# NihonGo: A Computer-Aided Learning Package of Japanese Language for Beginner Level

Boon Why Lee
Siew Hock Ow
Seow Ping Quek
Siti Nur Hidayah
Soon Sang Chong
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
MALAYSIA
Email: bwlee@fsktm.um.edu.my
Email: show@um.edu.my

## Abstract

*This paper describes the development of a computer-aided learning package, NihonGo that teaches Japanese Language in an easy way. It comprises eight main modules – Lessons, Exercises, Quizzes, Writing, Games, Dictionary, Karaoke and Maintenance. It is a bilingual system that allows the users a choice of either English or Malay language for the instructions. NihonGo was developed using the incremental evolution technique through rapid prototyping. The development tools used include VB.Net, Microsoft Access and Adobe Photoshop CS2. Adobe Photoshop CS2 is used to design the system's graphical user interface. VB,NET connects the application with the Microsoft Access database. The eight modules were first developed and tested individually before they were integrated and tested to form a complete system. User Acceptance Test (UAT) was conducted on the NihonGo.*

## 1. Introduction

Students who wish to pursue further study in Japan are required to master the Japanese language. As Japanese language have three types of writings: hiragana, katakana and kanji, most learners find it difficult to master this language [1].There are many Japanese learning packages in the market. These packages have a number of limitations. For example, the Declan's ReadWrite Hiragana only teaches pronunciations, writings, listening and readings of Hiragana words [2]. It does not teach the Japanese grammar usage. Therefore, the graphics in the Kana-Kun learning package provides unclear sequence of strokes of writing a Japanese character [3].

## 2. The Functionalities of NihonGo

To develop a good learning package to master a foreign language, it is important to provide writing, grammar and vocabulary functionalities. Hence, NihonGo has eight modules that meet these three important issues. These modules are described in detail below.

*Table 1: Review on Japanese Learning Packages*

| Software | Advantages | Limitations |
|---|---|---|
| Declan's ReadWrite Hiragana | 1. Animates Japanese writing; stroke-by-stroke .<br>2. Provides meanings of the Japanese words. | 1. Focusing on read and write only.<br>2. Very few lessons on grammar and particle.<br>3. No dictionary. |
| Kana-kun | 1. Use graphics to illustrate meanings of Japanese words.<br>2. Easy to use. | 1. Unattractive user interface.<br>2. Very few lessons on vocabulary and grammar.<br>3. Has one game only.<br>4. No dictionary. |
| Java Kanji Flashcards 500 [4] | 1. Has 500 kanji characters with stroke order animation.<br>2. Provides exercises.<br>3. Provides Kanji word search function. | 1. No pronunciation.<br>2. No exercises evaluation. |

### a. Maintenance

This module allows authorised users to add, edit and delete lessons, exercises, quizzes and vocabulary which are stored in the database.

### b. Writing

The Writing module enables learners to learn and write Japanese characters in Hiragana, Katakana and Kanji. The correct sequence of strokes of writing a Japanese character is illustrated using animated graphics.

### c. Lessons

Learners can either choose to have instructions in English or Malay language to learn Japanese language. NihonGo also has a text-to-speech synthesiser to teach Japanese pronunciation.

### d. Exercises

In Exercises module, learners can choose to do an exercise based on the lesson that they have studied. There are four types of questions: objective questions, multiple choice questions (MCQs), fill in the blanks with on-screen keyboard, and true or false questions. Each exercise has 10 questions. The questions are generated randomly. No marks are given in the Exercises module.

### e. Quizzes

Learners can choose a quiz to answer based on the lesson that they have studied. There are four types of questions which are same as with the Exercises module. Time allocated to answer Chapters 1 to 10, and Chapters 11 to 20 is 10 minutes and 15 minutes, respectively. Each quiz has 10 questions, which are generated randomly. When the time to answer is over, the quiz session will be terminated automatically. Marks will be given and the quiz result will be saved into the database. Only the top 10 results would be shown in the performance report.

### f. Games

Simple and interesting games create fun in learning. Two games, namely the Flash Card and the Boulders games are provided to help learners to memorise the Hiragana and Katakana writings, and Japanese vocabulary, respectively.

### g. Karaoke

To make learning interesting, the Karaoke module is provided to allow learners to sing in Japanese language. This can help learners to speak Japanese language fluently. Lyrics and their translations in English help learners to understand the Japanese song. Learners can thus learn and enjoy singing at the same time.

### h. Dictionary

In helping learners to understand Japanese words, the Dictionary module is incorporated to allow them to find the meanings of the Japanese words. Learners can search the meanings of a Japanese word either by key-in the romanised or hyperlink of the Japanese word.

## 3. The Functional and Non-Functional Requirements of NihonGo

Use case diagrams and use case narratives are used to describe the functional requirements of NihonGo. NihonGo has two groups of user – Administrator and Learner. Two use case diagrams were constructed to illustrate the functions that they are allowed to perform.

Basically, an Administrator can perform the functions: add, edit and delete lessons, exercises, quizzes and words in the dictionary. The functions that learners are allowed to perform include login; select language; view lessons; learn writings; do exercises; do quizzes; play games; play karaoke and view reports. An Administrator can perform all these functions and all the maintenance functions – add, edit and delete lessons, exercises, quizzes and words in a dictionary. Fig. 1 and Fig. 2 show the use case diagrams that illustrate the functions performed by the two groups of users.

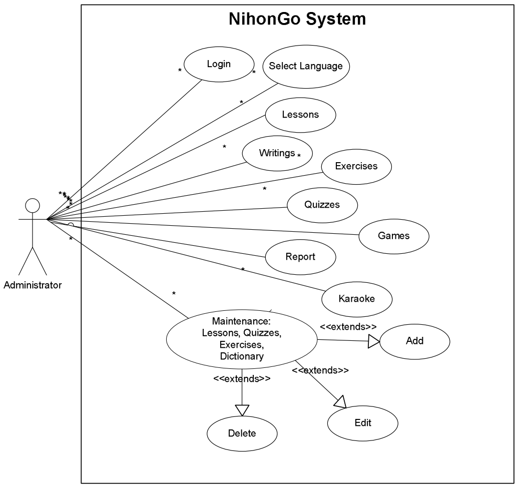


*Fig 1. Use case diagram for Administrator*

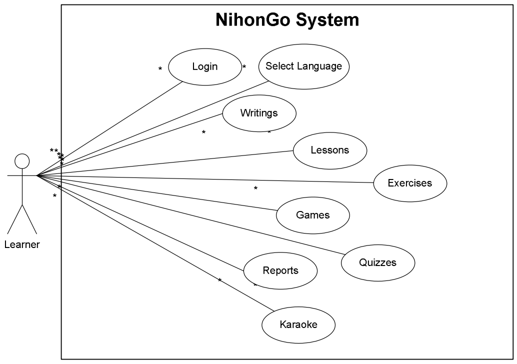


*Fig 2. Use case diagram for Learner*

Non-functional requirements present a systematic and pragmatic approach to "building quality into" software systems [5]. These specify the criteria that can be used to judge the operation of a system, rather than specific behaviours [6]. NihonGo was designed to fulfill the following non-functional requirements: correctness, user-friendliness, performance, security and understandability.

**a. Correctness**
The translations from Japanese into English language for all lessons and instructions were verified by the project supervisor who is proficient in English and Japanese language. This aims to achieve content correctness.

**b. User-friendliness**
In order to make the system easy to use, it is important to make the user interface as friendly, attractive, and as interactive as possible. These GUI issues are achieved through buttons, labels, messages, tooltips and links that are short, simple and clear. Simple English words are used so that all learners can understand and use NihonGo without difficulty.

**c. Performance**
In term of performance, it is important for a system to respond within a reasonable time to the user. In the Quizzes module, it takes less than 5 seconds to generate the results after a learner submitted his quiz answers. All processing in the eight modules were tested for their performance and found to respond within 10 seconds.

**d. Security**
In NihonGo, only authorised users are allowed to access to the modules that are relevant to them only. Also, every user is required to login using a password that has been encrypted using MD5 algorithm [7].

**e. Understandability**
NihonGo's program codes were written in modular format. Good programming practices such as standard naming convention and specifying program purpose at the beginning of program codes allow other programmers to understand the variables, program codes and coding method easily. Comments and indentation were included to help programmers to understand the code and its flow clearly.

## 3. Design of NihonGo

This section discusses the design of NihonGo which include the architectural, database and user interface designs.

### *3.1 Architectural Design*

The architectural design is a pictorial representation of the subsystem framework control and communication. It decomposes a system into several subsystems with its respective functionality [7]. Fig. 3 shows the architectural design of NihonGo.

The users (Administrators and learners) interact with the user interface to use the eight modules: Lessons, Writings, Exercises, Quizzes, Games, Karaoke, Dictionary and Maintenance based on the user's access level. The NihonGo system accesses directly to the database via ActiveX Data Object (ADODB) to get the data to carry out the process perform by of each module.

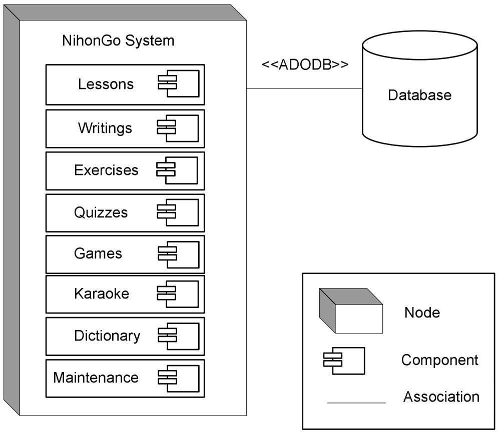


*Fig 3. Architectural Design of NihonGo*

### *3.2 Database Design*

There are two processes involved in developing the database of NihonGo.

- Creating the database

There are 11 main tables in the database – User, Administrator, Lessons, Hiragana Writing, Katakana Writing, Kanji Writing, Exercises, Quizzes, Games, Karaoke and Dictionary.

User – stores the user's name and password.
Administrator – stores the Administrator's name and password.
Lessons – stores the Japanese language notes and grammars.
Hiragana Writing – stores the Japanese Hiragana characters.
Katakana Writing – stores the Japanese Katakana characters.
Kanji Writing – stores the Japanese Kanji characters.
Exercise – stores the exercises and answers based on the lessons.
Quiz – stores the quizzes and answers based on the lessons.
Games – stores the games based on unique identification number.
Karaoke –stores the Japanese songs in AVI or WMV format.
Dictionary – stores the Japanese words together with translation in English.

Fig. 4 shows the logical database design of NihonGo. Two sub-tables, Quiz Score and Game Score are linked to the Quiz and Games tables, respectively. These two sub-tables store the learner's quiz result and the game score of the FlashCard game.

- Entering Data

The Administrator can enter data into NihonGo database using the *Maintenance* function.

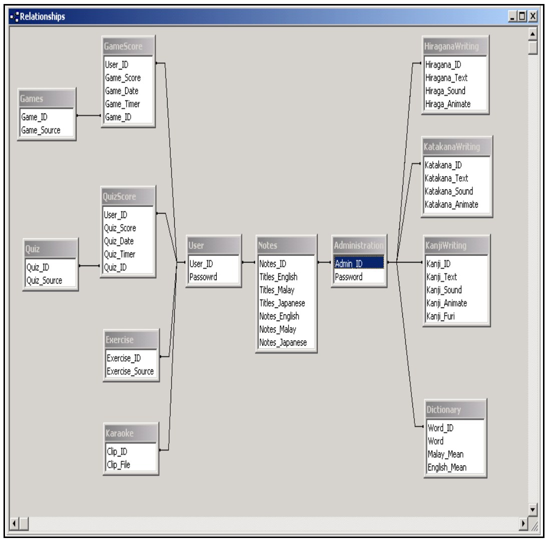


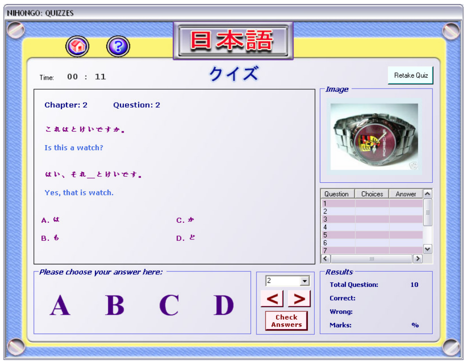
*Fig 4. Logical Database Design of NihonGo*

### *3.3 Graphical User Interface (GUI) Design*

One of the important aspects in software development is the user interface as it describes how the software interacts with the user.

NihonGo was designed to be easy to use with clear guidance, user-friendly data input interface and attractive animated screen designs. The design emphasises on the consistency in screen design, and interaction between the user and the system

As shown in, Fig. 5 and Fig. 6, the "Home" and "Help" buttons are consistently placed at the top left corner of the screen for easy navigation. Fig. 7 illustrates the FlashCard game of the Games Module. Simple and clear labels enable learners to play the game with fun and easily.



Time: 00 : 11

Chapter: 2 Question: 2

これはとけいですか。

Is this a watch?

はい、それ＿とけいです。

Yes, that is watch.

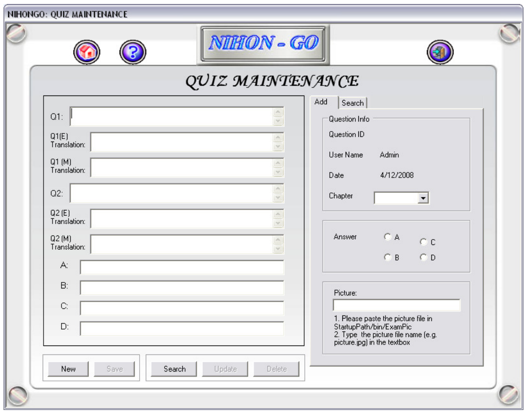
A. は
C. か
B. も
D. と

| Question | Choices | Answer |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |

2

Total Question: 10
Correct:
Wrong:
Marks: %

*Fig 5. GUI of Quizzes Module*



Q1:
Q1(E) Translation:
Q1 (M) Translation:
Q2:
Q2 (E) Translation:
Q2 (M) Translation:
A:
B:
C:
D:

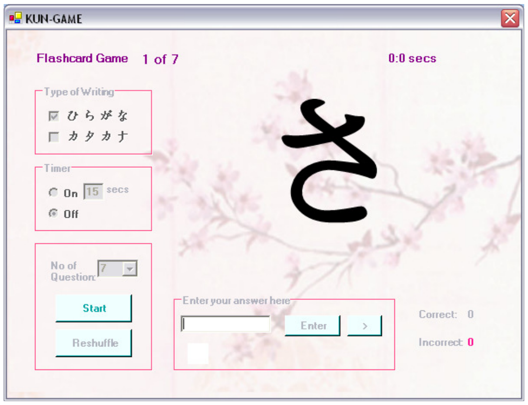
Question Info
Question ID
User Name Admin
Date 4/12/2008
Chapter

Answer ○ A ○ C ○ B ○ D

Picture:

1. Please paste the picture file in StartupPath/bin/ExamPic
2. Type the picture file name (e.g. picture.jpg) in the textbox

*Fig 6. GUI of Maintenance Module*



Flashcard Game 1 of 7 0:0 secs

Type of Writing
☑ ひらがな
□ カタカナ

Timer
○ On 15 secs
● Off

No of Question: 7

Enter your answer here

Correct: 0
Incorrect: 0

*Fig 7. GUI of Games Module*

## 4. Development of NihonGo

This section discusses the development tools used, program design and system testing.

### *4.1 Development tools used*

Table 2 shows the development tools used to develop NihonGo.

The NihonGo interfaces and modules were developed using the VB.NET programming language [9]. VB.NET provides a drag and drop function that allows users to draw the components (i.e. textbox, combo box and image) directly. In addition, the buttons, textboxes and other objects that are dragged into the window form will automatically recognise user actions such as mouse movements and button clicks. Thus, it is easy to design the interfaces and write the source codes for the system.

MS Access 2000 was used to develop the database for the NihonGo whereas the MS Japanese Language Text-to-Speech pronounces the Japanese words.

*Table 2: Development tools Used*

| Software | Application | Description |
|---|---|---|
| Microsoft® Windows XP | System Requirement | Operating system. |
| Microsoft® Access 2000 | Database | Builds tables, queries and forms to store and manipulate the data. |
| Microsoft® Visual Basic.NET (VB.NET) | Programming language for system development | System Coding. |
| Macromedia Dreamweaver MX 2004 | Create HTML | Generates web pages for lessons. |
| Adobe Photoshop CS | Interface and Contents Design | Creates attractive graphic components such as buttons, labels, application background and images. |
| Microsoft® Word 2003 | Documentation | System documentation and preparation of user manual. |
| Microsoft® Japanese Language Text-to-Speech | Convert the Japanese text to speech | Provide the pronunciation of the Japanese words. |

*4.2 Program design*

A standardised, uniform and organised coding style was used to create maintainable codes. It also makes the codes easy to read and understand.

Fig. 8 shows the program codes to search a Japanese word in the dictionary. The variables "con1" and "sql" are used to open a database connection and Sequel statement, respectively. Naming convention such as "cbxLang" indicates that this variable belongs to ComboBox in VB.NET. Comments (as shown in green colour) and indentation are used throughout coding to ease program readability and understand ability [8].

```
Dim con1 As New clsConn
con1.OpenConnect() ' open database connection
Dim sql As String
'select a Japanese word in the Dictionary table
If cbxLang.SelectedItem = "English" Then
    sql = "SELECT * FROM Dictionary WHERE
     TranslationE = '" & temp & "'"
ElseIf cbxLang.SelectedItem = "Japanese" Or
cbxLang.SelectedItem = "Jepun" Then
   If rbtHiraKata.Checked = True Then
     sql = "SELECT * FROM Dictionary WHERE
     Japanese = '" & temp & "'"
   ElseIf rbtKanji.Checked = True Then
     sql = "SELECT * FROM Dictionary WHERE
     Kanji = '" & temp & "'"
   End If
ElseIf cbxLang.SelectedItem = "Romanji" Then
     sql = "SELECT * FROM Dictionary WHERE
Romanji
     = '" & temp & "'"
End If
rs = New ADODB.Recordset
'open recordset
rs.Open(sql, conn, 1, 2)
```

*Fig 8: Program Codes for Searching a Japanese Word in the Dictionary*

As NihonGo was developed by a team of programmers, the good program design practices allow each team member to read and understand the codes easily. These also ease the debugging process when errors were detected in the system.

*4.3 System testing*

After the source codes were written, they were reviewed and verified to detect logic errors. Each module was tested with valid and invalid input data. Two types of testing were performed on NihonGo.

*a. White Box Testing*

White box testing evaluates the internal mechanism of a system or component [8, 10]. At the beginning of white box testing, the program codes were examined line by line to ensure that the algorithms used in the program are correct. For example, the processing paths and algorithms of the Lessons module to perform the chapter selection and TTS functionality were executed and tested thoroughly. Errors and deviations from coding standards were discovered and corrected.

*b. Black Box Testing*

In Black box testing, the testing focuses solely on the outputs generated in response to selected inputs and execution conditions [8, 10]. For example, in the testing of the Quizzes module, each answer was selected to test that only one correct answer is assigned to a question. The test results were compared to determine the test cases, generated results that matched with the expected results. Other test cases were developed to show that the inputs are correctly converted to the desired output.

During integration testing, the bottom-up test approach was adopted. All the lower level modules were coded and tested before integrating upwards.

Fig. 9 illustrates the unit and integration testing of NihonGo. For example, in Maintenance module, add, edit and delete quiz lessons were tested independently.

In integration testing, the next higher level modules were then integrated and tested with the modules that have already been tested independently. This step was repeated until all the integrated modules were tested.

The NihonGo system was tested thoroughly when all the modules were integrated. These testing took three days to complete as the testing covered all the eight main modules.

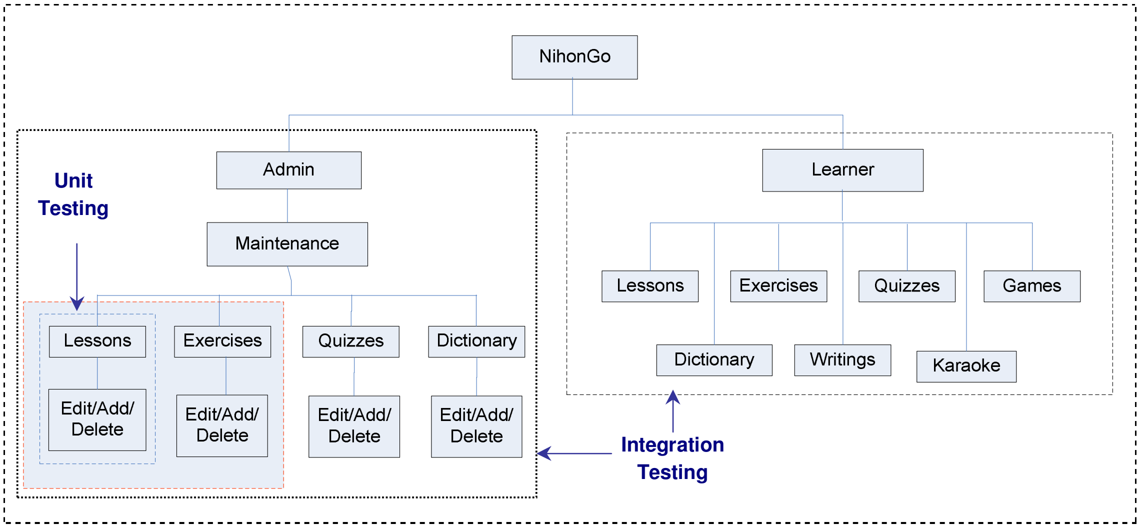


*Fig 9. Unit testing and integration testing of NihonGo*

### *c. User Acceptance Test*

User Acceptance Testing (UAT) is one of the important test activities in software development. It requires the participation of the end-users. An acceptance test plan was developed to guide the testing process. In order to conduct this testing, 26 users were selected at random to participate in the test. These participants were first given hands-on session to learn Japanese language using the system. They were then each given a set of questionnaire to answer to indicate their level of satisfaction on NihonGo. Fig. 10 and Fig. 11 show the user acceptance test results.

Fig. 10 shows that 4 (15.4%) and 14 (53.8%) students, totally agree and agree that Nihongo has sufficient contents covered for the Lessons Module, respectively. However, 7 (26.9%) students are not sure if the contents are sufficient to master the basic level of Japanese language. This could possibly be due to the fact that they were not given enough time to use NihonGo. Only 1 (3.9%) student totally disagrees with the other students' opinion.

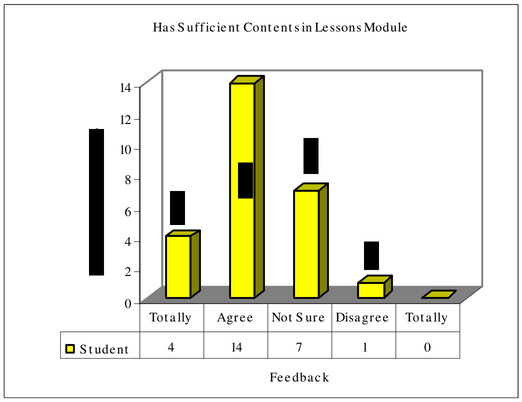


*Fig 10. Feedback on NihonGo - has sufficient contents in Lessons Module*

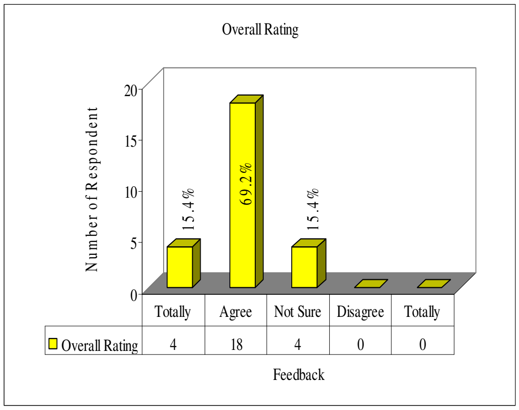


*Fig 11. Evaluation on NihonGo – Overall Rating*

Fig. 11 shows that 4 (15.4%) and 18 (69.2%) students, totally agree and agree that they are satisfied with the functions and features provided in NihonGo, respectively. Only 4 (15.4%) students indicated that they are not sure if NihonGo can help them to learn Japanese language. Overall, 22 (84.6%) of the students agree that NihonGo can assist them in the learning of Japanese language in an easy, fun and interesting way.

## 5. Discussion and Conclusion

NihonGo has a few system strengths that make it useful in the teaching and learning of Japanese language. These strengths include:

*a. Simple and User-friendly Interfaces*
NihonGo provides simple and consistent interfaces. Users can easily get familiar with the system. The buttons, labels, messages, tooltips and links are simple and clear. Hence, learners can use NihonGo easily without any difficulty.

*b. Text-to-Speech (TTS)*
TTS converts language text into speech that resembles human voice. Using the Microsoft Japanese Text-to-Speech Synthesiser, learners can learn the pronunciation of the Japanese words. This saves time and cost without having to pre-record the pronunciation of Japanese words and English translation when changes were made to the Japanese words in the Lessons and Writing modules.

*c. Interactive Learning through Games and Karaoke*
In NihonGo, the FlashCard game helps the learners to memorise the Hiragana and Katakana. On the other hand, the Boulder game helps learners to enrich his vocabulary. In Karaoke, learners can learn and enjoy singing the Japanese songs. The lyrics are in Japanese language and the translations are in English language.

*d. Maintenance*
Administrators are allowed to add, edit, and delete lessons, exercises, quizzes and Japanese words in the dictionary. This function allows the administrator to perform maintenance whereby this functionality is not available in other Japanese learning packages.

*e. Bilingual*
NihonGo has instructions in English language. Users can change the language anytime when they use the system. Most of the Japanese learning packages available in the market have instructions in English only.

*f. Dictionary*
Learners can search the meanings of a Japanese word from the dictionary. The meanings are explained in bilingual.

Besides the strengths mentioned above, NihonGo also has a few weaknesses. Some of the weaknesses and limitations identified include:

*a. Games*
There are only two games, the FlashCard and the Boulder games. To make learning interesting and fun, games such as Scrabble and Text Twister can be added into NihonGo. These two games would allow learners to improve their Japanese vocabulary.

*b. Lessons*
Currently, the Lessons module has twenty chapters only. An administrator can add more lessons through the Maintenance module to cover more topics for higher learning level.

*d. Graphical Illustration*
In Lessons module, 2D pictures are used to explain the lessons. One enhancement that could be made is to use 3D graphics as it provides better and clear illustration.

Although NihonGo has a few weaknesses and limitations, the functionalities that it provides has fulfilled the needs of beginners in learning Japanese language. It is an interesting learning package that allows users to learn Japanese language with ease, fun, and at their own pace at home.

## References

[1] Vitek, S. V. “Linguistic Problems with Translation of Japanese Patents to English.” Retrieved March 25, 2008, from http”//www.accurapid.comjournal40patents.htm

[2] Declan’s ReadWrite Hiragana Software: Japanese, 2004. Retrieved September 18, 2004, from http://www.declan-software.com/japanese

[3] Kana-Kun, 1994.
Retrieved September 18, 2004, from
http://www.eduthink.com/kanakun.htm

[4] Java Kanji Flashcards 500.
Retrieved September 20, 2004, from
http://www.nuthatch.com/kanjicards

[5] Chung, L., Nixon, B. A., Yu, E. and Mylopoulos, J. “Non-functional Requirements in Software Engineering”. Retrieved April 1, 2008, from: http://www.utdallas.edu/~chung/BOOK/book.html

[6] Pressman, R. S., 2005 “Software Engineering: A Practitioner’s Approach,” New York, McGraw-Hill Education.

[7] “An MD5 Compatible with PHP”, 2005.
Retrieved 18 August 2005, from
http://www.codeproject.com/KB/security/MD5FunctionPHP.aspx

[8] Schach, S. R., 2005. “Object-oriented & Classical Software Engineering,” Sixth Edition, Boston, MA, McGraw-Hill.

[9] Schneider, D. I., 2003. “*An Introduction to Programming Using Visual Basic.NET*,” New Jersey, Prentice Hall, Inc.

[10] Rajani, R. and Oak, P., 2004.” *Software Testing: Effective Methods, Tools and Techniques*,” New Delhi, Tata McGraw-Hill Publishing Company Limited.

Copyright © 2008 by the International Business Information Management Association (IBIMA). All rights reserved. Authors retain copyright for their manuscripts and provide this journal with a publication permission agreement as a part of IBIMA copyright agreement. IBIMA may not necessarily agree with the content of the manuscript. The content and proofreading of this manuscript as well as and any errors are the sole responsibility of its author(s). No part or all of this work should be copied or reproduced in digital, hard, or any other format for commercial use without written permission. To purchase reprints of this article please e-mail: admin@ibima.org.