
Software Requirements Specification

for

LEXI

Version 1.0 approved

Prepared by Timesha Frank, Esther Tanyi, and Emmanuel Isidahomen

CSDP 490

February 3rd, 2020

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 Product Scope	1
1.5 References	1
2.	Overall Description
	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3.	External Interface Requirements
	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4.	System Features
	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5.	Other Nonfunctional Requirements
	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6.	Other Requirements
	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version
Emmanuel Isi	2/6/2020	Worked on purpose and purpose scope	1.1

1. Introduction

LEXI is a web application that will be built and designed, after a conducted research, to aid individuals with literacy(reading) disabilities with ages ranging from three to twenty-one. For this web application to be of help to the mentioned age group, Lexi web application will have four features, which will include Voice Catcher (text to speech), Typing Speaker (speech to text), Word Booster (vocabulary helper), and Alpha Shot (spelling game). Each feature will be made to carefully respond to the needs of potential users. (1) **Voice Catcher** will allow the user to talk to the web application and it will write down the spoken phrase/word. (2) **Typing Speaker** will allow the user to input any textual content that will then be read out to them. (3) **Word Booster**, also known as the vocabulary helper/builder, will be the feature that will break words down by pronunciation- each syllabus of the word will be highlighted as it gives audio, gives a definition, and example of the word in a sentence. (4) **Alpha Shot game** is a fun way to test the knowledge of the learned words. The test will incorporate words as hot air balloons floating across the screen with one word spelled in four ways (three incorrect and one correct), the user will then shoot the incorrect spelling of the words each time it is seen. These features would enable users to identify words faster. Lexi will be built using HTML(hypertext Markup Language), CSS (Cascading Stylesheet), and JavaScript.

1.1 Purpose

This is a web application for students who face challenges seeking reading and writing tools to address their individual needs. Lexi will be designed to be a learning tool for students that may have a learning disability and may wish to keep their learning disability confidential.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

Lexi would be a learning tool for students that may have a learning disability and may wish to keep their learning disability confidential. Our web scope would have a Lexi student registration-portal

system where all our Lexi users can create their security pathway to access our web application page. Also, once registered to the Lexi application they can now utilize our implemented features that would aid them to improve their learning capabilities.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

All users of the LEXI website will need to create an account to access the features on the site. When creating an account, the user would create a username, a password, and an email address.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

For the Lexi web application, the users will be provided instruction on each feature page on how it is used in details and the purpose of the feature. Manuals are not necessary.

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

Lexi would be a web application that would be built with HTML, CSS, and Javascript. It would be able to store the users data on php that we help the users iterate using different data and information provided to from the users.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

- **System Feature 1**

- **System Requirements**

REQ-1: Lexi can be accessed using google chrome.

REQ-2: The webpage should occupy the entire screen.

REQ-3: The speech to the text page should request permission to use the user's microphone.

4.1.2 Functional Requirements

REQ-1: We will be providing the URL for the users to access our website.

REQ-2: Designing the website for the user to assess (user interface)

REQ-3: Creating authorization to policy to the users

REQ-4: Internet access needs to be available

REQ-5: Description of function and service(what they do, how they help)

REQ-6: Integrating speech with application

4.1 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- REQ-1: The Lexi web application will be available 99% of the time

REQ-2: 1% will be for updating the features. One week will be used for updates, one feature will be updated a day while other pages will be available for use.

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

5.6 Non-Funtional Requirements

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>