



Software Requirements Specification (SRS)

Project Title: *Game On*: Gamified Reading Learning for Grade 2 Students

Capstone Group: Bicomong, Catherine Jan G.; Espinosa, Paola Nicole D.; Portugal, Jelly Mae A.

Institution: Laguna State Polytechnic University Santa Cruz, Laguna

1. INTRODUCTION AND PURPOSE

1.1 Background

Game On is a mobile application developed to address the reading proficiency challenges faced by Grade 2 students at Tubigan Elementary School, Biñan, Laguna. With gamified learning gaining popularity as a pedagogical approach, this system seeks to make reading practice enjoyable and effective by incorporating rewards, challenges, and interactive reading tasks.

1.2 Scope

The application allows students to engage in reading tasks, while teachers can monitor progress, add reading materials, and generate insights. It supports student, teacher (admin), and parent participation. The platform is available as an Android application and includes machine learning-enhanced features for score prediction.

1.3 Objectives

- Develop a gamified application for reading skills enhancement.
- Implement and test key features: pronunciation, reading tasks, and short stories.
- Evaluate the impact of the system via pre- and post-test assessments.



- Assess user satisfaction and effectiveness using the Technology Acceptance Model (TAM).

2. OVERALL DESCRIPTION

2.1 User Classes and Characteristics

- **Students:** Engage with tasks, stories, and quizzes.
- **Teachers/Admins:** Manage content, track student progress, and view analytics.
- **Parents:** Provide support and monitor child's learning through observation.

2.2 Operating Environment

- **Platform:** Android (React Native)
- **Database:** Firebase
- **Libraries:** TensorFlow.js (ML), ChartKit (visualization)

2.3 Design and Implementation Constraints

- Internet connection required.
- Designed for Android smartphones and tablets.
- Mobile-first architecture.

2.4 User Needs

- Simple, engaging interface for young learners.
- Real-time progress feedback and motivation.
- Secure teacher/admin control over content.

3. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

3.1 Functional Requirements



- **Registration/Login:** Secure user creation and authentication.
- **Reading Tasks:** Interactive short story and word tasks.
- **Gamification:** Points, badges, leaderboards.
- **Admin Panel:** Story upload, student progress monitoring.
- **Score Prediction:** ML model forecasts future performance.
- **Visualization:** Graphs showing actual vs predicted scores.

3.2 Non-Functional Requirements

- **Performance:** <2s response time for user actions.
- **Usability:** Designed for non-literate users with icons and audio cues.
- **Security:** Authentication via Firebase; role-based access.
- **Scalability:** Modular design for future grade levels.
- **Compatibility:** Android OS, responsive UI.

4. SYSTEM FEATURES AND INTERFACES

4.1 User Interface

- Colorful, child-friendly screens.
- Audio-guided buttons.
- Score feedback and star rewards.

4.2 System Interfaces

- **Firebase Firestore:** Data storage and retrieval.
- **TensorFlow.js:** ML model for score forecasting.
- **ChartKit:** Graph generation.

5. ASSUMPTIONS AND CONSTRAINTS



- Target users have Android devices.
- Users have stable internet connection.
- Teachers are trained to manage content.
- ML model requires minimum 3 days of user data to predict scores.

6. USE CASE DIAGRAMS OR DESCRIPTIONS

Use Case 1: Student Reading Task

Actor: Student

Flow:

- Log in
- Choose and complete a reading task
- Earn points and badges
- View performance graph showing past and predicted scores

Use Case 2: Admin Story Management

Actor: Teacher/Admin

Flow:

- Log in
- Upload new short story
- Monitor class performance
- View predictions and generate feedback

Use Case 3: Score Prediction

Actor: System

Flow:



- Gather past scores
- Train regression model (TensorFlow.js)
- Display predicted scores via graph

7. TESTING TOOL DOCUMENTATION

7.1 Unit Testing

Tool: Jest (JavaScript unit testing framework)

7.2 Performance Benchmarking

Tool: Lighthouse (for app responsiveness and performance)

7.3 Compatibility Testing

Tool: BrowserStack (test across Android versions)

7.4 Rationale for Tool Selection

Tools are compatible with React Native development and ensure robust, reliable testing of performance, compatibility, and unit logic.

8. CONCLUSION

Game On brings an innovative, gamified approach to primary reading instruction. The mobile platform is tailored to young learners, delivering structured reading tasks and motivational tools. With integrated ML for personalized insights and real-time visualization of performance, the system enhances both learning and teaching experiences. The SRS ensures clear alignment across stakeholders, guiding the project's continued success.