# System Aiding Learning Japanese Language

Piotr Halama
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: piotr.halama@o2.pl

Joanna Janiszewska
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: joa.j711@gmail.com

***Abstract*—In this paper creating platform for language e-learning was discussed in both theory and as an example implementation. We present a discussion on used work-flows to show programming background of our project but we also discuss it from the user oriented quality of service. For these we present opinion results that our users gave us after using the applications. Finally we propose the way of development and possible future works.**

## I. Introduction

Web services are important software developments. There are several possibilities to implement each of them. In [10] was presented a survey on various possibilities for implementation of automated approaches used in on-line technology. Also a quality of service is very important especially for users, the better quality the more potential users will be interested in the service. In [8] was proposed a standard for composing dynamic services oriented on the user needs. In [1] was presented a discussion of work-flows management for web services. However it also crucial to think about the ontology of web services, a very interesting aspects of this were discussed in [11]. Also some models for verification of web services are very useful for quality of applications [4].

There are many techniques which directly adhere to practical aspects of web services. In [7] was presented a survey discussing their positive and negative aspects. In [3] authors proposed WFSM framework for web services. There are also frameworks devoted for special purposes. In [6] was discussed a framework OWLS-MX developed for semantic web services, while in [9] was proposed Meteor-s annotation framework.

One of most interesting applications of web services are these oriented on e-learning applications. In [12] were presented possible directions for the development in e-learning technologies, while in [14] were presented the uses and comments on the empirical results from e-learning technologies.

In this article we present our web service developed for e-learning of Japanese language. There are not many services that would allow user to learn Japanese language with the help of e-learning, especially given that this language due to its complexity needs individual treatment as opposed to most of the popular languages. We present an application of web service to help in learning of the Japanese in proposed annotations based on different alphabets. We have measured opinions from our users on the application developed for e-learning to evaluate which approach gives the best results. Our findings are presented in this article and discuss to show our approach.



## II. Introduction to Japanese alphabets

Japanese language usually uses three notations: hiragana, katakana and kanji. Hiragana is a Japanese syllabary, one character is one syllable, it is used in children's books and in textbooks before person's learn kanji.

Kanji are the adopted logographic Chinese characters, one character can mean one word, or be a part of the word. Each kanji have Onyomi and Kunyomi, the former being Chinese pronunciation at the time of its introduction, the latter Japanese pronunciation.

We let user choose if they want to use hiragana or kanji. If the user language knowledge is advanced enough, they can choose kanji since it is more common. Otherwise they can choose hiragana which is easier to learn but not used as common as kanji.

## III. Existing implementation and service overview

Data in the database is split into categories (called courses in the database and API), each category have assigned one of three types, that is vocabulary, kanji and definitions, which determines the data scheme in response body of an API requests. For instance, type "vocabulary" determines that response will contain entry id, Kanji notation, Kana notation and Polish notation, whereas in "Kanji" that would be entry id, Kanji notation, Kunyomi and Onyomi.

Both Android application and the website frontend retrieves data from the database using JSON API.
Our service uses one API for both website and Android application to get and save data to the database, Fig. 2 describes available API endpoints. This way we ensure that every application uses the exact same data served in the same format. However we need to copy most of the logic into every application since received data needs to be processed to format

used internally in given application and processed back to JSON format for writing data to the database through API.

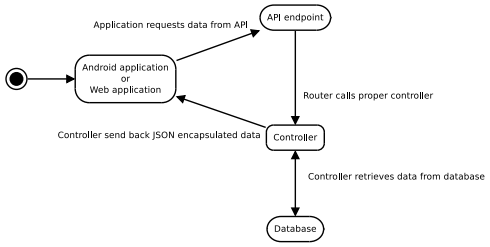


Fig. 1. Diagram showing data flow.

The Android application or web application sends a request to API using HTTP POST or GET methods depending on the need as shown on Fig 1. The server application that uses Model-View-Controller design pattern routes request to proper controller which retrieves data from the database, parse it and create JSON encapsulated response which is send back to the requesting application.

In the following example, we would like to retrieve data from API about all available categories. Android implementation written in Java, Gson and Retrofit libraries looks like follows:

```
private void getCategoriesFromInternet()
{
    String API_BASE_URL = "http://localhost/api/";
    OkHttpClient.Builder httpClient = new
        OkHttpClient.Builder();
    //retrieve JSON from API
    Retrofit.Builder builder =
            new Retrofit.Builder()
                    .baseUrl(API_BASE_URL)
                    .addConverterFactory(
                        GsonConverterFactory.create()
                    );

    Retrofit retrofit =
            builder.client( httpClient.build() ).build();

    //bind recieved data to Java class
    CategoriesClient client = retrofit.create(
        CategoriesClient.class);

    Call<List<Course>> call = client.coursesList(type
        );

    call.enqueue(new Callback<List<Course>>() {
        @Override
        public void onResponse(Call<List<Course>>
            call, Response<List<Course>> response)
            {
            //we were able to receive JSON and
                deserialize it
            courses=new ArrayList<Course>(response.
                body());
            }

            @Override
            public void onFailure(Call<List<Course>>
                call, Throwable t) {
                //failed to retrieve data
            }
        });
    }
```

Website implementation written in Vue.js that achieves the same goal as Android one looks like follows:

```
<template>
    <div class="container-cube col-md-4" v-for="
        category in this.categories">
        <div class="cube">
            Category: {{category.name}}
        </div>
    </div>
</template>

<script>

export default {
    components:{ },
  data () {
    return {
        categories: []
    }
  },
    methods: { },
    beforeCreate(){
      this.$http.get("courses").then( (response) =>{
        this.categories = response.body
      })
    }
}
</script>
```

We need to add following code to main Javascript file to inform Vue to use JSON format in communication and set correct ULR address used for communication with API:

```
Vue.http.options.root = 'http://localhost/api/';
Vue.http.options.emulateJSON = true;
```

API is written in PHP using Laravel framework and the corresponding API endpoint code looks like this:

```
/*database connection parameters are stored in Laravel
    application configuration*/
public function getAllCourses() {
    $coursesTableContents = DB::table('courses')->get();
    return response()->json($coursesTableContents);
}
```

As we can see, most of the logic have to be implemented in each application using API as stated previously.

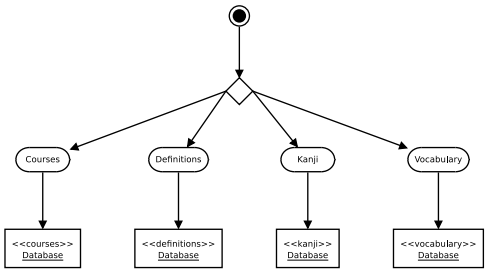


Fig. 2. Diagram showing API endpoints.

For instance, android application uses Retrofit2 library to retrieve data from API and GSON to convert it to existing Java classes.

Upon entering the website user sees list of three activities,as seen on Fig 3: quiz, flashcard and dictionary. After selecting activity a list of all categories is retrieved through API and shown to user to select from. After that, user selects direction of translation, for example from Kanji to Polish. List of words in given category is retrieved using API and selected activity is run.

Quiz shows one word in a given language and four buttons with answers in another language where only one contains correct answer. Upon pressing correct button one point is added to user score and next question is shown. After pressing wrong button one point is removed from user score and correct and selected answers are highlighted in green and red color respectively.

Flashcards shows one word in a given language and hidden translations as well as two buttons: ”I know” and ”I don’t know”. ”I know” button removes word from list and next word is presented to the user. ”I don’t know” button reveals translations of given word and adds the word at the end of the words list.

Dictionary is a table of words with corresponding translations.

As seen on activity flow for Android application (as seen on Fig. 4), the user sees list of three data types and an option menu. In options menu user can select activity they want to launch, direction of translation as well as additional help in form of hiragana notation. After selecting one of the types, a list of categories in given type is retrieved through API and shown to user to select. After that, user selects category, then list of words in given category is retrieved using API and selected activity is run.

Quiz shows one word in a given language and four buttons with answers in another language where only one contains correct answer. Upon pressing correct button one point is added to user score and next question is shown. After pressing wrong button one point is removed from user score and correct and selected answers are highlighted in green and red color respectively.

Flashcards shows one word in a given language and hidden translations as well as two buttons: ”I know” and ”I don’t know”. ”I know” button removes word from list and next word is presented to the user. ”I don’t know” button reveals translations of given word and adds the word at the end of the words list. Tapping field with hidden translation reveals it.

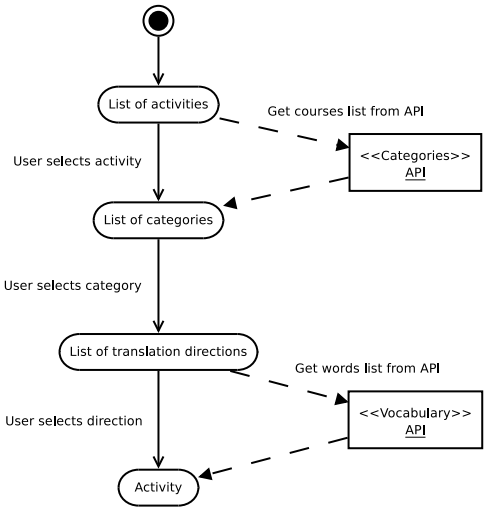


Fig. 3. Diagram showing user interface transitions in web application and API calls.

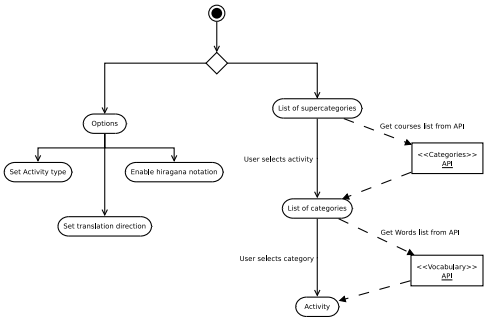


Fig. 4. Diagram showing user interface transitions in Android application and API calls.

## IV. Description of the traditional school

Learning process in traditional school is divided by difficulty level into semesters or even smaller units. Usually each unit is concluded with exam testing students’ knowledge and degree of their understanding of given topic. Students needs to have passed minimal percentage of correct answers on exam to proceed to the next learning unit. Teacher can also see student’s exam and provide them with custom exercises or explain them what they can change or pay more attention to.

## V. Introduction to e-learning

E-learning have a fair share of pros as well as cons over traditional learning and can supplement it if implemented correctly.

Physical location of the student does not matter. E-learning is a chance to learn for the people who live in places where traditional schools and language courses are unavailable. Users

can also learn from the specialist online, even if they live far away apart.

E-learning can be cheaper than traditional courses, not requiring physical copies of books and exercise books, making it affordable substitute for people with low income.

E-learning does not require a fixed schedule making it attainable for people with irregular lifestyle or unusual waking hours.

The best way to learn language however requires native speaker teacher which e-learning cannot provide without sacrificing some of its strength, for example not requiring to be online at fixed schedule.

User also have to be self-disciplined to learn regularly, which may be hard and can result in dropping e-learning.

Internet connection is usually required, given the the of the course it may be one-time download or constant connection for courses that use data from the web.

## VI. Implementation of traditional schooling in e-learning

Material of the e-learning platform can be split into units, as it is in traditional learning but we need to remember that some users would have previous contact with Japanese and we need to provide them some way to skip those units. We would suggest either asking user to take all exams prior to selected by them unit or automatically create new custom exam using knowledge from all previous units. On failure first approach would allow user to access some of the previous unit to the selected one given user's score. Second approach however would make it harder to differentiate which knowledge user already has and which not.

As a supplement to traditional learning, e-learning can provide a platform to both student and teacher, giving opportunity for teachers to see their students' marks and giving them advices and additional exams, like in traditional learning. Teachers could create custom units and use them instead of built-in ones, using e-learning platform as a framework rather than complete system with pre-existing knowledge base, taking time but giving more flexibility and tailoring learning curve specifically for their students or even for each student individually.

## VII. Implementation of theoretical considerations

Our implementation divides knowledge into smaller units as in traditional school, however we do not have requirement to pass all previous units to proceed to the next one. Our service does not implement user authentication and authorization so the usage of the service is anonymous but also forces user to keep track of their progress since our service does not have means to do it.

Our service have all aforementioned advantages of e-learning, being available online all the time and offline in a limited version since Android application have built-in database allowing it to operate without internet connection, however this database is smaller than the one available through online API to keep small application size.

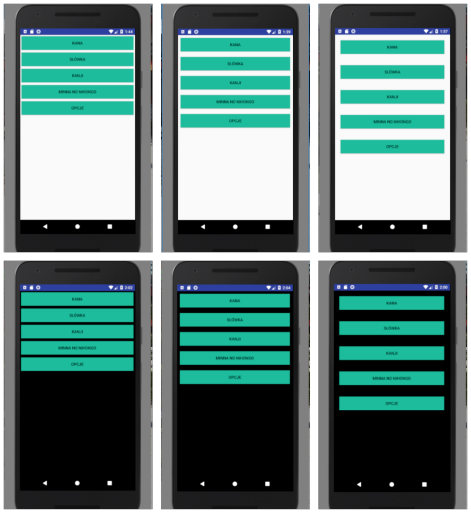

Fig. 5. Six interfaces used in questionnaire

## VIII. Interface questionnaire

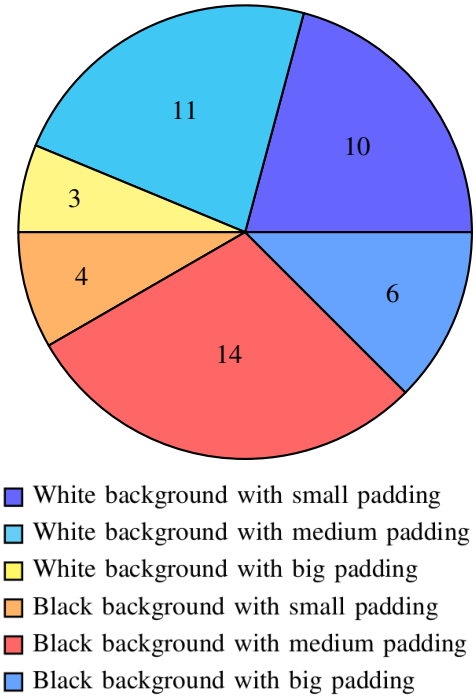


Fig. 6. Pie chart showing users' answers.

We asked users of mobile application which one of the 6 application interfaces (as shown on Fig.5) suits them the most and shown the collected data on a pie diagram (Fig 6). Out of 48 answers, votes for the white background and black background were split evenly. On white background smaller spaces between buttons were preferred, whereas on black background wider spaces were chosen more often. Considering only padding size, medium spacing was preferred with 25

votes, small gaps were chosen 14 times and big padding was selected by 9 respondents.

## IX. Conclusion

Depending on the student, e-learning may not be as effective as traditional learning but it can supplement it strengthening memorizing or in some conditions can be viable alternative to traditional courses.

The implementation of an e-learning platform can be used by students on very early development stages and with help of automated surveys help in development of new functionalities and in understanding users' needs and expectations as the survey results may be unexpected to the development team.

The results of our findings show these services are very useful in e-learning, especially when it comes to foreign languages. In the future we plan to develop our service for the latest technologies like virtual reality or other interactive approaches.

The use of API allows us to add new applications and replace the server application, Android application or web applications with different ones as long as common communication schema established in API is preserved thus allowing for easy migration of the individual applications.

## References


[1] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):281–308, 2004.

[2] R. Damaševičius, C. Napoli, T. Sidekerskienė, and M. Woźniak. Imf mode demixing in emd for jitter analysis. *Journal of Computational Science*, 22:240–252, 2017.

[3] D. Fensel and C. Bussler. The web service modeling framework wsmf. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[4] H. Foster, S. Uchitel, J. Magee, and J. Kramer. Model-based verification of web service compositions. In *Automated Software Engineering, 2003. Proceedings. 18th IEEE International Conference on*, pages 152–161. IEEE, 2003.

[5] T. Kapuściński, R. K. Nowicki, and C. Napoli. Comparison of effectiveness of multi-objective genetic algorithms in optimization of invertible s-boxes. In *International Conference on Artificial Intelligence and Soft Computing*, pages 466–476. Springer, 2017.

[6] M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 915–922. ACM, 2006.

[7] Y. Kun, W. Xiao-Ling, and Z. Ao-Ying. Underlying techniques for web services: A survey. In *Journal of software*. Citeseer, 2004.

[8] Y. Liu, A. H. Ngu, and L. Z. Zeng. Qos computation and policing in dynamic web service selection. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73. ACM, 2004.

[9] A. A. Patil, S. A. Oundhakar, A. P. Sheth, and K. Verma. Meteor-s web service annotation framework. In *Proceedings of the 13th international conference on World Wide Web*, pages 553–562. ACM, 2004.

[10] J. Rao and X. Su. A survey of automated web service composition methods. In *International Workshop on Semantic Web Services and Web Process Composition*, pages 43–54. Springer, 2004.

[11] D. Roman, U. Keller, H. Lausen, J. De Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied ontology*, 1(1):77–106, 2005.

[12] M. J. Rosenberg and R. Foshay. E-learning: Strategies for delivering knowledge in the digital age. *Performance Improvement*, 41(5):50–51, 2002.

[13] A. Venckauskas, A. Karpavicius, R. Damaševičius, R. Marcinkevičius, J. Kapočiūte-Dzikiené, and C. Napoli. Open class authorship attribution of lithuanian internet comments using one-class classifier. In *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on*, pages 373–382. IEEE, 2017.

[14] E. T. Welsh, C. R. Wanberg, K. G. Brown, and M. J. Simmering. E-learning: emerging uses, empirical results and future directions. *international Journal of Training and Development*, 7(4):245–258, 2003.

[15] M. Wróbel, J. T. Starczewski, and C. Napoli. Handwriting recognition with extraction of letter fragments. In *International Conference on Artificial Intelligence and Soft Computing*, pages 183–192. Springer, 2017.