# Group1 Members:

1.Joseph MUTANGANA 29061

2. Isaro Rubagumya Roxane 28762

3. Ishimwe Billy Joshua 28455

4. Rebeka patience 26277

5. Nsengiyumva George 28491

# Software Engineering Assignment

Group Project – SmartLearn Kids: An Interactive Learning App for Primary and Nursery Students

## Q1. Problem Statement

In Rwanda, many young learners in nursery and primary school lack access to engaging and structured digital learning resources that are aligned with their educational levels. Existing e-learning systems mainly rely on teacher uploads or text-based lessons, which are not suitable for very young learners who need interactive, visual, and guided learning experiences. To address this issue, the project proposes 'SmartLearn Kids' — a web and mobile-based interactive learning platform designed for nursery and primary students. It will provide categorized video lessons, animated storytelling, and simple exercises for each level (Nursery, P1 to P6). The developers will prepare and organize these lessons with age-appropriate visuals and narration. The goal is to help children learn languages, mathematics, social studies, and other courses of their level through play, storytelling, and interaction.

## Q2. Proposed Product Specification

**Functional Requirements:**

• The system categorizes learning content by level (Nursery, P1–P6).  
• Students can choose their grade and subject (Kinyarwanda, English, Math, Social Studies).  
• Interactive videos and storytelling lessons guide learners through each topic.  
• Exercises and quizzes follow videos to test understanding.  
• The system provides instant feedback after exercises.  
• Developers can add and organize new lessons and media.  
• Parents can track progress and time spent learning.  
• The app supports both web and mobile versions for flexibility.

**Non-Functional Requirements:**

• The interface must be colorful, animated, and simple for children to navigate.  
• Must support smooth playback of videos and animations.  
• Must provide offline access for pre-downloaded lessons.  
• The system should support both Kinyarwanda and English.  
• Security and privacy of student data must be ensured.

**Technical Requirements:**

• Backend: Python (Django or Flask)  
• Frontend: ReactJS or .NET Framework for web; Flutter for mobile app.  
• Database: PostgreSQL or Firebase.  
• Hosting: Cloud (AWS, Azure, or Firebase Hosting).  
• Version Control: GitHub with private repository for team collaboration.

## Q3. Preliminary Requirements Analysis

During the analysis, some challenges and dependencies were identified. Creating multimedia content will require graphic and audio design skills, and maintaining performance with video playback is a technical concern. Internet limitations in some areas suggest including an offline mode.  
  
**MoSCoW Prioritization:**  
Must Have – Grade and subject categorization, video lessons, exercises, and progress tracking.  
Should Have – Offline access, bilingual content, parental dashboard.  
Could Have – Gamification (stars, levels), AI voice tutor.  
Won’t Have (for now) – Real-time live classes or group chats.

## Q4. Key Deliverables

• Requirements Specification Document  
• System Architecture and Design Document  
• Interactive Lesson Prototype and Storyboard  
• User Interface Mockups  
• User Guide and Installation Manual  
• Test Report and Validation Results  
• Final Project Report and Deployment Package

## QQ5. Software Development Process

The Agile Software Process Model will be used for development. Each sprint will deliver specific functionalities, such as video lesson module, progress tracking, or quiz system. This iterative approach allows flexibility for feedback from teachers, parents, and students. Scrum suits educational app projects because content and interface designs often evolve after user testing.

## Q6. Overall Plan

**Main Tasks and Milestones:**  
• Requirements Elicitation – 1 week (Deliverable: Requirements Document)  
• Content Design – 2 weeks (Deliverable: Educational Videos, Storyboards)  
• System Design – 2 weeks (Deliverable: Architecture & UI Mockups)  
• Development – 4 weeks (Deliverable: Functional Application Prototype)  
• Testing and Validation – 2 weeks (Deliverable: Test Report)  
• Deployment & User Training – 1 week (Deliverable: Deployed System, User Manual)  
  
**Abstraction and Modularity:**  
The project will follow a modular architecture:  
• UI Layer – Displays categorized content and user interactions.  
• Content Engine – Manages videos, quizzes, and level-based organization.  
• Learning Tracker – Records student progress and results.  
• Database Layer – Stores user accounts and learning data.

## Q7. Version/Revision Control Strategy

GitHub will be the central platform for version control. Each developer will be responsible for one component (frontend, backend, content engine, etc.) and work in separate branches. Merges will occur after pull request reviews by the team leader. Tags will mark sprint completions and stable releases.

## Q8. Quality Measures

**Quality assurance will be achieved through:**  
• Usability testing with children and teachers.  
• Functional testing of video playback and exercises.  
• Accessibility testing for intuitive navigation.  
• Performance testing to ensure smooth video streaming.  
• Security testing to protect student data and access controls.

## Q9. Testing/Validation Strategy

Testing will combine black-box testing, usability testing, and pilot trials. Early prototypes will be demonstrated to teachers and a small group of learners to observe interaction flow. Unit and integration tests will validate system components. Automated tests will ensure lesson loading and quiz scoring function correctly across devices.

## Q10. Scope for Further Development

Future versions of SmartLearn Kids could introduce gamified learning badges, AI-driven personalized recommendations, more languages, and integration with national digital libraries. A teacher dashboard for uploading custom lessons could also be added to expand content diversity.

## References

• Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.  
• Rwanda Education Board (REB) Curriculum Documents, 2023.

• Sundar\_nycusa. (n.d.). MOSCOW Prioritization Tech. Scribd. Retrieved October 8, 2025, from [https://www.scribd.com/document/909469860/MOSCOW-Prioritization-Tech](https://www.scribd.com/document/909469860/MOSCOW-Prioritization-Tech?utm_source=chatgpt.com)

•GASHEMA, G. (2025). Lecture 3: Software Process and Software Process Models. Adventist University of Central Africa.

•Devineni, S. K. (2020). Version Control Systems (VCS): The Pillars of Modern Software Development – Analyzing the Past, Present, and Anticipating Future Trends. International Journal of Science and Research (IJSR), 9(12), 1816–1829.  
Retrieved from <https://www.researchgate.net/publication/378490782_Version_Control_Systems_VCS_the_Pillars_of_Modern_Software_Development_Analyzing_the_Past_Present_and_Anticipating_Future_Trends>

•UNIT‑III. (n.d.). UNIT III: Design Concepts and Abstraction [Document]. Scribd. Retrieved October 8, 2025, from [https://www.scribd.com/document/741583673/UNIT-III](https://www.scribd.com/document/741583673/UNIT-III?utm_source=chatgpt.com)