

# BITLANGUAGE: A JAVA LANGUAGE LEARNING APP

**Kurt Christian Lopez**<sup>1</sup>

*Computer Science Program  
School of Engineering, Architecture, and Information  
Technology Education, University of Saint Louis  
Tuguegarao City, Cagayan*

**Danzelle Rae Bajet**<sup>2</sup>

*Computer Science Program  
School of Engineering, Architecture, and Information  
Technology Education, University of Saint Louis  
Tuguegarao City, Cagayan*

**John Kenneth Jacinto**<sup>3</sup>

*Computer Science Program  
School of Engineering, Architecture, and Information  
Technology Education, University of Saint Louis  
Tuguegarao City, Cagayan*

**Diane Jenalyn Datul**<sup>4</sup>

*Computer Science Program  
School of Engineering, Architecture, and Information  
Technology Education, University of Saint Louis  
Tuguegarao City, Cagayan*

**Abstract**— Learning programming is considered to be a daunting task by many. Students would invest their time and money in books, boot camps, or university programs but would still find it hard to master the basics of programming. This problem is especially true in Java programming language, where it can take 1-2 years for an individual to master it. This study aimed to look into the adoption of an approach called the Digital Assessment Proficiency Exam to develop a mobile app that can help students learn the basics of Java. The iterative model was used in the development of the app. Six university instructors evaluated the app using the Great App Checklist with a mean score of 4.17 for purpose, 4.17 for alignment, 3.83 for pedagogical framework, 4.17 for personalization, 4.42 for ease of use, and 4.28 for app citizenship. With an evaluation ranging from 3.83 to 4.17, the product of this study is a highly effective tool for learning Java.

**Keywords**— *Java, programming language, digital assessment proficiency exam, frequent testing, flowcharts.*

## I. INTRODUCTION

Computer Science is a relatively unforgiving field. It all comes down to programming. A programming language is needed to create nearly all application software. However, learning to program from scratch is challenging, tiring, and demanding. Learning the logic behind the syntax of a programming language is just like understanding the grammar of a natural language but without ambiguity. Notably, according to a study in 2022 [1], it would take six months to learn coding from a coding boot camp, 12 months of self-study, or more than four years from a college degree.

Flowcharts have been agreed to be the best way to study algorithms. A flowchart is a type of diagram used to represent the logic behind the algorithm. Flowcharts help programmers develop the most efficient code because they can see the whole picture of the algorithm and know where the data will end up [2]. According to an online encyclopedia [3], flowcharts make

it easier for programmers to communicate complex rationale inside a framework. Likewise, flowcharts can be used as a guideline to plan the next project program. Many developers use flowcharts to assist them with troubleshooting.

Programming languages have often been compared to natural languages. There are a lot of similarities between programming languages and natural languages, such as how they can identify the differences between syntax and semantics, the existence of composition, and the reason for communication [4]. One prominent app in the Google Play store offers an effective way to learn natural languages called Duolingo. Duolingo is an app for learning foreign languages, like English, French, Mandarin, et cetera. Duolingo has been proven by a team of PhDs dedicated to studying the effects of Duolingo on its learners. According to a survey [5], the content is so good that Duolingo ranks among the best software for learning a language.

One feature of Duolingo is frequent testing. Researchers found that frequent and repeated tests can boost learning. A study in cognitive science and psychology [6] shows that proper and frequent tests can be an effective way to learn. Repeating tests can produce better memory recall and offer a deeper understanding than education without examinations.

The researchers aim to develop a Mobile Application similar to Duolingo's approach but with Java Language Learning Content. Java is used to create mobile apps, websites, web apps, desktop programs, games, et cetera [7]. From experience, the researchers believed that Java is the best language for beginners to learn. Java is a strongly typed language that will catch beginner mistakes [8]. There are already similar applications available on the Google Play Store that offer programming proficiency exams, like Codingguru. However, the researchers did not find any study backing up the credibility of these applications. Moreover, these applications have a different style of approach from Duolingo, which has a

basic and straightforward process that the researchers would like to follow. Another gap that separates this app from the apps available in the market is that it was developed using Google's native language, Kotlin, and Google's recommended development software, Android Studio. These translate to code efficiency and better software-to-hardware compatibility.

This study compiled the adoption of flowcharts, Duolingo's approach, and frequent testing into one mobile app to learn Java more quickly. The development product will serve as the starting point of future related software for learning programming languages.

## II. RELATED WORKS

Enki is an app available in the Play Store that teaches coding, excel, SQL, and data skills. It uses a coding playground where students can freely run their code with mobile devices [9]. Enki was founded by Kirill Makharinsky after he saw the impact of team-wide data training at the hotel booking company he co-founded previously. It was a big reason why they grew from \$100m in annual sales to almost \$1 billion while becoming profitable. He also saw a huge impact from internal data training and a strong data culture at some of the top companies in the world, like Airbnb, Stripe, and Revolut.

Sololearn is a programming education app designed to teach the basics of software development. It breaks down the programming concepts into daily lessons. Furthermore, it offers a range of programming language courses [10]. Sololearn has the world's largest collection of FREE programming courses to learn how to code. Learn Python, C++, JavaScript, Java, jQuery, machine learning, data science, and more. You receive a certificate for each course that you complete. Choose from thousands of programming topics to learn coding concepts, brush up on your programming skills, or stay informed about the latest coding trends. Challenge yourself and others with head-to-head coding competitions. Write and test your code on iOS's #1 mobile code editor.

Grasshopper is an educational app designed to teach Javascript. The students work with blocks of real javascript codes rather than coding concepts or code-like blocks [11]. Grasshopper is the best way to start the coding adventure with fun, quick games on your phone that teach you to write real JavaScript. Players can move through progressively challenging levels as they develop their abilities, then graduate with fundamental programming skills for their next step as a coder.

## III. METHODS

### A. Project Design and Specification

In the history of programming languages, Visual Programming Language holds the most optimistic approach to building a software application, and it has been the goal of Human-Computer Interaction. Perhaps the current most promising Visual Algorithm is flowgorithm. Flowgorithm.org [12] is a website that offers a visual programming language to

aid beginners in understanding the logic behind the syntax of a spring-based algorithm.

Flowgarithms are tools that allow users to write programs using flowcharts. Some programmers consider flowgarithm to be a programming language. However, the purpose of flowgarithms is to assist practitioners in understanding the concepts and logic behind the program. According to a study in 202 [13], a developer could create a complete plan using a flowchart only. The article added that with the help of a flowchart, the developer could know every step of the process in the software. Hence, practitioners can easily transition from flowgarithm software to high-level string-based programming languages like Java or Python at the advanced level.

Through software development, flowcharts and visuals are used to aid beginners in grasping the syntax behind an algorithm. Moreover, it is, in fact, beneficial and effective. For example, A study in 2006 [14] found that Visual programming using a flowchart is the best way for beginners to create a program for general purposes. It is superior to the conventional programming system regarding readability, ease-to-debug, effectiveness, and user-friendliness. This study is helpful to the researchers as it adds credibility to determining the efficacy of flowcharts as learning material.

A study in 2018 [15] attempted to answer if the essentials of algorithms and programming can be taught more effectively using computer algebra systems, spreadsheets, flowgarithms, animations, and other visual techniques. In two different surveys, they found out that flowgarithm is an effective tool in teaching. They found that flowgarithm is an effective tool for lecturing and presenting algorithms and results.

These different studies about flowcharts built a strong foundation for the application, as flowcharts are one of the main features of BitLanguage.

Today, the modern natural language learning approach uses many virtual flashcards. These flashcards have different questions like multiple choice, matching type, identification, et cetera. Numerous studies have been invested in this kind of approach, and most of them got positive results.



Fig. 1. A Duolingo digital flashcard.

Figure 1 shows an example of a digital flashcard from Duolingo. Duolingo is a free mobile learning app that introduces courses in foreign languages. A study in 2012 [16] shows the effectiveness of Duolingo and found that the vast majority of its participants found the app helpful in their foreign language studies. Most of them succeeded in learning Spanish. Based on their findings, a novice user would take 26 to 49 hours of study with Duolingo to cover an entire semester of college.

Duolingo is a mobile learning application that offers skill trees and lessons that use digital flashcards, multiple-choice questions, and listening exercises to introduce learners to new words and phrases. Memrise is also a mobile learning application similar to Duolingo, with unique features. Babbel is another learning application that uses multiple-choice questions. It also uses conversational examples to show how words and phrases are used in real scenarios [17]. A study published in 2022 [18] shows that with Duolingo's digital flashcards and multiple-choice questions, learners without prior proficiency in a target language reached low intermediate proficiency in reading and novice high proficiency in listening. Duolingo was the only tool the learners used in their study. The reading and listening scores of the learners were comparable with students from their 4th year of the semester. They concluded and suggested that Duolingo is an effective tool for studying foreign languages. Indeed, Duolingo's learning approach helps students and learners of natural languages master their courses quickly and effectively. This learning approach was implemented in the BitLanguage mobile application.

Frequent exams and testing is another learning approach that uses quizzes, exams, tests, and challenges to introduce knowledge to students instead of traditional reading assessments. Frequent exams and testing are critical features of BitLanguage, where there are only minimal reading assessments inside the mobile app. Various studies have reported the effectiveness of this approach in learning. According to a study in 2019 [19], frequent exams and tests have a better effect on learning than repeated studying. Subject students who used frequent testing scored seven percent higher and earned half the number of failing grades. They stated that frequent testing was even more potent when using multiple-choice and free-response questions.

Giving quizzes and short exams to students results in better memory and understanding of complex lessons rather than a one-time extended comprehensive test. With repeated testing, learners are more likely to read the lectures and master the material in a short period. This principle is referred to as frequent testing or retrieval practice [20]. A study in 2009 [21] compared two learning techniques: Readiness assessment tests (RATs) and frequent testing. The study suggested that if a university instructor wants his students to learn the material in any way possible, he should consider frequent testing. The study added that frequent testing is an effective strategy to address the discrepancy in learning styles. A study in 2013 [22] stated that using computers to take a quiz online can act as an aid to teaching, not just a distraction. The study further mentioned that frequent testing reduces students' gaps in achievements. Moreover, according to an article in 2018 [23],

the testing effect is a style of learning that involves answering questions from a lesson rather than restudying the material. Taking tests during the learning phase supplements the retrieval of knowledge from long-term memory [24]. Frequent testing and the testing effect play a significant role in BitLanguage. This learning style instigates students to better retrieval of memory for remembering the syntaxes of an algorithm.

### B. App Design

The app development was completed using Android Studio, DB Browser, Adobe Photoshop, and Audacity. Kotlin was the primary programming language that handled the software's back-end processes. Photoshop and Android XML were used to design the UX and UI. SQLite was selected as the primary database of the app.

Android activity is a Kotlin or java code representing one screen on an android app. An activity consists of 2 components. The first component is a Java/Kotlin file that handles the background process of the activity. The second component is an XML (Extensible Markup Language) file that defines the activity's layout. Each layout has several methods that the app needs to function.

BitLanguage consists of seven Kotlin files, six activity layouts, 35 code divisions, 28 methods, and 2243 lines of code.

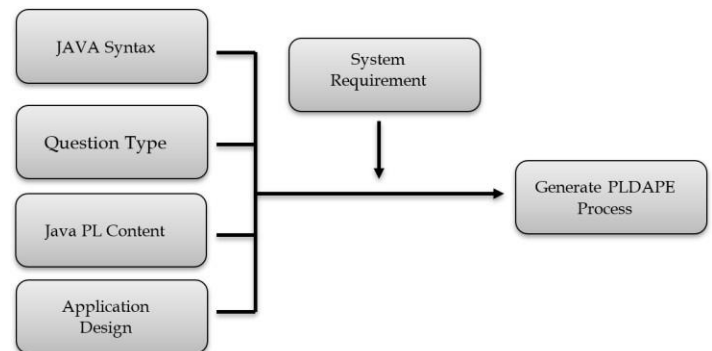


Fig. 2. Conceptual Framework.

As illustrated in Fig. 2, the overall goal is to produce a JDAPE (Java Digital Assessment Proficiency Exam) process. The researchers identified four variables to achieve the output. These variables are JAVA Syntax, Question Type, Java Programming Language Learning Content, and Application Design. Furthermore, one moderator variable can affect the outcome of the process. This variable is the system requirement of the application.

The learning process of the application is based on the tree-based approach of Duolingo. The application offers different question modes like multiple-choice, true or false, and matching. The learning tree of the application provides an essential introduction to the Java language. The app first assesses the student's prior knowledge before starting. The student can start on any lessons he wants.

<b>N</b>	Java Syntax
<b>N+1</b>	Java Comments
<b>N+2</b>	Java Variables
<b>N+4</b>	Java Data Types
<b>N+4</b>	Java Type Casting
<b>N+5</b>	Java Operators
<b>N+6</b>	Java Strings
<b>N+7</b>	Java Math
<b>N+8</b>	Java Booleans
<b>N+9</b>	Java If...Else
<b>N+10</b>	Java Switch
<b>N+11</b>	Java While Loop
<b>N+12</b>	Java For Loop
<b>N+13</b>	Java Break/Continue
<b>N+14</b>	Java Arrays

Fig. 3. Java Lessons.

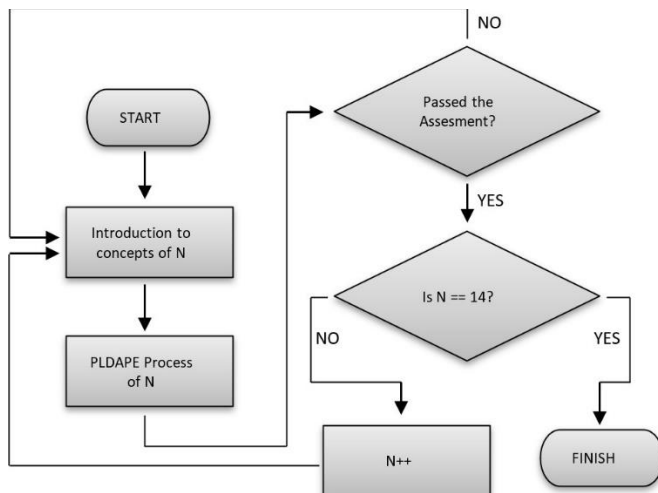


Fig. 4. Learning Process Flow.

Fig. 3 and Fig. 4 describe the learning process of BitLanguage. BitLanguage follows a linear skill tree where lessons start from the basic to intermediate lessons in Java. This study's data is qualitative and cannot be measured in numbers. Instead, the researchers only want a single answer: Can the researchers develop an app that implements the three learning styles to teach new Java practitioners? The Software Development Life Cycle's iterative model is implemented to answer this study's objectives. The system software, hardware, and the internet are used as the primary material of this study.

The researchers thoroughly studied the requirement specifications of the language. The development tools that the researchers used are Android Studio, XML, Google Kotlin, SQL Lite, and Adobe Photoshop. The mobile application is tested on the Android 5 (Lollipop) emulator of Android Studio.

The graphical interface of the application was heavily based on Duolingo. XML or Extendable Markup Language is

the main API the programmers implemented to develop the User Interface and graphics. The testable parts of the software are referred to as units. Units may be a subprogram, function, or procedure.

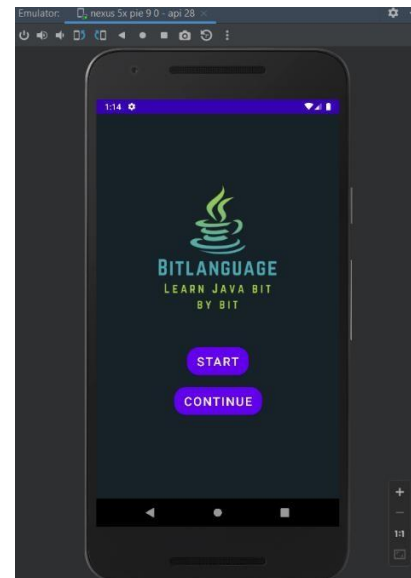


Fig. 5. BitLanguage Home Screen running on Nexus 5x Pie 9.0 Emulator.

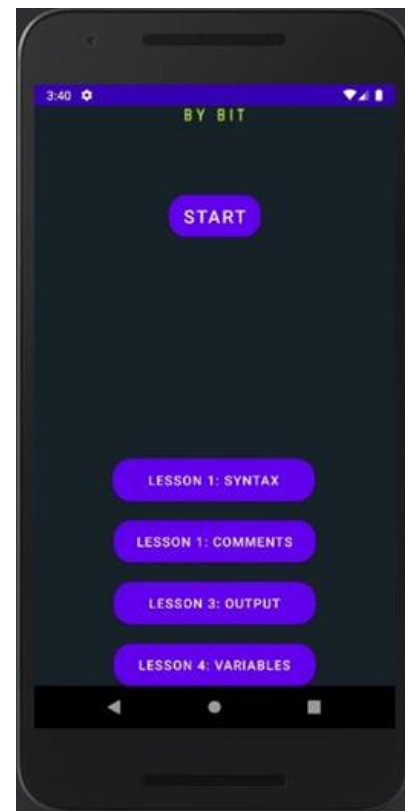


Fig. 6. Linear Skill Tree.



Fig. 7. Information Digital Flashcard.

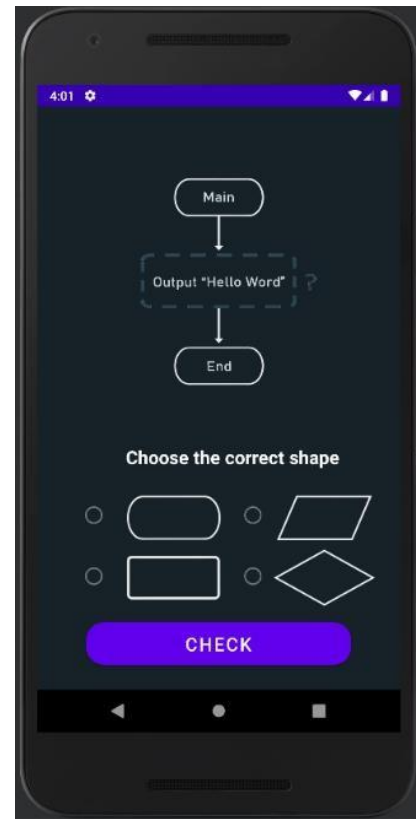


Fig. 9. Flowchart-type digital flashcard.



Fig. 8. Multiple Choice-Question Flashcard.

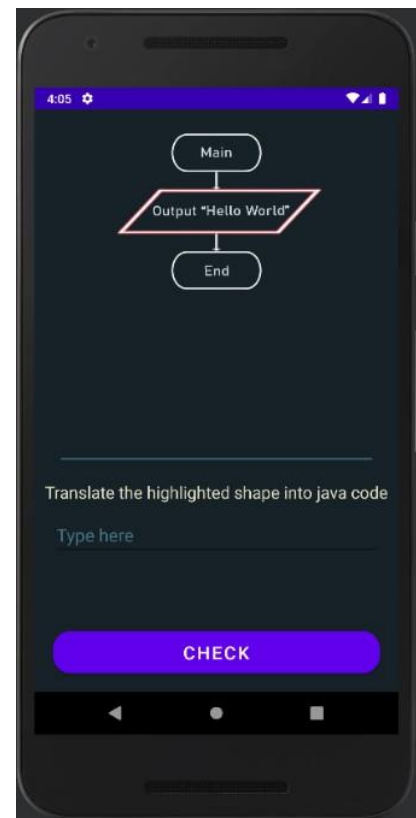


Fig. 10. Real-time coding digital flashcards.

Fig. 5 shows the home screen of the app. The app starts on the home screen when launched. Fig. 5 also shows that the testing and prototyping were done in the Nexus 5x Pie 9.0 Emulator. Successful testing in Android 9 Pie means that the application can run on 98% of Android devices according to the android studio itself, the software the researchers used. Each function was tested. Bugs and errors are frequent but have been debugged. After testing, all the units developed in the previous phase were integrated into the systems. Post integration, the entire app was tested for any faults and failures. Fig. 6 shows the skill tree when the user scrolls down from the homepage. Also, the user can freely choose a topic he wants to start. Fig. 7 shows a digital information flashcard. Its purpose is to introduce new knowledge of Java to users and assist them with what to expect from the following questions. Fig. 8 shows an example of a multiple-choice question flashcard. The user only needs to choose the correct answer from the 3 choices. If the user gets the correct answer, the app will proceed to the next question. Fig. 9 shows a flowchart-type digital flashcard. Here, the user needs to choose the correct share representing the code in the flowchart to get the correct answer. Fig. 10 shows a digital flashcard asking the user to type the correct syntax of the code of the highlighted shape of the flowchart.

### C. Testing and Evaluation

The researchers adapted the Great App Checklist [25] to evaluate the educational quality of the app. The Great App checklist is a tool for teachers and university instructors to assess if the app is appropriate for the students. It asked if the app has attained the different learning factors like purpose, alignment, pedagogical framework, personalization, sharing, ease of use, privacy, app citizenship, and accessibility.

## IV. RESULTS AND DISCUSSION

Six technical experts from the ITE department evaluated the mobile app. They were asked to open the link on Google Drive and download the Android Package Kit (APK) of Bitlanguage. The researchers gave instructions to the technical experts about how to install and use the app. Then, the instructors used the application without any supervision from the researcher. Google Forms was used as the evaluation medium. A five-point scale rating was used in the evaluation, which was based on the University. The lowest score that the participant can give is one, which corresponds to poor, while the highest score is five, which corresponds to excellent. The mean of the scores per category was computed after evaluating all the participants.

Table I shows the instructor's evaluation of the mobile app. The app scored a mean of 4.171 in all categories. The Pedagogical Framework category scored less than 4, while the rest of the categories got a score above 4. All scores correspond to the "Good" description on the five-point scale.

Bit language is a Java learning mobile application. Its purpose is to instill the concepts of Java Programming Language to beginners. BitLanguage implemented three approaches to learning namely: flowcharts, the modern natural

language learning approach, and frequent testing. The application proved helpful to both instructors and students of ITE at the University of Saint Louis.

TABLE I. RESULT OF THE EVALUATION

Category	Mean per Category	Description
Purpose	4.1667	Good
Alignment	4.1667	Good
Pedagogical Framework	3.8333	Good
Personalization	4.1667	Good
Ease of Use	4.4167	Good
App Citizenship	4.2778	Good

The learning contents of BitLanguage relate to the curriculum. It is leveled to the student's skills, and it fits the University's standards. The app is logical and simple, and it is readily used by students without much effort. On the other hand, the Pedagogical Framework category scored the least. The researchers speculate that the flowchart and programming flashcards in the app need to be improved.

Frequent testing is a relatively new concept. It is a learning approach and it works for a student repeating a test or an exam until that student masters every question. Frequent exam has a better effect on a student than repeated studying [16]. Frequent testing is even more effective when using multiple-choice and short-answer questions. Another advantage of frequent testing is that it prompts learners to read lectures and master the material in a short period [17]. Taking tests during the learning phase makes better retrieval of information from memory [21].

## V. CONCLUSION

Information is getting more and more readily accessed since the introduction of the world wide web. A normal person can learn any foreign language he wants without enrolling in a school or university with the help of a digital learning application. BitLanguage is a Java learning mobile application to help beginners learn the basic concepts of algorithms of Java, the programming language. The BitLanguage mobile app has successfully implemented the three approaches to learning the research has looked into. It developed a Java Digital Assessment Proficiency Exam feature that houses flowcharts, flashcards, and code-typing.

Frequent testing has massive potential. Thus, it is recommended to future researchers not just look at Computer Science but also to look more into frequent testing as a way of learning. Frequent testing is especially effective on courses that require tons of memorization, like biology, chemistry, or even social sciences. The University of Saint Louis is already implementing frequent testing in the form of Mock-Board Examinations, weekly quizzes, and three-term exams in a semester.

The researchers recommend to future developers who will dwell through frequent testing to make more mobile or web applications that implement digital proficiency exams in different courses like engineering, medicine, education, and more.

## REFERENCES

- [1] J. Weinstein, "How long does it take to learn coding?", [online document], 2021. Available: <https://careerkarma.com/blog/how-long-does-it-take-to-learn-coding/>. [Accessed: September 07, 2021].
- [2] M. C. Canaces, "Programming Flow Charts: Types, Advantages and Examples", [online document], 2021. Available: <https://zipurl.eu/cbPDL>. [Accessed: September 10, 2021].
- [3] Integrify, "What is a Flowchart, and Why Is It Important?", [online document], 2022. Available: <https://www.integrify.com/what-is-a-flowchart/>. [Accessed: September 25, 2021].
- [4] Chakray, "Programming Languages: Types and Features: Differences between natural language and programming language" 2022. [Online]. Available: <https://tinyurl.com/p6es6dsj> [Accessed: November 5, 2021].
- [5] J. Duffy, "Duolingo Review", [online document], 2022. Available: <https://tinyurl.com/2p9d4mbh>. [Accessed: September 25, 2021].
- [6] A. M. Pau, "Researchers Find That Frequent Tests Can Boost Learning", [online document], 2015. Available: <https://tinyurl.com/33uvschh>. [Accessed: November 7, 2021].
- [7] W3schools, "Java Tutorial", Refsnes Data 1999-2022. [Online]. Available: <https://www.w3schools.com/java/>. [Accessed: December 3, 2021].
- [8] jayvinpaul, "Why Java is the best Programming language for Beginners to Learn Coding?", [online document], 2020. Available: <https://tinyurl.com/2p92amsu>. [Accessed: December 17, 2021].
- [9] Enki. [Google Play store] United States: Enki.com, 2015.
- [10] SoloLearn. [Google Play store] United States: Y. Hyusyan, D. Kocharyan, 2014.
- [11] Grasshopper. [Google Play store] United States: grasshopper.app, 2022.
- [12] Flowgorithm, [online learning website], 2021. Available: <http://flowgorithm.org/>. [Accessed: January 03, 2022].
- [13] L. Angelica, "Everything You Need to Know about Flowgorithm", [online document], 2022. Available: <https://mockitt.wondershare.com/flowchart/flowgorithm.html>. [Accessed: January 23, 2022].
- [14] R. R. Gajewski, "Algorithms, Programming, Flowcharts and Flowgorithm" 2018 [Abstract]. Available: <http://weinoe.old.us.edu.pl/node/1432>. [Accessed: January 23, 2022].
- [15] Charntaweekhun, K., and Wangsiripitak, S., "Visual Programming using Flowchart", October 2006 [Abstract]. Available: <https://ieeexplore.ieee.org/document/4141379>. [Accessed: January 25, 2022].
- [16] R. Vesselinov and J. Grego, "Duolingo Effectiveness Study" Conclusion, p. 19, December 2012 [Online Serial]. Available: [https://static.duolingo.com/s3/DuolingoReport\\_Final.pdf](https://static.duolingo.com/s3/DuolingoReport_Final.pdf). [Accessed: January 31, 2022].
- [17] E. Ravenscraft, "500 Days of Duolingo", [online document], 2019. Available: <https://tinyurl.com/2p8jfwct>. [Accessed: February 03, 2022].
- [18] X. Jiang, J. Rollinson, L. Plonsky, E. Gustafson, B. Pajak, "Evaluating the reading and listening outcomes of beginning-level Duolingo courses", January 2022 [Abstract]. Available: <https://onlinelibrary.wiley.com/doi/10.1111/flan.12600>. [Accessed: March 1, 2022].
- [19] J. Morphew, "Frequent mastery testing with second-chance exams leads to enhanced student learning in undergraduate engineering", September 2019 [Abstract]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/acp.3605>. [Accessed: March 4, 2022].
- [20] University of Texas and Austin, Practice and Frequent Testing, [online document], 2022. Available: <https://ctl.utexas.edu/frequent-testing>. [Accessed: April 7, 2022].
- [21] S. Weinstein and S.-W. Wu, "Readiness Assessment Tests versus Frequent Quizzes: Student Preferences", 2009 [Abstract]. Available: <https://files.eric.ed.gov/fulltext/EJ899304.pdf>. [Accessed: April 7, 2022].
- [22] B. Carey, "Frequent Tests Can Enhance College Learning, Study Finds", [online document], November 2013. Available: <https://tinyurl.com/2p8jfwct>. [Accessed: April 12, 2022].
- [23] S. Greving and T. Richter, "Examining the Testing Effect in University Teaching: Retrievability and Question Format Matter", December 2018 [Abstract]. Available: <https://www.frontiersin.org/articles/10.3389/fpsyg.2018.02412/full>. [Accessed: April 25, 2022].
- [24] J. Schwieren, J. Barenberg and S. Dutke, "The Testing Effect in the Psychology Classroom: A Meta-Analytic Perspective", March 2017 [Abstract]. Available: <https://tinyurl.com/29j5xsmx>. [Accessed: May 1, 2022].
- [25] S. McQuiggan, J. McQuiggan, L. Kosturko and J. Sabourin, "Mobile Learning", The great App Checklist, p.358, 2015. Available: <https://tinyurl.com/bdbr366f> [Accessed: May 7, 2022].