
    - Notification Preferences: Settings for how and when notifications are sent.
    - Theme Preferences: User’s choice of interface themes or color schemes.
* **Academic Data:**
  + Course Information:
    - Course Name: Name of the course.
    - Course Code: Unique identifier for each course.
    - Instructor Name: Name of the course instructor.
    - Credits: Number of credits for the course.
  + Assignment Information:
    - Assignment Name: Title of the assignment.
    - Assignment Description: Detailed description of the assignment.
    - Due Date: Deadline for the assignment submission.
    - Priority Level: Importance level of the assignment.
    - Status: In progress, completed, etc.
  + Grades and Performance:
    - Assignment Grades: Grades for individual assignments.
    - Exam Grades: Grades for exams.
    - Overall GPA: Cumulative Grade Point Average.
    - Daily, Weekly, and Monthly Summaries: Aggregated data on tasks completed and GPA trends.
* **Calendar and Scheduling Data:**
  + Calendar Events:
    - Event Name: Title of the event.
    - Event Date and Time: When the event is scheduled, assignment due, exam date, etc.
    - Reminders: Notification settings for event reminders.
  + Study Schedule:
    - Study Sessions: Planned study times.
    - Session Duration: Length of each study session.
    - Subjects Covered: Subjects/topics to be studied during each session.
* **Gamification Data:**
  + Achievements and Badges:
    - Badge Name: Title of the badge earned.
    - Criteria Met: Conditions that were satisfied to earn the badge.
    - Date Awarded: Date when the badge was earned.
  + Motivational Quotes:
    - Quote Text: The actual quote.
    - Author: Author of the quote.
* **Security and Compliance Data**
  + Audit Logs:
    - Log ID: Unique identifier for each log entry.
    - Timestamp: Date and time of the log entry.
    - Activity Description: Description of the activity, login, data update, etc.
  + Privacy Settings:
    - Data Sharing Preferences: Users’ choices regarding data sharing with third parties.
    - Consent Records: Records of user consent for data collection and usage.

**Non-functional Requirements:**

* **Performance:**

The system must show efficient operation and timely responsiveness to user commands, even in periods of elevated activity.

* **Reliability:**

Users must encounter minimal disruptions or errors when using the system, thus guaranteeing its consistent availability and functionality.

* **Scalability:**

The system must expand alongside the user base and effectively manage increased data volumes without compromising performance.

* **Usability:**

The user interface design should prioritize intuitiveness and user-friendliness to minimize the requirement for extensive training and promote efficient operation.

* **Accessibility:**

The system must adhere to accessibility standards to accommodate users with disabilities and support assistive technologies.

* **Security:**

Implementing robust security measures is necessary to protect user data from unauthorized access, breaches, and cyber threats.

* **Compatibility:**

The system must showcase seamless compatibility across various web browsers and operating systems to achieve universal functionality.

* **Maintainability:**

The maintenance and updates of the system will be straightforward because of the presence of clear documentation and a modular design that facilitates future enhancements.

* **Compliance:**

The system must adhere to relevant laws, regulations, and industry standards, specifically data protection and privacy policies.

* **Performance Monitoring:**

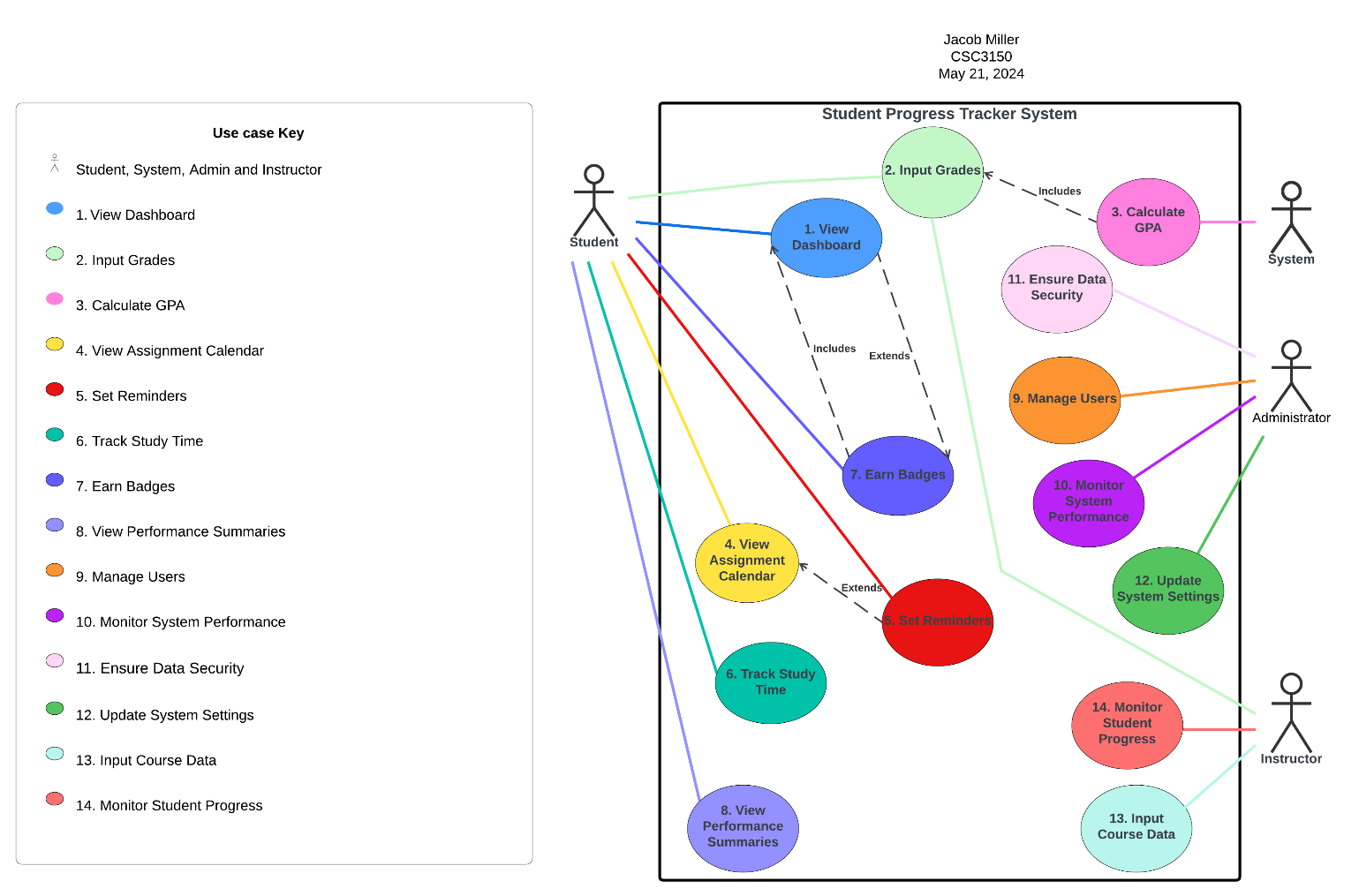
Regular monitoring and evaluation of performance ensure the consistent functioning of the system, facilitating the identification of areas needing improvement.

**5.0 Requirements Model**

**Introduction:**

This section presents the Requirements Model for the Student’s Progress Tracking App. It includes a Use Case Diagram and supporting documentation to illustrate the interactions between the system and its users. A Use Case Diagram visually represents the system’s functionality and interactions with various actors. In this context, “actors” refer to the users or other systems interacting with the application. Each “use case” represents a specific function or goal the system can perform. The diagram and descriptions provided will help you understand how the system will be used and what functionality it must have. The relationships between use cases, such as “extend” and “include,” will also be explained to show optional, conditional, and reusable behaviors.

**Use-Case Diagram:**



Link:

https://lucid.app/lucidchart/b4a9661c-6c80-4078-8220-258c9c9093ba/edit?viewport\_loc=-1205%2C-534%2C3066%2C3522%2C0\_0&invitationId=inv\_f8561710-e9c0-46ee-b9bb-6672041a9423

**Use-Case Descriptions:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: View Dashboard | | **ID**: UC-001 | **Importance**: Must Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Parents: Interested in monitoring their child’s academic progress. * School Administration: Interested in overall student performance metrics. | | | |
| **Brief Description**:  This use case allows students to access their personalized dashboard, which provides an overview of their academic performance, including grades, GPA trends, and achievement badges. | | | |
| **Trigger**: Student logs into the app.  **Type** (mark one): Temporal | | | |
| **Relationships**:  **Association**: Existing Student communicates with View Dashboard.  **Extend**: Earn Badges. | | | |
| **The Normal Flow of Events**:   1. Student logs into the application. 2. System verifies student credentials. 3. System displays the personalized dashboard. 4. Student views their grades, GPA trends, and earned badges. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   1. If the student’s login credentials are incorrect, the system will prompt the student to re-enter their credentials. 2. If the student has no grades entered, the system will display a message indicating no data is available. | | | |
| **Special Requirements:**   * Performance: Dashboard must load within 5 seconds. * Security: Only authenticated users can access the dashboard. * User Interface: The dashboard should be intuitive and display data clearly. | | | |
| **To do/Issues:**   * Ensure real-time data synchronization for accurate dashboard display. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Input Grades | | **ID**: UC-002 | **Importance**: Must Have |
| **Primary Actor**: Student, Professor | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Teachers: Interested in the accuracy of student’s self-reported grades. | | | |
| **Brief Description**:  This use case enables students or the instructors to enter the grades for various assignments and exams, which the system will use to calculate their GPA. | | | |
| **Trigger**: Student selects the option to input grades.  **Type** (mark one): External | | | |
| **Relationships**:  **Association**: Student communicates with Input Grades.  **Includes:** Calculate GPA | | | |
| **The Normal Flow of Events**:   1. Student navigates to the grade input section. 2. Student selects the course and assignment/exam. 3. Students enter the grade received. 4. System saves the grade and updates the GPA. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   1. If the student enters an invalid grade format, the system will prompt for a correct format. 2. If the student tries to enter grades for a non-existent course, the system will prompt an error. | | | |
| **Special Requirements:**   * Performance: Grade input must be saved within 3 seconds. * Security: Ensure only the authenticated student can input their grades. | | | |
| **To do/Issues:**   * Validate grade format before saving to the database. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Calculate GPA | | **ID**: UC-003 | **Importance**: Must Have |
| **Primary Actor**: System | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Students: Interested in accurate GPA calculation. * School Administration: Needs accurate GPA for records and reporting. | | | |
| **Brief Description**:  This use case allows the system to automatically calculate the student’s GPA based on the grades entered. | | | |
| **Trigger**: Grades are entered or updated.  **Type** (mark one): External | | | |
| **Relationships**:  **Association**: System calculates GPA upon grade entry.  **Includes:** Input Grades. | | | |
| **The Normal Flow of Events**:   1. Student inputs grades. 2. System retrieves all entered grades. 3. System calculates GPA based on the retrieved grades. 4. System updates the student’s dashboard with the new GPA. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   1. If there is an error in retrieving grades, the system will log the error and notify the student. | | | |
| **Special Requirements:**   * Performance: GPA must be calculated within 2 seconds after grade entry. * Accuracy: Ensure GPA calculation adheres to the school’s grading policy. | | | |
| **To do/Issues:**   * Verify calculation formula with school grading policy. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: View Assignment Calendar | | **ID**: UC-004 | **Importance**: Must Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Teachers: Interested in ensuring students are aware of deadlines. * Parents: Interested in helping students manage their assignments. | | | |
| **Brief Description**:  This use case allows students to view a calendar with all upcoming assignments and deadlines. | | | |
| **Trigger**: Student navigates to the assignment calendar.  **Type** (mark one): External | | | |
| **Relationships**:  **Association**: Student communicates with View Assignment Calendar.  **Extend:** Set Reminders. | | | |
| **The Normal Flow of Events**:   1. Student logs into the application. 2. Student navigates to the assignment calendar section. 3. System displays the calendar with upcoming assignments and deadlines. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * If there are no upcoming assignments, the system will display a message indicating no assignments are due. | | | |
| **Special Requirements:**   * Performance: Calendar must load within 3 seconds. * User Interface: Calendar should be visually appealing and easy to navigate. | | | |
| **To do/Issues:**   * Ensure reminders are set correctly and integrated with the calendar. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Set Reminders | | **ID**: UC-005 | **Importance**: Should Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Parents: Interested in ensuring students meet deadlines. * School Administration: Interested in student punctuality and responsibility. | | | |
| **Brief Description**:  This use case allows students to set reminders for upcoming assignments and deadlines. | | | |
| **Trigger**: Student selects an assignment from the calendar.  **Type** (mark one): External | | | |
| **Relationships**:  **Association**: Student communicates with Set Reminders.  **Extend:** View Assignment Calendar. | | | |
| **The Normal Flow of Events**:   1. Student navigates to the assignment calendar. 2. Student selects an assignment. 3. System prompts student to set a reminder. 4. Student sets the reminder for the chosen assignment. 5. System saves the reminder and schedules a notification. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the reminder cannot be set, the system will prompt an error and ask the student to try again. | | | |
| **Special Requirements:**   * Performance: Reminder must be set and saved within 2 seconds. * User Interface: Reminder setting should be intuitive and quick. | | | |
| **To do/Issues:**   * Ensure integration with device notification systems for reminders. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Track Study Time | | **ID**: UC-006 | **Importance**: Could Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Parents: Interested in their child’s study habits. * School Counselors: Interested in supporting students’ time management skills. | | | |
| **Brief Description**:  This use case enables students to log and monitor the amount of time they spend studying. | | | |
| **Trigger**: Student selects the option to log study time.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Student communicates with Track Study Time.  **Extend:** | | | |
| **The Normal Flow of Events**:   1. Student logs into the application. 2. Student navigates to the study time tracking section. 3. Student starts a new study session. 4. System logs the start time. 5. Student ends the study session. 6. System logs the end time and calculates the total study duration. 7. System updates the student’s study time log. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to log the study session, it will prompt the student to try again. | | | |
| **Special Requirements:**   * Performance: Study time must be accurately logged. * User Interface: Study time tracking should be easy to start and stop. | | | |
| **To do/Issues:**   * Ensure the accuracy of time tracking. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Earn Badges | | **ID**: UC-007 | **Importance**: Could Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Parents: Interested in their child’s achievements. * Teachers: Interested in motivating students through recognition. | | | |
| **Brief Description**:  This use case allows students to earn achievement badges for reaching specific academic milestones. | | | |
| **Trigger**: Student achieves a milestone.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Student communicates with Earn Badges.  **Extend:** View Dashboard. | | | |
| **The Normal Flow of Events**:   1. Student achieves an academic milestone. 2. System recognizes the milestone achievement. 3. System awards a badge to the student. 4. Student views the new badge on their dashboard. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to award a badge, it will log the error and notify the student. | | | |
| **Special Requirements:**   * Performance: Badge awarding must be instantaneous upon achievement recognition. * User Interface: Badges should be visually appealing and prominent on the dashboard. | | | |
| **To do/Issues:**   * Define and implement criteria for various badges. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: View Performance Summaries | | **ID**: UC-008 | **Importance**: Should Have |
| **Primary Actor**: Student | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Parents: Interested in a summary of their child’s academic performance. * School Administration: Interested in tracking overall student performance trends. | | | |
| **Brief Description**:  This use case allows students to view summaries of their academic performance over different time periods (daily, weekly, monthly). | | | |
| **Trigger**: Student selects the option to view performance summaries.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Student communicates with View Performance Summaries.  **Extend:** | | | |
| **The Normal Flow of Events**:   1. Student logs into the application. 2. Student navigates to the performance summaries section. 3. System displays the summaries for the selected time period (daily, weekly, or monthly). | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If there is no performance data available for the selected time period, the system will display a message indicating no data is available. | | | |
| **Special Requirements:**   * Performance: Summaries must be generated and displayed within 3 seconds. * User Interface: Summaries should be easy to read and interpret. | | | |
| **To do/Issues:**   * Ensure accurate and up-to-date performance data is used for summaries. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Manage Users | | **ID**: UC-009 | **Importance**: Must Have |
| **Primary Actor**: Administrator | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * School Administration: Interested in maintaining accurate and up-to-date user records. * Teachers: Interested in ensuring all students and faculty have access to the system. | | | |
| **Brief Description**:  This use case allows administrators to add, edit, and delete user accounts within the system. | | | |
| **Trigger**: Administrator selects the option to manage users.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Administrator communicates with Manage Users.  **Extend:** | | | |
| **The Normal Flow of Events**:   1. Administrator logs into the application. 2. Administrator navigates to the user management section. 3. Administrator selects to add, edit, or delete a user. 4. System performs the requested user management action. 5. System confirms the action was successful. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to manage the user, it will log the error and notify the administrator. | | | |
| **Special Requirements:**   * Security: Only authenticated and authorized administrators can manage users. * User Interface: User management functions should be intuitive and efficient. | | | |
| **To do/Issues:**   * Ensure proper validation for user data input. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Monitor System Performance | | **ID**: UC-010 | **Importance**: Should Have |
| **Primary Actor**: Administrator | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * School Administration: Interested in ensuring system reliability and performance. * Students: Interested in uninterrupted access to the system. | | | |
| **Brief Description**:  This use case allows administrators to monitor the system’s performance and reliability. | | | |
| **Trigger**: Administrator selects the option to monitor system performance.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Administrator communicates with Monitor System Performance. **Extend:** | | | |
| **The Normal Flow of Events**:   1. Administrator logs into the application. 2. Administrator navigates to the system performance monitoring section. 3. System displays performance metrics and reliability data. 4. Administrator reviews the data. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to retrieve performance data, it will log the error and notify the administrator. | | | |
| **Special Requirements:**   * Performance: Performance data must be updated in real-time. * User Interface: Performance metrics should be clear and easy to interpret. | | | |
| **To do/Issues:**   * Define performance thresholds and alerts for system performance issues. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Ensure Data Security | | **ID**: UC-011 | **Importance**: Must Have |
| **Primary Actor**: Administrator | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Students: Interested in the protection of their personal and academic data. * School Administration: Interested in compliance with data security regulations. | | | |
| **Brief Description**:  This use case allows administrators to manage security protocols and ensure the protection of user data. | | | |
| **Trigger**: Administrator selects the option to manage data security.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Administrator communicates with Ensure Data Security. **Extend:** | | | |
| **The Normal Flow of Events**:   1. Administrator logs into the application. 2. Administrator navigates to the data security management section. 3. Administrator updates security settings and protocols. 4. System applies the new security settings. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to update security settings, it will log the error and notify the administrator. | | | |
| **Special Requirements:**   * Security: Ensure compliance with data protection regulations. * Performance: Security settings updates must be applied immediately. | | | |
| **To do/Issues:**   * Regularly update security protocols to address new threats. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Update System Settings | | **ID**: UC-012 | **Importance**: Should Have |
| **Primary Actor**: Administrator | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * School Administration: Interested in maintaining optimal system configuration. * Students: Interested in a well-configured and functioning system. | | | |
| **Brief Description**:  This use case allows administrators to update and configure system settings as needed. | | | |
| **Trigger**: Administrator selects the option to update system settings.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Administrator communicates with Update System Settings. **Extend:** | | | |
| **The Normal Flow of Events**:   1. Administrator logs into the application. 2. Administrator navigates to the system settings section. 3. Administrator updates the desired settings. 4. System applies the updated settings. 5. System confirms the updates were successful. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to update settings, it will log the error and notify the administrator. | | | |
| **Special Requirements:**   * Performance: System settings updates must be applied without causing downtime. * User Interface: Settings should be easy to update and configure. | | | |
| **To do/Issues:**   * Document all system settings and their implications. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Input Course Data | | **ID**: UC-013 | **Importance**: Must Have |
| **Primary Actor**: Instructor | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Students: Interested in up-to-date course information. * School Administration: Interested in maintaining accurate course records. | | | |
| **Brief Description**:  This use case allows instructors to add details about courses, including assignments and exams. | | | |
| **Trigger**: Instructor selects the option to input course data.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Instructor communicates with Input Course Data. **Extend:** | | | |
| **The Normal Flow of Events**:   1. Instructor logs into the application. 2. Instructor navigates to the course data input section. 3. Instructor enters details about the course, assignments, and exams. 4. System saves the course data. 5. System confirms the data was saved successfully. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to save the course data, it will log the error and notify the instructor. | | | |
| **Special Requirements:**   * Performance: Course data must be saved within 3 seconds. * User Interface: Data entry should be straightforward and user-friendly. | | | |
| **To do/Issues:**   * Ensure validation of course data input. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Monitor Student Progress | | **ID**: UC-014 | **Importance**: Should Have |
| **Primary Actor**: Instructor | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:   * Students: Interested in receiving feedback on their progress. * Parents: Interested in their child’s academic progress. | | | |
| **Brief Description**:  This use case allows instructors to monitor and review the academic progress of their students. | | | |
| **Trigger**: Instructor selects the option to monitor student progress.  **Type** (mark one): External | | | |
| **Relationships**:  **Association:** Instructor communicates with Input Course Data. **Extend:** | | | |
| **The Normal Flow of Events**:   1. Instructor logs into the application. 2. Instructor navigates to the student progress monitoring section. 3. System displays the academic progress of students. 4. Instructor reviews and monitors student progress. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * + If the system fails to retrieve student progress data, it will log the error and notify the instructor. | | | |
| **Special Requirements:**   * Performance: Progress data must be displayed within 3 seconds. * User Interface: Progress data should be clear and easy to interpret. | | | |
| **To do/Issues:**   * Ensure real-time data synchronization for accurate progress monitoring. | | | |

**6.0 System Evolution**

The developers envision several enhancements and upgrades for the Student Progress Tracking App beyond the initial Minimum Viable Product (MVP). These future features aim to expand the app’s capabilities and ensure it remains a valuable tool for students, educators, and administrators.

One key feature planned for future versions is generating detailed reports. This functionality will allow users, particularly administrators and instructors, to create comprehensive reports on student performance, assignment completion rates, and additional academic metrics. These reports will be customizable based on specific criteria such as date ranges, course codes, and individual student performance, providing valuable insights that can inform data-driven decisions and strategies to support student success.

Another significant enhancement is a sophisticated notification system. This system will inform students of essential updates, assignment deadlines, upcoming exams, and other critical academic events through various channels, including email, SMS, and in-app alerts. Effective communication is crucial for student success, and automated notifications will help ensure students stay informed and engaged, reducing the likelihood of missed deadlines and improving overall academic performance.

The development team also plans to include advanced analytics in future app versions. These capabilities will offer deeper insights into student performance trends, identify at-risk students, and suggest personalized improvement plans based on historical data. By leveraging advanced analytics, educators can proactively address issues and tailor interventions to individual student needs, fostering a more supportive and effective learning environment.

Future versions will integrate seamlessly with popular Learning Management Systems (LMS) such as Canvas, Blackboard, and Moodle. By integrating seamlessly with popular Learning Management Systems (LMS) such as Canvas, Blackboard, and Moodle, future versions will automatically synchronize grades, assignments, and course data between the tracking app and the LMS. By reducing the need for manual data entry and ensuring system consistency, we can simplify data management for educators and improve the user experience.

We also plan to enhance the mobile app, incorporating features like offline access, push notifications, and an improved user interface design specifically designed for mobile devices. These improvements will ensure students can access and use the system effectively from anywhere, increasing engagement and usability.

Regarding system infrastructure, as the user base grows, server upgrades will be necessary to handle increased data storage and processing demands. This may involve incorporating more robust servers, expanding storage capacity, and implementing load-balancing solutions. We will schedule regular software updates to introduce new features, fix bugs, and improve system security. These tasks involve upgrading the underlying database management system, applying the most recent security patches, and optimizing the system’s performance.

We will develop comprehensive training materials and support resources to help users maximize the benefits of new features. The offering comprises a variety of resources, including online tutorials, user manuals, and dedicated customer support channels. The aim is to ensure a seamless transition to new features and foster user confidence and satisfaction with the system.

By planning for these future enhancements and upgrades, we aim to create a dynamic and adaptable system that develops in response to user needs and technological advancements, maintaining its relevance and effectiveness in supporting student success over the long term.

**7.0 Conclusions and Recommendations**

**Appendices:**

Berkeley, U. (n.d.). Application system development guidelines. Application System Development Guidelines | Information Security Office. [https://security.berkeley.edu/application-system-developmentguidelines](https://security.berkeley.edu/application-system-development-guidelines)

Lamping, P. (2023, November 17). What are business applications? benefits, careers, & more.

University of Cincinnati. <https://online.uc.edu/blog/what-are-business-applications/>

Database Development. (2023, August 24). How do you understand and define your application’s data requirements? How to Define Your Application’s Data Requirements in 6 Steps. <https://www.linkedin.com/advice/1/how-do-you-understand-define-your-applications#:~:text=Data%20requirements%20are%20the%20specifications,application’s%20functional%20and%20performance%20needs>.

Relationships in use-case diagrams. UML relationships in use-case diagrams. (n.d.). <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-relationships-in-use-case>

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2019). Systems analysis and design in a Changing World 6th edition. Cengage Learning.

**Glossary:**

**Functional Requirements:** A system must carry out specified tasks and actions to effectively accomplish its desired goal, which involves engaging with users and processing data.

**Non-functional Requirements:** When considering a system’s overall effectiveness and usability, essential attributes and characteristics include performance, reliability, and security.

**System Proposal:** A comprehensive document delineates a proposed system’s or project’s scope, objectives, and specifications.

**User Interface (UI):** The point of interaction between a user and a computer system encompasses graphical components such as screens, pages, and buttons.

**Data Processing:** Raw data is transformed into meaningful information through operations and transformations.

**MVP: Minimum Viable Product:** The initial product version with the minimum features required to satisfy early customers and provide feedback for future development.

**LMS: Learning Management System** - A software application used for the administration, documentation, tracking, reporting, and delivery of educational courses or training programs.

**Analytics:** The process aims to identify patterns, trends, and insights by systematically analyzing data or statistics.

**Integration:** Combining different software systems and applications allows them to function collectively, often through APIs (Application Programming Interfaces) or middleware.

**Push Notifications:** Messages or alerts sent from a server to a user’s device, even when the user is not actively using the application.

**Server Upgrades:** Improvements or enhancements to a server’s hardware or software components to improve its performance, reliability, or security.

**Software Updates:** New versions or releases of software applications that include bug fixes, security patches, and new features or enhancements.

**User Engagement:** The measure of how actively involved users interact with a system or application.

Training Materials: Resources and materials educate users on effectively using a system or application.

**Support Resources:** Tools, documentation, and help provided to users to help them troubleshoot issues or get answers to their questions.

**Bibliography:**

* CSC 3150 notes
* Class Lectures
* Class Recordings
* Group Projects
* Larman drawing tool