Software Requirements Specification

for

Mnemonic Device

Version 1.0 approved

Prepared by Mnemonic Device Team

Mnemonic Device Team

2-9-2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

The Product outlined in this document is menmonicstudy.com version 1.0. The scope that is covered in this document is the planning for the entire web application.

1.2 Document Conventions(optional)

<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.>

The intended audience for this document is our lovely CPS 298 professor, Jing Bai. The rest of the SRS contains our plan for implementing menmonicstudy.com and can be read in a normal numerical order as suggested. Filled in portions are highlighted and underlined.

1.4 Product Scope

Mnemonicstudy.com is a website where users can create and store mnemonic devices in a simple and easy to access place.

1.5 References(if it applies)

<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.>

This website is a new self-contained product. Originally it was a way for people to generate their own mnemonics for niche and personal information which they needed help memorizing. It would automatically generate a mnemonic device when a study concept would be entered. This was sparked by a comment received by one of the developers from a friend that they found mnemonic devices helpful, but were often unable to think of their own. And that there was not a good place to keep mnemonics other than their own hand-made notecards, or a cumbersome note on their phone.

2.2 Product Functions

Our product aims to allow users to create and login to their accounts, from there users will be able to create their own mnemonic flashcards. Users will be able to make their own mnemonics private or public/searchable on creation. Users will be able to also search for other users' mnemonic flashcards by searching up tags. Users will be able to save flashcards to a collection on their profiles which include their own created mnemonics, which will be clickable to a viewable study page where the flashcards will flip.

Future Directions: Users will also be able to auto-generate mnemonic flashcards. The product will eventually feature background music and an upvote system on search.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

This product is aimed mainly towards students, as it is presented in a classic flashcard style.

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.>

The operating environment is an Ubuntu server on a Digital Ocean Droplet. The product runs on any web browser, such as Chrome, Firefox, and Safari.

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).>

The two main constraints this project has are time and available knowledge. There was a hard deadline of 4/27/22, so the team needed to prioritize what we could realistically get done in that timeframe. Also, the technology chosen to build and operate this product were chosen based on what the team was most comfortable using already, as having to learn entire new systems would further limit the time available for completing the product.

2.6 User Documentation(optional)

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

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.>

The most important part of our UI is the navigation bar that will appear at the top of every screen. It will be able to take you to each of the necessary pages, including home page, log in, search results, profile, and create (user and mnemonic), from anywhere on the site.

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.>

Software and services we will be using for our project will be Java and the Spring Framework for backend work, SQL for database management, as well as HTML/JSP, and CSS for the frontend of this project.

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

This website will use HTTP to communicate with the web browser.

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.>

4.1 System Feature 1 - Mnemonic Device Flashcard Creation

Mnemonic device flashcard creation use case:
Actors: User, Application, Database
Inputs: Title, Study terms, Mnemonic device terms, Checkbox click, Tags
Outputs: Mnemonic device flashcard set
Normal operation:
The user inputs the desired study terms they would like to memorize. They are then prompted to input a matching set of mnemonic device terms. The user submits the set and, if correct, the flashcard set is created. Tags can be added to aid search. If private checkbox is selected, this flashcard will not be shown in search results.

Exception:
Invalid flashcard: Both sides of flashcard must have valid entries before the user is allowed to save it.

4.2 System Feature 2 - Login

Login use case:
Actors: User, Application, Database
Inputs: Username, password

Outputs:
Normal operation:
The user inputs his/her username and password into the website. If correct, he/she is presented with their account dashboard.
Exception:
Invalid login: User is presented with an error message, and prompted to enter the correct password. If login fails after X attempts, the account is timed-out and the user is presented with the option to reset their password.

4.3 System Feature 3 - …

Search use case:
Actors: User, Application, Database
Inputs: Search content
Outputs: Search results with flashcard sets
Normal Operation:
The user clicks the search field and enters the desired search criteria. Search results appear from matches in the database.

4.4 System Feature 4 - …

Save use case:
Actors: User, Application, Database
Inputs: User click
Outputs: Saved flashcard set
Normal Operation:
Once the user has a flash card set selected or created, the user clicks the save button and can then view the flashcard set in their profile.
Exception: If the user already has this flashcard saved, it will return nothing.

4.5 System Feature 5 - …

View use case:
Actors: User, Application, Database
Inputs: User click
Outputs: Flashcard set
Normal Operation:
Once the user has a flash card set selected or created, the user clicks that flashcard icon in the profile page to view the flashcard, clicking the flashcard itself will flip to show the other side.

4.6 System Feature 6 - …

Delete use case:
Actors: User, Application, Database

Inputs: User click
Outputs:
Normal Operation:

Clicking the delete button on the flashcard icon on the profile page will delete the card from that user's page.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

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.>

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.>