Software Requirements Specification

for

LexiCross

Version 1.0 approved

Prepared by 2019510004-MUSAB ALBARGHOUTH  
2020510110-SHAKHOBIDDIN URINOV  
2020510121-GÜLNAZ HİLMİOĞLU  
2020510158-ALİ ÖZGÜR İNEP  
2021510082-HALİL İBRAHİM İRAN

<MAHGONIX>

23.03.2025

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 2

1.5 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. System Features 5

3.1 System Feature 1 5

3.2 System Feature 2 (and so on) 6

4. External Interface Requirements 9

4.1 User Interfaces 9

4.2 Hardware Interfaces 9

4.3 Software Interfaces 10

4.4 Communications Interfaces 10

5. Other Nonfunctional Requirements 11

5.1 Performance Requirements 11

5.2 Safety Requirements 11

5.3 Security Requirements 13

5.4 Software Quality Attributes 14

6. Other Requirements 15

Appendix A: Glossary 16

Appendix B: Analysis Models 19

Appendix C: Issues List 19

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Purpose

## The purpose of this SRS document is to define and document all software requirements for the LexiCross project. That is Crossword Puzzle Game, a mobile application offering various crossword challenges. This report has been prepared for submission to the relevant authorities at 'Worsixsoft' software company.

## Document Conventions

In writing the LexiCross Software Requirements Specification (SRS), the following conventions were observed:

### *Font Styles: Different font styles were employed for clarity and emphasis:*

### *Bold Font: Highlights headings, section titles, and key terms.*

### *Italic Font: Used for citations, references, and emphasis.*

### *Monospace Font: Reserved for code snippets and technical specifications.*

Organization and Format: Sections, subsections, and appendices are organized in a hierarchical and logical order to increase readability. These conventions ensure readability and consistency, aiding all stakeholders involved in the LexiCross project.

## Intended Audience and Reading Suggestions

This document was prepared with the following audiences in mind:

### Developers: Team members who will work on system architecture, API integrations, and modular design details.

### Test Engineers: To be used as a reference for requirements validation for functional and performance testing.

### Project Managers and Stakeholders: Executives and investors who want to learn about the overall goals, scope, and critical requirements of the project.

### Technical Writers: To serve as guides in the preparation of user guides and other documentation materials.

### Reading Recommendations: Before you start reading the document, reviewing the "Overall Description" section will help you understand the overall structure and operation of the project. After that, it is recommended that you work on the sections that include detailed requirements and system specifications.

## Project Scope

LexiCross is an interactive word puzzle game that will work on mobile platforms (Android and iOS). The project scope includes the following main features:

### Puzzle Game Experience: Users will be presented with crossword puzzles prepared in various themes, difficulty levels and sizes.

### Daily Challenge: A special puzzle module that is renewed every day and where users can compete with each other.

### User Progress: Users will be able to track puzzle solving time, number of errors and progress, and continue puzzles from where they left off.

### User Verification: Secure login and data synchronization will be provided with Google account integration and a standard email-based registration system.

### Data Analytics: The system is aimed to be continuously improved by monitoring and analyzing user behavior and puzzle performance.

## References

**[1]** **GfG. (2023, September 20). Software Requirement Specification (SRS) Format. GeeksforGeeks.** [**https://www.geeksforgeeks.org/software-requirement-specification-srs-format/**](https://www.geeksforgeeks.org/software-requirement-specification-srs-format/)

**[2]** **Alison Weber. (2016, September 11). Software Engineering: Chapter 3 SRS explained [Video]. YouTube.** [**https://www.youtube.com/watch?v=FvxO8SLoua8**](https://www.youtube.com/watch?v=FvxO8SLoua8)

**[3]** **Department of English, Linguistics, and Writing Studies at SLCC. (2020, September 10). Software Requirements Specification (SRS). Pressbooks.**

**https://slcc.pressbooks.pub/technicalwritingatslcc/chapter/software-requirements-specification-srs/**

# Overall Description

This section provides the overall framework for the LexiCross application. It comprehensively defines where the application is located, what features it offers, who will use it, in what environment it will operate, design constraints, user documentation, and assumptions and dependencies. This information creates a holistic understanding of the project and serves as a common reference source for both technical and business stakeholders.

## Product Perspective

LexiCross is a new application that will be developed independently and will run on mobile platforms (Android and iOS). The application is positioned in an integrated structure with the following elements:

Technological Ecosystem: User security and data synchronization will be optimized by integrating with third-party services such as Google authentication and payment systems.

Context and Competition: The application is positioned as a modern mobile version of traditional word puzzle games. In this context, it is aimed to offer flexibility and timeliness that will meet the expectations of both classic puzzle lovers and the younger generation.

System Components: The application includes a number of subcomponents such as user interface, data management, game engine, advertising and subscription modules. These components will be integrated together in a way that directly affects user experience and game performance.

## Product Features

The key features LexiCross offers aim to provide users with a comprehensive puzzle experience:

**Crossword Puzzle Games:**

Crosswords in different themes, sizes, and difficulty levels.

An interactive game engine where users can choose puzzles and start solving them.

* ***Daily Challenge:*** A special puzzle that is refreshed daily allows users to compete against each other on a daily basis. Completed daily puzzles include leaderboards where users can compare their performance.
* ***User Progress and Profile Management:*** A record is kept of users’ puzzle-solving times, error counts, and puzzles completed. Thanks to Google account integration, user data is synchronized securely and quickly.
* ***Subscription and Premium Content:*** While basic features are offered in the free version, premium membership provides additional benefits such as special themes, background music, ad-free experience and unlimited hints.
* ***Data Analytics:*** Data is collected about user behavior and puzzle performances, and analyses are performed for continuous improvement of the application.

## User Classes and Characteristics

LexiCross is designed to appeal to a broad range of users. The different needs and usage habits of each user group are considered:

* **Teenagers (Ages 13-19):**Young users looking for short-term, fun, and mobile-friendly gaming experiences.Colorful and dynamic interfaces and fast interactions are preferred.
* **Young Adults (Ages 20-35):**Tech-savvy users looking for mental exercise and competitive gaming.Features such as social media integration and leaderboards are prominent.
* **Adults (36-55 Years):**Users who are interested in traditional puzzle games and have a more planned and strategic approach.Ease of use and detailed progress tracking are important for this user group.
* **Seniors (55+ Years):**Users who prefer simple, understandable interfaces and large fonts.

Ease of use, accessibility and simple design are priorities.

## Operating Environment

The application will be a mobile-based application that can be accessed from any mobile or tablets with an internet connection. The system will be compatible with all major operating systems such as Android and IOS. The system will be hosted on a cloud server with appropriate security measures in place.

## Design and Implementation Constraints

The System's design and implementation are constrained by the following:

* Modern software development techniques must be used in the system's design to ensure high-quality and maintainability.
* The system needs to be scalable in order to deal with a lot of data and users.
* The interface needs to be simple to use and open to all users.
* To ensure the accessibility, privacy, and reliability of data, the system must abide by industry-standard safety measures.

## User Documentation

* Every user of the system will have access to an online help center and user manual.
* Detailed instructions for using the functions and features of the system must be included in the user manual.
* There will be troubleshooting guides and commonly asked questions in the online help center.

## Assumptions and Dependencies

**Assumptions:**

* The system's hardware and software infrastructure will be dependable and accessible throughout its development and operation.
* Every user of the system would receive instruction in efficient system use and have a basic understanding of mobile technology.
* The system will be built with scalability and easy maintenance in hand.
* The system will be built with security as a priority to avoid unauthorized access or data breaches.
* The data saved in the system will be accurate and up to date.

**Dependencies:**

* The availability of funds and resources to support the development process is a necessary condition for the successful completion of the system.
* Both developers' and end users' access to reliable internet connections will be necessary for the system to function.
* The system may require to integrate with other platforms or systems.
* It may depend on the availability of third-party hardware or software, like servers or databases.
* Regulations pertaining to safety or data privacy may need to be complied with, which could have an impact on the system's development and functionality.

# System Features

This section details the system features that provide the core functionality of the LexiCross application. Each feature is described with user interactions (stimulus/response), priority level, and related functional requirements.

## System Feature 1

### Crossword Puzzle Gameplay

Description and Priority

This feature, as the main function of the application, allows users to solve selected puzzles. The user chooses from puzzles presented in different categories and difficulty levels, starts the solution process and makes progress. This feature is a high priority as it is the core function of the application.

### Stimulus/Response Sequences

* **Start:** The user opens the application, is directed to the "Choose Puzzle" screen in the main menu.
* **Selection:** The user selects a puzzle through filter options such as category, difficulty, and theme.
* **Game Start**: The selected puzzle is loaded by the system, the game timer is started, and the user is presented with a blank template of the puzzle.
* **Progress and Feedback**: The system checks for errors at each move as the user solves the puzzle, provides instant feedback on correct or incorrect entries, and records completed sections.

### Functional Requirements

**REQ-1:** The system shall present a list of available puzzles categorized by themes, sizes, and difficulty levels.

**REQ-2:** The system shall record, in real time, the inputs made on the selected puzzle, the number of errors, and the time taken to complete it.

**REQ-3:** The user shall be able to save the progress of a solved puzzle and later continue from where they left off.

**REQ-4:** During puzzle solving, the system shall automatically provide error notifications and hint suggestions.

## System Feature 2 (and so on)

**3.2 Daily Challenge**

**3.2.1 Description and Priority:**  
 The Daily Challenge feature provides a special puzzle that is refreshed daily for all users. By solving this puzzle, users can compete with others and view their rankings on a leaderboard. This feature increases user engagement and encourages daily returns to the application.

**3.2.2 Stimulus/Response Sequences:**

* **Daily Display:** When the application is launched or when the Daily Challenge tab is accessed, the system automatically loads the puzzle of the day.
* **Solution Process:** While the user is solving the puzzle, the system activates the timer and error tracking.
* **Result and Comparison:** After the puzzle is completed, the system calculates the user's performance and automatically adds them to the daily leaderboard.

**3.2.3 Functional Requirements:**

* **REQ-5:** The system shall refresh the puzzle of the day every 24 hours and present it to the user.
* **REQ-6:** The completion time, error count, and user performance of the daily puzzle shall be collectively listed in a leaderboard.
* **REQ-7:** The user shall be able to view the results and ranking of the daily puzzle in the in-app statistics section.

## User Authentication and Profile Management

### **Description and Priority:**

This feature enables users to securely log in, create an account, and manage their profile information. By offering easy and fast verification through Google integration, data synchronization and user progress are ensured. This feature is critical for maintaining data integrity and providing a personalized experience within the application.

### Stimulus/Response Sequences:

* **Login Selection:** When the user opens the application, they choose between “Sign in with Google” or “Create Your Own Account” on the login screen.
* **Verification:** The system verifies the user credentials based on the selected authentication method and loads the user profile upon a successful login.
* **Profile Management:** The user can update personal information, view progress, and track achievements in the profile settings screen.

### Functional Requirements:

* **REQ-8:** The system shall ensure secure login for users through Google OAuth integration.
* **REQ-9:** The user shall be able to view and edit their personal information and progress data via the in-app profile management panel.
* **REQ-10:** An automatic backup mechanism shall be established to enable data synchronization across different devices for users.

## Subscription and Premium Content

### **Description and Priority:**

This feature includes the premium content and subscription system to support the application’s revenue model. Premium users benefit from an ad-free experience, exclusive themes, background music, and unlimited hints. The core game experience remains equal for all users; premium features only enhance visual and auditory comfort.

### Stimulus/Response Sequences:

* **Subscription Initiation:** The user navigates to the in-app subscription section to view the premium membership options and selects the desired plan.
* **Payment Process:** For the selected plan, the system initiates the transaction via integrated payment APIs; once the payment is approved, the premium content becomes active immediately.
* **Content Access:** Premium users gain access to exclusive themes, music, and unlimited hints within the application, and advertisement displays are disabled.

### Functional Requirements:

* **REQ-11:** The system shall differentiate between free and premium users by restricting access to certain features for non-premium users.
* **REQ-12:** During payment transactions, the security of user data shall be ensured, and premium content must be activated immediately after the transaction is completed.
* **REQ-13:** The selection of premium content and subscription status shall be updated in the user profile and accessible by all related modules.

## Analytics and Data Collection

### Description and Priority:

This feature monitors user behavior and puzzle performance, providing data analytics aimed at the continuous improvement of the application. The collected data will be analyzed to make enhancements in game design, optimize difficulty levels, and increase user experience. This feature is strategically important for the long-term success of the application.

### Stimulus/Response Sequences:

* **Data Collection:** While the user is solving a puzzle, the system automatically collects metrics such as solution time, error count, number of hints used, and interaction durations.
* **Data Analysis:** The collected data is periodically analyzed in the back-end system; the results are presented to stakeholders in the form of reports and graphs.
* **Feedback:** The analysis results are used as feedback for in-app improvements and enhancements to the user experience in subsequent updates.

### Functional Requirements:

* **REQ-14:** The system shall automatically record key metrics related to user interactions, puzzle results, and application performance.
* **REQ-15:** The collected data shall be transferred to the back-end database and reported at defined intervals for analysis.
* **REQ-16:** The analysis results shall be presented as accessible reports through the application management panel and supported by data visualization tools where necessary.

# External Interface Requirements

This section defines all the external interactions between LexiCross and its surrounding systems, hardware, and communication channels. It specifies how the system will interface with users, devices, other software components, and networks.

## User Interfaces

The User Interface (UI) is the primary touchpoint for end-users, and it must be designed to provide an intuitive, responsive, and aesthetically consistent experience. Key aspects include:

* **Design and Layout:**
  + The UI should follow modern mobile design principles with clean layouts, clear typography, and easy navigation.
  + It must be responsive to different screen sizes and orientations on both Android and iOS devices.
* **Screen Components:**
  + **Login and Registration Screens:** These include options for Google sign-in and manual account creation.
  + **Puzzle Selection and Gameplay Screens:** Interactive screens for choosing puzzles, displaying puzzle grids, tracking progress, and providing real-time feedback.
  + **Daily Challenge and Leaderboard Screens:** Sections that automatically load the daily puzzle and display rankings in a clear, competitive layout.
  + **Profile and Settings Screens:** Areas where users can manage their account details, view progress, and adjust preferences (e.g., music on/off, notifications).
* **Usability Requirements:** The interface should provide clear prompts, error messages, and confirmations. Accessibility features such as high-contrast themes, scalable fonts, and voice-over support must be considered, especially given the broad age range of users.

## Hardware Interfaces

LexiCross is designed to operate on modern mobile devices and must integrate seamlessly with the hardware capabilities of these devices. Key requirements include:

* **Device Compatibility:** The system must be compatible with a wide range of smartphones and tablets running Android and iOS. It should automatically adjust to various screen resolutions and aspect ratios.
* **Touch and Sensor Inputs:** The application must fully support touch-based interactions including gestures like tap, swipe, and pinch-to-zoom.
* **Performance Considerations:** Efficient resource management is necessary to ensure smooth performance even on devices with lower processing power or limited memory. Battery usage should be optimized, especially during prolonged gameplay sessions.

## Software Interfaces

LexiCross must interact with several external software systems to deliver a seamless and feature-rich experience. These interfaces include:

* **Authentication Services:** Integration with Google OAuth for secure user authentication, enabling users to log in quickly using their Google accounts.
* **Payment and Subscription APIs:** The system will interact with third-party payment processing services to handle subscription transactions securely and reliably.
* **Backend and Database Systems:** Communication with cloud-based databases for storing user profiles, puzzle data, progress logs, and analytics. The client application will use RESTful APIs to retrieve and update data, ensuring that data is synchronized across multiple devices.
* **Advertisement and Analytics:** Integration with advertising networks to display ads to free users while ensuring ad content is dynamically updated. Utilization of analytics APIs to collect and report user interaction data, which will be processed for further insights into game performance.

## Communications Interfaces

The communications interfaces define how LexiCross exchanges data with external systems over networks. Essential aspects include:

* **Network Protocols:**
  + The system shall use secure HTTPS protocols for all client-server communications to protect user data during transmission.
* **API Communication:**
  + RESTful web services will be employed to allow the mobile application to communicate with backend servers, retrieve updates, submit user progress, and handle subscription validations.
  + Data exchanged between the client and server will be formatted in common data interchange formats such as JSON or XML.
* **Push Notifications and Real-Time Updates:**
  + The system should support push notifications for real-time updates, such as daily challenge alerts or subscription status changes.
  + Notifications should be delivered reliably using standard messaging protocols (e.g., Firebase Cloud Messaging for Android and Apple Push Notification Service for iOS).
* **Error Handling and Timeout Management:**
  + The communication layer must include robust error handling to manage network disruptions, latency issues, and timeouts.
  + Automatic retries and fallback mechanisms should be implemented to ensure continuous data flow even under suboptimal network conditions.

# Other Nonfunctional Requirements

This section defines constraints and quality attributes that ensure the software meets expectations in areas beyond core functionality. These requirements focus on performance, safety, security, and overall software quality.

## Performance Requirements

 **Response Time:**

The system shall load puzzles, transition between screens, and execute user interactions within a maximum of 3 seconds under normal network conditions.

 **Throughput and Scalability:**

* The application must support a high volume of concurrent users without significant performance degradation.
* The architecture should be scalable, both horizontally and vertically, to accommodate an increasing number of users and data loads.

 **Resource Utilization:**

* The application should be optimized to use minimal CPU and memory resources, ensuring smooth performance even on lower-end mobile devices.

 **Data Synchronization:**

* Real-time synchronization of user progress, daily challenges, and analytics data must occur seamlessly, with minimal delay or disruption.

## Safety Requirements

### ****Safety of User Data:****

**User data must be encrypted using the AES-256 encryption algorithm when stored in the database.**

### ****Authorization and Authentication Mechanisms:****

**Secure OAuth 2.0-based authorization mechanisms should be used for user logins.**

### ****Protection Against External Attacks:****

**The application should be equipped with protective measures such as security firewalls and intrusion detection systems to safeguard against**

* 1. **Firewall Security:**

Security firewalls should be added to the server infrastructure behind the application. These firewalls filter incoming requests and block unwanted traffic, providing protection against attacks.

* 1. **Protection Against Login Attacks:**

Measures should be taken to protect against user login attacks at entry points. For instance, additional security layers such as temporarily locking accounts after failed password attempts or implementing CAPTCHA verification can enhance security.

* 1. **SQL Injection Protection:**

To secure database queries within the application, precautions against SQL injection attacks must be taken. This can be achieved by using parameterized queries and securely processing input data.

* 1. **XSS (Cross-Site Scripting) and CSRF (Cross-Site Request Forgery) Protection:**

User inputs and outputs, data entry points, and session authentication mechanisms should be safeguarded against XSS and CSRF attacks. Adhering to security standards and employing proper coding and validation techniques is essential.

* 1. **Security Updates:**

Regularly check for security vulnerabilities in the libraries and components used by the application and apply updates. Additionally, keep the server operating system and other dependencies up to date.

* 1. **Logging and Monitoring:**

Detailed logging of all activities on both the application and server sides is crucial. This helps detect attempted attacks and improves response.

## Security Requirements

### Application Security and Data Integrity:

* The application should be regularly tested for security vulnerabilities, and any identified vulnerabilities should be promptly addressed.
* Security updates and patches should be applied swiftly to prevent potential attacks.

Protection of User Data:

* User data must be securely stored and processed. This requires implementing database and server security measures.
* User data should only be accessible to authorized users.

### Authorization and Authentication:

* The application should employ a robust authorization and authentication mechanism to verify user identities.
* Additional security measures, such as one-time passwords or second-factor authentication, can be implemented for user logins.

### Data Encryption:

* All user data and communication should be encrypted using secure communication channels. For example, HTTPS protocol should be used.
* User passwords and sensitive information should be securely stored and encrypted using strong encryption algorithms like AES-256.

## Software Quality Attributes

Software quality attributes are crucial factors that determine the overall performance, reliability, and usability of the application. In this section, let's specify some specific features and test criteria that need to be considered to enhance the quality of the application:

* **Usability:**
  + The application should have a user-friendly interface design. To achieve this, the user interface design should be shaped based on data obtained from user tests and feedback.
  + User feedback should be regularly analyzed to continuously improve the user experience, and interface enhancements should be made based on this feedback.
* **Performance Optimization:**
  + The speed and response time of the application should be continuously evaluated and improved with specific benchmarks. For example, each page should load within a certain time frame.
  + Determining how the application should behave under high traffic scenarios is crucial. This may involve mechanisms such as automatic scaling.
* **Security:**
  + AES-256 encryption algorithm should be utilized for securing user data within the application. Additionally, secure OAuth 2.0-based authentication mechanisms should be implemented to safeguard user logins.
  + The application should be equipped with protective measures such as security firewalls and intrusion detection systems to protect against external attacks.
* **Portability:**
  + Ensuring seamless operation of the application across different platforms (iOS, Android, etc.) is essential. This includes selecting appropriate cross-platform development frameworks and testing compatibility with various screen sizes and resolutions.
* **Feedback Mechanism:**
  + Forms or surveys should be created within the application to allow users to easily provide feedback. This feedback should be regularly analyzed and utilized to enhance the user experience.

# Other Requirements

This section captures additional requirements that do not fall under the previously defined categories. These may include database specifications, legal and regulatory constraints, and internationalization requirements, among others.

* Database Requirements:  
  The system shall employ a scalable and secure database for storing user profiles, puzzle data, progress logs, and analytics. Data access must be efficient to support real-time interactions and ensure synchronization across devices.
* Regulatory and Compliance Requirements:  
  LexiCross must adhere to all applicable data privacy and security regulations, such as GDPR. The application should implement measures to protect user data and ensure transparent data handling practices.
* Internationalization:  
  Although the initial release may target a specific language or region, the design should be flexible enough to support multiple languages and cultural contexts in future updates.
* Additional Design Constraints:  
  Other factors, such as device-specific limitations, third-party API constraints, and potential future integrations, shall be documented here and monitored throughout the project lifecycle.

Appendix A: Glossary

* **Appendix A: Glossary**

The Glossary provides clear definitions for key terms, acronyms, and abbreviations used throughout this document. This helps ensure consistent understanding among all stakeholders.

* **Premium User:**  
  A user who subscribes to premium features, granting them access to exclusive content such as special themes, background music, ad-free experience, and unlimited hints.
* **Daily Challenge:**  
  A unique puzzle that is refreshed every 24 hours, allowing users to compete and compare their performance on a leaderboard.
* **OAuth:**  
  An open-standard authorization protocol used for secure user authentication (e.g., Google OAuth).
* **API (Application Programming Interface):**  
  A set of routines, protocols, and tools for building software and allowing different software components to interact.
* **GDPR (General Data Protection Regulation):**  
  A regulation in EU law on data protection and privacy for all individuals within the European Union

Appendix B: Analysis Models

A diagram of a crossword puzzle game

Description automatically generated

Figure 1. Class Diagram

A diagram of a software project

Description automatically generated with medium confidence

Figure 2. Sequence Diagram

A diagram of a company

Description automatically generatedFigure 3. Activity Diagram

A diagram of a crossword puzzle game

Description automatically generated

Figure 4. Use Case Diagram

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>

Appendix D: Meetings List

List of Meetings done to gather information. (Meeting No – Meeting Date – Stakeholders other than team members)