Active-Reading Assistant
Software Requirements Specification

A. Hornof (ajh) - 3-31-2022 - v0.1

Table of Contents

1. SRS Revision History

Date	Author	Description
3-31-2022	ajh	Created the initial document using the template from https://classes.cs.uoregon.edu/22S/cis422/Templates.html

2. The Concept of Operations (ConOps)

This document is the Software Requirements Specification (SRS) for the Active-Reading Assistant (ARA). ARA is a computer program that is designed to support the SQ3R reading method.

ARA is a software tool that guides students through the use of the SQ3R active reading technique. SQ3R stands for Survey, Question, Read, Recide, Review. SQ3R provides students (and other readers) with a structured approach for reading textbooks and other technical material. Learning such material is best accomplished by first skimming the high-level structure of a chapter (or other section of text), and then generating questions, and then reading the material primarily to answer the initial questions (McKeachie & Svinicki, 2013).

The goal of SQ3R is to motivate students to engage in *active learning* by first generating questions, and then move through the text with the goal of answering those questions. In the first pass through a chapter, the student should skip over a lot of text, such as to only (a) read the headings and topic sentences and (b) study the figures and diagrams. It is expected that students will eventually read the entire chapter, but only after first constructing a mental understanding of the structure of the chapter, and how its major pieces fit together (McKeachie & Svinicki, 2013).

The basic concept of ARA is that it will provide a structured, hierarchical, note-taking facility that guides students through the use of SQ3R active reading technique, partly by providing appropriate prompts for the next step in following the SQ3R, and partly by providing a structured organization of text fields in which the user can type their notes.

A key feature of ARA is that it should store the user's notes on a server that can be accessed from anywhere with full internet access. This permits the user to use multiple different computers to use the SQ3R method to read a textbook, and always add to their previous notes rather than starting fresh.

2.1. Current System or Situation

There are substantial digital-document ad paper-based resources available to assist students in the use of SQ3R. For example: Michigan State University (2022), McKeachie & Svinicki (2013), Robinson (1946), and University of Oregon (2022). A brief internet search suggests that there are few if any pieces of software created specifically for helping students learn and use SQ3R.

2.2. Operational Features of the Proposed System

The key operational features of ARA include guiding the user through the use of SQ3R, providing a hierarchy of text fields in which the user can do SQ3R tasks (such as writing questions and answers), and storing all of the data on a server.

The system will specifically follow the SQ3R technique that is described by Michigan State University (2022), which is also copied at University of Oregon (2022). One distinction of this specific form of the technique is that the *reading* phase is focused on answering questions, not reading each section carefully.

2.3. User Classes

There are two user classes:

1. A "student" who is attempting to use SQ3R, with the assistance of ARA, to engage in active learning while reading. The "student" could be anyone, not necessarily someone in school, but the SRS will refer to this user as the student.

2. A system administator who sets up the server.

2.4. Modes of Operation

The system has one primary mode of operation, in which the server is running, and students run ARA which gain client access to the server.

2.5. Operational Scenarios (Also Known as “Use Cases”)

(These will be developed.)

3. Specific Requirements

The basic functionality must include the following:

1. The program provides a visual framework for taking notes.
2. The notes should be hierarchical as follows:
    Book Title (single line of text, entered by user)
        Chapter Title (single line, entered by user)
            Notes (scrolling text, with some kind of clear separation between lines.)
3. There should be no login required for this initial version of the software. Any login information should be built in to the software. (To assist with this, it is acceptable if the initial version of the software has only a single user.)
4. The software should run on a laptop or desktop machine. The program should be designed for use with a real keyboard, not a smartphone.
5. The program should guide the user through the use of SQ3R with prompts such as:
    “SURVEY: Glance over the headings in the chapter to see the few big points.”
        5a. “Prompts” should guide the user through the use of SQ3R. Prompts should be non-modal: They should not block any text the user types, should not require the user to read them or provide any input, and should not be interative. For example, the prompts could appear as headings above the text fields.
        5b. The user should be able to turn the prompts on or off, making the prompts either present or absent.
6. The system should save all notes that the user enters, and should never delete user data without a warning.
7. Data should be stored on a server using either mysql or mongo.

Build-Related Constraints

1. Target Platform
The system must run on Macintosh OSX 12.1.

2. System Document File Formats
All system-related and system-development-related documents that are intended for human reading must be in either plain text or PDF. For example, Microsoft Word, Microsoft Excel, or markdown language documents must be converted into plain text or PDF.

3. Programming Constraints
- The system may be built in C/C++, the C++ standard library, Cocoa, and no other components. (Note that an XCode command line tool could fulfill many of the requirements.)
- The system may be built using Python 3 along with The Python Standard Library https://docs.python.org/3/library/index.html, but no other imports except for mysql or pymongo. This means that the only GUI package that can be used is tkinter.
- The system may be built using Java along with Java Standard Edition modules https://docs.oracle.com/en/java/javase/12/docs/api/index.html, but no other imports.
- C++ code must comply with C++11.
- Python code must run in Python 3.7 through 3.10.
- Java code must run in Java 7 or 8.
- Instructions must be provided for how to compile the code.
- No server connections may be required for either installing or running the software, except for the server that you provide instructions to set up.
- No virtual environments may be used.
- No gaming engines such as Unity may be used.

8.4. Installation
- There can be at most 20 user actions to compile the code and run the program.
- An experienced computer programmer should not require more than 30 minutes working alone with the submitted materials to compile and run the code.

4. References

McKeachie, W., & Svinicki, M. (2013). McKeachie's teaching tips. Cengage Learning.

Robinson, F. P. (1946). Effective study.

University of Oregon. (2022). Tutoring and Academic Engagement Center. (2022)., https://engage.uoregon.edu/learning-resources/. Accessed 3-31-2022.

Michigan State University. (2022). Reading a Textbook Effectively. https://natsci.msu.edu/students/current-students/student-success-resources/academic-success/habits-to-develop-outside-of-class/study-strategies/reading-a-textbook-effectively/. Accessed 3-31-2022.

5. Acknowledgements

This SRS builds on the template from https://classes.cs.uoregon.edu/22S/cis422/Templates.html.