Development of E-learning Contents for Exchange Students

Kuniaki YAJIMA[#1], Jarmo Seppälä[*2], Akihiro NITTA[#1]

[#]*Department of Information networks, Hirose campus, Sendai College, Institute National of Technology*
Sendai, Miyagi, Japan
[1] yajima@sendai-nct.ac.jp
[3] s1100731@sendai-nct.jp
**Department of Information Communication Technology, Helsinki Metropolia University of Applied Sciences*
Espoo, Finland
[2] jarmo.seppala@metropolia.fi

Abstract- **The outline for this project is to create a Web-based learning environment for exchange students. The service's objective was to create an ever-growing database of words, utilize an interactive learning environment and endorse self-driven way of learning using text, sound and images. This was achieved by using semantic web technologies and a database, which allow users to create their own data, edit or delete data regardless of the end device used.**

Keywords- **e-learning, web-services, mobile, usability, database**

I. INTRODUCTION

The purpose of the study is to implement a Web-based language learning environment where student can input, view and study Japanese words. All of the learning contents are made by the students themselves, whereas the teacher is mainly acting as a moderator.

In the word adding section, students are able to input new words in to the database and create memory rules with each word in the shape of example sentences, images and audio.

Data can be then viewed in the word-bank, where the words are divided by their word type. For example, onomatopoetic words and nouns have their own dedicated sections.

From these self-created words, students can create a quiz and study them in a section dedicated for this purpose. The quiz sequentially displays the words like a set of flashcards, bearing all the student made information.

II. TOOLS AND METHODS USED

User interface, or front end of the service was developed by using HTML5, CSS3 and JavaScript mark-up languages, as these technologies provide mobile scalability and overall increased usability [1].

Server side, or the back end of the service utilizes PHP and MySQL mark-up in order to store and retrieve the word-entries from the database. Ease of use and reliability makes MySQL one of the most popular database systems used with PHP. [2]

As for increased speed and user-friendliness between client and server, AJAX calls are frequently used. Ajax is a client-side script that communicates to and from a server/database without the need for a post back or a complete page refresh.

In other words, AJAX is the method of exchanging data with a server, and updating parts of a web page-without having to reload the entire page. Another major advance to JavaScript and Ajax is the JavaScript object library called jQuery which is a JavaScript library designed to simplify the client-side scripting of HTML. [3]

The service also utilizes an API (application programming interface), which fetches an external audio-file in a JSON format from "Forvo" pronunciation service to each corresponding word if the sound file is available.

JSON is short for JavaScript Object Notation, and is a way to store information in an organized and easy-to-access manner. It gives a human-readable collection of data that can be accessed in a logical manner. [4]

Using all these above mentioned developing tools, creating, manipulating, fetching and deleting data over the web is made possible. Below is a flowchart which describes the workflow of the service.

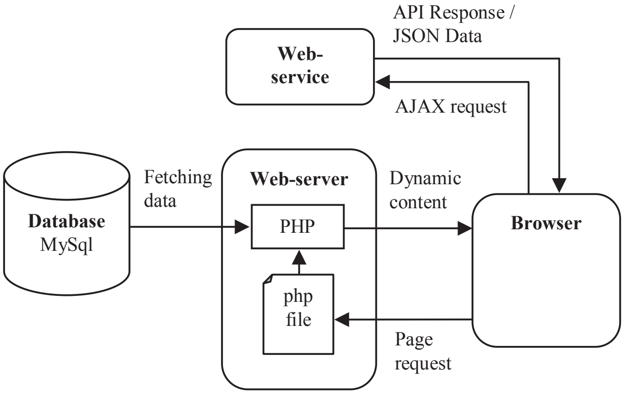
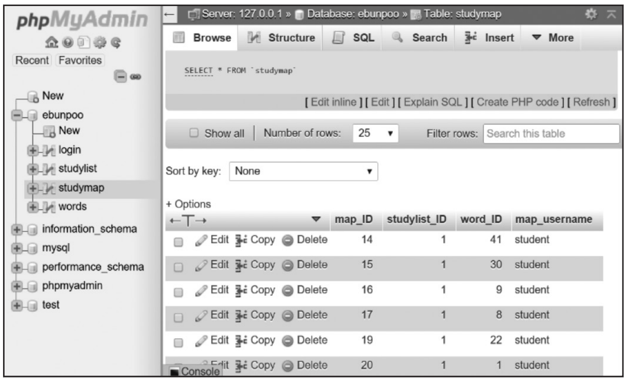


Fig. 1 Wireframe of the service's functionality

Accessing the database through browser is possible by using service called "phpMyAdmin", which is a tool written in PHP intended to handle the administration of MySQL over the web. [5]



```
SELECT * FROM `studymap`
```

	map_ID	studylist_ID	word_ID	map_username
□ Edit Copy Delete	14	1	41	student
□ Edit Copy Delete	15	1	30	student
□ Edit Copy Delete	16	1	9	student
□ Edit Copy Delete	17	1	8	student
□ Edit Copy Delete	19	1	22	student
□ Edit Copy Delete	20	1	1	student

Fig 2. phpMyAdmin control panel. Data tables can be accessed here using a graphical interface.

III. USING THE SERVICE

A. Overview of the user interface

When arriving to the service's main page first time, user has to login with an existing account or create a new one.

All new users are required to make a profile in order to use the service, since each user has their own self-created words and memory rules.

After successful registration, user is redirected in to the main page, where the three core functionalities of the service can be seen. These functionalities are:

- Word adding section, where the users are able to create their learning materials
- Word storing section, where all of the words are stored and can be later viewed
- Word learning section dedicated for learning user chosen words

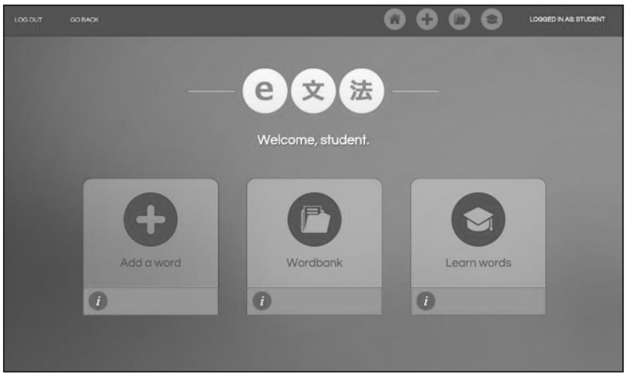


Fig. 3 Main page as seen in desktop computers

The service can also be used in mobile devices. Media queries, which were introduced in CSS3, is a technique which alters the appearance of the website when it meets "breakpoints" of certain width.

For example, when the screen width is 768 pixels wide, the page will reorder its contents to fit the end device's screen promptly.

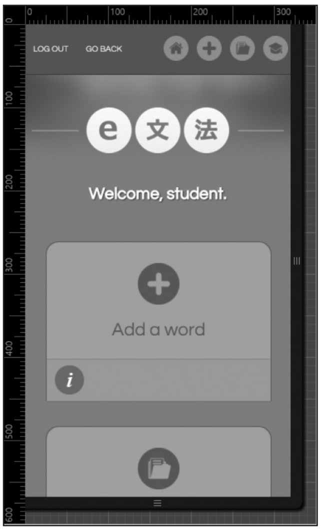


Fig. 4 Main page as seen on iPhone 5's resolution (640×1136).

Moving from one place in to another inside the service happens by clicking the icons in the navigation bar, which is on top of each page.

This allows mobility and freedom to the user what comes to transitioning between pages regardless of the user's current location.

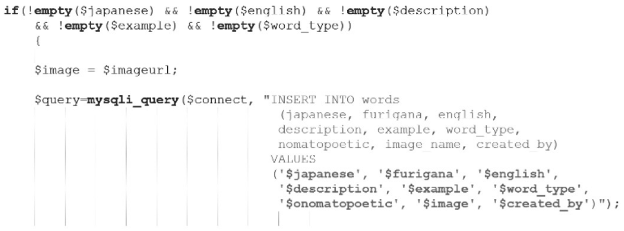
B. Word adding

In the word adding section, new data is created by using HTML form. User needs to fill all the necessary input fields before submission. The input fields consist of the following fields:

- Japanese input in kanji
- Japanese input in hiragana
- English translation input
- Word description field
- Example sentence field
- Word type checkmark
- Image uploading field
- Onomatopoeia checkmark

Image input and onomatopoeia checkmark are not required. jQuery script checks if the necessary fields are filled, and then makes the submission button available. Submitted data is then processed by an external PHP file, which connects and inserts the received data in to the database.

If the submission was successful, user can choose to add another word or return to the main page.



```
if(!empty($japanese) && !empty($english) && !empty($description)
    && !empty($example) && !empty($word_type))
    {

    $image = $imageurl;

    $query=mysqli_query($connect, "INSERT INTO words
                                    (japanese, furigana, english,
                                    description, example, word_type,
                                    nomatopoetic, image name, created by)
                                   VALUES
                                   ('$japanese', '$furigana', '$english',
                                    '$description', '$example', '$word_type',
                                    '$onomatopoetic', '$image', '$created_by')");
```

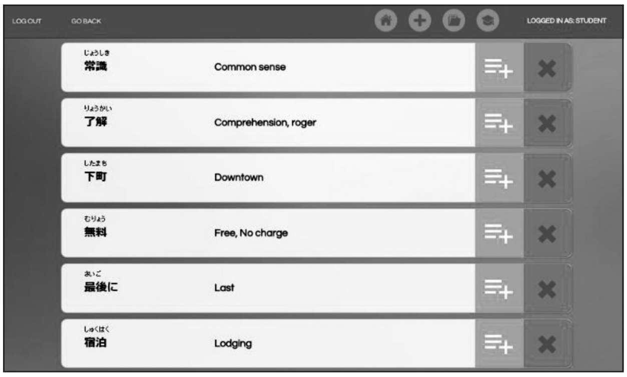
Fig. 5 Data-entry. Example of PHP & MySQL mark-up. Input entries being inserted into the database.

C. Word storing and displaying

Words are stored in a dedicated place called word-bank. All of the user created word data can be viewed in this section of the service.

Words have been categorized in eight different types, which each have their own dedicated page. These word types are adjectives, adverbs, verbs, nouns, pronouns, particles, conjunctions and onomatopoetic words. User can also choose to view all of the words in one page.

Upon clicking on a desired category, user is directed in to a dynamically created page, where the words designated to that specific category are displayed.



じょうしき 常識	Common sense
りょうかい 了解	Comprehension, roger
したまち 下町	Downtown
むりょう 無料	Free, No charge
さいご 最後に	Last
しゅくはく 宿泊	Lodging

Fig. 6 Viewing all the nouns

As seen in the figure above, all of the nouns are displayed and are listed on top of each other in alphabetical order. Each word have their own container, which is accompanied by two buttons:

- add to quiz - button
- delete word - button

Using the green "add to quiz"-button, user is able to add words in to a quiz for later studying purposes. When this button is clicked, a small notifier at the bottom of the page informs that the word was added in to the quiz list in the database.

These words are later displayed in the quiz section in the same order as the green buttons were clicked.

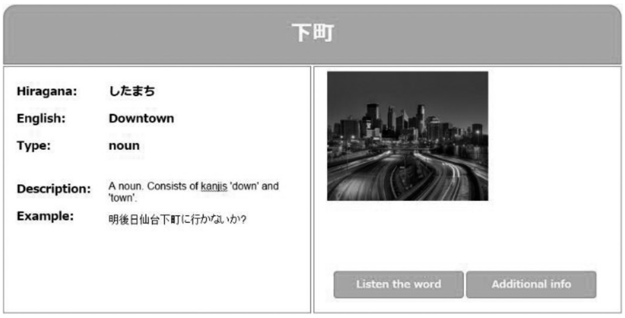
Upon clicking the red delete button, jQuery script runs an AJAX call which connects to a specific PHP file. This PHP script then receives the word's unique identification number. Using this number, the PHP script then connects in to the database and finally deletes that specific word-entry permanently.

As AJAX is a client-side script that communicates with the server/database asynchronously, the page doesn't need to be refreshed every time a word is added in to quiz list or deleted completely.

D. Displaying individual words

Since all the words are given a unique identification number upon their creation, word-specific fetching and dynamic links can be made.

As seen in figure no. 6, white area around each word container has link. When this white area is clicked, browser URL takes that word's identification number as its parameter and displays the word individually in a new page.



下町

Hiragana: したまち
English: Downtown
Type: noun

Description: A noun. Consists of kanjis 'down' and 'town'.
Example: 明後日仙台下町に行かないか?

Fig. 7 How individual words are displayed after clicking any of the words in the word-bank. In this case, the fetched word is "downtown".

In this dynamically created page, user can view the words individually. As seen in figure no. 7, all the example sentences, descriptions and images which were created in the word-adding section earlier are now displayed.

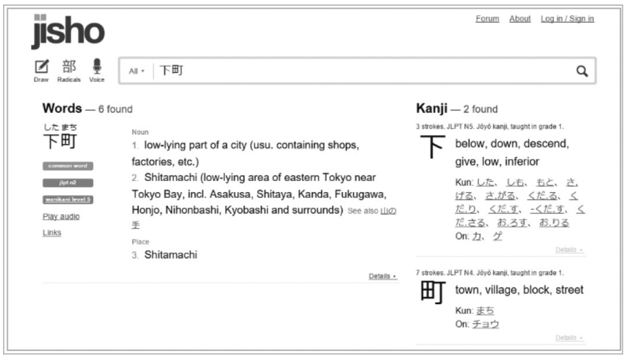
In this page, altering of the word data can also be practiced. By simply touching the area where the word data is (for example, description), the text field lights up in a different colour and becomes editable.

After the user has finished editing, jQuery sends an AJAX call to tell the database to update information. As this editing manoeuvre happens asynchronously, page doesn't need be refreshed.

"Content editable" element is one of the new attributes introduced in HTML5 mark-up. Accompanied by AJAX/PHP scripts connecting the database, this element makes editing of HTML data easy and fast, as new word data can be written in on the fly. [6]

Besides viewing and editing the word data, user can choose to look up additional information about the word from internet dictionary called "jisho".

By clicking the "Additional info" button at the right, a new tab in the browser opens. Address of this page has the word used in its search parameter.



Words — 6 found

したまち
下町

Noun
1. low-lying part of a city (usu. containing shops, factories, etc.)
2. Shitamachi (low-lying area of eastern Tokyo near Tokyo Bay, incl. Asakusa, Shitaya, Kanda, Fukugawa, Honjo, Nihonbashi, Kyobashi and surrounds) See also 山の手

Place
3. Shitamachi

Kanji — 2 found

3 strokes. JLPT N5. Jōyō kanji, taught in grade 1.
下 below, down, descend, give, low, inferior
Kun: した、しも、もと、さ.げる、さ.がる、くだ.る、くだ.り、くだ.す、-くだ.す、くだ.さる、お.ろす、お.りる
On: カ、ゲ

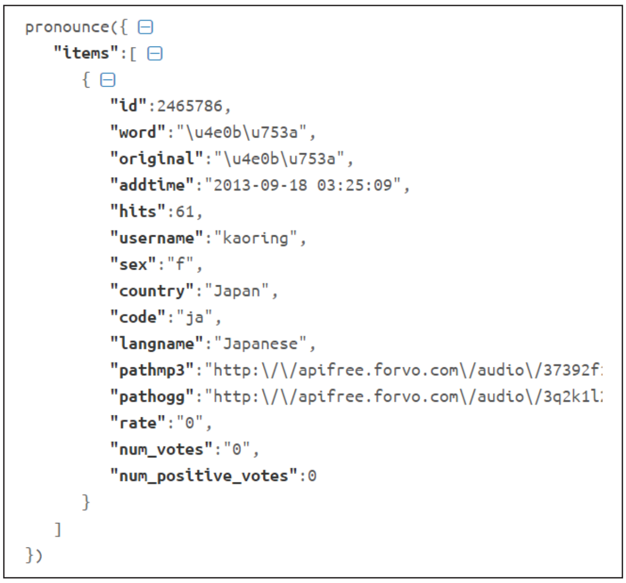
7 strokes. JLPT N4. Jōyō kanji, taught in grade 1.
町 town, village, block, street
Kun: まち
On: チョウ

Fig. 8 Searching additional information from "Jisho" with the chosen word: downtown.

Listening the word is also possible for available words. When clicking the "Listen the word" - button, jQuery script runs a function which connects to the Forvo pronunciation service, fetches the desired word in JSON format if available and then plays it once in the browser.

As the JSON data returns all the information it can get from the server regarding to that specific word, the desired part needs to be selected in the JavaScript.

As only the link to the sound file is wanted, the script does a query which only points to "pathmp3". From the figure below, format in which the JSON returns data can be seen.



```
pronounce({ ⊟
   "items":[ ⊟
      { ⊟
         "id":2465786,
         "word":"\u4e0b\u753a",
         "original":"\u4e0b\u753a",
         "addtime":"2013-09-18 03:25:09",
         "hits":61,
         "username":"kaoring",
         "sex":"f",
         "country":"Japan",
         "code":"ja",
         "langname":"Japanese",
         "pathmp3":"http:\/\/apifree.forvo.com\/audio\/37392f:
         "pathogg":"http:\/\/apifree.forvo.com\/audio\/3q2k1l:
         "rate":"0",
         "num_votes":"0",
         "num_positive_votes":0
      }
   ]
})
```

Fig 9. JSON representation describing the fetched word "downtown".

E. Word learning

The actual learning of the words is designed to be practiced in this section of service. All the words which were added in to the quiz list back in the word bank are shown in this page in a separate word containers, which appear to the user one by one.

The quiz works in similar sense as flash cards - one side has the word in English and the other one in Japanese. User can also choose a reversed order, where the question side is Japanese and answer side in English.

User navigates through the quiz by clicking one of the two buttons, which are dedicated for the purpose if the user remembers the word or has forgotten it. After either "remember"-, or "not remember" - button has been clicked, next word is displayed.

It is up to the users to decide whether they remember the words or not. Choosing either one of the buttons will not affect the order in which the cards are displayed.

Users can also use hints, if they don't remember the word. These hints consist of user-made descriptions, example sentences and uploaded images.

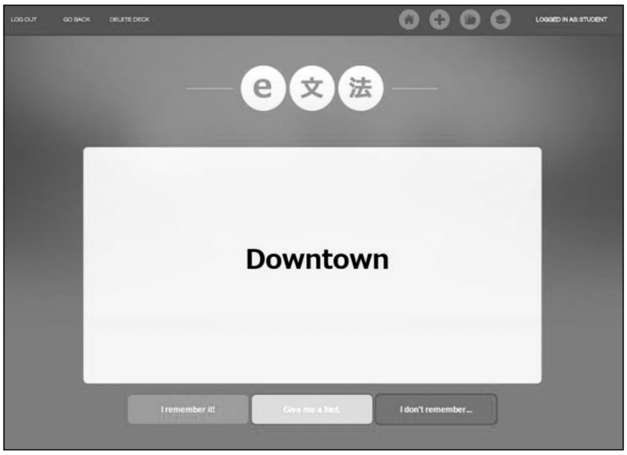
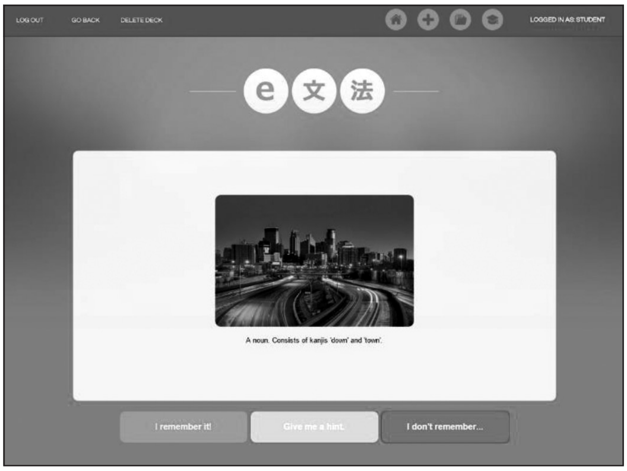


Fig. 10 The quiz section.



A noun. Consists of kanjis 'down' and 'town'.

Fig. 11 Using a hint

When the quiz has reached its end, users are shown how many words they remembered. Users can also take the quiz again, or delete the study deck using the button located in the navigation bar.

IV. CONCLUSION

As of now, the service delivers all the functionalities which are required to store, edit and delete data. UI of the software is near completion, but several features require further work and polishing, such as image uploading as it's not possible to add images later after the initial input.

The pronunciation API might be replaced with an alternative method, as its usage costs - own voice recording system is a possibility.

ACKNOWLEDGMENT

The author would like to acknowledge with appreciation and gratitude the invaluable help and guidance of Mr. Kuniaki Yajima, teacher of Networking systems and project supervisor at the Sendai National College of Technology.

REFERENCES

[1] R. Ritchie, "The iMore website," 2012. [Online]. Available:http://www.imore.com/its-not-%20about-html-%20or-skeumorphism-%20its-about-usability.

[2] "The W3Schools," 2016. [Online]. Available:website,http://www.w3schools.com/php/php_mysql_intro.asp.

[3] W. Fote, "Blog on Seguetech," 2013. [Online]. Available:www.seguetech.com/blog/2013/03/12/what-is- ajax-and-where-is-it- used-in- technology.

[4] J. Lengstorf, "The Copterlabs website," 2015. [Online]. Available: http://www.copterlabs.com/json-what- it-is- how-it- works-how- to-use-it/.

[5] "The phpMyAdmin website," 2016. [Online]. Available:https://www.phpmyadmin.net/.

[6] "Developer page on Mozilla," 2016. [Online]. Available:https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_Editable.