Feasibility Study Report

SAMS (Student Academic Management System By Lazrek Nassim)

Project Duration: 11 Weeks

Contents

1 Technical Feasibility

1.1 Required Tools, Software, and Hardware

The SAMS project requires specific development tools and infrastructure to support both backend and frontend development. This section outlines all necessary technical resources.

Development Environment:

- **Visual Studio Code**
- **Git** - Version control system for code management and collaboration
- **Postman** - API testing and endpoint validation tool
- **PostgreSQL** database administration and query interface

Backend Technology Stack:

- **Java Development Kit (JDK) 1** - Required for Spring Boot compatibility
- **Spring Boot Framework** - Core backend framework including:
  - Spring Web (REST API development)
  - Spring Data JPA (Database interaction layer)
  - Spring Security (Authentication and authorization)
- **PostgreSQL 14 or higher** - Relational database management system for data persistence

Frontend Technology Stack:

- **Node.js (v18 or higher)** - JavaScript runtime for frontend tooling
- **Vue.js 3.x** - Progressive JavaScript framework for user interface development
- **Vue Router** - Client-side routing for single-page application navigation
- **Axios** - HTTP client library for API communication

1.2 Skills Assessment

This assessment evaluates existing competencies against project requirements to identify learning needs and potential challenges.

Skill/Technology	Required Level	Current Level	Gap Analysis
HTML/CSS	Intermediate	Intermediate	None - Can build responsive layouts
JavaScript	Intermediate	Intermediate	Comfortable with ES6+ syntax and concepts
Vue.js	Intermediate	Basic	Need to learn component composition, routing, and state management patterns
Java Programming	Intermediate	Beginner	Currently learning in parallel course - basic syntax understood but OOP concepts still developing
Spring Boot	Intermediate	None	Complete framework unfamiliarity
REST API Design	Intermediate	Basic	Understand HTTP methods but lack practical implementation experience
PostgreSQL	Intermediate	Theoretical	Know relational database concepts but no hands-on SQL query writing or administration
Spring Security	Basic	None	Authentication/authorization concepts understood theoretically but no implementation experience
Git Version Control	Basic	Basic	Familiar with commit, push, pull operations

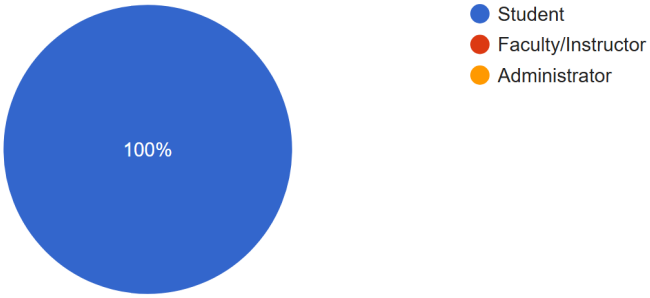
2 Market / User feasibility:

Section 1: About You

Copy chart

1. What is your role?

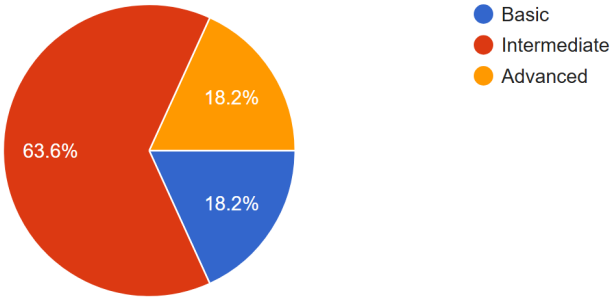
11 responses



Copy chart

2. Tech comfort level?

11 responses

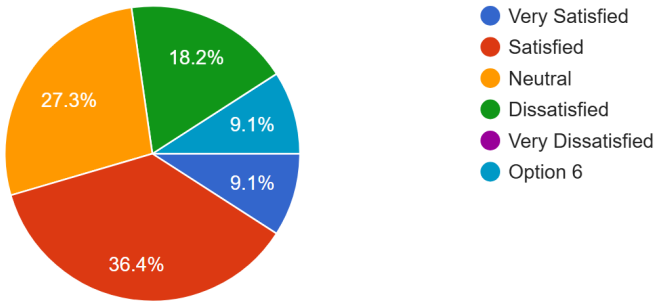


Section 2: Current Experience

Copy chart

3. How satisfied are you with your current academic system?

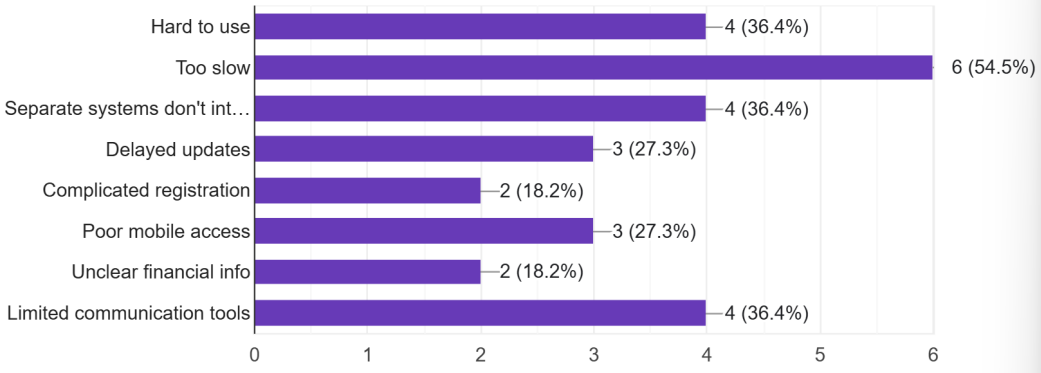
11 responses



Copy chart

4. Top 3 problems with current systems? (Check up to 3)

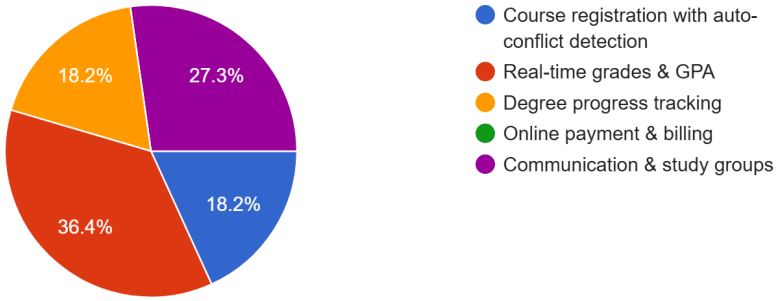
11 responses



Copy chart

5. which features are important for you? (at least 3)

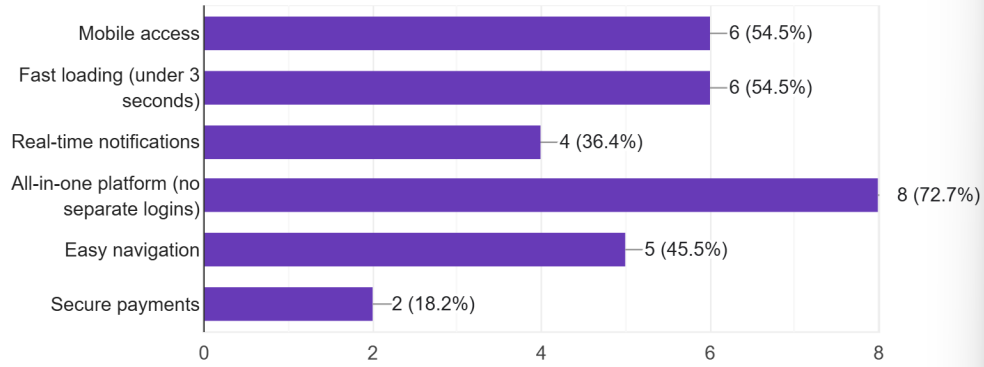
11 responses



Copy chart

6. Must-have features? (Check all that apply)

11 responses

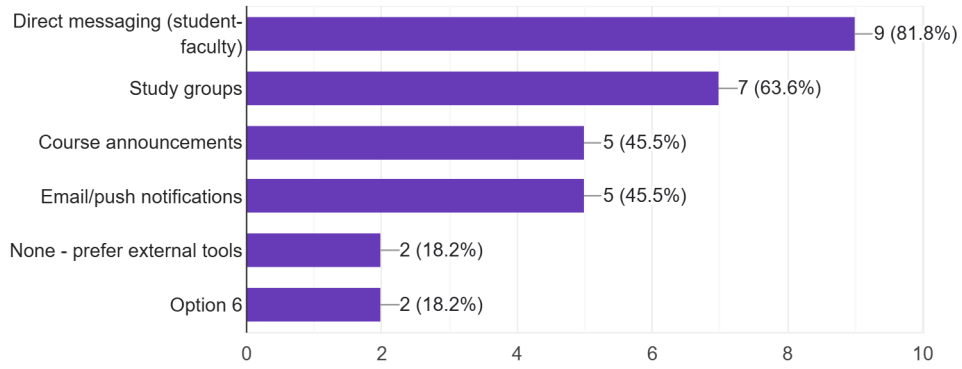


Section 5: Communication

Copy chart

9. Which communication features would you use? (Check all that apply)

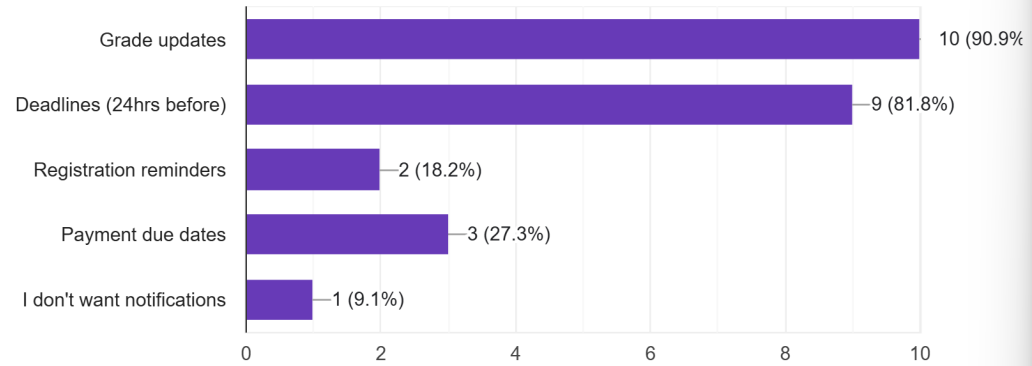
11 responses

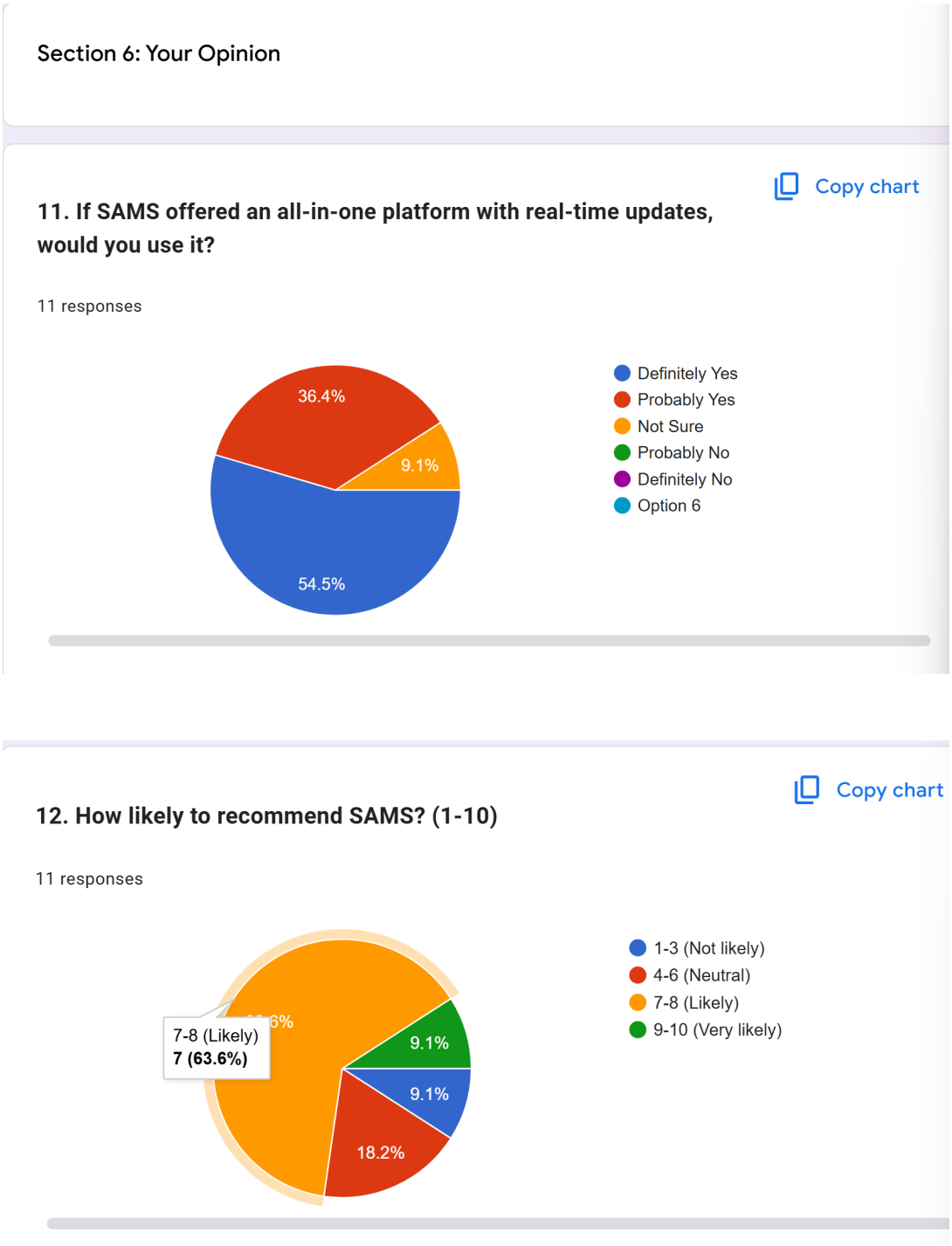


Copy chart

10. Notification preferences? (Check all that apply)

11 responses





3 Schedule Feasibility

3.1 Project Timeline Overview

The project spans 11 weeks divided into four main phases.

Phase Distribution:

- **Phase 1 (Weeks 1-3):** Foundation & Planning - 3 weeks (27%)
- **Phase 2 (Weeks 4-7):** Core Backend Development - 4 weeks (36%)

- **Phase 3 (Weeks 8-10):** Frontend & Integration - 3 weeks (27%)
- **Phase 4 (Week 11):** Polish & Documentation - 1 week (10%)

3.2 Task Breakdown with Duration Estimates

Week	Phase	Primary Tasks	Estimated Hours
Week 1	Foundation	Requirements engineering, user personas, user stories, scope definition	10 hours
Week 2	Foundation	Database schema design, ER diagrams, REST API design, system architecture	10 hours
Week 3	Foundation	Development environment setup, basic Spring Boot project, User entity, basic authentication	10 hours
Week 4	Backend Core	User management system - Repository, Service, Controller layers, unit testing	10 hours
Week 5	Backend Core	Course management - Course entity, relationships, enrollment entity, search functionality	10 hours
Week 6	Backend Core	Enrollment business logic - prerequisite validation, schedule conflicts, capacity management	10 hours
Week 7	Backend Core	Grade management, GPA calculations, JWT security, API documentation	10 hours
Week 8	Frontend	Vue.js project setup, routing, authentication components, dashboard layouts	10 hours
Week 9	Frontend	Core UI implementation - course browsing, enrollment forms, grade management interfaces	10 hours
Week 10	Integration	Integration testing, bug fixes, error handling, performance optimization	10 hours
Week 11	Polish	Technical documentation, user manuals, deployment guides, presentation preparation	10 hours

Total Estimated Effort: 110 hours over 11 weeks (average 10 hours per week)

3.3 Feasibility Within Semester Timeline

Can this project be completed in 11 weeks?

The project is **feasible but challenging**. Success depends on maintaining consistent weekly progress and making the Week 8 frontend decision promptly. The JavaFX backup plan significantly improves overall feasibility by providing a realistic alternative if the Vue.js integration proves too time-consuming.