Software Requirements Specification for Brightpath E-Learning Platform

Abdelrhman Karam Abdelrsoul Ibrahim Hegazy (120220126)

Aliaa Maamoun Ibrahim (120220255)

Kamal Elsawah (120220005)

Marchellino Michel (120220317)

Fatma Mahmoud Abd Elrahman (120220355)

Ziad Mahmoud Hanafi (120220096)

May 22, 2025

Egypt Japan University of Science and Technology

Contents

Revision History

Version	Date	Description
1.0	May 22, 2025	Initial complete draft with improvements and added glossary, revision history, appendix, and enhanced PDF metadata.

1 Introduction

1.1 Purpose and Objective

This SRS defines requirements for Brightpath - a gamified e-learning platform addressing:

- Transition from passive memorization to interactive learning in Egyptian national schools
- Reduction of parental concerns (70% reported) through family engagement features
- Bridging classroom technology gaps (only 25% use interactive tools) via AI personalization and VR
- Improving engagement for students aged 6-15 through adaptive learning paths

1.2 Scope

The platform shall include:

- Gamified lessons with points/badges/leaderboards
- VR modules for STEM subjects
- Parent portals for progress tracking and event scheduling
- AI-driven adaptive learning engine
- School licensing integration
- Cross-platform support (Web/iOS/Android)

2 Stakeholders

- **Primary Users:** Students 6-15 years (national schools)
- **Secondary Users:** Parents, Teachers, School Administrators
- **Key Partners:** National Schools, UX Designers, AI/ML Developers
- **Third-Party Services:** Google OAuth, Cloudflare CDN, TensorFlow Lite
- **System Admins:** Platform maintenance and user management

3 User and System Requirements

3.1 User Requirements

- Students 6-11: Access game-like lessons with visual rewards (e.g., math challenge badges)
- Students 12-15: Receive AI-adjusted learning paths based on quiz performance
- Parents: Schedule learning events + receive weekly progress reports
- Teachers: Create VR lessons and view class engagement heatmaps
- Admins: Monitor system health through Grafana dashboards

3.2 System Requirements

- **Frontend:** React.js (Web), Flutter (Mobile), Unity (Game Engine)
- **Backend:** Django REST API, Redis (Leaderboards), PostgreSQL
- **AI Engine:** TensorFlow Lite for on-device personalization
- **VR:** Three.js framework with ¡100ms latency
- **Scalability:** Kubernetes cluster handling 10,000 concurrent users
- **Security:** AES-256 encryption, COPPA compliance for under-13 users

4 Functional Requirements

- **Gamification Module:**
  - Point system for lesson completion
  - Redeemable badges for learning streaks
  - Class/group leaderboards
- **VR Learning:**
  - Interactive 3D simulations (e.g., virtual chemistry lab)
  - Collaborative VR projects
- **Parent-Teacher Interface:**

  - In-app messaging system
  - Automated performance alerts
  - Event scheduling calendar

5 Non-Functional Requirements

- **Performance:** 60 FPS VR rendering on mid-tier devices
- **Reliability:** 99.9% uptime SLA with auto-scaling
- **Compliance:** COPPA, GDPR, and Egyptian MOE standards
- **Accessibility:** WCAG 2.1 AA compliance
- **Scalability:** Horizontal scaling for 500% userbase growth

6 Class Diagram

This section defines the main classes, their attributes, and methods used in the Brightpath system.

Main Classes

- **Class: User**
  Attributes: userID, name, email, role (Student, Parent, etc.)
  Methods: login(), logout(), updateProfile()
- **Class: Student (extends User)**
  Attributes: age, badges[], points
  Methods: takeLesson(), viewLeaderboard(), receiveAIContent()
- **Class: Parent (extends User)**
  Attributes: children[]
  Methods: scheduleEvent(), trackProgress(), receiveAlerts()
- **Class: Teacher (extends User)**
  Attributes: subjects[], classes[]
  Methods: createVRLesson(), viewHeatmap()
- **Class: Lesson**
  Attributes: lessonID, title, type (VR/Gamified), difficulty
  Methods: startLesson(), submitAnswer()

- **Class: Badge**
  Attributes: badgeID, name, criteria
  Methods: awardBadge()
- **Class: AIEngine**
  Attributes: modelVersion
  Methods: personalizeLesson(), analyzePerformance()
- **Class: VRModule**
  Attributes: moduleID, subject, latency
  Methods: startSimulation(), joinCollabProject()

System Structure Overview

User Hierarchy:

- **User (base class):** Shared properties for all roles — student, parent, teacher, admin.
- **Student:** Can access lessons, earn rewards, and use AI-driven content.
- **Parent:** Linked to children; can view reports and schedule events.
- **Teacher:** Can create VR content and view class analytics.
- **Admin:** Can monitor the system and manage user roles.

Educational Content:

- **Lesson:** Basic learning content (title, subject, etc.).
- **VRLesson:** Extends Lesson; adds interactive, 3D virtual elements.

AI and Progress:

- **AILearningPath:** Generates adaptive lessons based on student performance.
- **Progress:** Tracks scores, streaks, and updates.

Gamification:

- Tracks points, badges, and leaderboard ranks for each student.

Associations:

- Students are linked to lessons, progress, gamification, and AI paths.

- Parents are connected to their children.
- Teachers are associated with the VR content they create.

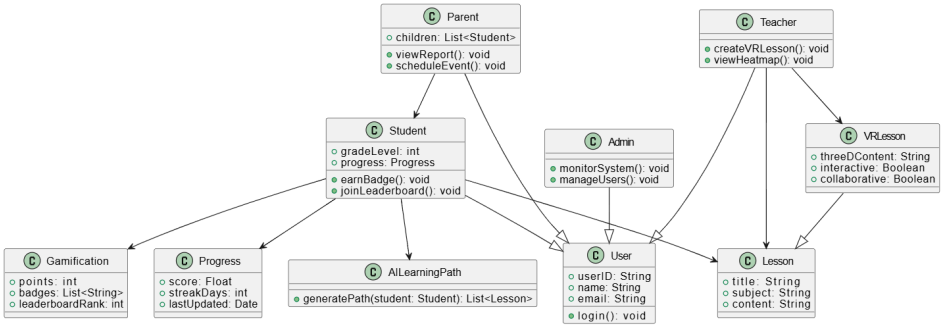


Figure 1: UML Class Diagram for Brightpath System

7 Use Case Diagram

Actors

- Student (6–11, 12–15)
- Parent
- Teacher
- Administrator
- System Admin

Use Cases

- View gamified lessons
- Earn points/badges
- View leaderboard
- Receive AI-personalized content
- Use VR modules
- Schedule learning events (Parent)

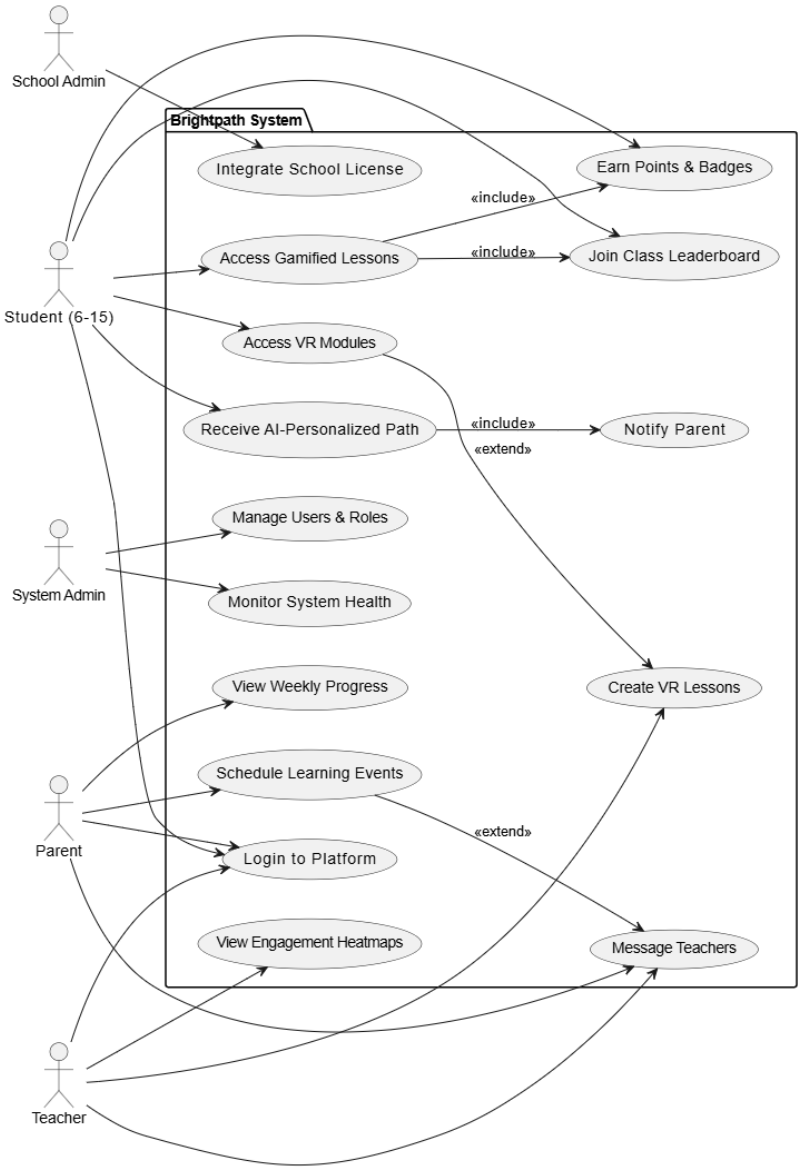
- Track student progress (Parent)
- Create VR lessons (Teacher)
- View engagement heatmaps (Teacher)
- Monitor system (Admin)
- Manage users (SysAdmin)

Actor to Use Cases Mapping

Actor	Use Cases
Student (6–15)	Login to Platform (UC0), Access Gamified Lessons (UC1), Earn Points & Badges (UC2), Join Class Leaderboard (UC3), Access VR Modules (UC4), Receive AI-Personalized Path (UC5)
Parent	Login to Platform (UC0), View Weekly Progress (UC6), Schedule Learning Events (UC7), Message Teachers (UC8)
Teacher	Login to Platform (UC0), Create VR Lessons (UC9), View Engagement Heatmaps (UC10), Message Teachers (UC8)
School Admin	Integrate School License (UC13)
System Admin	Monitor System Health (UC11), Manage Users & Roles (UC12)

Relationships Table

Primary Use Case	**Related Use Case**	**Relation Type**	Purpose
Access Gamified Lessons (UC1)	**Earn Points & Badges (UC2)**	`<<include>>`	Every gamified lesson awards points and badges
Access Gamified Lessons (UC1)	**Join Class Leaderboard (UC3)**	`<<include>>`	Lessons contribute to leaderboard scores
Access VR Modules (UC4)	**Create VR Lessons (UC9)**	`<<extend>>`	Optionally includes teacher-created VR content
Receive AI-Personalized Path (UC5)	**Notify Parent (UC14)**	`<<include>>`	Notifies parents based on student performance
Schedule Learning Events (UC7)	**Message Teachers (UC8)**	`<<extend>>`	Optionally includes messaging during event setup


School Admin
Brightpath System
Integrate School License
Earn Points & Badges
«include»
Access Gamified Lessons
«include»
Join Class Leaderboard
Student (6-15)
Access VR Modules
Receive AI-Personalized Path
«include»
Notify Parent
«extend»
Manage Users & Roles
System Admin
Monitor System Health
View Weekly Progress
Create VR Lessons
Schedule Learning Events
«extend»
Parent
Login to Platform
View Engagement Heatmaps
Message Teachers
Teacher

8 Activity Diagram

This diagram shows the step-by-step flow a student goes through when using Brightpath:

Actors

- Student (6–11, 12–15)
- Parent
- Teacher
- Administrator
- System Admin

Flow Breakdown

1. Login to Brightpath: Student starts by signing in.
2. Check Age:
    - If 6–11 years old: Student is guided through gamified lessons, earns points/badges, and joins the leaderboard.
    - If 12–15 years old: Student receives an AI-personalized learning path, completes those lessons, and still earns gamification rewards.
3. Access VR Modules: All students may access immersive VR content.
4. Check for Teacher-Created Content: If available, the student engages with custom VR lessons.
5. Log Progress: The student's activity is saved.
6. Send Weekly Report to Parent: Automated system sends a summary to the parent.
7. Update Analytics: Data is used to monitor engagement and inform teachers/AI models.

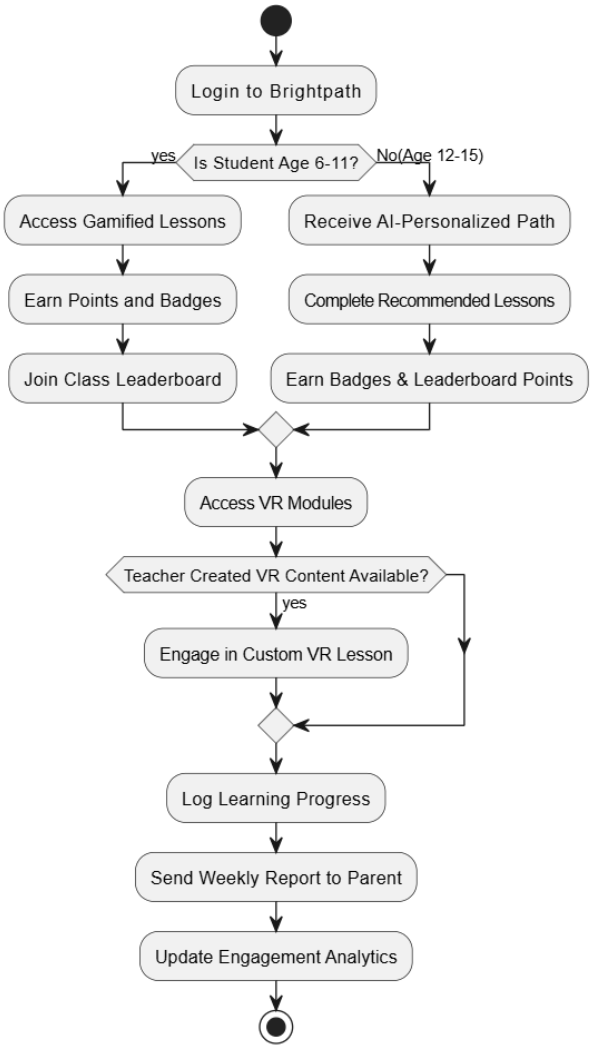


Figure 3: UML activity diagram for Brightpath System

9 System Architecture

9.1 User Layer

- Mobile Apps (Flutter)

- Web Portal (React.js)
- VR Interface (Three.js/WebXR)

9.2 Application Layer

- Django Microservices
- AI Recommendation Engine
- Gamification Service (Redis)
- Content Delivery Network (BunnyCDN)

9.3 Data Layer

- PostgreSQL: User profiles, courses
- MongoDB: Game logs, analytics
- Amazon S3: Media storage

10 Glossary

- **AI:** Artificial Intelligence, technology for machine learning and adaptive behaviors.
- **COPPA:** Children's Online Privacy Protection Act, US regulation protecting privacy of children under 13.
- **GDPR:** General Data Protection Regulation, EU privacy law.
- **Kubernetes:** Open-source platform for automating deployment, scaling, and management of containerized applications.
- **React.js:** JavaScript library for building user interfaces, mainly web.
- **Redis:** In-memory data structure store, used as a database, cache, and message broker.
- **TensorFlow Lite:** Lightweight machine learning library for mobile and embedded devices.
- **VR:** Virtual Reality, technology to create immersive simulated environments.
- **WCAG:** Web Content Accessibility Guidelines, international standards for making web content accessible.

11 Conclusion

Brightpath addresses national education challenges through gamification, AI adaptivity, and family engagement. The MVP will focus on math gamification and parent dashboards, followed by VR integration and school licensing. Development will follow Agile methodology with bi-weekly sprints.

Appendix

References

[1] ISO/IEC/IEEE 29148:2018, *Systems and software engineering — Life cycle processes — Requirements engineering*, ISO, 2018.

[2] FTC's Children's Online Privacy Protection Rule, https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa