e JAVA Chatbot for Learning Programming

Article · September 2020

CITATIONS
0

READS
152

1 author:

Noor Hasimah Ibrahim Teo
Universiti Teknologi Mara (Melaka)
11 PUBLICATIONS **19** CITATIONS

SEE PROFILE

ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter67872020.pdf
https://doi.org/10.30534/ijeter/2020/67872020

e-JAVA Chatbot for Learning Programming Language: A Post-Pandemic Alternative Virtual Tutor

Siti Hawa Mad Daud[1], Noor Hasimah Ibrahim Teo[2], Nurul Hidayah Mat Zain[3]
[1]Universiti Teknologi MARA, Malaysia, sitihawamaddaud08@gmail.com
[2]Universiti Teknologi MARA, Malaysia, shimateo@uitm.edu.my
[3]Universiti Teknologi MARA, Malaysia, nurul417@uitm.edu.my

ABSTRACT

A chatbot is a tool that provides online communication platform with an agent and is usually used by organizations as a customer service agent to promote their products and deliver help services on their web, apps or instant messaging platforms like Telegram and WhatsApp. It also has been increasingly used in education to assist students as a virtual tutor in different subject areas. However, there are many questions about the potential and the limits of the use of chatbots for learning a programming language. Nowadays, as education is shifting towards Education 4.0, its system must adapt to new circumstances and changes in roles. The use of chatbot as a substitution in the process of learning may act as a virtual learning tutor to fulfil the need of education towards Education 4.0. This project exploits a rule-based technique, to generate a solution for finding a suitable control structure for a given computational problem. While text matching is deployed to automatically give instant responses to the users based on template-based questions such as greetings and general theoretical questions. The initial purpose of this project is to develop a virtual tutoring tool that provides support to students on JAVA programming language problems. The result shows there is a significant output produced by the e-JAVA chatbot.

Key words: Chatbot, COVID-19 pandemic, JAVA programming, learning, question generation, virtual tutor.

1. INTRODUCTION

Programming language is a vital for developing software applications and to solve various computer problems. It is an essential subject for computer science courses, and it is crucial for computer science students to master the fundamental theory on how to use programming to solve computer problems. Many students experience difficulties in learning a programming language and to apply programming on computer problems. Nowadays, programming languages are introduced as early as in pre-school level to overcome students' assumption of difficulty in learning computer programming[1]. There are many programming languages designs with some of them used for specific purpose like, for logic manipulation, system software, web application, mobile applications and many more.

The conventional face-to-face teaching techniques have been proven to be an optimum way to teach programming language. Students are equipped not only through lectures but during practical sessions in the lab as well. However, not all students are able to catch up with their lessons due to the traditional education teaching approaches where students must learn according to lecturers' teaching style while students have different learning styles and ways of approaching new materials [2][3]. In addition, students and lecturers also find there is time constraint to have one-to-one sessions or small group discussions due to increasing number of students and job scopes of lecturers that not only focus on teaching but also research work. It would be helpful for any students to learn under close supervision and to get immediate response or feedback on the given problems and explanations of challenging topics [4][5].

Therefore, e-JAVA chatbot is proposed in order to support students in learning JAVA programming. A chatbot could provide flexibility and easy to access learning for everyone [6]. This research is guided by two research objectives, 1) to identify if there is a need for support for learning the programming language, and 2) to develop e-JAVA chatbot for learning the JAVA programming language.

The next sections will discuss how chatbot relieves pressure during COVID-19 pandemic, requirement gathering, a preliminary survey on needs to have support for learning and system development and design.

2. HOW CHATBOT RELIEVES PRESSURE DURING COVID-19 PANDEMIC

Chatbots are programs that interact with people through voice or text in their natural languages [7][8]. The existence of well-known chatbots are Cortana produced by Microsoft, Siri

from Apple, and Alexa from Amazon. Some of the tasks of Chatbots are to assist us to get information quickly, support health that impacts behaviour and reduce the psychological damage triggered by fear [9]. Many domains have utilized the benefit of this new trend. The following subsection discusses three sectors that implement chatbot as pressure reliever during Covid-19 pandemic.

2.1 Health sector

Throughout Covid-19 pandemic, the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) have initiated applying chatbots to share info, advise behaviours, and offer emotional supports [9]. During Covid-19 pandemic, reliable information sources are vital. Chatbots provide concise information from trustworthy sources. This situation is essential because fake news spread online faster than truthful news [9].

The features that set in chatbots give advantages in this pandemic situation where it can respond in natural language, enhance accessibility to people who have difficulties in reading and using the internet. People may access chatbots anytime to ask a question regarding the latest information. Moreover, a chatbot can implement tasks such as symptom screening in Covid-19 pandemic because people with stigmatized conditions always avoid going to the hospital [10].

2.2 Tourism sector

Covid-19 pandemic gives significant impacts on tourism sector and the full impact on global tourism which is currently still not completely determined. Automation technology, such as chatbot, is a useful tool to enforce social distancing during a pandemic. Through chatbot, physical human contact is eliminated thus reduces infection [11]. In the case of infected guests in a particular hotel need to be quarantined, the chatbot can be used as a medium to interact with the hotel's services such as the hotel room is installed with voice-activated technologies [12]. This scenario can maintain an effective communication between the guests and the company [13].

Besides, most hotels have offered self-check-in registration for guests [14], [15]. The decision to use chatbot is important not only to restructure its operations, cut costs, and be more price competitive but also to implement social distancing as well as avoidance of handshakes during Covid-19 pandemic.

2.3 Business and marketing sector

Covid-19 has affected every person in daily life. One of the impacts is that many business operations have shifted to virtual environments. Therefore, business managers will provide an interactive online shopping experience [16]. As an example, the *Whole Food* business offers a chatbot that assists its customers options to choose ingredients to shop from the store. When a customer is in urgency, they can send a food Emoji [17]. The integration of chatbots in online businesses can be implemented using private messaging apps, personalized offers and deliver interactive contents to the customers [18]. In addition, chatbot also help businesses marketing in economical way and save time and effort for resources. A chatbot has been used for digital marketing, sales and advertising businesses in [19] to increase revenue and brand visibility.

3. e-JAVA CHATBOT: REQUIREMENT GATHERING

This section discusses the literature of related works and components in e-JAVA chatbot development which includes an existing chatbot, control structures in programming, and Artificial Markup Language (AIML) for chatbot development.

3.1 Chatbot

A chatbot can be classified as an open domain chatbot which can respond appropriately with a general topic and closed domain chatbot that focuses on a specific field of knowledge. A chatbot also can be categorized based on purposes, for instance, personal assistant, entertainment, simplification, and single-function chatbot. Table 1 shows some existing chatbots that provide different features.

Table 1: Existing chatbots and their functionality

Feature	Chatbot	Functionality
Personal assistant	SIRI[20]	Uses voice queries to help users of iPhones.
	CLIPPY[21]	Automated customer service for businesses developed by Microsoft.
	[1]Skycivic COVID-19	Is a guiding tool to assess individual risk based on Covid-19 symptoms
Proactive	[1]COVID-19 cases Tracker	Helps to track the number of reported cases in your country/state/city or region.
Single function	[1]COVID-19 FAQ	Used to increase awareness about the pandemic by healthcare organizations or other businesses
Simplification	Duolingo[22]	Helps users to learn a new language includes French, Spanish and German.
	[2]DoNotPay	Helps users to deal with simple legal problems
Entertainment	[3]ALICE	A program that engages in a conversation with a human on a general topic.

There is massive development of chatbots in the web, but the focus of this research is in education sector to support student learning especially in the post-pandemic phase. This project developed a chatbot for learning JAVA programming language which is the core subject of Computer Science students. This chatbot adapts features of simplification to help

[1] https://hellotars.com/chatbot-templates/healthcare/
[2] https://donotpay.com/
[3] http://www.alicebot.org/anatomy.html

students understand specific knowledge in JAVA programming language on control structure topic. The next section discusses further on the different forms of control structures in programming language implemented in this project.

3.2 Control structure in programming language

There are three ways a program can be processed, which is in sequence, selection, and repetition. Statements are executed based on certain conditions are called selection, which consists of if statement, if-else statement, nested if statement, and switch statement. The program keeps repeating the statement in a certain number of times based on certain conditions known as repetition. Repetition consists of several conditions which are for loop, while loop and do-while loop. The following discusses different types of selection and repetition control structure.

A. An "if" statement

The "if" statement is used in one-way selection in a programming language. A statement can be a single or compound statement, followed by a parenthesis. Figure 1 illustrates the if-statement execution flow. A statement is executed if the logical expression is true.

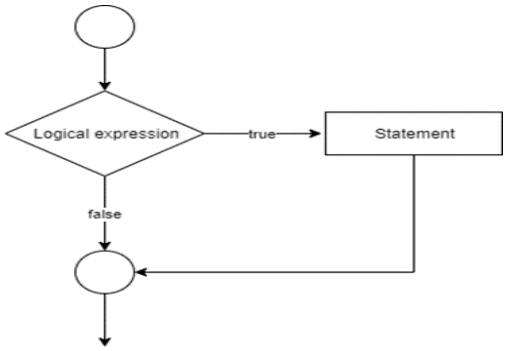


Figure 1: A one-way "if" statement

B. An "if-else" statement

The "if-else" statement is a two-way selection that provides two alternation options for a solution. If the value of a logical expression is true, then Statement 1 executes and if it is false then Statement 2 executes. Figure 2 illustrates the execution flow of two-way selection.

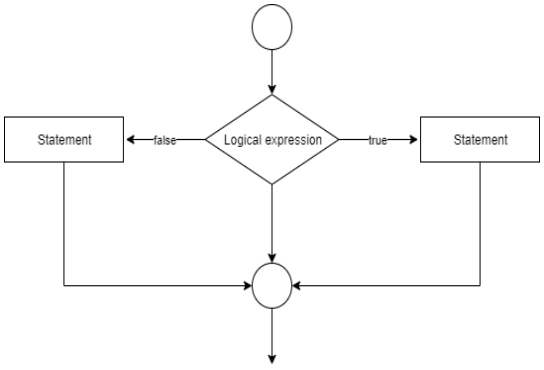


Figure 2: A two-way "if" statement

C. Switch statements

Switch statements do not require the evaluation of a logical statement. The switch structure of JAVA allows the computer to choose from many cases or alternatives. Figure 3 shows the switch case execution flow. Control is passed to the statement or switch, while the break statement is used to stop the execution of a statement in the case.

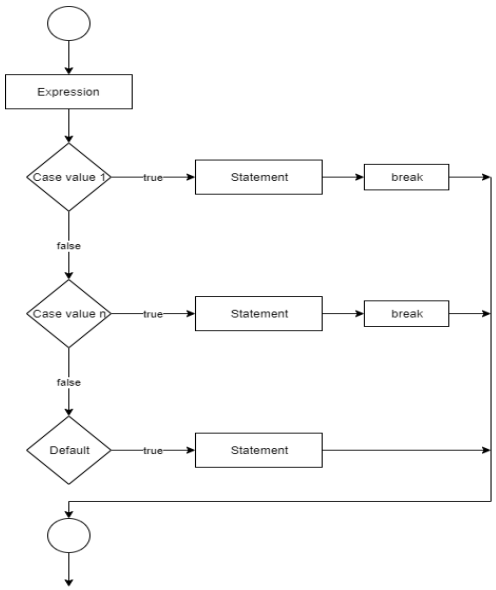


Figure 3: A switch-case selection

D. A for loop

A for loop is a control flow statement for specifying iterations, to enable codes to be executed repeatedly. The structure of the loop statement consists of three parts namely initializations, conditions, and increment or decrement of the loop. Figure 4 shows the for-statement execution flow.

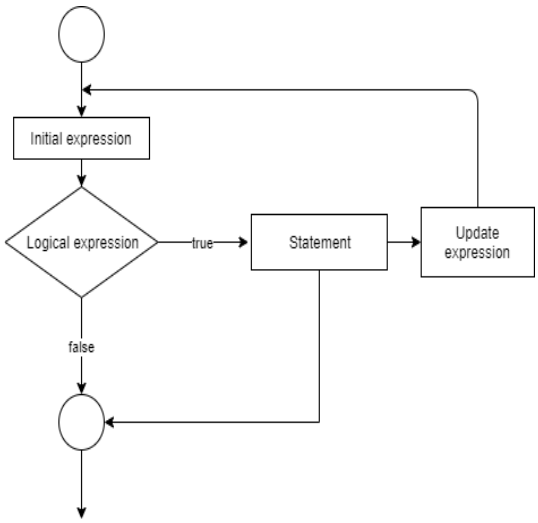


Figure 4: A for loop execution flow

E. A while loop

A while loop is a control structure that allows a target statement to execute repeatedly until the condition is met. The statement can be a single statement or a block of a statement. Figure 5 shows the flow execution of the while loop.

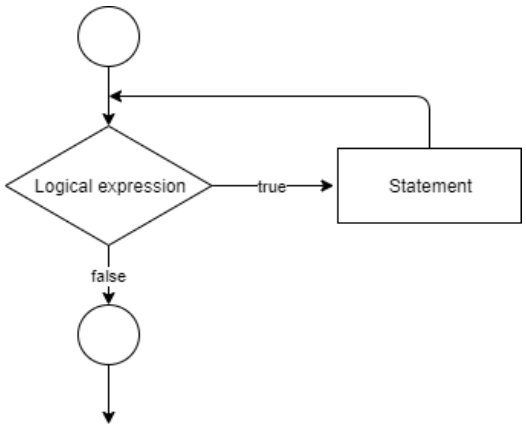


Figure 5: A while-loop flow execution

F. A do-while loop

A do-while is a control flow that executes a block of codes at least once before it checks the condition. The difference with while the statement is, the conditional expression is tested after the execution of the loop, rather than before the executions, which is at least one execution of do-while is tested. Figure 6 illustrates the execution flow of the do-while statement.

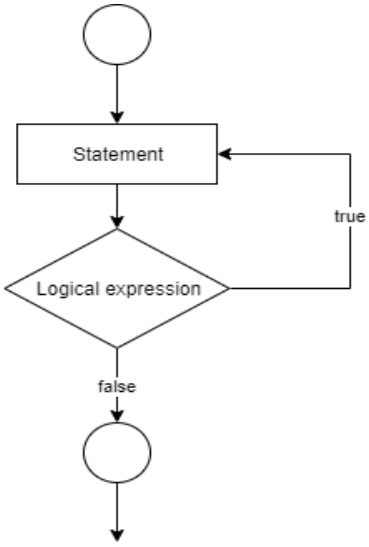


Figure 6: A do-while execution flow

The next section discusses on the markup language used in the development of chatbots.

3.4 Artificial Markup Language (AIML)

AIML is used for pattern matching and to compare user's input with a response of the chatbot Knowledge Base. It is now commonly used in software development, which uses natural languages in communication with users. Category, pattern, and template are the most important objects among the AIML objects. Category tag works to define the conversation knowledge unit, known as categories. There are two categories: templates and patterns. Pattern tag identifies the input and template tags used to react to the user input. The basic forms of AIML that consist of category, pattern, and template tag. Artificial markup language consists of three different types of categories: default, recurrent and atomic category [18].

Atomic categories: Patterns without wildcard (*) symbols.

```
<category>
<pattern>HI THERE</pattern>
<template>Hello! What's your name?</template>
</category>
```

Default categories: The pattern of words consists of wildcard symbols (* or _). If it does not find an atomic pattern in the previous input, the preceding input '10 Dollars' is assumed, then it will try to find the default pattern in a category.

```
<category>
     <pattern>10 *</pattern>
     <template>It is ten.</template>
</category>
```

Recursive categories: Contains symbolic reduction and artificial recursive intelligence tag, which is referred to <srai> and <sr> tags. This type of categories offers several applications including symbolic reduction, reducing complicated grammar to simple, or dividing and conquering inputs into different sub-sets.

```
<category>
     <pattern>HIYA</pattern>
     <template><srai>Hello</srai></template>
</category>
```

4. PRELIMINERY SURVEY ON THE NEED OF CHATBOT FOR LEARNING PROGRAMMING LANGUAGE

The survey was conducted to 43 students from the Faculty Computer and Mathematical Sciences at Universiti Teknologi MARA, Malaysia. These students were majoring in different programs including Computer Science, Mathematics, Networking, Multimedia and Electrical Engineering. The survey was conducted using an online

platform and distributed to first semester students in the faculty. It is conducted to obtain three objectives, 1) to identify problems faced by students in learning a programming language, 2) to identify the preferred sources to get help and 3) to identify the students' acceptance toward chatbots for learning programing language.

4.1 Evaluation metric

The first objective was evaluated by asking three questions with three answer options: Yes, No, Others. The questions were:

- Are you having any difficulty in understanding presented programming problems?
- Are you having any problem in understanding and visualizing programming tasks and designing their algorithmic solutions?
- Are you having any problem in remembering JAVA programming language syntax?

The second objective identified the preferred sources for students to find a solution for their programming problems. The selection of sources included lecturers, websites, forum, tutors, books, and others. Each participant was only allowed to choose one from the selection list.

The third objective asked students' opinions on whether they like to use a chatbot for learning programming. The three-scale answer options used were Yes, No, and Others.

4.2 Results

The result obtained from this survey is used to indicate whether chatbots could be used as an alternative in learning programming. The result obtained for the first objective shows that the students had problems in understanding and finding a solution for a programming problem as shown in Figure 7. The result shows that the students who had problems in understanding programming language outperformed students that indicated no problems in all three questions asked.

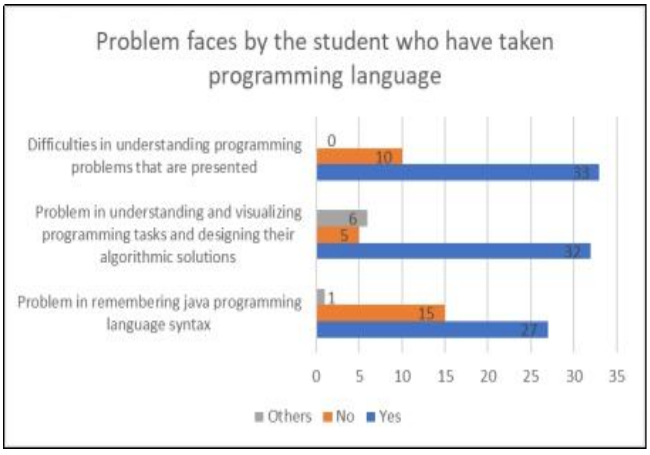


Figure 7: Distribution of students having problems in a programming language

The result obtained for the second objective shows that more than a quarter students preferred to find the solution from the websites compared to other sources. The result also indicates that students preferred to find a solution using an online platform by 50 per cent of the students chose forums or websites as shown in Figure 8. The reason is the solution to their problems was more accurate and correct through this medium. The respondents did not likely refer books as their source to get solutions and used other sources including asking their friends or watching YouTube videos.

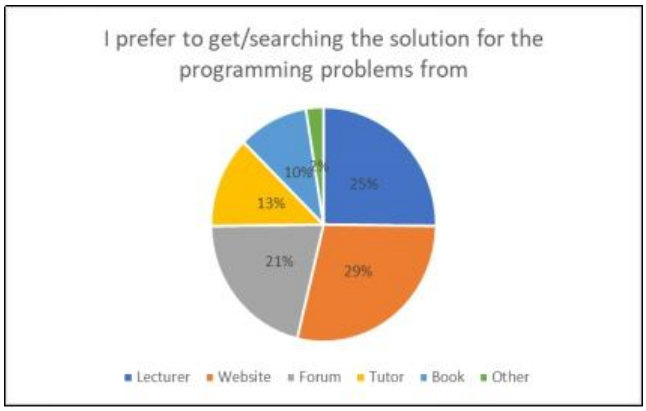


Figure 8: Distribution of students' preferred sources to find a solution

The result obtained for the third objective shows that over three-quarter respondent agreed that chatbots were beneficial for them and 5 per cent of the respondents did not agree, where the five percent respondents referred to the persons who did not face difficulties in learning programming language as illustrated in Figure 9.

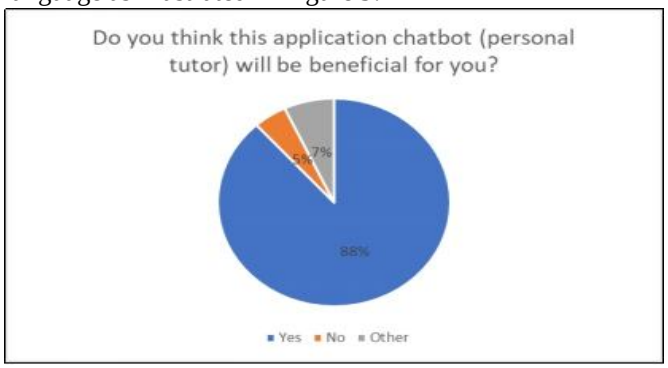


Figure 9: Distribution of students who think chatbot is beneficial for learning a programming language

In the next section, we discuss the e-JAVAchatbot design and system architecture.

5. JAVABOT SYSTEM DESIGN AND IMPLEMENTATION

The submitting author is responsible for obtaining the agreement of all co-authors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

5.1 Topic Coverage

The e-Java chatbot initial development includes control structure topic, with both factual and coding questions. The user of the e-JAVA chatbot can ask coding from a chatbot, given the problems inserted in a table form. Figure 10 shows the coverage of subtopic implemented in e-JAVA chatbot.

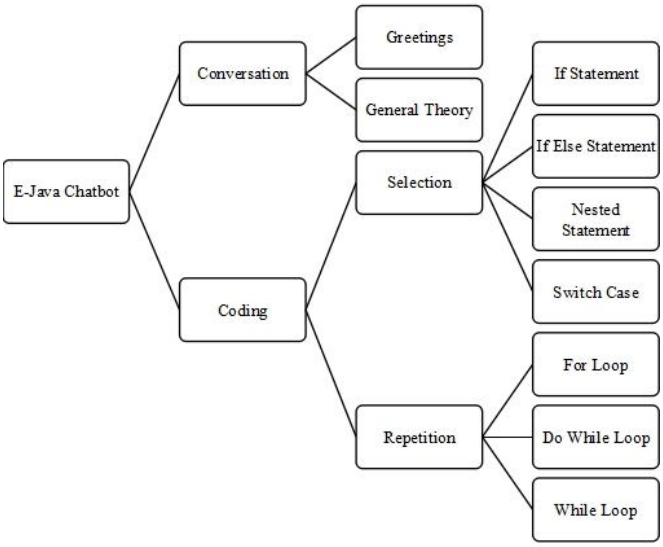


Figure 10: Topic coverage in e-Java chatbot

5.2 System Architecture

The e-JAVA constructed with three modules namely, conversation, selection and repetition. Students could enter questions in the form of text into a text input field for a conversation module or table form for selection and repetition modules. The user input is matched to the pattern stored in the three modules to process the appropriate solution to the students' questions. Figure 11 shows the system architecture of e-Java chatbot that received user input and produced programming solutions based on JAVA language syntax.

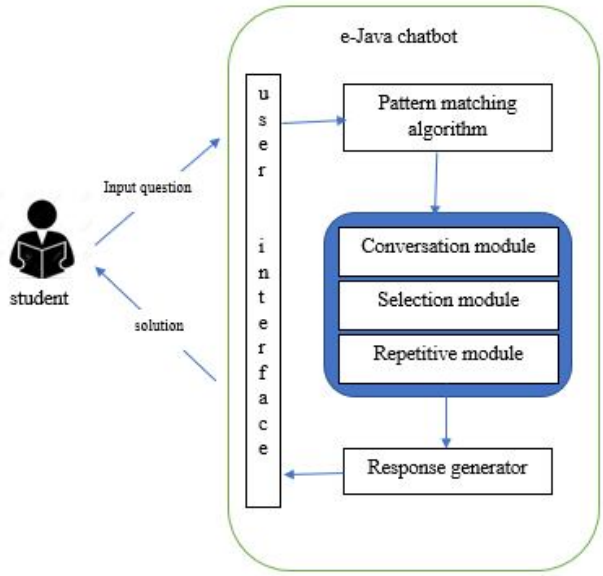


Figure 11: e-JAVA system architecture

5.3 Algorithm design

The algorithm structure is used to control the output based on user selection. A user might select either to learn selection control structure or repetitive control structure. The output is presented based on the input pattern entered by the user. The input for both selection and repetition is in a table form. Table 2 and Table 3 show the table format for selection and repetition, respectively.

Table 2: Sample user input for the selection control structure

Categories	Condition	Output
n	< 0	Negative numbers
n	>= 0	Positive numbers

The output generated is based on the input pattern which is indicated by a number of row and column as shown in Figure 12.

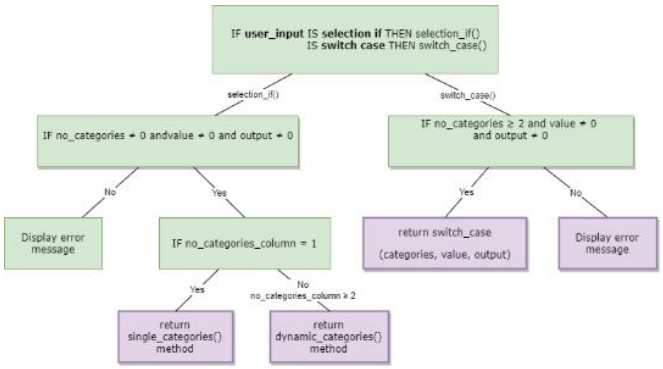


Figure 12: Input pattern flow for selection control structure

Table 3: Sample user input for the selection control structure

Row/Col	Col[1]	Col[2]	Col[m]
Row[1]	*	*	
Row[n]	*		

The table will create a 2D array and the number of row and column for the table is dynamic, which also means that the user can add as many rows and columns as they want. E-JAVA will generate the code when at least one cells of the table is filled, otherwise, e-JAVA will throw an error message to ask the user to enter an input. Furthermore, the flow in Figure 13 shows the condition for a single loop, difference in size of a table row and column and table of the same size.

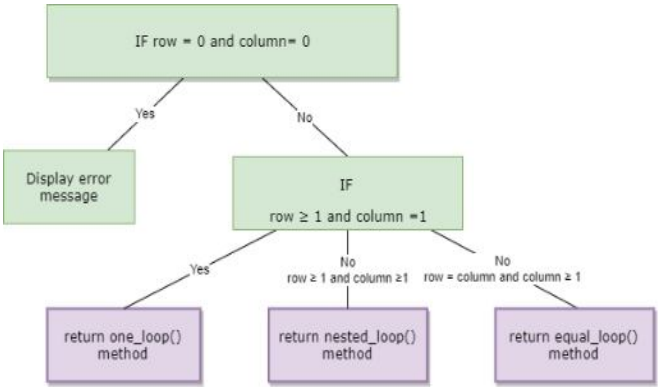


Figure 13: Input pattern flow for repetitive control structure

5.4 Interface Design

This application has a simple design of interface which can be easily understood and used by any user. The first scene of this application is the application main page as shown in Figure 14. The main page of the application consists of the name of the application and the menu button for the user options. The button conversation is for the messaging interfaces between the user and the bot. While the button guides are for the samples guide for the user manual for better user understanding and the about button provide application's information.

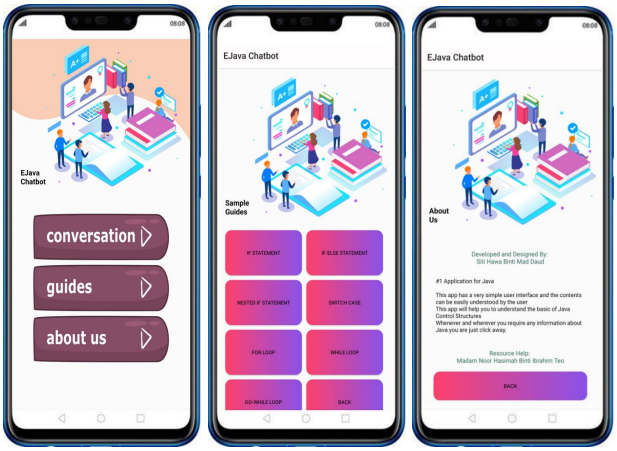


Figure 14: Interface for a) Main page, b) Guides and c) About us

The main function of e-JAVA lies on the conversation menu button, where the learning of the control structure takes place. There are three more sub-menus in the conversation page which are:

Conversation for a fact explaining about selection and repetitive control structure. This menu is using text field input.

Selection is for asking sample code on three selection control structures: 1) one-way if, 2) two-way if, and 3) switch-case. The input from the user is in a table form template as shown in Figure 15.

Figure 15: Table form template for user input on the control structure menu

Repetition is for asking a sample code on three repetitive control structures: 1) for loop, 2) while loop and 3) do-while loop. The input from the user is also in a table form template.

6. E-JAVA SYSTEM TESTING RESULTS

This section discusses the results of the system testing using test cases. The test was conducted to find out whether e-JAVA could help students in 3 situations:

i) Which control structure is used for the given problems?
ii) How to code the given problem?
iii) How different forms of selection and repetition are used to solve the same problem?

The validation of e-JAVA output is conducted based on five test cases as shown in Figure 16-20.

Test Data	Table for Selection Statement: age, Variable 2, age group
Expected Result	If Else
	`if (age > 18) { System.out.println("adult"); }`
	Nested If
	Use of nested statement are not suitable
	Switch case:
	Use of switch case are not suitable
Actual Result	Same as expected result
Remarks	Passed

Figure 16: Test case #1 for selection control structure

The result in Figure 16 shows the solution was generated when a question input was entered by the user. This test case shows that e-Java could suggest a possible solution to a given problem. While Figure 17 shows the output of an error message was displayed when no input from a user was inserted. A user only can add more rows to the table but not column. The column is fixed into item name, condition, and output only.

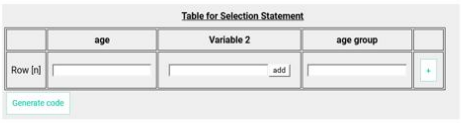
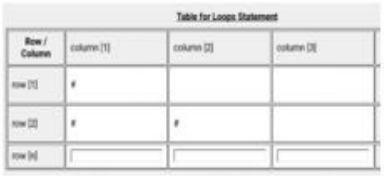
Test Data	
Expected Result	Empty field: error message display
Actual Result	Same as expected result
Remarks	Passed

Figure 17: Test case #2 for selection control structure

Figure 18 shows the solution obtained for inserting a pattern into a table for repetitive control structure. The result shows that an e-JAVA was able to provide a solution for all three repetitive control structures for the same problem. This indirectly could help a student to understand a different form of codes for the same problem given.

Test Data	

Expected Result	While loop
	`int row = 1;`
	`while (row <= 2)`
	`{`
	`  int column = 1;`
	`  while (column <= row)`
	`  {`
	`  System.out.print('#');`
	`  column++`
	`  }`
	`  System.out.println();`
	`  row++;`
	`}`
	For loop
	`for(int row=1; row<= 2; row++)`
	`{`
	`  for(int column=1; column<=row; column++) {`
	`  System.out.print('#');`
	`  }`
	`  System.out.println();`
	`}`
	Do-While loop
	`int row = 1;`
	`do`
	`{`
	`  int column = 1;`
	`  do`
	`  {`
	`  System.out.print('#');`
	`  column++`
	`  } while (column <= row)`
	`  System.out.println();`
	`  row++;`
	`} while (row <= 2)`

Actual result	Same as expected result
Remarks	Passed

Figure 18: Test case #3 for a repetition control structure

Meanwhile, Figure 19 shows the incorrect pattern inserted into the table and it returned an error message. The input problem inserted in the table must have a proper pattern and in order. It cannot be scattered around the table cell. The number of row and column are scalable, and students can add as many rows and columns.

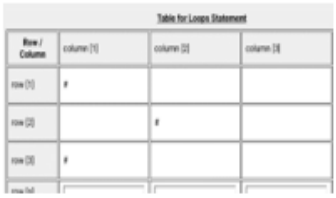
Test Data	
Expected Result	Error message display
Actual Result	Same as expected result
Remarks	Passed

Figure 19: Test case #4 for a repetition control structure

The evaluation also has been conducted for a general conversation on greeting and guide menu for asking about facts. The result in Figure 20 shows that e-JAVA was able to produce the output as expected.

Test Data #5	User input: Hello!!!
Expected Result	Random: Greetings messages such as hi! Any questions about java control structures
Actual Result	Same as expected result
Remarks	Passed
Test Data #6	User message: Question related about java control structures Example: What is control structures?
Expected Result	Related answer corresponds to the questions
Actual Result	Same as expected result
Remarks	Passed
Test Data #7	User message: Question not related about java control structures Example: What is your name?
Expected Result	Error respond message corresponds to user input Example: Sorry, I don't think that can get your questions.
Actual Result	Same as expected result
Remarks	Passed
Test Data #8	User message: Unclear questions Example: Syntax?
Expected Result	Bot respond message correspond to user input. Example: Sorry, can you please your questions clearly?
Actual Result	Same as expected result
Remarks	Passed
Test Data #9	User message: Questions related to java that does not exist in the bot template. Example: code use to calculate the odd and prime number using repetition?
Expected Result	Error respond message corresponds to user input Example: Sorry, I don't think that can get your questions.
Actual Result	Same as expected result
Remarks	Passed

Figure 20: Test cases for general questions

The system testing conducted shows that e-JAVA chatbot could be used as a virtual tutor to help students finding a solution for their programming problems. Furthermore, it could help the user to learn and understand the different ways to code control structure for the same problem by generating codes of different forms as shown in Figure 18.

7. CONCLUSION

Learning programming language is the key and fundamental theory in Computer Science study. Learning through virtual tutors may relieve students' pressure and avoid students from bothering their lecturing for just a single code that they need

to understand. E-JAVA Chatbot may help students in understanding the topic they found difficult to understand by reducing the time taken to get help from lecturers personally and provide the exact answer for their problem rather than an example of similar code from the web. However, this e-JAVA was constructed using limited contents and cover and is meant to be enhanced in future with full topics' coverage for programming language.

ACKNOWLEDGEMENT

This work is supported by Universiti Teknologi MARA Melaka Branch, under the TEJA 2020 Internal Grant (GDT2020-41).

REFERENCES

1. L.P. Jack, N. Khamis, C. Salimun, D. M. Nizam, Z. Haslinda, and A. Baharum. Learn **Programming Framework for Malaysian Preshoolers.** *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE).* vol. 8, no.1(6), pp. 431–436, 2019. https://doi.org/10.30534/ijatcse/2019/6281.62019
2. S. Aithal. **Impact of On-Line Education on Higher Education System**. *International Journal of Engineering Research and Modern Education*, 7(1), 225–235, 2011.
3. A. Buran and A. Evseeva. **Prospects of Blended Learning Implementation at Technical University**. *Procedia - Social and Behavioral Sciences*, *206*(November), 177–182, 2015.
4. M. Choy, U. Nazir, C. K. Poon and Y.T. Yu. **Experiences in Using an Automated System for Improving Students' Learning of Computer Programming**. *Lecture Notes in Computer Science*, *3583*(6980041), 267–272, 2005.
5. R. Romli, S. Sulaiman, and K.Z. Zamli. **Improving Automated Programming Assessments: User Experience Evaluation Using FaSt-generator**. *Procedia Computer Science*, *72*, 186–193, 2015. https://doi.org/10.1016/j.procs.2015.12.120
6. M. Dahiya. **A Tool of Conversation: Chatbot**, *International Journal of Computer Sciences and Engineering (JCSE)*, *5*(5), 2347–2693, 2017.
7. A. S. Miner, A. Milstein, S. Schueller, R. Hegde, C. Mangurian, and E. Linos, **Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health,** JAMA Intern. Med., vol. 176, no. 5, pp. 619–625, 2016.
8. G. M. Lucas, J. Gratch, A. King, and L.-P. Morency, **It's only a computer: Virtual humans increase willingness to disclose**, Comput. Human Behav., vol. 37, pp. 94–100, 2014. https://doi.org/10.1016/j.chb.2014.04.043
9. S. Vosoughi, D. Roy, and S. Aral**, The spread of true and false news online, Science** (**80-.).,** vol. 359, no. 6380, pp. 1146–1151, 2018.
10. M. Berger, T. H. Wagner, and L. C. Baker, **Internet use and stigmatized illness**, Soc. Sci. Med., vol. 61, no. 8, pp. 1821–1827, 2005.
11. S. Ivanov, C. Webster, E. Stoilova, and D. Slobodskoy, **Biosecurity, automation technologies and economic resilience of travel, tourism, and hospitality companies**, 2020. https://doi.org/10.31235/osf.io/2hx6f
12. T. Bittendorfer, J. Bunt, L. Grunder, D. Riedel, B. Magnus, and T. Salzlecher, **Technology in tourism: How voice assistants influence the hospitality industry**, *ISCONTOUR*, pp. 328–338, 2019.
13. I. Nica, O. A. Tazl, and F. Wotawa, **Chatbot-based Tourist Recommendations Using Model-based Reasoning.**, in ConfWS, 2018, pp. 25–30.
14. C. Liu and K. Hung, **Understanding self-service technology in hotels in China: technology affordances and constraints,** *in Information and Communication Technologies in Tourism* 2019, Springer, 2019, pp. 225–236.
15. W. Wei, E. Torres, and N. Hua, **Improving consumer commitment through the integration of self-service technologies: a transcendent consumer experience perspective,** *International Journal Hosp. Manag.*, vol. 59, pp. 105–115, 2016. https://doi.org/10.1016/j.ijhm.2016.09.004
16. R. Y. Kim, **The Impact of COVID-19 on Consumers: Preparing for Digital Sales**, *IEEE Eng. Manag. Rev.*, 2020.
17. B. Morgan, **20 Fresh Examples of Customer Experience Innovation,** Forbes. [Online]. Available: https://www.forbes.com/sites/blakemorgan/2019/10/21/20-fresh-examples-of-customer-experience-innovation/#4634233d7c41. [Accessed: 19-Jun-2020].
18. A. W. Ding and S. Li, **National Response Strategies and Marketing Innovations during the COVID-19 Pandemic**, *Business Horizon*, 2020.
19. S. Sujata, M. Nikita and S. Shubham. **Applications of Chatbots in Marketing: use Cases, Impacts, Challenges and Drivers.** *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE).* vol. 8, no.1(6), pp. 195–200, 2019. https://doi.org/10.30534/ijatcse/2019/3081.62019
20. M. Broussard. **Chatbot-Like Siri Patent Includes Intelligent Image, Video, and Audio Recognition within Messages.** *MacRumors*, May, 11, p.7, 2017.
21. E. Luehning, **Microsoft tool 'Clippy' gets a pink slip**. *CNET News*, 2001.
22. L. Von Ahn, **Duolingo: learn a language for free while helping to translate the web**. In *Proceedings of the 2013 international conference on Intelligent user interfaces* (pp. 1-2), 2013. https://doi.org/10.1145/2449396.2449398