Software Requirements Specification (SRS) for *natsu*

TophUwO

December 2023

Contents

Version History

Below is a chronological list of changes, alongside metadata. Version *1.0* marks the first release version of this document.

> **Note**
>
> Dates used in this table are of the format ***MM/DD/YYYY***.

Date	Version	Contributor(s)	Changes
12/07/2023	0.1	TophUwO	initial commit
12/07/2023	0.1.1	TophUwO	**ADD** content structure for *Introduction* section; **ADD** text for *Purpose...*, *...Conventions*, and *...References*
12/07/2023	0.1.2	TophUwO	**ADD** initial content for *...Rationale* sub-section
12/07/2023	0.1.3	TophUwO	**ADD** contents for product feature overview

1 Introduction

This chapter serves as a general introduction to the product, its features, and its working environment. The features and requirements mentioned here will be discussed and elaborated on in chapter 3 and 4.

1.1 Purpose of This Document

This document describes the software requirements specification (SRS) for a standalone desktop and notebook application dedicated to enhancing the user's *Hanzi*/*Kanji*/*Hanja*[1] learning experience and efficiency. The application is codenamed *natsu* (jp. for *Summer*) and will be referred to as such throughout the remainder of this document. This document was created in accordance with the *IEEE 830* standard, is part of the internal developer documentation, and serves as a benchmark of what the application is intended to do and be used for.

1.2 Document Conventions

In this document, technical and other domain-specific terms relevant to the implementation are in *italics* while table heads are rendered in **boldface**. See section *Glossary* for an explanation of important technical terms used in the context of the product's domain. Other nomenclature (also in *italics*) may be further explained by either footnotes or in the section *External References*. Terms in *italics* are also hyperlinks to other sections within this document.
This document manages its own version history (see section Version History). Each commit holds data such as *Date*, *Version*, *Contributor(s)*, and *Changes*. Changes begin with a verb in infinitive form, all-caps, in boldface, describing the change's category, e.g. **ADD**, **REMOVE**, **FIX**, etc.
Figures such as GUI mock-ups and OOA diagrams can be found in the Appendix A and B, respectively.

1.3 Background and Rationale

Languages are a very prevalent subject in today's globalized society. For the current and following generations of people, mastery of different languages appears mandatory in a variety of industries that thrive in an international environment. Particularly Eastern-Asian languages are popular in the Western world due to interest, requirements, or simply popular culture. What many of the most popular languages like Mandarin, Japanese, and to an extent (South-)Korean have in common is the use of logographic characters of ancient Chinese origin. While these characters are referred to as Kanji (漢字) in Japanese, regional names may differ.
This SRS describes an application that is supposed to substantially relieve some of the frustrations commonly encountered when learning Kanji. While there are high-quality solutions aimed at learning Kanji in context, no solution of this quality exists for writing practice. This is the gap natsu will fill. I believe that the application will be useful to those who want to learn Japanese hand-writing in a controlled and customizable environment.

1.4 Product Feature Overview

At its core, the application should be a software that manages flashcards by the means of an SRS (*spaced repetition* system). These flashcards should contain information on the Kanji character, such as *radicals*, *readings*, *stroke information*, *common vocabulary*, etc. These flashcards can be created and modified by the user. Flashcards are part of a *deck* that can be imported/exported from either the proprietary natsu deck format, or other formats such as *CSV* or SQL-based databases. Progress on each item must be saved persistently in a database. Optionally, items can be grouped together in a *level order* which allows for certain subjects to be studied to retention until other items are processed. The application will be shipped with a large number of pre-installed Kanji characters, sourced from the excellent *KanjiVG* project. There should be a way to add new characters to the repository, either by pre-made data or manual creation by the means of drawing. In the review, the user is now required to write the subject whose meaning/hint is shown. The application relies on self-evaluation in order to determine progress.
Furthermore, the application should be focused on customization via application and icon styles. It should be possible to set a global accent color that will be used for icons, focus indicators and miscellaneous items present in the GUI.

[1] logographic characters of Chinese origin; used to this day in Chinese languages, Japanese, (South-)Korean, etc.

1.5 Stakeholders and Audience

Since this is a personal project, the only stakeholder is the project manager, developer and maintainer, TophUwO. This document is also meant for future contributors of the software described within this document.

1.6 Glossary

Below is a curated list of context-specific nomenclature used throughout this document, alongside brief explanations.

Term	Explanation
radical	graphical parts of a Kanji used for indexing
reading	a way of pronouncing a character; context-sensitive
stroke	smallest distinguishable grapheme in a Kanji/radical

1.7 External References

Below is a non-exhaustive list of useful resources that are likely to be frequently consulted throughout the conceptualization and implementation of the software described by this document.

Note

The resources presented beneath may serve as a starting point but do in no way represent a complete and necessarily sufficient resource for the entirety of the subject. Additional research is expressly endorsed.

Resource	URL
IEEE 830 specification	`http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf`
Qt 6 documentation	`https://doc.qt.io/qt-6/`
spaced repetition	`https://en.wikipedia.org/wiki/Spaced_repetition`
KanjiVG project	`https://kanjivg.tagaini.net/`