**BrainBoost: An Intelligent Academic Task Manager and Learning Scheduler**

**August 12, 2025**

Submitted by:

**Carlos, Mary Lovensky M.**

**Gonzalvo, Mariz Nicole C.**

**Juan, Jonalito A.**

**Lozada, Paul Andre N.**

**Vidal, Jelyn M.**

Submitted to:

**Engr. Lance Aeron F. Guevarra**

**BrainBoost: An Intelligent Academic Task Manager and Learning Scheduler**

**1. Introduction**

**1.1 Purpose**

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for BrainBoost, an intelligent academic task manager and learning scheduler designed specifically for college students.

**1.2 Scope**

BrainBoost is a comprehensive study management application that combines task tracking, intelligent scheduling, progress monitoring, and AI-powered recommendations to help college students improve their academic productivity and time management.

**1.3 Definitions and Abbreviations**

* **AI:** Artificial Intelligence
* **API:** Application Programming Interface
* **UI:** User Interface
* **UX:** User Experience
* **SRS:** Software Requirements Specification

**2. Overall Description**

**2.1 Product Perspective**

BrainBoost addresses the critical problem of academic task management among college students by providing a centralized, intelligent platform that combines traditional task management with AI-powered study optimization.

**2.2 Product Functions**

* Task and assignment tracking with deadline management
* Intelligent study schedule generation
* Automated reminders and notifications
* Academic progress tracking and analytics
* Study recommendations and optimization

**2.3 User Classes and Characteristics**

**Primary Users: College Students**

* Age range: 18-25 years
* Tech-savvy with smartphone and computer access
* Managing multiple courses simultaneously
* Varying levels of time management skills
* Need for centralized academic organization

**2.4 Operating Environment**

* **Platform:** Mobile Application (iOS and Android)

**3. Functional Requirements**

**3.1 Task and Assignment Tracker**

**3.1.1 Task Creation**

* Users shall be able to create new tasks with title, description, due date, priority level, and subject/course
* Users shall be able to categorize tasks (Assignment, Quiz, Project, Exam, Reading)
* Users shall be able to set estimated completion time for tasks

**3.1.2 Task Management**

* Users shall be able to mark tasks as complete
* Users shall be able to delete tasks
* Users shall be able to view tasks in list, calendar, and kanban board views
* Users shall be able to filter and sort tasks by date, priority, subject, or completion status

**3.2 Smart Study Scheduler**

**3.2.1 Schedule Generation**

* System shall automatically generate personalized study schedules based on task deadlines and user availability
* System shall consider task priority and estimated completion time when creating schedules
* Users shall be able to input their free time blocks and preferred study hours
* System shall distribute study sessions to avoid cramming and optimize retention

**3.2.2 Schedule Management**

* Users shall be able to view their study schedule in daily, weekly, and monthly views
* Users shall be able to manually adjust suggested study sessions
* System shall automatically reschedule when tasks are completed or deadlines change
* Users shall be able to block out unavailable time slots

**3.3 Reminders and Notifications**

**3.3.1 Notification Types**

* System shall send deadline reminders (customizable: 1 day, 1 week, custom intervals)
* System shall send study session start notifications
* System shall send break reminders during study sessions
* System shall send motivational progress notifications

**3.3.2 Notification Delivery**

* Users shall be able to choose notification methods (push, email, SMS)
* System shall respect user's quiet hours and do-not-disturb settings

**3.4 Progress Tracker**

**3.4.1 Analytics and Statistics**

* System shall track and display completion rates by subject and time period
* System shall track time spent studying per subject
* System shall show productivity trends and patterns
* System shall display upcoming deadlines and overdue tasks dashboard

**3.4.2 Visual Reports**

* System shall generate visual charts and graphs for progress tracking
* Users shall be able to view weekly, monthly, and semester progress reports
* System shall provide goal-setting and achievement tracking

**3.5 AI-Based Suggestions**

**3.5.1 Study Optimization**

* System shall analyze user study patterns to suggest optimal study times
* System shall recommend break intervals based on task difficulty and duration
* System shall suggest review sessions for completed topics based on spaced repetition principles
* System shall recommend task prioritization based on deadlines and workload

**3.5.2 Personalization**

* System shall learn from user behavior to improve recommendations over time
* System shall adapt suggestions based on user performance and completion rates
* System shall provide warnings for potential schedule conflicts or over commitment

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* System response time shall be less than 2 seconds for all user interactions
* System shall support concurrent access by up to 10,000 users
* Mobile app shall launch within 3 seconds
* Data synchronization across devices shall occur within 5 seconds

**4.2 Security Requirements**

* User data shall be encrypted in transit and at rest
* System shall implement secure user authentication (OAuth 2.0 or equivalent)
* System shall implement role-based access control

**4.3 Mobile Usability Requirements**

* New users shall complete on-boarding and create their first task within 3 minutes
* App shall work offline for core task management (viewing, creating, editing tasks)

**4.4 Reliability Requirements**

* System shall have automated daily backups
* System shall recover from failures within 15 minutes
* Data loss shall not exceed 5 minutes of user activity

**5. System Architecture**

**5.1 High-Level Architecture**

* **Frontend:** React Native for cross-platform mobile development
* **Backend:** Node.js with Express.js REST API
* **Database:** MongoDB for user data, SQLite for offline storage
* **AI/ML:** Cloud-based Python microservice with TensorFlow Lite for on-device processing
* **Cloud Infrastructure:** Firebase for authentication and real-time sync
* **Push Notifications:** Firebase Cloud Messaging (FCM) for Android, Apple Push Notification Service (APNs) for iOS

**5.2 Key Mobile App Components**

* User Authentication & Profile Management
* Task Management Module
* Smart Scheduling Engine
* Push Notification Handler
* Offline Data Sync Manager
* AI Recommendation Engine
* Analytics & Progress Tracking
* Settings & Preferences Controller

**6. User Interface Requirements**

**6.1 Mobile App Design Principles**

* Native Mobile Experience: Touch-optimized interface with gesture support
* Portrait-first design with landscape support for calendar views
* Thumb-friendly navigation with bottom tab bar
* Dark mode support for better battery life and night studying
* Offline-first design with sync when connectivity restored

**6.2 Key Mobile App Screens**

* Dashboard/Home with today's tasks and quick actions
* Task creation with camera integration for assignment photos
* Calendar view with swipe gestures and haptic feedback
* Study timer with focus mode and break notifications
* Progress dashboard with interactive charts
* Settings panel with notification preferences and sync options

**7. Data Requirements**

**7.1 User Data**

* Profile information (name, email, school, major)
* Authentication credentials
* Preferences and settings
* Usage analytics and behavior patterns

**7.2 Academic Data**

* Tasks and assignments
* Course and subject information
* Deadlines and schedules
* Progress and completion data
* Study session logs

**7.3 System Data**

* Notification logs
* Performance metrics
* Error logs and debugging information

**8. Integration Requirements**

**8.1 Mobile App Integrations**

* Device Integration: Camera for document scanning, calendar access, contacts
* Platform Integration: App Store/Google Play for distribution and updates
* System Integration: Push notifications, widgets, shortcuts, Siri/Google Assistant
* Cloud Services: iCloud/Google Drive for backup, Google/Apple sign-in
* Educational Platforms: Canvas, Blackboard mobile APIs where available

**9. Constraints and Assumptions**

**9.1 Technical Constraints**

* Mobile application must work on devices with limited storage (minimum 1GB available)
* Must be compatible with both iOS and Android platforms
* Must function with intermittent internet connectivity
* AI features require minimum dataset for personalization

**9.2 Assumptions**

* Users have basic smartphone/computer literacy
* Stable internet connection for most usage scenarios
* Users will actively input their academic tasks and schedules

**10. Acceptance Criteria**

**10.1 MVP Acceptance Criteria**

* Users can successfully create, edit, and complete tasks
* Basic study schedule generation works accurately
* Notification system delivers timely reminders
* Progress tracking displays accurate completion statistics

**11 Project Schedule**

**BrainBoost Gantt Chart (August – November 8, 2025**)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase / Task** | **Week 1 Aug** | **Week 2 Aug** | **Week 3 Aug** | **Week 4 Aug** | **Week 1 Sep** | **Week 2 Sep** | **Week 3 Sep** | **Week 4 Sep** | **Week 1 Oct** | **Week 2 Oct** | **Week 3 Oct** | **Week 4 Oct** | **Week 1 Nov** | **Week 2 Nov** |
| Requirements Gathering | ████ | ████ |  |  |  |  |  |  |  |  |  |  |  |  |
| User Research & Analysis |  | ████ | ████ |  |  |  |  |  |  |  |  |  |  |  |
| System Architecture Design |  |  | ████ | ████ |  |  |  |  |  |  |  |  |  |  |
| UI/UX Design & Prototyping |  |  |  | ████ | ████ |  |  |  |  |  |  |  |  |  |
| Database Design |  |  |  |  | ████ | ████ |  |  |  |  |  |  |  |  |
| Core Mobile Development |  |  |  |  |  | ████ | ████ | ████ | ████ |  |  |  |  |  |
| AI Features Development |  |  |  |  |  |  | ████ | ████ | ████ | ████ |  |  |  |  |
| Advanced Features |  |  |  |  |  |  |  |  |  | ████ | ████ | ████ |  |  |
| Unit Testing |  |  |  |  |  |  | ████ | ████ | ████ | ████ |  |  |  |  |
| Integration Testing |  |  |  |  |  |  |  |  |  | ████ | ████ | ████ |  |  |
| User Acceptance Testing |  |  |  |  |  |  |  |  |  |  | ████ | ████ | ████ |  |
| Bug Fixes & Optimization |  |  |  |  |  |  |  |  |  |  |  | ████ | ████ | ████ |
| App Store Preparation |  |  |  |  |  |  |  |  |  |  |  | ████ | ████ |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  | ████ | ████ |
| Final Deployment |  |  |  |  |  |  |  |  |  |  |  |  |  | ████ |

**12. Appendices**

**12.1 Glossary**

* **Spaced Repetition:** Learning technique that incorporates increasing intervals of time between subsequent review
* **MVP:** Minimum Viable Product
* **Kanban Board:** Visual workflow management method using cards and columns

**12.2 References**

* Educational psychology research on study habits
* Time management best practices for students
* Mobile app design guidelines