

日本一語 (Nihon-Go!)

Dylan Morgen

Project Overview

For this project I intend to use the opportunity to create the databases for my capstone project. The intent of the capstone project is to produce a piece of software to assist people in learning the Japanese language with a focus on grammar and written communication.

The scope of this project applicable to this course is the data structures; the dictionary database and user databases will be all I work on in this course. I will be making a simple web interface to search for words and view your personal 'set' of stuff to study (indexed by user id).

Database Design

This project will pose unique challenges to implementation due to Japanese not using a Latin alphabet. The database will likely have to be Unicode. It might make more sense however to use JIS X 0212 encoding.

The dictionary database will be populated by a variety of sources:

JMDICT

EDICT

KANJIDIC2

KRADFILE

While the number of sources here to be parsed and added seems large, I believe each adds a valuable component to the dictionary entry and will be essential for functionality in the end app. Furthermore, I hope to link entries in a variety of ways helpful to language learning. Some ways I hope to link data include cognates, morphemes, and characters related in concept and appearance. To do this I will most likely write scripts in Java or another language I am familiar

with to edit the SQL server directly. At some point I will be adding a table for user submitted mnemonics, but most likely not for the database produced in this course.

Kanji in said table will be linked to a 'readings' table. Each Kanji entry will most likely have several readings, which will contain the phonetics and meaning of each reading.

Words in said table will include links to all Kanji used. Here is a simple example of a kanji and its links:

本 (kanji db) -> 本 [ほん] (reading db) -> 本 [ほん] (word db)¹

Another link path is:

本 (kanji db) -> 本 [ほん] (reading db) -> 日本 [にほん] (word db)

It might seem redundant to have both the reading and word tables, but it should be noted that every kanji will have one or more readings, but a word may have no kanji (and therefore no reading entries). The reading table will store very little information, it will be the bridge between kanji and words. The relationship between the kanji and readings table will be one to many. The relationship between the readings table and word table will be one to many.

The user database will store the ID of a word in the dictionary, index of the reading, and with a few metrics for *how* the user knows it. Do they know it well? Can they recognize the character, but not the reading? How many times has it been shown to them?

User Interface

My end program will most likely be written using Flutter, a cross platform framework which allows for a unified codebase. It doesn't seem difficult to create a webapp using this, but if necessary, I will resort to using either plain javascript or node. I don't anticipate much trouble with this part of the assignment, as Flutter has libraries for SQL and other common databases. I plan on hosting the databases on Heroku using the free dynamos allocated to a basic account. Access will likely be using a premade SQL hosting library using HTTP requests for the actual

¹ 本 is the character; ほん is the pronunciation written in a phonetic script

content. For the capstone I will be transitioning this to proper authentication.

Controls available in this web interface will include dictionary search, clickable links on word entries for related words, and the option to add entries to your ‘study list’. I hope to have another page for viewing entries, along with some primitive controls for bumping entries up or down the scale of ‘how well you know them’.