JARVAN: AN AUTOMATIC CHECKER FOR REUSED STACK OVERFLOW CODE SNIPPETS IN JAVA SOFTWARE PROJECTS

จาร์วาฬ เครื่องมือตรวจสอบซอฟต์แวร์ภาษาจาวาแบบอัตโนมัติเพื่อค้นหาโค้ดที่นำมาใช้ซ้ำจาก
สแต็กโอเวอร์โฟลว

BY

MR. PHATTHARAPONG	**POOLTHONG**	**5888136**
MISS PANAYA	**SIRILERTWORAKUL**	**6088164**
MISS KANIKA	**WONWIEN**	**6088176**

ADVISOR
DR. CHAIYONG RAGKHITWETSAGUL

A Senior Project Submitted in Partial Fullfillment of the Requirement for

THE DEGREE OF BACHELOR OF SCIENCE
(INFORMATION AND COMMUNICATION TECHNOLOGY)

Faculty of Information and Communication Technology
Mahidol University

2020

ACKNOWLEDGEMENTS

We would like to give acknowledgement to people who has been associated with the development and the accomplishment of this project and thank them for their support during the completion of this project, especially Dr. Chaiyong Ragkhitwetsagul, our advisor for this project. We would like to thank the Faculty of Information and Communication Technology, its instructors, staff members, and members of Mahidol University for the assistance given to us as well. Ultimately, we would also like to thank our families for giving us encouragement while conducting this research This project would not have been this successful without those mentioned and their support that was offered.

Mr. Phattharapong Poolthong
Miss Panaya Sirilertworakul
Miss Kanika Wonwien

JARVAN: AN AUTOMATIC CHECKER FOR REUSED STACK OVERFLOW CODE SNIPPETS IN JAVA SOFTWARE PROJECTS

PHATTHARAPONG	POOLTHONG	5888136 ITCS/B
PANAYA	SIRILERTWORAKUL	6088164 ITCS/B
KANIKA	WONWIEN	6088176 ITCS/B

B.Sc.(INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. CHAIYONG RAGKHITWETSAGUL

ABSTRACT

Today, software development has become one of the essential fields for globalization with the goal to harmonize technology and industry. Through this approach, many people are learning and studying to perform programming. As there are many people learning about programming, some channels have sprung up where people can communicate, learn and exchange knowledge in order to comprehend some concepts of coding with regards to programming, such as Stack Overflow, a popular programming Q&A website. Nevertheless, by exchanging knowledge and solutions, it could cause some people to duplicate some code fragments from Stack Overflow into their own projects. This process is called “code cloning”. Code clones could possibly affect the overall performance of programs. For example, if a code fragment contains defects and is updated, all code cloned fragments previously before the update would have the similar flaws as the original one.

Due to the problem explained above, the developers of this project became motivated to create a software tool called “JARVAN”, which is an automated checker for reused code snippets on Stack Overflow in Java software projects. JARVAN can help to analyze the usage of code clones from Stack Overflow within a software project. Additionally, JARVAN can help to detect software license violation in a Java software project as well because all of code snippets on Stack Overflow being part of a CC BY-SA 4.0 license. JARVAN can create a report of the analysis to a user, which includes an analysis of a reused code used in a software project and warnings for possible software license violations. With regards to this evaluation, JARVAN is sufficiently precise and accurate when on reporting reused code snippets from Stack Overflow. Moreover, JARVAN is fast when searching for reused code snippets from Stack Overflow. Finally, the developers of this project believe that JARVAN can help Java software developers to be aware when reusing the source code from Stack Overflow answers as they have to give an attribution to avoid license violation.

KEYWORDS: CODE CLONE SEARCH, CODE CLONE DETECTION, REUSED CODE, STACK OVERFLOW

93 P.

จาร์วาพ เครื่องมือตรวจสอบซอฟต์แวร์ภาษาจาวาแบบอัตโนมัติเพื่อค้นหาโค้ดที่นำมาใช้ซ้ำจากสแต็กโอเวอร์โฟลว

ภัทรพงษ์	พูลทอง	5888136 ITCS/B
ปณยา	ศิริเลิศวรกุล	6088164 ITCS/B
กนิกา	วนเวียน	6088176 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. ชัยยงค์ รักขิตเวชสกุล

บทคัดย่อ

ณ ปัจจุบันนี้ การพัฒนาซอฟต์แวร์ได้รับการยอมรับให้เป็นหนึ่งในสาขาที่มีความจำเป็นต่อโลกในยุคโลกาภิวัตน์สำหรับการทำให้ภาคอุตสาหกรรมและเทคโนโลยีเป็นอันหนึ่งอันเดียวกัน ดังนั้นผู้คนมากมายจึงเริ่มที่จะศึกษาการพัฒนาโปรแกรม เมื่อผู้คนศึกษาการเขียนโปรแกรมมากขึ้น ก็อาจจะมีช่องทางต่าง ๆ ให้ผู้คนได้ติดต่อสื่อสารแลกเปลี่ยนความรู้ด้านการเขียนโปรแกรม อย่างไรก็ตาม การที่จะแลกเปลี่ยนความคิดเห็นและองค์ความรู้ต่าง ๆ นั้น อาจทำให้เกิดการทำซ้ำกันของโค้ดต่าง ๆ ในซอฟต์แวร์ของคนเหล่านั้นได้ กระบวนการนี้เรียกว่า “โคลนโค้ด” โคลนโค้ดนั้นสามารถส่งผลกระทบต่อภาพรวมการทำงานของซอฟต์แวร์ต่าง ๆ ได้ ตัวอย่างเช่น ถ้าส่วนของโค้ดชุดหนึ่งมีข้อผิดพลาดอยู่หากแต่ได้ถูกแก้ไขแล้ว แต่โค้ดในซอฟต์แวร์ต่าง ๆ ที่ได้ทำการคัดลอกมาจากส่วนของชุดคำสั่งนี้ก่อนการได้รับการแก้ไข จะยังคงมีข้อผิดพลาดเดิมอยู่ ซึ่งข้อผิดพลาดนั้นอาจทำให้ประสิทธิภาพการทำงานของซอฟต์แวร์โดยรวมลดลง จากปัญหาดังกล่าว ผู้พัฒนาได้มีความคิดที่จะสร้างเครื่องมือซอฟต์แวร์นามว่า “จาร์วาพ” ที่ซึ่งสามารถตรวจหาโค้ดที่ล้าสมัยได้ จาร์วาพสามารถวิเคราะห์หาการใช้งานของโค้ดที่ถูกนำมาใช้ซ้ำในซอฟต์แวร์ที่มาจากคำตอบบนเว็บไซต์สแต็กโอเวอร์โฟลว นอกจากนั้น จาร์วาพสามารถตรวจสอบการละเมิดใบอนุญาตในซอฟต์แวร์ภาษาจาวาได้ จาร์วาพสามารถรายงานผลการวิเคราะห์ดังกล่าวให้กับผู้ใช้งานได้อีกด้วย จากผลการประเมินประสิทธิภาพของจาร์วาพสามารถสรุปได้ว่าจาร์วาพนั้นมีความแม่นยำมากพอที่จะตรวจสอบหาการใช้ชุดคำสั่งซ้ำจากสแต็กโอเวอร์โฟลวในซอฟต์แวร์ภาษาจาวา มากไปกว่านั้นจาร์วาพสามารถวิเคราะห์ซอฟต์แวร์ภาษาจาวาได้อย่างรวดเร็ว ท้ายที่สุดนี้ นักพัฒนาเชื่อว่าจาร์วาพจะสามารถช่วยเหลือผู้ที่กำลังพัฒนาซอฟต์แวร์ภาษาจาวานั้นได้หลีกเลี่ยงการใช้ซ้ำของชุดคำสั่งจากสแต็กโอเวอร์โฟลว และพวกเขาควรต้องระบุแหล่งที่มาตามเงื่อนไขเพื่อหลีกเลี่ยงการละเมิดใบอนุญาต

93 หน้า

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

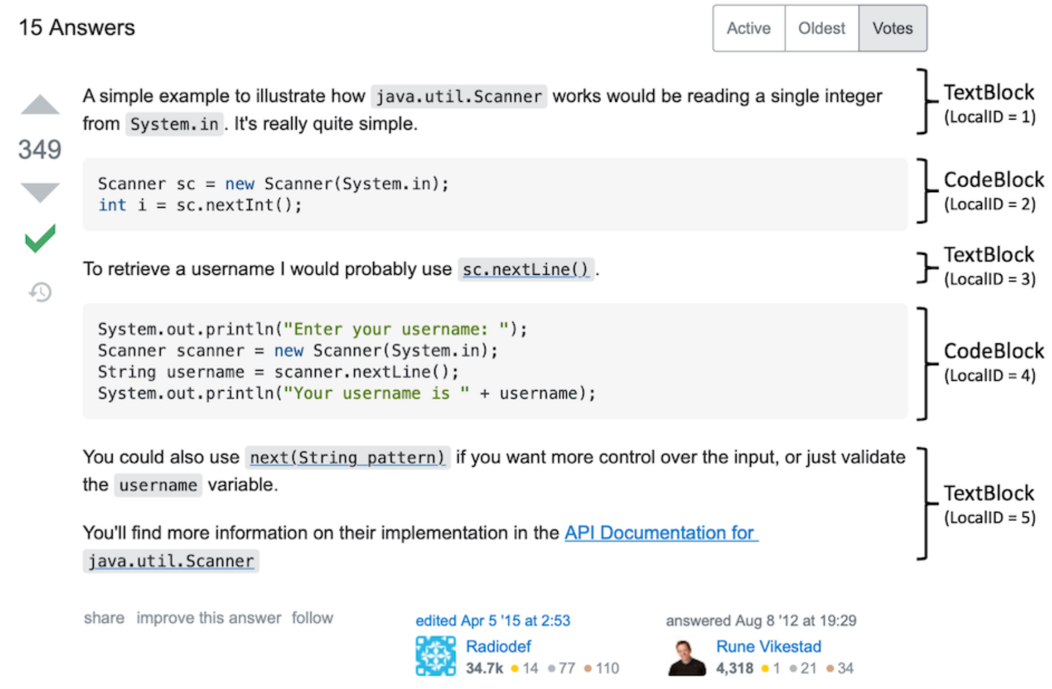
CHAPTER 1
INTRODUCTION

This chapter offers a synopsis of the senior project report. It consists of the motivation for conducting this research, the problem statement, the objectives of this project, the scope of this project, and the target users for this tool correspondingly. Ultimately, the chapters for this report are laid out with clearer details in the report structure.

1.1 Motivation

Code cloning is a process of reusing code fragments which are copied from a project and pasted to another location in the same project or to another project with or without changes. These reused parts are also called “code clones”. Nowadays, code cloning is commonly found in a software development environment. Code cloning can affect both the software maintenance and development process. For instance, if the cloned code fragment contains a bug, all duplicated code fragments would probably contain the same faults as the original code. Typically, a proportion of codes in software industries have been cloned. Previous research shows that between approximately 7 and 23 percent of codes in software developments have been cloned [1]. Additionally, with the rise of the internet, developers can effortlessly access many sources of code snippets online.

Stack Overflow is a large online open-source platform where software codes are collected and developers can discuss several problems they face related to typical software development, programming languages, and other topics associated with computer programming [2]. There are more than 20 million questions asked on Stack Overflow with over 120 millions people visiting the website monthly, which makes Stack Overflow one of the most visited websites in the world [3]. This is mainly because developers can post or answer questions on Stack Overflow. Generally, users can query and look for the best solution that can solve their problems. Among the questions on Stack Overflow, users usually focus on the code block in answers to look out for their solutions (see



15 Answers

349

A simple example to illustrate how `java.util.Scanner` works would be reading a single integer from `System.in`. It's really quite simple.

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

To retrieve a username I would probably use `sc.nextLine()`.

```
System.out.println("Enter your username: ");
Scanner scanner = new Scanner(System.in);
String username = scanner.nextLine();
System.out.println("Your username is " + username);
```

You could also use `next(String pattern)` if you want more control over the input, or just validate the `username` variable.

You'll find more information on their implementation in the API Documentation for `java.util.Scanner`

share improve this answer follow

edited Apr 5 '15 at 2:53
Radiodef
34.7k ● 14 ● 77 ● 110

answered Aug 8 '12 at 19:29
Rune Vikestad
4,318 ● 1 ● 21 ● 34

Figure 1.1: An example of Stack Overflow answers with text blocks, code blocks, and inline code. The LocalID represents the position in a post.

Figure 1.1). Therefore, Stack Overflow is a popular source for searching for solutions for software problems [4, 5, 6, 7].

According to a study by Ragkhitwetsagul et al. [4], many developers who post a source code to answer a question on Stack Overflow are not aware of the license connected to that code, and as a result, they do not include the references for those sources. This duplicating codes from one project to another project with a different license can cause a software license violation [8].

Additionally, reusing source codes from Stack Overflow may lead to several problems if they are outdated codes [4]. By having software that can recognize the source of a code in a project and notify the developers when a code has been updated, the developers can continue to work with their code efficiently and be assured that the code they are using is up to date. Nevertheless, Stack Overflow is not always a perfect source for programming solutions. There is evidence showing that code snippets from Stack Overflow can possibly cause problems to software developers [4]. The reason for this is that it may contain bugs or be outdated. Furthermore, developers do not regularly cite the source where they took the code snippets from. Hence, it is difficult to identify where those code snippets come from, and perhaps the code snippets may be obsolete.

During software development, the idea of notifying developers to recognize where they take code snippets from can be beneficial for developers. For instance, they can be reminded that the codes used are taken from an online source. This may avoid software license violations. Second, it can be more advantageous for developers if they can receive a notification that states that their code snippets taken from Stack Overflow are obsolete, as the code might have already been modified on Stack Overflow. Third, it can help developers to practice good programming behaviors by giving credibility to the owner of the code snippets. According to Stack Overflow, the code in the answers is subject to the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. This CC BY-SA 4.0 license requires attribution by giving credit to the owner of the answer.

With the reasons asserted previously, this project presents "JARVAN" as a tool that assists developers to detect code snippets copied from Stack Overflow to help acknowledge source codes from Stack Overflow. Additionally, it can also inform developers to realize that the code snippets that are taken from Stack Overflow are already obsolete. The developers of this project believe that JARVAN can decrease the number of obsolete codes in real-world software development. Moreover, they anticipate that it can let developers practice a norm so that owners of original code can be honored rather than copied.

1.2 Problem Statement

This project deals with the following problems of software development:

1. The copied code snippets from Stack Overflow posts can become outdated after being copied to a local software project.

2. The copied code can violate the CC BY-SA 4.0 license if the developers do not give attribution to the Stack Overflow posts.

1.3 Objectives of the Project

The project focuses on completing the following objectives.

1. To create a software tool that can detect the Java code that the developers have reused from Stack Overflow.

2. To create a software tool that can report if the cloned Java code has been updated, edited, or modified on Stack Overflow while at the same time checking whether the software license conflicts.

1.4 Scope of the Project

The scope of this work is as follows:

1. The proposed tool only uses the data from Stack Overflow website.

2. The proposed tool allows users to work with GitHub repositories only.

3. The proposed tool is designed to work with Java source code files only.

1.5 Target Users

The target users of this project are Java developers who usually adopt code snippets from Stack Overflow to their projects. The project would be beneficial for these developers to review their source code. By using an automated tool that can check the outdated version of the source code and return the result as a report, developers would be allowed to obtain an analysis report of their code. Moreover, the developers can use this tool to check which parts of the source code come from Stack Overflow and add attributions to those parts of code to avoid software licensing violations.

1.6 Report Structure

This report contains three chapters. The first chapter is the introduction. It contains the motivation, problem statements, objectives, scope, and target users of this project, and the report structure. Secondly, Chapter 2 involves the background concepts for this project and its related work. Thirdly, Chapter 3 discusses the design and architecture of this project, and the system components, along with the use case diagram and the context diagram. Next, Chapter 4 explains the implementation of JARVAN by clarifying how the system and the techniques and tools that have been used can be implemented.

Next, Chapter 5 contains an evaluation of the JARVAN system including its precision results, performance evaluation, and user evaluation. Ultimately, Chapter 6 concludes the important issues in this project along with problems and a plan for future research.

CHAPTER 2
BACKGROUND

2.1 Code Clones

Code clones are the result of copying a part of code and pasting it from one project to another project with or without adaptation. This activity leads to the creation of two copies (or more) of the same source code. When this has been completed, the copied piece of code will have some similarities to the original piece of code. These two similar code fragments form a "clone pair". There are four types of clones. Type-1 clones are identical code fragments that are different through the comments, white spaces, and layouts they contain. A Type-2 clone has syntactically identical code fragments with additional variations in identifier names, literals, and types. A Type-3 clone is a copied part of code with minor modifications that are changed, added, or removed statements in any lines of code. Lastly, a Type-4 clone has a similar logic computation compared to the original one, but it is written in a different syntax.

2.1.1 Clone Detection Process

The following section will provide basic information about the basic steps in the code clone detection process. First, there is the pre-processing step which starts with the removal of uninteresting parts after that the source code has become a set of disjointed fragments called source units. Source units are divided into smaller units depending on the comparison technique. The second is the transformation step through which the raw source code is transformed into an intermediate representation which is often called "extraction" in the reverse engineering community. Extraction is done through tools that support the normalizing of transformations such as tokenization, parsing, and control and data flow analysis. For simple normalizations, the process should be very fast as it uses a short time to remove white-space and comments, normalizing identifiers, pretty-printing of source codes, and structural transformations. The third is match detection.

This involves the transformed source code being used in a comparison process to find matches with the original code. After this, the list of transformed code clone pairs are formatted to align with the original codebase. Next, in the Post-processing/Filtering step, clones are ranked by using manual analysis. The last step involves the aggregation that combines with clone pairs into clone classes.

2.1.2 Techniques and Tools

The techniques and tools used for code clone detection can be classified into five categories:

1. Textual approaches or text-based techniques: The source code has undergone minor changes or no changes before they are compared to the original. Often time original source code is directly used to find pairs of code clones.

2. Lexical approaches or token-based techniques: First, the source code to be compared is converted into "tokens". The tokens are then scanned in the same subsequence to return the corresponding units as clones.

3. Syntactic approaches: This approach utilizes abstract syntax trees (ASTs) or a parser to convert source programs into parse trees. They are processed to search clones through Tree matching approaches (Tree-based techniques) or Metrics-based approaches (Metrics-based techniques).

4. Semantic approaches(Graph-based): The source code is shown through a program dependence graph (PDG). The PDG has nodes and edges with suggested expressions and data dependencies. The nodes of the graph represent expressions and statements. The edges of the graph represent control and data dependencies. This representation abstracts from the lexical order in which expressions and statements occur to the extent that they are semantically independent.

5. Hybrids: A hybrid approach is a syntactic approach combined with a semantic approach.

2.1.3 The Issues with Code Clones

Code cloning is efficient and advantageous across various approaches. However, it can be dangerous to evolve and maintain the software as there are many issues with reused source codes from the Internet. For example, Stack Overflow is a very popular website among the developer community, but at the same time, it is also the website through which codes are mostly reused by millions of users [4]. A code clone search tool called "Siamese" found 72,365 Java codes of clone pairs on Stack Overflow. Moreover, a study by Ragkhitwetsagul et al. showed numerous issues that arose from Stack Overflow answers. For example, these included outdated solutions, incorrect solutions, mismatched solutions, and bugs from the code. The researchers conducted a survey among developers that had contributed to the Stack Overflow website using questionnaires. Over half of the developers answered that they had been notified of outdated codes before and that they had never checked for licensing. However, the developers never fixed the source codes which led to other visitors or the developers themselves getting unfavorable effects from the used source codes as the source codes became toxic code snippets.

2.1.4 Code Clones from the Internet

Nowadays, the Internet or online platforms are becoming the largest place where a plethora of source codes can be stored and proliferated. There are several developers who prefer to reuse source codes from the Internet and adapt them by copying, duplicating, and pasting them into their projects. As a result, projects often contain code clones. Research by Roy et al. showed that between 7% and 23% of the code in a typical software system had been cloned [1].

2.2 Issues from Stack Overflow Code Snippets

Toxic code snippets are frequently created when developers post their software project's source code onto the Internet or open-source online platforms such as Stack Overflow, and GitHub. That code is then copied and reused for another project by other developers. There are many source codes for several programming languages that the developers can search for easily on the Internet. However, as time changes, certain technologies and knowledge might develop more, so those source codes may become obsolete, leading to the code requiring an update [5]. TThis is the reason why the developers should be aware and notice that not all the source codes they find on the Internet can be utilized. According to Zhang et al. 58.4% of code block answers were already obsolete on Stack Overflow [6]. Besides, from those obsolete answers, only 20.5% of them had ever been updated after they had been posted [6]. As a result, the developers who may not have noticed the obsolete or reused codes may run into several problems afterwards.

2.3 Code Clone Detection vs. Code Clone Search

When looking at the definition of code clone detection, code clone detection is the process of finding identical or similar pieces of code known as “code clones” within a source code. Code clone detection only looks up clones within its repositories. Nevertheless, it is important to recognize that a code clone search is different from code clone detection. The reason for this is that a code clone search requires a query from a user to look up clones in its repositories which match a query. Therefore, the major difference between a code clone detection and a code clone search is that code clone detection only finds clones in its repositories, but a code clone search requires a query to find clones in its repositories which are similar or identical to the given query.

2.4 Siamese: Code Clone Search Tool

“Siamese” is a tool that can search for pairs of code clones with a high accuracy and scalability through the use of multiple code representations [9]. It was developed with Java programming language and can therefore detect code clones in both Java and Python programming languages. In addition, it uses Elasticsearch 2.2.0 to index and retrieve the source codes that need to be searched for clones. Consequently, Siamese can detect all types of clones. Moreover, Siamese adopts the clone detection process to transform raw source codes into intermediate representation formats. Finally, it can be noted that Siamese can be executed by using the command line.

2.4.1 Architecture of Siamese

This section will explain the architecture of Siamese in details. Figure 2.1 provides a visual representation of the system architecture of Siamese. There are in total 6

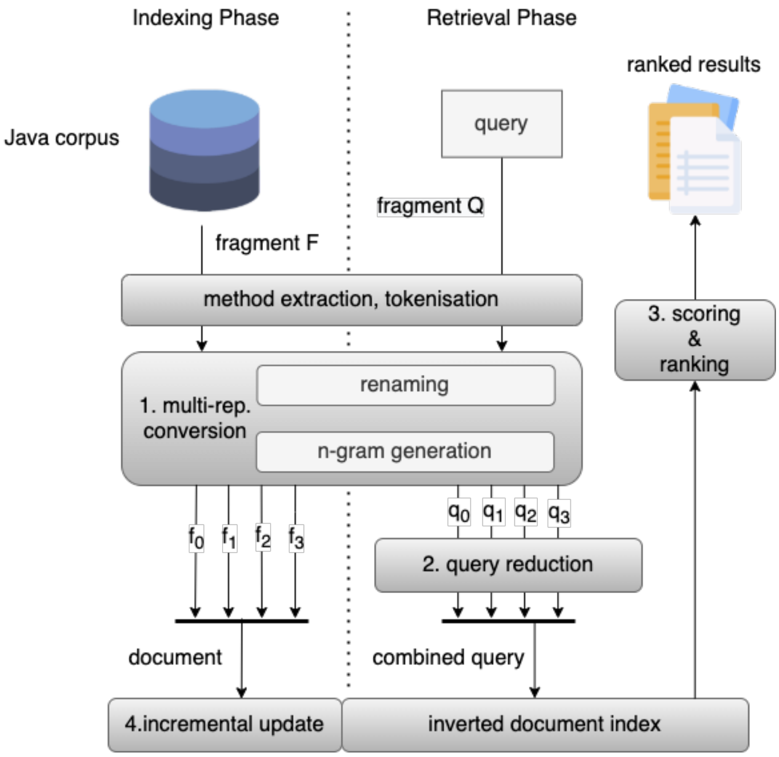


Figure 2.1: The Siamese system architecture

main components that belong to the Siamese architecture.

Indexing Phase

During the indexing phase, Siamese will create a searchable code index by using computations from the given source code base. Because Siamese is a token-based tool, and is tough to incomplete or uncompilable code fragments, if the method parsing fails, it falls back to store the source code at a file level. Consequently, each input code as a fragment will be tokenized into a stream of tokens and transformed into a multi representation so that it can generate four types of code representations to capture different levels of code structure before they are kept in the index. This process is burdensome, however, and therefore occurs a lot less than the querying phase.

Querying Phase

Querying is the primary activity of Siamese. During the querying phase, the user can query a code through the search tool and the tool then returns the clone result back to the user. This is similar to the indexing phase, for which the source code is also provided and prepared in the same way. After a code has been queried, a query reduction module will prevent the query from having a format with multi representations format and rewrite it. After that, it will merge 4 reduced queries into a single search request and perform a search for it through the search engine. Finally, Siamese will retrieve code fragments that were indexed and examined and match those with the combined query and calculate their ranking based on closeness before presenting the result to the user.

Multi Representations

There are four representations of detection clones that can be used to present each clone type with a high precision.

1. Original representation: a representation that contains tokens from the original source code to a stream of tokens.

2. Type-1 representation or Type-1 clone search: a representation that contains tokens from the original source code to a stream of n-grams.

3. Type-2 representation or Type-2 clone search: a representation that represents tokens through normalized n-grams, such as an identifier, literals, and type tokens, which will be replaced by a representative tokens.

4. Type-3 representation or Type-3 clone search: a representation that normalized n-grams with all tokens are replaced by the representative tokens, except Java punctuators.

Query Reduction

Siamese uses a query reduction technique to rewrite the query from the user so that only the rare tokens are chosen, and the time occur frequency is omitted by removing the frequent ones. Query reduction can reduce the occurrence of highly irrelevant code retrieval, increase the search speed, and avoid false-positive results.

Scoring and Ranking

Siamese applies the ranking function and Apache Lucene's scoring method to created a list with all clone results ranked by the similarity. They use the scoring and

```
// r0 (n-gram size = 1)
public static int binarySearch1 ( int arr [ ]
, int key , int imin , int imax ) ... ;
else return imid ; }

// r1 (n-gram size = 4)
publicstaticintbinarySearch1 staticintbinarySearch1( intbinarySearch1(
int binarySearch1(intarr (intarr[ ...
;elsereturnimid elsereturnimid; returnimid; }

// r2 (n-gram size = 4)
publicstaticDW staticDW( DW(D W(DW (DW[ DW[ ] W[ ],
[ ],D ],DW ,DW, DW,D W,DW ,DW) DW) W){if ){if( ...
);elsereturn ;elsereturnW elsereturnW; returnW; }

// r3 (n-gram size = 4)
KKDW KDW( DW(D W(DW (DW[ DW[ ] W[ ], [ ],D ],DW ,DW,
DW,D W,DW ,DW) DW){ W){K ){K( K(W K(WO (WOW ...
KK(W WOV, OV,W V,W) ,W); W);K ;KKW KKW; KW; }
```

Figure 2.2: An example of multi representations

ranking technique are based vector space model (VSM) representation technique to convert the documents into k-dimensional weight vectors. A query vector and a document vector are computed as the result of the relevance score received by Apache Lucene, which will make the speed faster when searching and ranking. It looks at the highest to lowest scores to choose the candidates. If the clone results have the same score, Siamese will use an alphabetical order technique to rank the results and return the list with rankings as top n result to the users.

Incremental Updates

Siamese has an advantage as it allows its index to be incrementally updated, so it can maintain a large-scale code repositories index. The index is able to be modified without having to re-index all repositories again. Because Siamese is flexible, it allows users to modify the index by adding, editing or deleting the code without damaging the data in the indexes.

2.4.2 Siamese Implementation

The implementation of Siamese uses Elasticsearch for scalable code indexing and retrieval. Siamese consists of preprocessing, the multi-representation module, query reduction module, and scoring module on top of the Elasticsearch. The Java method parsing is done by the Java parser and the tokenization is done by using the Antlr4 lexer with Java 8 grammar. The parser, tokenizer, and normalizer are language-dependent while the multi-representation module, query reduction module, and scoring and ranking modules are language independent.

Selection of N-Gram Sizes

Siamese selects the size of an n-gram of 4 for the three code representations in the multi-representation module (Type-1 representation, Type-2 representation, and Type-3 representation). The size of 4 is enough to capture code sequences but still allows small modifications within a statement. However, the original representation chooses 1- gram to function as a keyword search, which is advantageous when looking for a specific token among the cloned fragments.

Choosing the Query Reduction Thresholds

Siamese uses two data sets, which are Bellon's clone benchmark and Qualitas corpus, to select the optimal threshold values for the query reduction module [9]. An observation revealed that the document frequency of the original representation would drop sharply and start rapidly converging to one for approximately 10% of all the documents in the corpus. The results show similarities when compared to the observation of the Type-1 representation. The document frequency of the Type-2 representation and the Type-3 representation were also converged to one although they drop to one quite slower than the original representation and the Type-1 representation because of the token normalization. However, they are almost distinct for 10% of all the documents. These results occurred with both the two data sets. Therefore, Siamese picked the same query reduction threshold for all representations to be at 10

2.5 Related Work

This part focusses on discussing the research papers and tools that are related to this project. The selected papers present closely similar interesting ideas of studies but focus on different aspects. However, they all focus on studying code clones to analyze problems from source codes that have been reused or outdated on Stack Overflow answers.

2.5.1 Siamese: a scalable and incremental code clone search via multiple

code representations

This research paper by Ragkhitwetsagul and Krinke, published online in 2019, studied clones by focusing on the differentiation type of the code clones to order the specific clones through their ranking results. They also implemented a clone search tool called 'Siamese', which is able to evaluate a clone data set on three established sites with a high search accuracy and scalability. They then conducted an experiment by setting a question and using the Siamese code clone search tool across with more than 100,000 projects from GitHub, Stack Overflow, and other several open sources to demonstrate that Siamese could find the code clones efficiently.

The result from the experiment shows that for a large number of code clones, the code was reused from several open-source sites including Stack Overflow when using the 4 code representation techniques to detect the different types of the clone. Therefore, it can be said that Siamese offers a high-level code clone search engine that allows the developers to find similar codes and detect plagiarism in their software.

2.5.2 Stack Overflow in Github: Any Snippets There?

This research paper by Yang et al., published in 2017, discussed the clones found on Stack Overflow, the largest online platform among the developer community, and GitHub. Specifically, they focus on the Python snippets shared between Stack Overflow and GitHub. They are interested in studying how to find code snippets, when the code snip- pets are used, and how developers reuse those code snippets.

The result of the paper presented an analysis of the 1.9M Pythons snipped on Stack Overflow. The quantitative analysis shows that there are a rare number of exact code clones. However, they found 405k of the source codes that seemed to be code clones from Stack Overflow and GitHub. This is still a significant number, so it was concluded that Stack Overflow and GitHub both had source codes that had undergone code cloning.

2.5.3 An Empirical Study of Obsolete Answers on Stack Overflow

This research paper by Zhang et al., published in 2019, studied the obsoleted code answer on the Stack Overflow. They collected 15 million questions, 23 million answers, and 62 million comments on Stack Overflow. The developers perform the analysis manually by reading the comments in each Stack Overflow post. They also discussed and gave an example of findings from their experiment including the user complaints, the updating answers, or the unaccepted answers. Lastly, they gave the explanation of the reasons why the codes become obsolete.

The purpose of the paper was to highlight the problem of the obsolete answers occurring on Stack Overflow so that Stack Overflow could improve its mechanics to handle and maintain outdated code problems. The finding of those outdated codes could help with the identification of valid versions for users and five reasons why the approaches methods should be used in the future to detect obsolete answers in Stack Overflow.

2.6 Chapter Summary

This chapter gave information about previous studies regarding code clones, which in particular looked at the clone detection process, the techniques and tools used for clone detection, and issues with code clones and those available on the internet. Moreover, the issues with Stack Overflow code snippets and the differences between code clone detection and a code clone search are described as well. Moreover, an explanation of how the Siamese tool works was offered by depicting the architecture of the Siamese, including each phase in its architecture. Additionally, the implementation of Siamese was also described. This covers the selection of N-gram sizes and choosing the query reduction thresholds. Lastly, this chapter provided related pieces of research that are associated with this project.

CHAPTER 3
ANALYSIS AND DESIGN

This chapter provides an analysis of the design of this project including definitions of outdated Stack Overflow code snippets, an overview of the proposed system, its use cases, data flow, system architecture, and finally the components, techniques and tools that belong to the JARVAN system. Moreover, it provides a timeline for this project from the beginning to the end in detail. Lastly, a brief summary of the chapter is offered.

3.1 Definition

An Outdated Stack Overflow code snippet stands for a code snippet that is copied from Stack Overflow and reused in a software project. The version of the code snippet that is on Stack Overflow is newer than the clone in the software project.

3.2 Overview of the Proposed System

The proposed system is called “JARVAN” and is known as an automatic checking system to search for outdated Stack Overflow code snippets in a Java software project. JARVAN can find where source code snippets come from by using the code clone search technique presented by Siamese. It is based on the Stack Overflow data from the Stack Exchange Data Dumps and the SOTorrent dataset and includes the tracking of both the versions and code blocks in Stack Overflow answers. The tool is believed to be able to help software developers in the following ways. First, it helps to avoid the issues caused by obsolete copied codes. This is mainly because JARVAN can detect the version of the source code and notify the developers that the source code in their projects have been updated compared to the version of that source code on Stack Overflow. If those source code snippets have been updated, the developers can then access the latest source code on Stack Overflow and update their local source code to the latest version. Second, JARVAN helps developers to avoid software licensing violations caused by the developers rarely providing references to or citations of original source codes that they reused from

Stack Overflow.

The objective of this chapter is to present the JARVAN system and its components in detail. Based on the background studies and related work covered in Chapter 2, the developers come up with a methodology on how to implement the system. This chapter allows readers to understand the in-depth details of the JARVAN system and provides the conspicuous methods that were used to develop the system. This chapter will provide a further analysis of JARVAN's design by providing its use case diagram, context diagram, system architecture and system components.

3.3 The JARVAN System Use Cases

Figure 3.1 illustrates the use cases for the JARVAN system. There are primary and secondary actors involved with the system which are the developers and GitHub respectively. There are four use cases in the system, which will be described below as follows:

The first use case is 'Log into GitHub'. During this process, developers must provide their GitHub account to JARVAN. This use case is important because the system needs to obtain the authorization from the developers in order to retrieve the developers' GitHub repositories in the next use case.

The second use case is 'Get a repository'. This use case happens when developers specify which Java repository in their own GitHub repositories they want JARVAN to analyze. After getting their selected repository, GitHub acts as the secondary actor which takes a request from JARVAN. Thenceforth, it returns authorization for the developers to access the specified repository in the system.

Another use case is 'Get report'. When developers perform this use case, the system takes the specified repository that the developers selected to perform a code clone search for the source code snippets in the selected software projects and the code snippets on Stack Overflow. Once the searching is done, the system generates a report for the developers that describes the analysis result in detail.

The last use case is 'Get issue'. After the system has performed the code clone search, the system will request that an issue report gets created on the developer's GitHub repository. The request is passed to GitHub, which in this case is the primary actor.

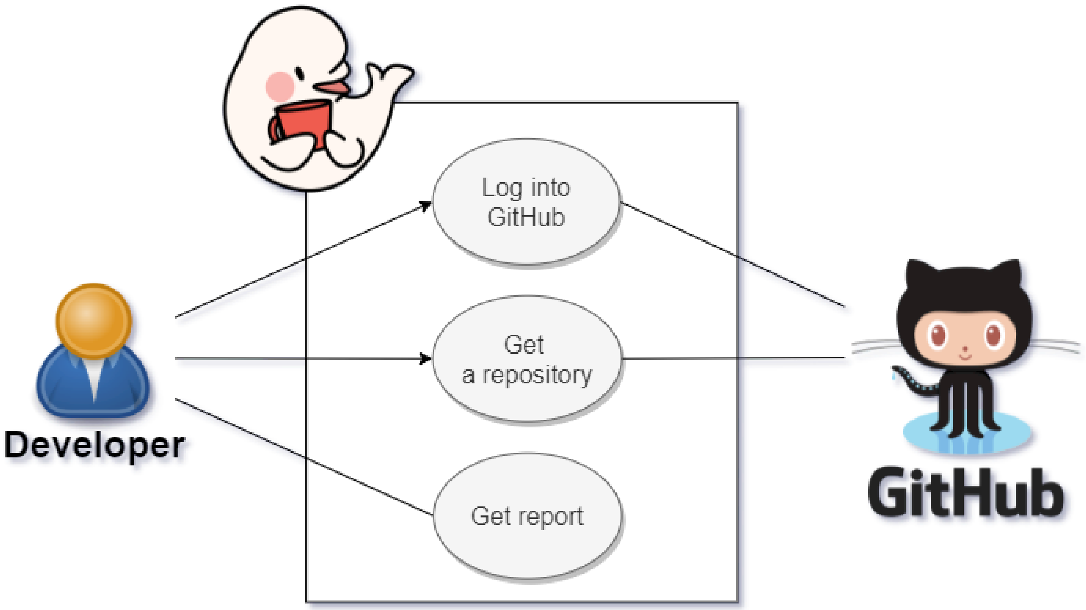


Figure 3.1: The use case diagram of the JARVAN system

3.4 Data Flow Analysis (Context Diagram)

Figure 3.2 depicts the data flow analysis by showing the context diagram of the JARVAN system. The data comes from two external entities which are the developers and GitHub. The developers must provide their GitHub account in order to use the system. The system then passes the developer's GitHub account onto GitHub. After the authorization is successful, GitHub returns the authorization token to the system. Then, the system sends a request to GitHub to retrieve the developers' GitHub repositories. GitHub then sends back the repositories to the system. Next, the system shows a list of repositories to the developers, and then the developers select the repository that they want and perform a code clone search with Stack Overflow on it. Once the developers have selected the repository, the system sends another request to GitHub to clone a source code from that specific repository after which GitHub sends back the source code to the system. The system can then process the code clone search. Once the process has been completed, the system will generate a report for the developers and also make a request to GitHub to open an issue.

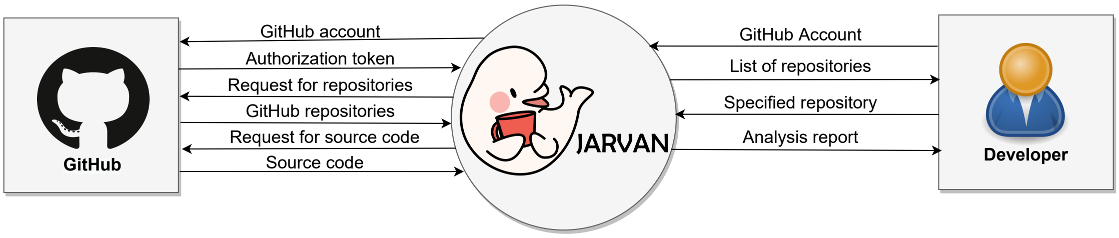


Figure 3.2: The context diagram of the JARVAN system

3.5 System Architecture

Figure 3.3 represents the system architecture of the JARVAN system. It can be seen JARVAN consists of 5 main processes that should be considered when the system is developed. The first process involves the conducting of an analysis of Stack Overflow data. The Stack Overflow data is first retrieved from the Stack Exchange Data Dumps and SOTorrent dataset [7]. Then, the filtering process is applied to the data retrieved from the Stack Exchange Data Dumps and SOTorrent. The filter only receives posts on Stack Overflow which relate to Java and are accepted answers only, while other posts are ignored. After the filtered data has been retrieved, the dataset for JARVAN is ready. Secondly, posts on Stack Overflow have a version history, and the goal is to develop linkages between different versions of the same posts, which can be older or newer versions. After the linkages have been developed, the versions of posts in the dataset can become traceable. For this, a version-traceable Stack Overflow Java dataset is used to build up the JARVAN search index on Siamese. This will enable the search index to be able to track the history of Stack Overflow posts. Thirdly, JARVAN will incorporate the JARVAN GitHub connector to connect to GitHub's repositories to retrieve Java repositories so that they can be processed on the JARVAN system. The fourth component involves the creation of a JARVAN query system. The query system clones Java source code from the GitHub repositories from the previous component and uses the code snippets in the cloned project to search for code clones within the version- traceable Stack Overflow Java search index. The last process involves the development of the reporting system. The reporting system is used to generate a clear and understandable representation of the result from the query system which can then be delivered to a user.

Siamese is a part of the JARVAN engine. The tasks that Siamese is responsible

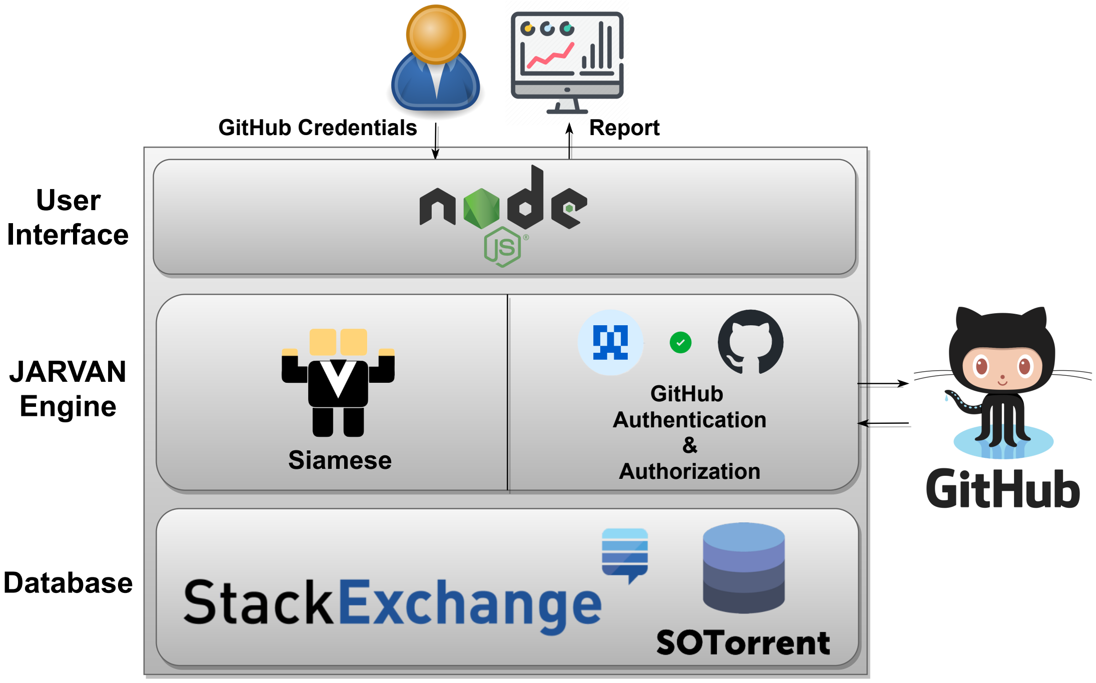


Figure 3.3: The system architecture for JARVAN

for include the indexing of the source code, and the searching for source codes. With regards to other tasks, JARVAN is responsible for tasks through the back-end in Node.js as it handles the SQL, HTTP requests, the front-end implementation and user interfaces, the reports, the database connections, and also the GitHub authentication.

Next, in the system components section, each component will be described in more detail.

3.6 System Components

This section explains the system components of the JARVAN system in more detail.

3.6.1 The creation of a Stack Overflow Database

JARVAN uses the Stack Exchange Data Dumps and SOTorrent dataset to extract data from the Stack Overflow. Both the Stack Exchange Data Dumps and SOTorrent contain a collection of questions and answers from Stack Overflow and provide access to the content version history of Stack Overflow posts [7]. Based on the full Stack Exchange Data Dumps and SOTorrent dataset, JARVAN consists three tables, which can be seen

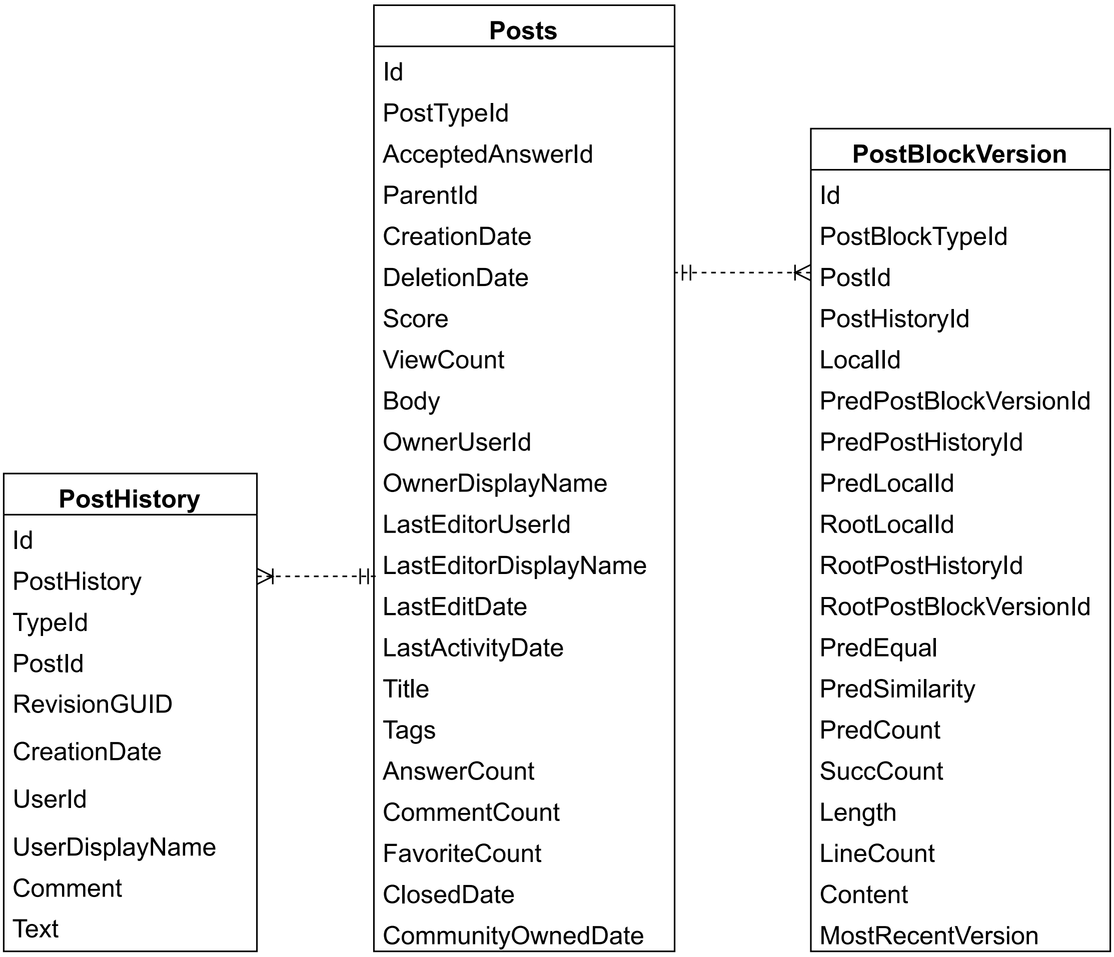


Figure 3.4: The entity relationship diagram of the SOTorrent dataset

in Figure **??**. Figure **??** shows that JARVAN uses the PostHistory, Post, and Tags from the Stack Exchange Data Dumps and also the PostBlockVersion table from SOTorrent to get the relevant data. The PostHistory table is a table that holds the history of posts in Stack Overflow. JARVAN uses this table to find out for which posts the code has been updated. The Posts table holds the content of questions and answers of the posts on the Stack Overflow. JARVAN uses this table as the main table to retrieve the required data from. The PostTags table is the table that lists the types of posts that JARVAN can use to select Java posts. The PostBlockVersion table is the table that used to collect the edited versions of Post blocks on Stack Overflow. From these three tables, JARVAN is able to find posts that are the accepted answers in Java along with all of their history.

3.6.2 Creating Stack Overflow History and Search Index Creation

After JARVAN receives the data from the first step, JARVAN gets data that contains both text blocks and code blocks. Text blocks are parts of texts in Stack Overflow answers and code blocks are parts of Java code snippets in the answers. JARVAN selects only the code blocks. It then uses Siamese to index the data and stores the selected versions of the code blocks before it extracts all versions of the code blocks from the database. Once this is done, files are created in JARVAN to categorize the versions of the source code. A file for one version should include the Name of PostID, LocalID, and HistoryID so that the version of a source code can be defined starting from the original version to the recent version. This process is continued until it has covered all the versions. After that, JARVAN uses the data to create a search index. JARVAN also uses a Siamese code clone searching tool to create a search index. When JARVAN is creating an index, the system will store the indexed code blocks in the form of a “document”. Document is a term that is usually used for data stored in a search engine, which can consist of several properties. The developers design the properties of a code block document in the system so that it includes the following information: id, name, code, pointer to the previous version, and pointer to the next version of the source code. With this design, JARVAN can track the versions of the source code by looking at the relationships between the documents. When JARVAN finds another version, the system will determine whether the source code is an updated or outdated version. Otherwise, if either the next version or the previous version attribute is null, JARVAN can determine that the source code is either the latest version or the original version.

3.6.3 GitHub Connector

Figure 3.6 illustrates how the GitHub Connector component works. This component is used to communicate with the external service of JARVAN, which is GitHub. There are 3 steps in this component which will be described as follows.

1. JARVAN uses OAuth as an authorization framework for the system [10]. Firstly, a user must provide their GitHub account to the system. Then, the system will communicate with the GitHub API to request for authorization from GitHub. After

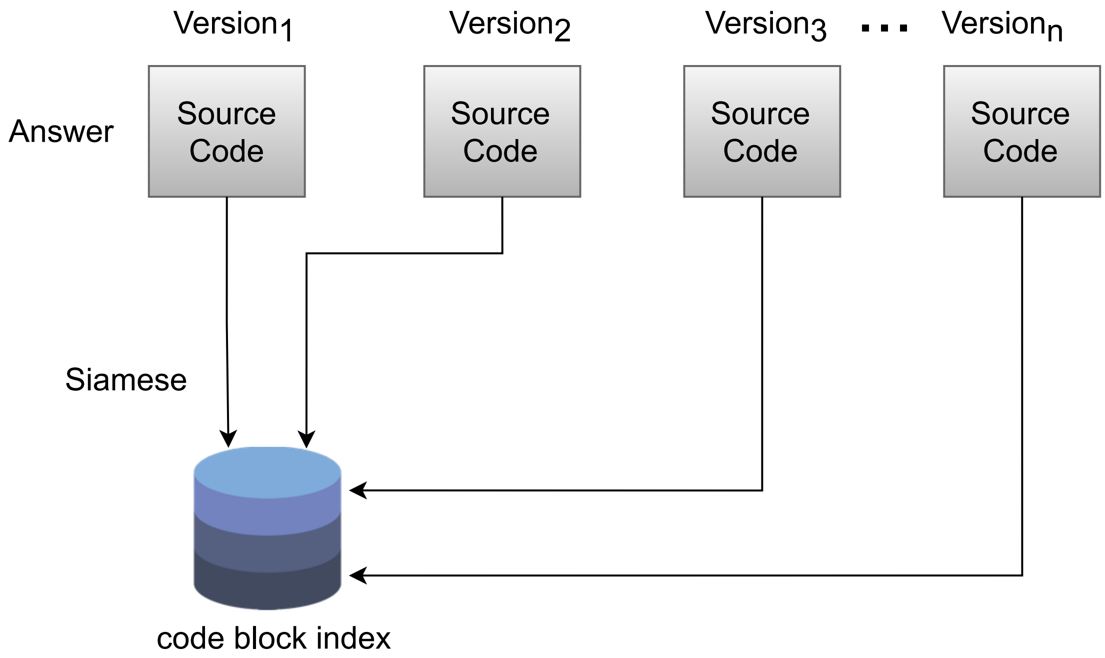


Figure 3.5: The diagram showing the indexing process where each Stack Overflow code block with its complete history is inserted into the Siamese code block index.

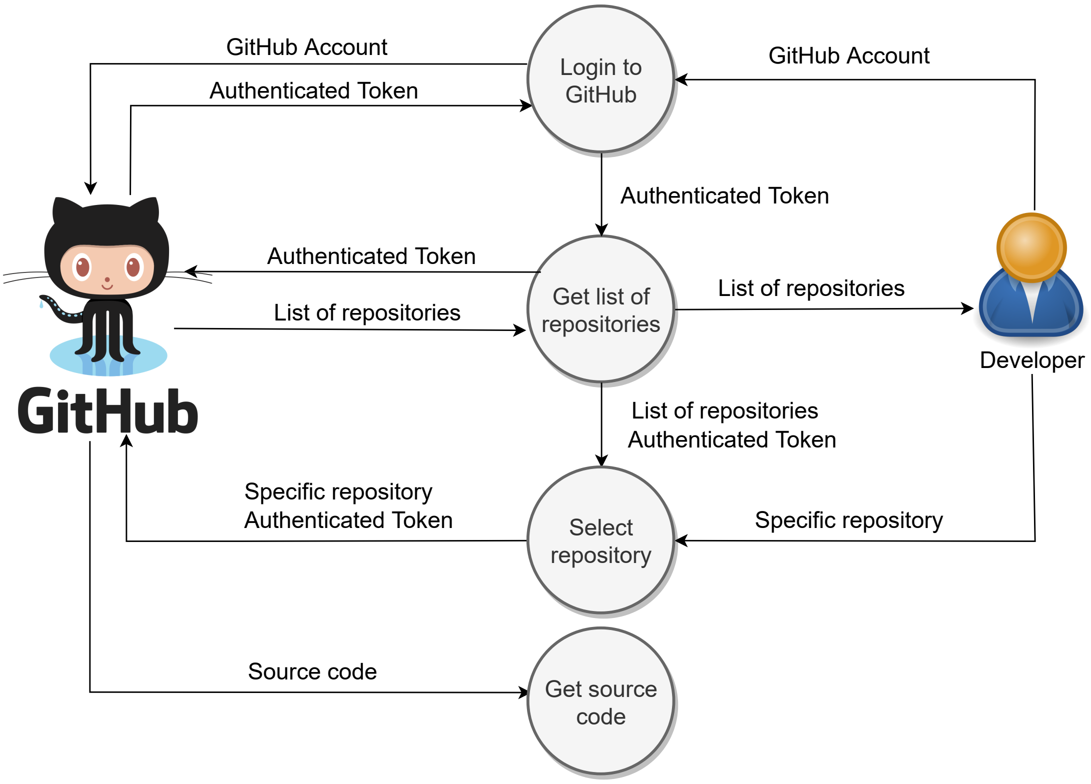


Figure 3.6: The data flow diagram describes how JARVAN connects to GitHub

that, if GitHub grants authorization, it should provide the system with an authentication token to confirm that the system is allowed to access the user's GitHub account. Once this has been done, a confirmation will be shown to the user to confirm that authentication was granted by GitHub.

2. After JARVAN retrieves the authentication from the user to access the user's GitHubaccount, it makes a request to the GitHub API to obtain the user's repositories. Consequently, GitHub will provide a list of the user's repositories to the system. Eventually, the system will show the obtained repositories to the user for the last step.

3. Once the user sees the list of repositories, the user can select a repository through the system. The system then communicates with GitHub again to request for a specific repository. The GitHub should subsequently provide the user's repository with its source code to the system. At this point, JARVAN has obtained the user's repository to conduct further work on it in the next step.

3.6.4 Code Clone Search

For this component, there are two main steps which the system performs.

1. As said before, at the beginning of this step, JARVAN obtains the user's repository. The system then looks for the Java files to conduct the search in. Figure 3.7 illustrates this as an example. The system finds the Java file and after that is done, it extracts the file into methods. After that, the system uses the extracted methods as queries to search for code clones. It passes the queries to the code clone search engine after which the engine does through the dataset to search for the query. As a result, the engine produces n results which show the code clones found in the database.

2. Once the results are obtained from Siamese, the obtained results are looked into further to consider whether the results are reused codes from Stack Overflow. Moreover, the system applies a filter through which it filters out outdated codes on Stack Overflow. As a result, the system will develop two sets of results which

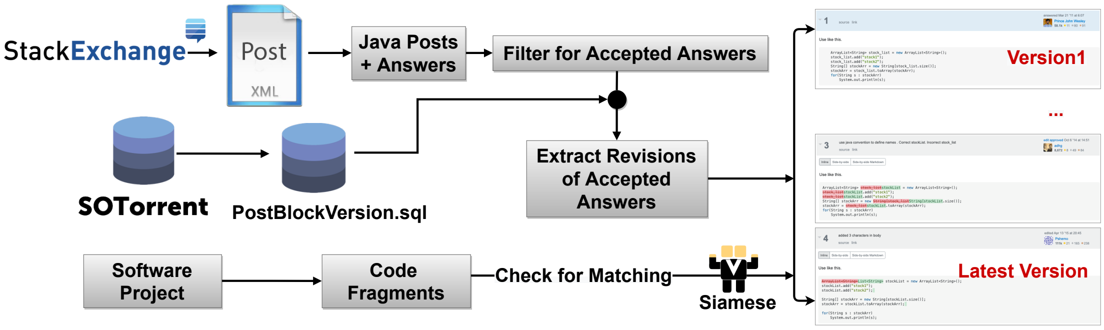


Figure 3.7: The diagram showing how code clone searching works

are the reused code result and the outdated code result. It should be noted though that the reused code result subsumes the outdated code result.

The time complexity of this code clone search function can be analyzed as follows. According to Wang [11] , the time complexity of the inverted index, which is the underlying data structure of Siamese, is O(|q|*|L|), where |q| is the number of terms in the query and |L| is the average number of documents for each posting (list of documents containing the term). Thus, the average time complexity of JARVAN is also O(|q|*|L|). It can be seen that this depends on the size of the query and the average number of documents in each posting. The searching time can be slower if the code search index is larger, but the rate of slowing down rate should be low as it is based on the average number of documents in each posting, and not all the documents in the index.

3.6.5 Report Creation

When the two sets of results have been collected, JARVAN then creates a report for the user. The report consists of a table representation with an explanation of the code clones found and warning messages regarding license conflicts.

3.7 Techniques and Tools

The following section includes the tools and techniques that the project has used during the development of the JARVAN system.

3.7.1 GitHub OAuth

GitHub OAuth is an application that uses GitHub as an authenticator after a user has been granted access to the application. This project uses GitHub OAuth as a gateway to authenticate permissions for users. When a user logs in with one's GitHub account, GitHub OAuth provides an access token for the system to access the user's personal information, including the user's GitHub repositories. This shows that the system uses GitHub OAuth to obtain the user's GitHub repositories.

3.7.2 MySQL database

MySQL is a popular open source SQL database management system for which the distribution and development is supported by the MySQL. MySQL databases are relational databases that store data in separate tables. The SQL part of MySQL stands for "Structured Query Language" which is a standardized language used to access databases and defined by the ANSI/ ISO SQL Standard. This project uses MySQL databases as a server to allow multiple users to manage and create massive databases through MySQL. The database allows the user to create a server that can be used to connect to the database they need to manage the database.

3.7.3 Node.js

Node.js is a cross platform used to run on a web browser by using a JavaScript that was created through Chrome's V8 JavaScript engine. It allows the developers to write a JavaScript code in the command line tools and server-side scripting. Node.js can then be utilized to create the dynamic web page content before the page is sent to the user's web browser.

3.7.4 Siamese

Siamese (Scalable, incremental, and multi-representation) is a code clone search system powered by Elasticsearch 2.2.0 [9]. The system offers high-level code clone search approaches, including code normalization, n-grams, and query reduction techniques. Moreover, it can search for clones of Type-1, Type-2, Type-3, and Type-4 from a large compilation of Java source codes. This project adapts Siamese to help search for code clones and compare them with the SOTorrent database which is a collection of history posts of code block answers on Stack Overflow. When Siamese finds the code clones of a project, it can analyze whether the source code may have been reused from posts on Stack Overflow. After that, it can approach the original source code and find any updates.

3.8 Chapter Summary

This chapter explains the analysis and design of the JARVAN system including the use case diagram of the JARVAN system, the data flow analysis, system architecture, system components, tools and techniques. For the system component, the paper focuses on 5 steps. They include the creation of the Stack Overflow database, the Stack Overflow history creation and the search index creation, the connection with GitHub connector, the code clone search, and the creation of a report. Lastly, the chapter explains the tools and techniques that have been used in the development of the JARVAN system. They included GitHub OAuth, MySQL database, Node.js, and Siamese.

CHAPTER 4
IMPLEMENTATION

This chapter contains the implementation details of this project. There are four main sections in this chapter which are the preparation, back-end implementation, front-end implementation, and cloud migration.

4.1 Preparation

There are two main parts that are needed to be prepared in order to implement the JARVAN application. These parts include the gathering of data and the configuration of the Siamese system.

4.1.1 Data gathering

To implement the JARVAN system, data is a major component that needs to be gathered before any other step is conducted. The main task of the JARVAN application is to use the Stack Overflow website as a source to get the code snippets from for indexing. There are in total more than 50 million questions on Stack Overflow.

Figure 4.2 illustrates the number of the questions on Stack Overflow. The developers aimed to find code snippets on Stack Overflow website containing Java language and accepted answers. Based on the statistics, there were more than 2.5 million Java questions on Stack Overflow. Figure 4.3 shows the number of Java questions on Stack Overflow with details. The reason why they only selected Java language is because Java is one of the most popular languages used on Stack Overflow website. It is believed that there might be a high possibility that many people might copy code snippets from the website. In addition, the accepted answer is also known as a post on Stack Overflow so that the owner of the question can mark the question of as being solved. Therefore, there might be a high chance that other people will use this answer as their solution as well. Figure 4.1 illustrates the overall steps how to obtain Java accepted answer code snippet files. There are two data sets where the required data can be gathered from which are

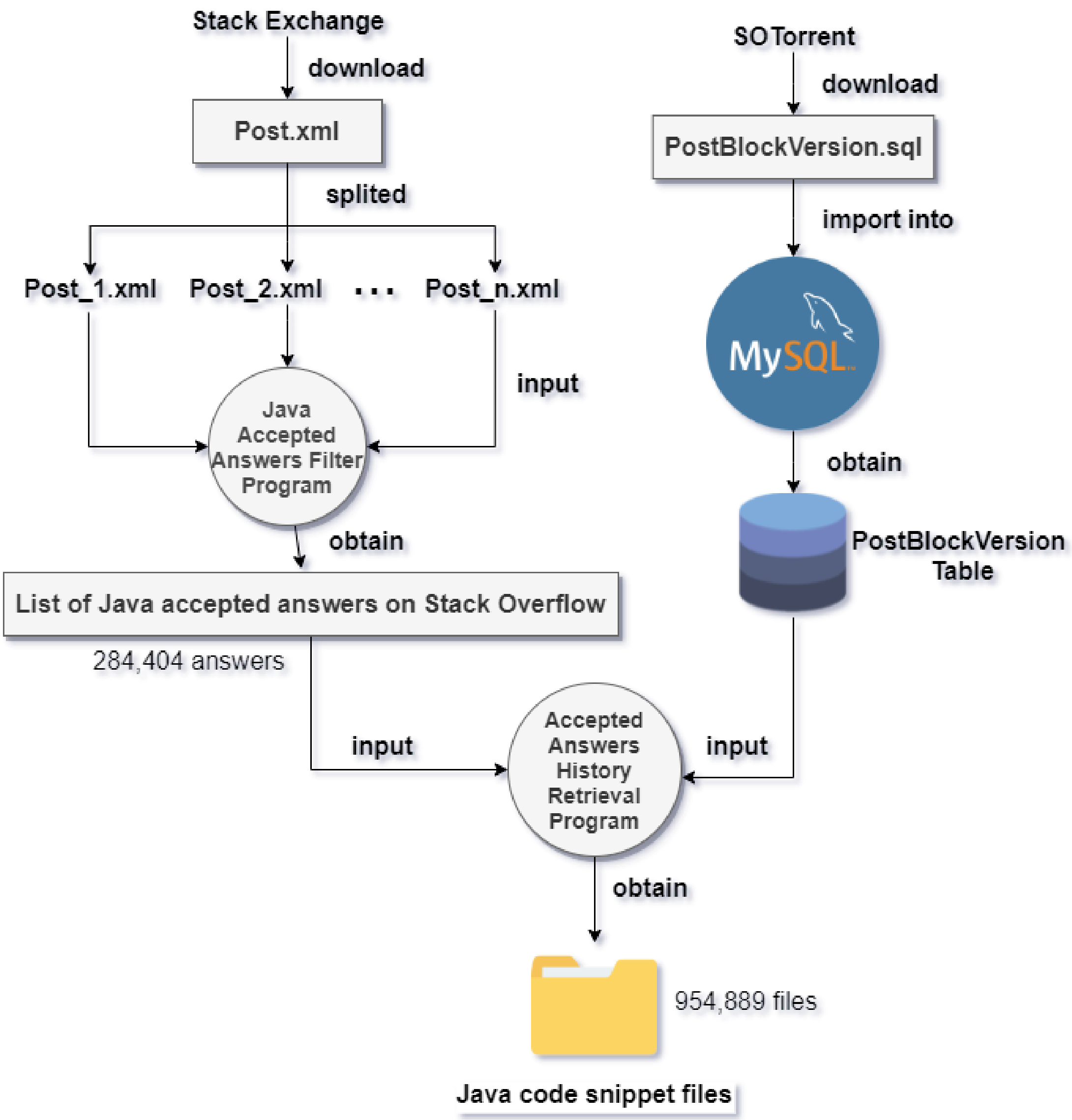


Figure 4.1: The diagram shows the steps how to obtain Java accepted answer code snippet files

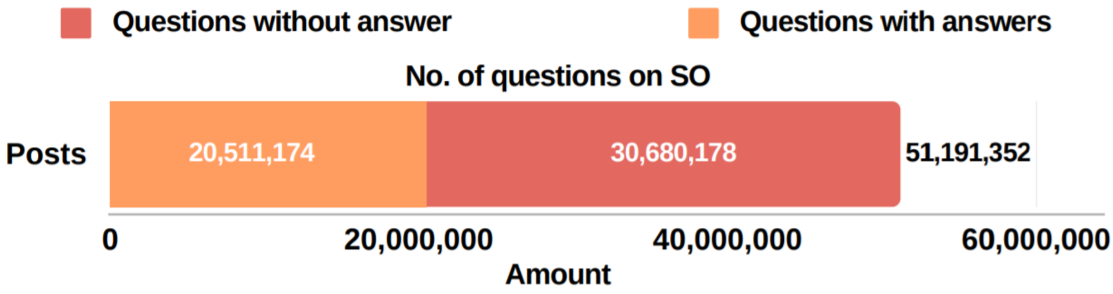


Figure 4.2: The statistics of Stack Overflow questions

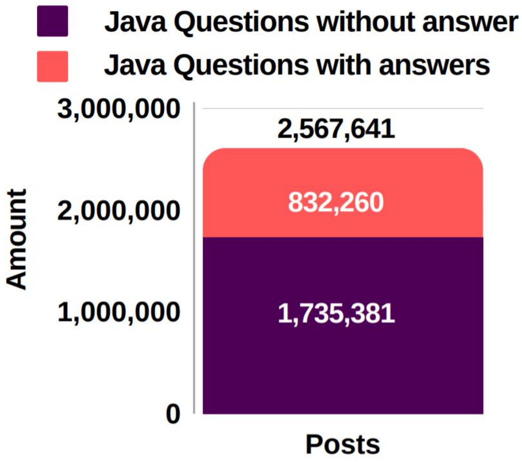


Figure 4.3: The statistics for the Java questions on Stack Overflow

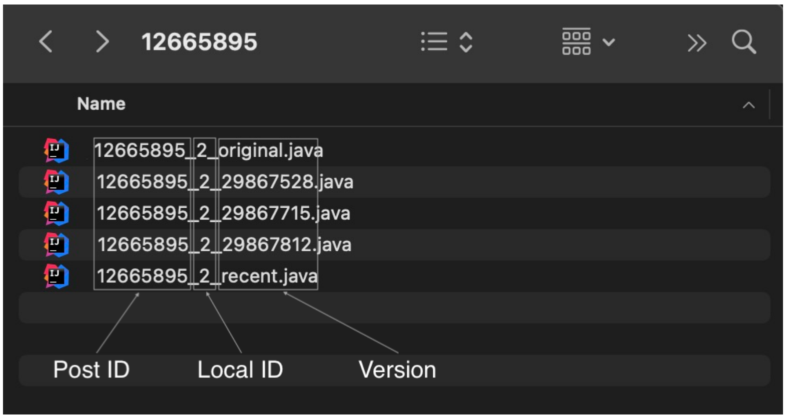
Stack Exchange [12] and SOTorrent [7]. The data sets contain an XML file and a SQL script file used for the importing of data into the database. From the Stack Exchange, the Post.xml was retrieved and the SQL script file, PostBlockVersion.sql was retrieved from SO Torrent. Nevertheless, the XML file size was enormous as it contained over hundreds of gigabytes. Therefore, the developers developed a Java program to split that huge file into smaller files in order to overcome the size limitation. However, it was still extremely difficult to extract the day a into a local database and read all necessary data from those files. Hence, the developers decided to filter out the necessary data from those smaller files. In addition, the developers implemented another Java program to process those smaller XML files through which the program would read all the smaller XML files line by line, and then only store essential data in the local storage area.

Initially, the Post.xml file was processed. This file contains several attributes that are essential for the JARVAN application, such as PostTypeId, AcceptedAnswerId, and Tags. The Java program will then only look for the data record for which PostTypeId is 1, and the Tag element contains the word ‘Java’, and the AcceptedAnswerId component of that data record is not null. The reason why the PostTypeId must be 1 is because number 1 represents a question type post. The question type post will only be a data record when the AcceptedAnswerId and Tags attributes are not null. Furthermore, the Tag attribute of the file should contain the word ‘Java’ because the JARVAN application will only focus on the Java language. Additionally, that specific record must contain code snippets in its answer, as otherwise it will be ignored. Lastly, the AcceptedAnswerId attribute should not be null because that data will be used in the next process. Eventually, after the Java program is able to run with all of the smaller Post.xml files, it will generate a text file that contains a list of AcceptedAnswerIds. The list of AcceptedAnswerIds contains 284,404 items which will be processed using a local database which contains data from the PostBlockVersion.sql file.

The second step involved the importing of data from the PostBlockVersion.sql file into the database. It takes several hours to import the PostBlockVersion.sql into the local MySQL database. In the database, the PostBlockVersion table which contains several columns which are needed for JARVAN application. There are the PostId, PostHistoryId, LocalId, PostBlockTypeId, Content, and MostRecentVersion columns. The PostId column was used to match with the list of AcceptedAnswerId. The PostHistoryId and LocalId were reserved for the file naming later. PostBlockTypeId indicated the type of post blocks. There were two possible post blocks which were the text block which that contains only text and the code block which contains only a code snippet. The Content column stored the content for those specific post blocks. Lastly, the MostRecentVersion contains two binary values which indicate whether that post block is the most recent version or not.

After the database was imported, the developers created another Java program to process the list of AcceptedAnswerId to conduct queries within the database. The Java program will read the AcceptedAnswerId list file and create a command to query the database. It is a simple command because it involves selecting all columns where the PostId matches with the AcceptedAnswerId on the list and the PostBlockTypeId comes in code block type. When the query is executed, there will be a list of results returned to the Java program. Next, the program will generate Java files from the returned list. The method to generate a Java file is that the program will write a file with the Content column from the return list.

The naming of the Java files was done through 'PostId_LocalId_version.java' for which the version depended on several conditions. Figure 4.4 shows an example of the naming process. If that post is the original version, then the version will be named 'original'. If that post is the most recent version, then the version will be named 'recent'.



12665895

Name

12665895_2_original.java
12665895_2_29867528.java
12665895_2_29867715.java
12665895_2_29867812.java
12665895_2_recent.java

Figure 4.4: An example of how to name Java files in JARVAN

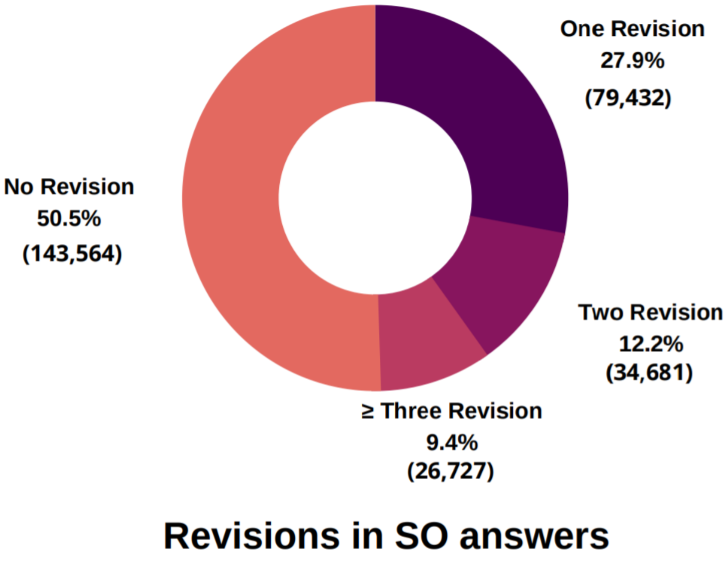


Figure 4.5: The statistics for the Java accepted answer revisions

Otherwise, the version will be named by its PostHistoryId. As a result, the Java code snippet files, for which the versions can be tracked are retrieved for this process.

Eventually, a list of Java code snippets was obtained and some statistics regarding this were collected during this process. Figure 4.5 illustrates the number of Java accepted answer revisions on Stack Overflow. From all of 284,404 Java accepted answer posts on the Stack Overflow website, there were 50.5% of answers with no version (a post with no edit). Moreover, 49.5% of answers were from different versions (a post which had been edited at least once). Finally, a list of Java code snippets (with all their versions on Stack Overflow) were used for indexing by Siamese as can be seen in the next step.

4.1.2 Siamese Configuration

Once the list of Java code snippets was ready for indexing, the developers used the list as a source for Siamese to index those files into its system. Siamese already has a built-in command function to perform the indexing. Therefore, the developers only needed to run that command to start the indexing process. Even though there were 284,404 Java accepted answers, the actual number of files which were indexed in Siamese was 954,888 files. This number is larger than the number of Java accepted answers because they contain versions of themselves too. This process took approximately half an hour to complete.

Additionally, to make Siamese perform appropriately for the JARVAN application, the developers had to adjust, and test the performance of the JARVAN application. There were a hundred projects randomly selected from GitHub to adjust the accuracy and precision of the results. After performing a search for those projects by gradually changing the options in the Siamese configuration, the developers came up with several configurations for Siamese.

First, the developers set the 'simThreshold' option to 78. This simThreshold helps determine the similarity threshold for the Siamese clone search. The number 78 means Siamese will only consider methods as clones when the similarity to them is higher or equal to 78 percent. The developers found that with this number Siamese provided the best result since it offered a minimum number of false positive results and the highest number of true positive results.

In addition, the developers set the 'minCloneSize' option to 10. The minCloneSize lets the user select a method to search through Siamese with a minimum line of code. The developers found that if this option is set at less than 10, it usually generates boiler plate results through getter and setter methods. Hence, this configuration helps to reduce the number of those boiler plate results. The 10 lines of minimum clone size is also considered the preferred size when searching for clones in a large code corpora [13].

4.1.3 Boilerplate Code Filtering

This section explains about how the developers improved Siamese and the results that came from searches. At first, Siamese returned results that were accurate as they were similar or exactly the same code snippets as the query. However, the developers found that some of the results were "boilerplate codes" as they were generally generated by IDE or developers must follow a predefined pattern. Figure 4.6 shows the examples of boilerplate codes. Therefore, the developers decided to modify Siamese to ignore methods which were usually considered to be boilerplate codes. In order to do that, the developers improved Siamese to accept text files when filtering out common clone methods. The text file contained a list of method names for which the developers empirically found that they were usually common methods and usually reported as clones, but they did not give much value to the results. For instance, the list contained a total of 8 method names which were run, start, stop, getId, getName, getView, loop, and setup. The developers considered these methods as boilerplate codes because during the manually validating of the JARVAN application, these methods came out as results frequently. Additionally, by checking meticulously, the developers also found these codes in GitHub repositories and Stack Overflow were usually generated by IDE.

Therefore, this configuration was used to prevent Siamese to return results that were not considered as clones. Therefore, before Siamese performs a search, the method that comes from the search query will be checked to see whether it is the method's name is on the list or not. If yes, Siamese will ignore and not perform a search for that method, but otherwise it will perform the search as usual.

4.1.4 Software License Violation Checking

Code copying and modifications can cause software license violations if the original license of the code has conflicts with the target software license. As mentioned previously, the copied code can be 100% the same or slightly different from the original code. However, whether the copied code violates the original code's software license or not is not up to JARVAN to decide. This is because any suspected violation has to go through a legal investigation and be decided on by the court according to the restrictions mentioned for each software license type. The results provided by JARVAN can only

```
    public static void main(String args[]) {
            /* Set the Nimbus look and feel */
            //<editor-fold defaultstate=”collapsed” desc=” Look and feel setting code (optional) ”>
            /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
             * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
             */
            try {
                for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if (”Nimbus”.equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(info.getClassName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {
                java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
                java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
                java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
                java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            }
            //</editor-fold>
            /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new NewJFrame().setVisible(true);
                }
            });
        }
```

Figure 4.6: Boilerplate code example from the Nimbus project

be used as evidence if a legal investigation occurs. Nonetheless, a 100% similarity score offers a stronger guarantee that the code is actually copied.

When talking about software licensing, it also involves copyright. Copyright is obtained automatically when a creative piece of work is created as the original owner of the work can claim their copyright on the work immediately. However, other people who would like to use it, must ask for a license from the person who owns the copyright. Thus, it is important to note that if a source code is reused without following the restrictions in the license, it violates the software license but not the copyright.

When the user analyzes one's repository with JARVAN, JARVAN retrieves the license of the user's repository from GitHub and evaluates whether it is compatible with Stack Overflow's CC BY-SA 4.0 license or not. If it is not compatible, JARVAN will notify the user through the results page. There are only two licenses which are compatible with Stack Overflow's CC BY-SA 4.0 license which are CC-BY-SA 4.0 itself and the GPLv3 license [1].

4.2 JARVAN Back-end Implementation

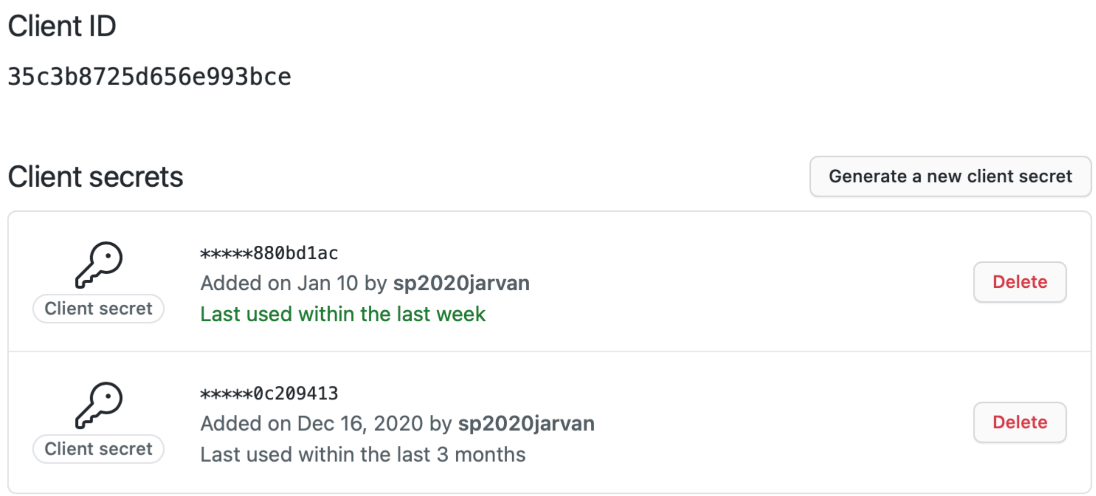
For the implementation of JARVAN, the back-end side is important to maintain the JARVAN application. It acts as a server to accept requests from users and responde back to them. There are two major implementations for back-end side, which are the GitHub Authentication and Node.js.

4.2.1 GitHub Authentication

GitHub Authentication is required for the JARVAN Application, and is considered important because it will be used for the authentication process during the authorization of users. Additionally, it will also provide useful resources for JARVAN, such as user tokens, user repositories, etc. Next, there will be an explanation of how to the GitHub Authentication was set up for the JARVAN application.

In order to allow JARVAN to use the GitHub Authentication, the developers had to create a GitHub OAuth application first. This GitHub OAuth Application can be

[1] Reference: https:// creativecommons.org/ share-your-work/ licensing-considerations/ compatible-licenses



Client ID

35c3b8725d656e993bce

Client secrets

*****880bd1ac
Added on Jan 10 by sp2020jarvan
Last used within the last week

*****0c209413
Added on Dec 16, 2020 by sp2020jarvan
Last used within the last 3 months

Figure 4.7: The client ID and client secret keys for JARVAN

created by going to the ‘developer’ menu located in the GitHub settings. Some information was required to create the GitHub OAuth application, which in this case, was the JARVAN GitHub OAuth application, such as application name, Homepage URL, and Authorized callback URL. From the beginning, the Homepage URL and Authorized callback URL were set as localhost with the port number 8080 because the developers were developing the JARVAN application on a local machine. Once all of information was provided, GitHub generated a client ID and client secret key. Figure 4.7 shows the client ID and client secret keys used for JARVAN. These client IDs and client secret keys were then implemented into the front-end side later.

4.2.2 Node.js

Node.js is the core structure used for the JARVAN application. It handles all requests from users, responses to users and also creates requests on GitHub. As can be seen, there are several tasks that Node.js needs to do and handle. This section will explain the steps to set up Node.js and also explain all of the tasks that Node.js performs in detail.

Setting up the Node.js

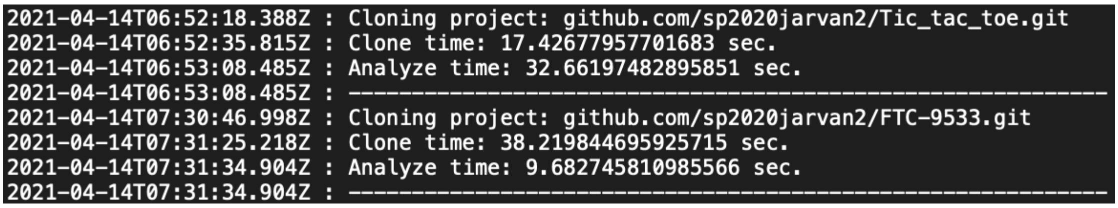
There are several modules needed for Node.js in order to function well for JARVAN. There is the ‘axios’ module which helps Node.js to make HTTP requests easily. In addition, there is he ‘express’ module, which is a minimal and flexible Node.js web application framework. It provides a robust set of features for the web application which are necessary for the JARVAN application. Then there is the ‘ShellJS’ module, which is also required for the JARVAN application. This module can create shell commands to be executed. This is necessary because JARVAN will perform searches through the help of Siamese through the shell. The ‘fs’ module is needed for JARVAN to read files from a given path. This module will be used for the retrieval of result files. Lastly, there is theNode.js port, which is set at 8080. This port is usually used for running a web server as a non-root user.

Web Page Redirection

Apart from handling user requests, Node.js can also perform web page redirections. The developers used this advantage of Node.js so that the JARVAN application would be able to show several web pages to users. There are a total three web pages for the JARVAN application which will be explained in depth in the front-end section. Node.js will be waiting for HTTP GET requests from a web browser which the users use. Then, based on those requests, it will send back the web page that the user requested to the user’s web browser.

HTTP POST for Making Git Clone Request on GitHub

This is a HTTP POST request for performing cloning repository on GitHub. The request will be sent from the front-end side with three parameters, which are the repository cloning URL, repository full name, and the license of that repository. The repository full name and license parameters will be stored as variables in Node.js. These variables will be returned to the front-end side when there is a request later. Once the repository cloning URL has been completed, it will declare the time start variable. This variable is then used to report the time of cloning. Then, Node.js will invoke shellJS to perform a shell command. The shell will use the repository URL and clone the repository to a specific directory. The location of the cloned repository is then shared among Siamese, so that Siamese can perform a search on this repository later. Once the cloning process is done, the time stop variable will be declared. These time variables will be used to calculate the time spent on the cloning process. Finally, the cloning process will be done, and the Node.js will then send back a response to the front-end side to indicate that this



```
2021-04-14T06:52:18.388Z : Cloning project: github.com/sp2020jarvan2/Tic_tac_toe.git
2021-04-14T06:52:35.815Z : Clone time: 17.42677957701683 sec.
2021-04-14T06:53:08.485Z : Analyze time: 32.66197482895851 sec.
2021-04-14T06:53:08.485Z : ------------------------------------------------------------
2021-04-14T07:30:46.998Z : Cloning project: github.com/sp2020jarvan2/FTC-9533.git
2021-04-14T07:31:25.218Z : Clone time: 38.219844695925715 sec.
2021-04-14T07:31:34.904Z : Analyze time: 9.682745810985566 sec.
2021-04-14T07:31:34.904Z : ------------------------------------------------------------
```

Figure 4.8: The time performance log example for JARVAN

request has been performed successfully.

HTTP POST Siamese Execution Request

This request is continuous from the clone request. Once the cloning request has been completed, the front-end side will immediately request for this request to be executed. Through this request, a search for the repository which was previously cloned will be performed. However, before it performs the search, the time start variable will be declared. After this, the Node.js will invoke the shellJS module again to perform a shell command, which will in turn call on Siamese to perform a search of the repository. When the search has been completed, Siamese will write out a result file in a specific destination. This destination is shared through the get result request, so that the result can be sent back to the front-end side later. Moreover, the time end variable will also be declared. Therefore, the time used for the analyzing process can be calculated. Node.js will then send back a response to the front-end side to indicate that this request has been performed successfully.

Performance Time Log

During the cloning and executing requests, the time variables are declared so that the time used for each request can be stored. Figure 4.8 shows an example of a time performance log for JARVAN. In the log, three important things are reported, which are the repository name, which is represented in the form of a cloned URL, the time used on the cloning and analysis of the time used. This information will be useful for a performance evaluation later.

HTTP GET Result Request

When the execute request is done, Siamese will create a result file. This request is called when the front-end side demands to get the result from the search conducted by Siamese. Node.js will then go to the directory where the result file is stored, read the result file as a text and then respond back to the front-end side with the result text.

HTTP GET Repository Full Name Request

During the clone request, several pieces of information are stored in the Node.js. One of them is the repository's full name. The repository's full name is needed for the front-end side to be displayed to the users and also to be used to perform other requests with GitHub. When there is this request from the front-end side, the Node.js will respond by immediately sending the repository's full name back to the front-end side.

HTTP GET Repository Software License Request

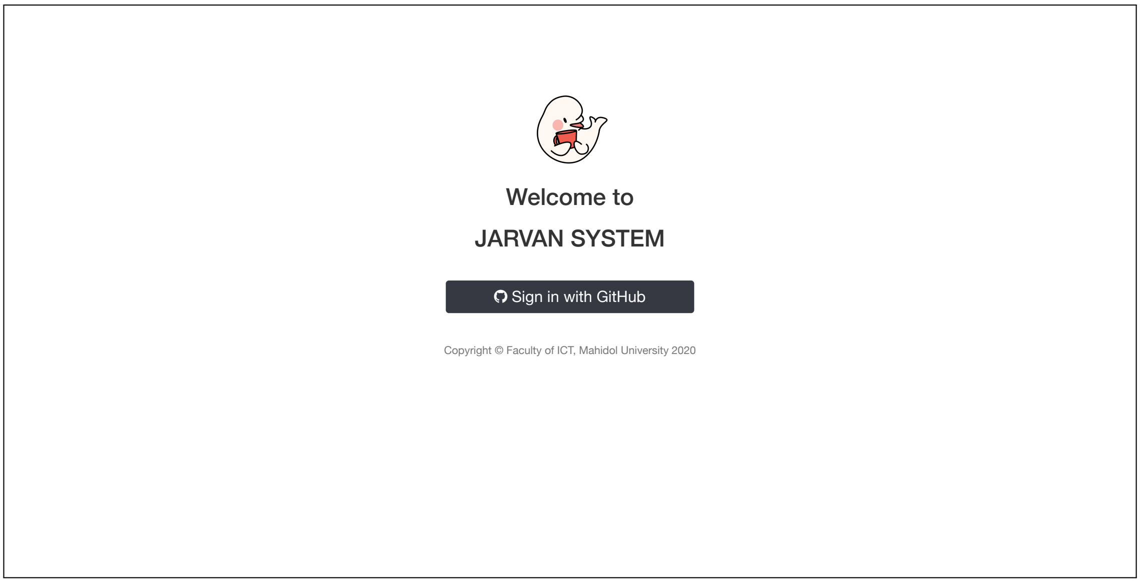
Similar to the get repository full name request, during the clone request, a repository software license is stored in the Node.js. When this request is received from the front-end side, the Node.js will then respond with the repository license instantly.

4.3 JARVAN Front-end Implementation

In order to provide a service to JARVAN users smoothly, there must be a front-end side where users can interact with the system easily. With regard to the JARVAN Front- end, there are three main pages which the users will see, which are the login page, repository page, and results page. This section will clearly explain how those pages were implemented and what they show to users.

4.3.1 Login Page

The login page displays a welcome message and provides a login button for the users. Figure 4.9 shows the interface for this page. Users can click on the 'Sign in with GitHub' button to log in with their GitHub account. This button is linked to the JARVAN OAuth application which was created earlier. The client ID and client secret key will then be used once this button has been pressed to link them to the application. Once the user has clicked the button, they will be redirected to the GitHub Login Page. Figure 4.10 shows the interface of the GitHub login page. On that page, the users must provide their GitHub credentials to log into their GitHub account. After that, the GitHub authorization page will be shown to the user. Figure 4.11 illustrates the interface of the



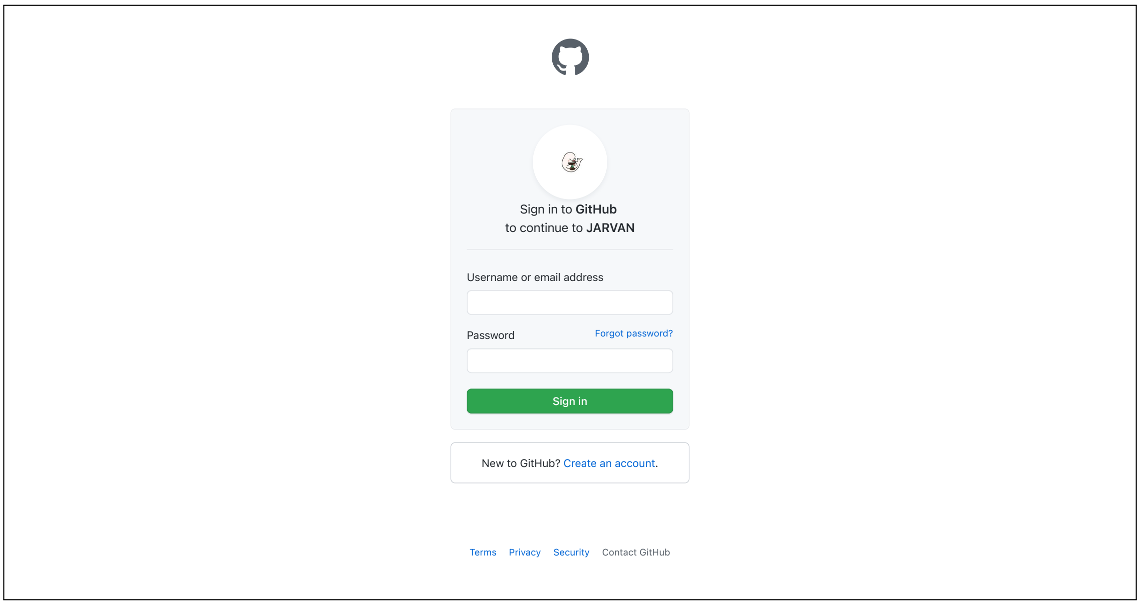
Welcome to
JARVAN SYSTEM

Figure 4.9: The interface for login page

GitHub authorization page. The user must then give the authorization to the JARVAN OAuth application for the application to access the user's public and private repositories on GitHub. After the user has authorized for JARVAN, they will be redirected back to then JARVAN system again, and then the repository page will be shown. Additionally, GitHub also provides the user token. This token is useful as it can be used to make requests through GitHub later.

4.3.2 Repository Page

After the user has given authorization to JARVAN, they will be redirected to the repository page. This page will show details about the user's repositories. Figure 4.12 represents the interface of the repository page. At the top of this page, the owner's GitHub account name is shown. As can be seen from the table, the details of the user's repositories are displayed. This information can be obtained by making a HTTP GET request to GitHub directly through a token. As a result, a list of user's repositories can be obtained, and JARVAN then populates the table with the information obtained. The information includes counted numbers, repository names, and the repositories' licenses. Additionally, JARVAN also indicates whether a repository is public or private by putting a lock or unlock symbol in front of a repository name. The lock symbol specifies that a repository is private, and the unlock symbol specifies that a repository is public. At the



Sign in to GitHub
to continue to JARVAN

Username or email address

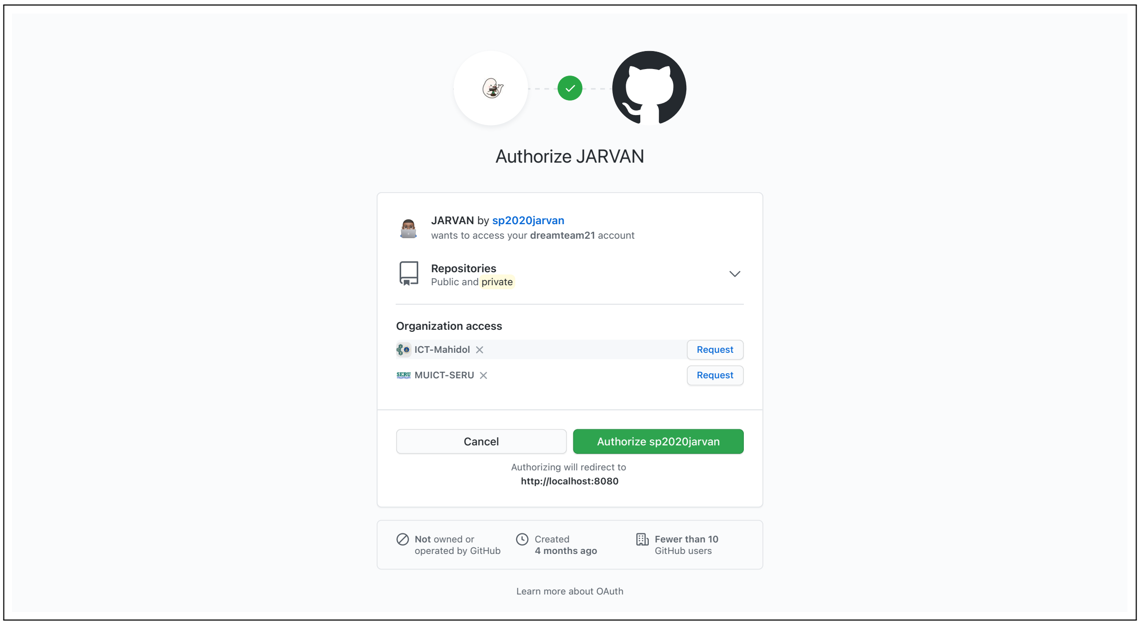
Password Forgot password?

Sign in

New to GitHub? Create an account.

Terms Privacy Security Contact GitHub

Figure 4.10: The interface for the GitHub login page



Authorize JARVAN

JARVAN by sp2020jarvan
wants to access your dreamteam21 account

Repositories
Public and private

Organization access

ICT-Mahidol × Request

MUICT-SERU × Request

Cancel Authorize sp2020jarvan

Authorizing will redirect to
http://localhost:8080

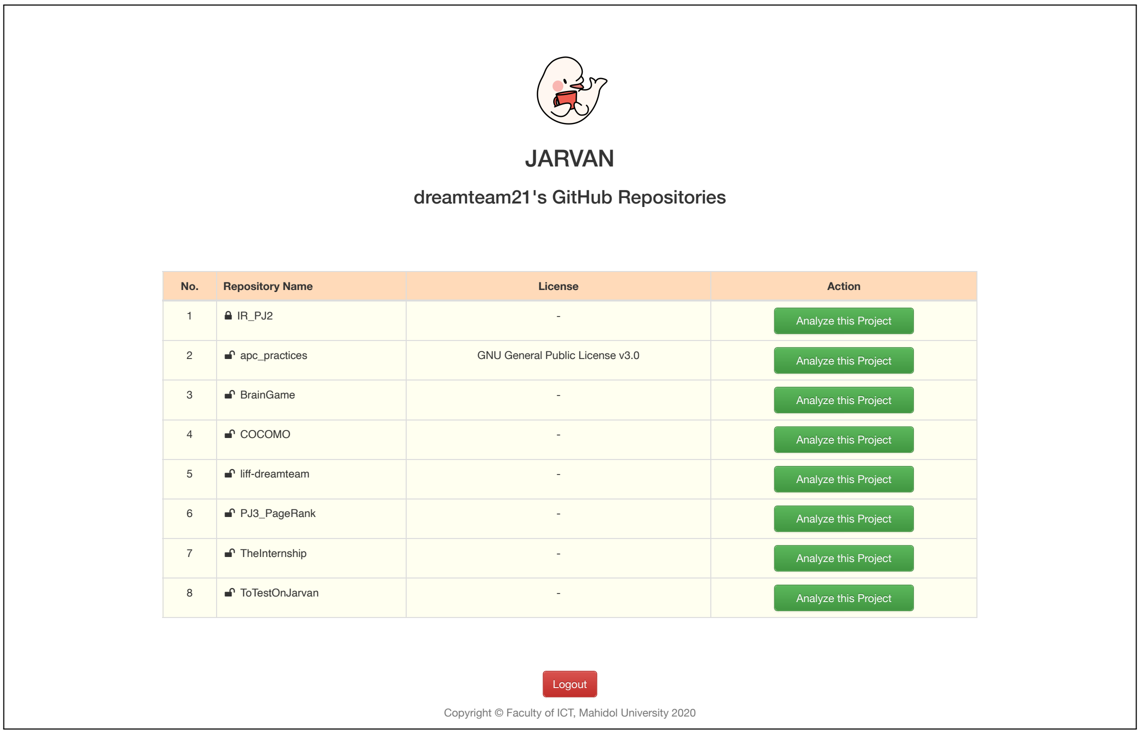
Not owned or operated by GitHub

Created 4 months ago

Fewer than 10 GitHub users

Learn more about OAuth

Figure 4.11: The interface for the GitHub authorization page



JARVAN

dreamteam21's GitHub Repositories

No.	Repository Name	License	Action
1	🔒 IR_PJ2	-	Analyze this Project
2	🔓 apc_practices	GNU General Public License v3.0	Analyze this Project
3	🔓 BrainGame	-	Analyze this Project
4	🔓 COCOMO	-	Analyze this Project
5	🔓 liff-dreamteam	-	Analyze this Project
6	🔓 PJ3_PageRank	-	Analyze this Project
7	🔓 TheInternship	-	Analyze this Project
8	🔓 ToTestOnJarvan	-	Analyze this Project

Logout

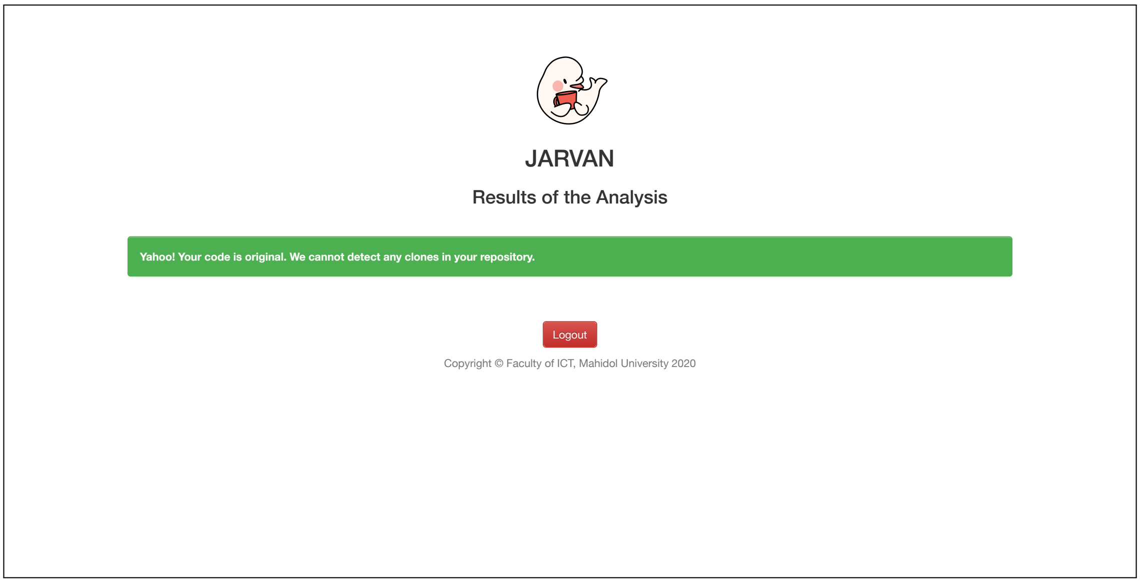
Copyright © Faculty of ICT, Mahidol University 2020

Figure 4.12: The interface for the repository page

end of each row in the table, it can be seen there are buttons shown. These buttons are used to select a specific repository which can then be analyzed. When one of the buttons is clicked on, a clone function will be called and a HTTP POST request will be sent to the back-end side. After that, when the request has been made, the user will be redirected to the last page which is the results page. Lastly, at the bottom of the page, the user may use the logout button to log out from both the JARVAN application and GitHub.

4.3.3 Results Page

The results page is the final page of the JARVAN application to which every user will be led. This page will depict the analysis results. When the user is redirected to the results page, the front-end will make a HTTP get request for a result which was produced after conducting a search. In addition, the front-end side also makes HTTP GET re- quests to get a repository's full name and license. There are three different versions of the result page depending on the result from the analysis. In this section, each version of the result page will be explained in detail.



JARVAN

Results of the Analysis

Yahoo! Your code is original. We cannot detect any clones in your repository.

Copyright © Faculty of ICT, Mahidol University 2020

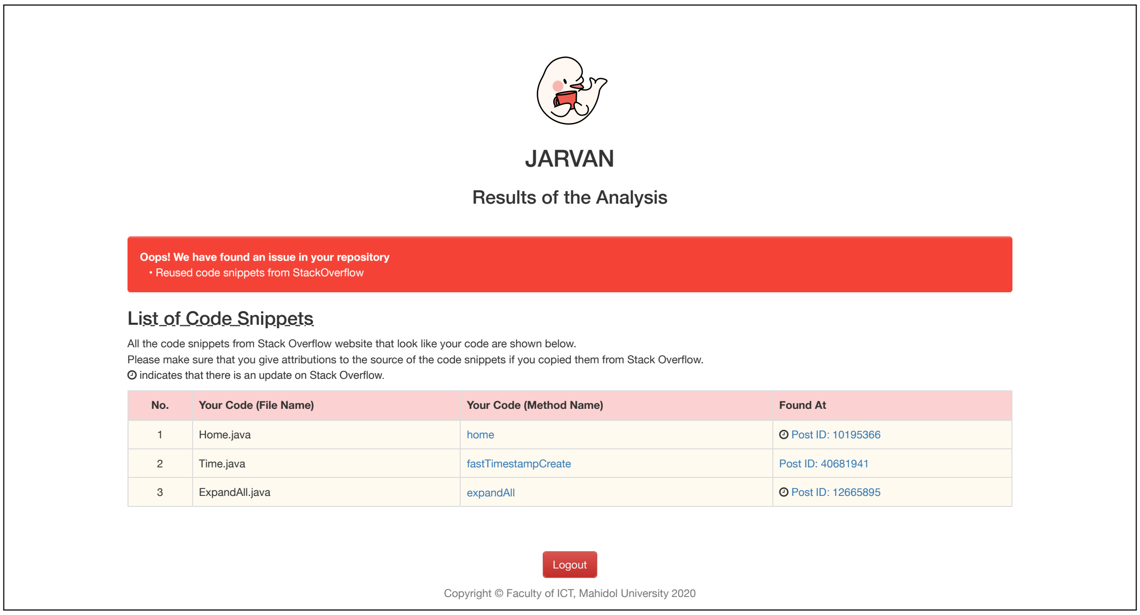
Figure 4.13: The interface of the result page when JARVAN cannot find any clones

Results with no Stack Overflow clones found

Figure 4.13 shows the first version of the result page when no clones are found. To analyze whether JARVAN found a clone or not, the front-end processes the result received from the request. If the result is empty, then it means that JARVAN found that nothing was cloned. Therefore, this version of the result page will be shown to the user. A message in a green box will say that JARVAN cannot find clones in the user's repository. Lastly, at the bottom of the page, there is a logout button. The user may use the logout button to log out from the JARVAN application as well as from GitHub.

Results with Stack Overflow clones found

Figure 4.14 shows the second version of the results page when clones are found in the user's repository, but they do not violate any license. When the front-end side received a result from the back-end side, it found that the result was not empty, which in turn meant that JARVAN had found clones in the user's repository contains clones. The result page will as a result show a message in a red box to warn the user that the user's repository contains clones. From the result received, the front-end will then populates a table with the information to show the result. The table will contain the counted numbers, file names that clones were found in, method names that clones were found in, and the location on the Stack Overflow website where the clones were found. With regards to the



JARVAN

Results of the Analysis

Oops! We have found an issue in your repository
- Reused code snippets from StackOverflow

List of Code Snippets

All the code snippets from Stack Overflow website that look like your code are shown below.
Please make sure that you give attributions to the source of the code snippets if you copied them from Stack Overflow.
🕒 indicates that there is an update on Stack Overflow.

No.	Your Code (File Name)	Your Code (Method Name)	Found At
1	Home.java	home	🕒 Post ID: 10195366
2	Time.java	fastTimestampCreate	Post ID: 40681941
3	ExpandAll.java	expandAll	🕒 Post ID: 12665895

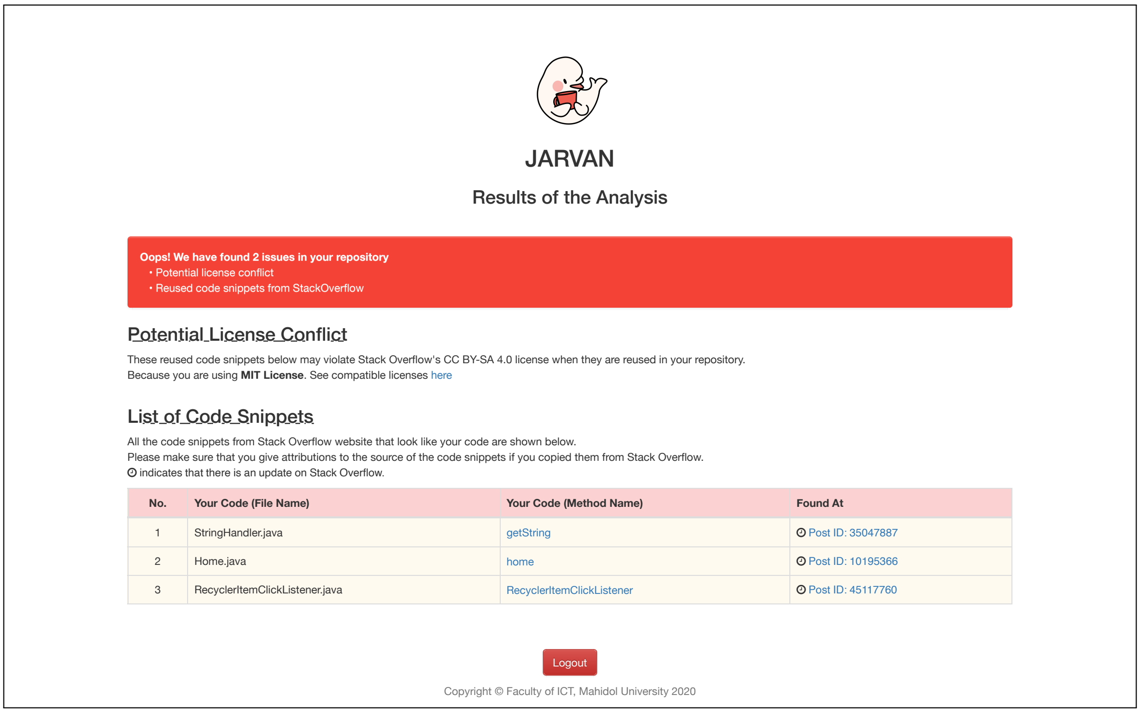
Copyright © Faculty of ICT, Mahidol University 2020

Figure 4.14: The interface of the result page when JARVAN can find clones, but there is no license conflict

method names, links are provided, which can link the user's files to the user's repository on GitHub. It will locate to the specific lines and methods that are considered as clones. Apart from the location being found on the Stack Overflow website, links are provided as well. When these links are clicked on, the users will be redirected to the posts that clones were found in on the Stack Overflow website. Additionally, in front of the location, a clock symbol might be provided to indicate that a post on Stack Overflow has received an update and that the existing code in the repository may no longer be up-to-date.

Results with Stack Overflow clones and potential license conflicts found

Figure 4.15 represents the last version of the result page when clones are found, and they violate the software licensing rules for codes on Stack Overflow. Similar to the second version of the results page, this version will show the search results in the form of a a table. Nevertheless, because the repositories in question contain licenses which do not adhere to the CC BY-SA 4.0 license, this page will display more information about the issues regarding the license conflict. In a red box, a message warns that the repository may contain a license conflict. There is also a new section displayed to the user to highlight potential license conflicts. Therefore, this section will show the user's repository license. In addition, the user may be provided with a link that redirects the



JARVAN

Results of the Analysis

Oops! We have found 2 issues in your repository
- Potential license conflict
- Reused code snippets from StackOverflow

Potential License Conflict

These reused code snippets below may violate Stack Overflow's CC BY-SA 4.0 license when they are reused in your repository.
Because you are using **MIT License**. See compatible licenses here

List of Code Snippets

All the code snippets from Stack Overflow website that look like your code are shown below.
Please make sure that you give attributions to the source of the code snippets if you copied them from Stack Overflow.
⏲ indicates that there is an update on Stack Overflow.

No.	Your Code (File Name)	Your Code (Method Name)	Found At
1	StringHandler.java	getString	⏲ Post ID: 35047887
2	Home.java	home	⏲ Post ID: 10195366
3	RecyclerItemClickListener.java	RecyclerItemClickListener	⏲ Post ID: 45117760

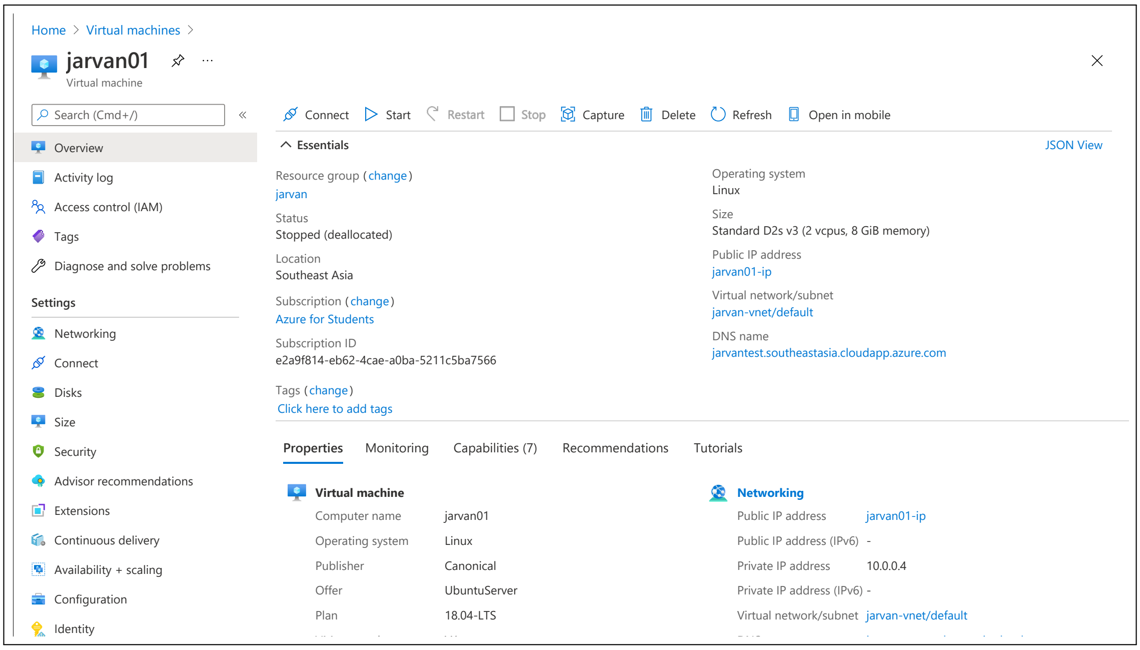
Copyright © Faculty of ICT, Mahidol University 2020

Figure 4.15: The interface of result page when JARVAN can find clones and there is a license conflict

user to a page where one can learn more about compatible licenses on Stack Overflow.

4.4 Cloud Migration

This section will explain how to move the JARVAN application from local deployment to cloud. After the JARVAN implementation had been completed on a local machine, the developers decided to move JARVAN to a cloud machine. This was because they wanted the JARVAN application to be accessible from anywhere. Additionally, it was believed that moving JARVAN to a cloud machine would benefit the developers when they wished to appraise the JARVAN system when conducting speed and user evaluations. The JARVAN system can be used from anywhere with the same exact machine all the time, so the developers would always get the same result from the evaluation when they separately perform the evaluation on JARVAN. In order to migrate JARVAN to cloud, first, the developers must decide which cloud provider they would like to use. For this project, the developers decided to use Microsoft Azure to operate the JARVAN application. The reason why they selected Microsoft Azure is because it provides free credits for university students and it is easy to configure a platform with.



Resource group (change)
jarvan

Status
Stopped (deallocated)

Location
Southeast Asia

Subscription (change)
Azure for Students

Subscription ID
e2a9f814-eb62-4cae-a0ba-5211c5ba7566

Tags (change)
Click here to add tags

Operating system
Linux

Size
Standard D2s v3 (2 vcpus, 8 GiB memory)

Public IP address
jarvan01-ip

Virtual network/subnet
jarvan-vnet/default

DNS name
jarvantest.southeastasia.cloudapp.azure.com

Virtual machine

Computer name	jarvan01
Operating system	Linux
Publisher	Canonical
Offer	UbuntuServer
Plan	18.04-LTS

Networking

Public IP address	jarvan01-ip
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	jarvan-vnet/default

Figure 4.16: The portal page of a virtual machine on Microsoft Azure

There are several configurations needed for setting up JARVAN on the cloud. First, the operating system for the cloud needs to be set up. The Ubuntu server 18.04 LTS Gen1 is used for this. The size of the cloud is based on a Standard D2s v3 plan. This plan comes with 2 virtual CPUs and 8 GB of memory. Once that was done, the network configuration needed to be set up. The NIC network security group was set to be none as this setting would allow other machines from different networks to access this cloud machine so that other machines could access the JARVAN application through their web browser more easily. Once all the configurations are set, the developers can create the cloud machine to host the JARVAN application. Figure 4.16 shows the portal page for the JARVAN cloud machine after it was created successfully. In order to access the cloud, Microsoft Azure provides a ssh private key to access to the cloud via an ssh protocol. Once the developers can cloud machine, the setting up of the cloud machine for the JARVAN application can be done similarly to the local machine. When the setting up is completed, the JARVAN application can be accessed from anywhere through the Internet directly from their web browser.

CHAPTER 5
EVALUATION AND DISCUSSION

This chapter focuses on the testing of the JARVAN system, including an evaluation of the system's search precision and a performance evaluation across hundreds of Java software projects.

5.1 Environment Setup

The JARVAN system was run as a web application on the cloud by using a service offered by the Azure Virtual Machine created by the Microsoft Corporation. The plan used for this cloud machine is the Standard D2s v3 plan. The reason why the developers selected this plan for the cloud machine was because JARVAN does not require a high-performance machine to perform on. This is because the developers wanted to demonstrate that with an adequate performance machine, JARVAN could still perform efficiently and effectively. The details of the machine for this setup were shown in Table 5.1 below.

Table 5.1: The specifications of the cloud machine used for the evaluation

Item	Description
Operating System	Linux 18.04-LTS
Model	Microsoft Corporation Virtual Machine
Processor	Intel Xeon E5-2673 v4 @ 2.29 GHz 1 processor, 2 threads
Memory	7.78 GB

5.2 Project Selection

To evaluate the performance of JARVAN, the developers selected software projects from GitHub. This section will explain the methods of how the software projects for evaluation on JARVAN were obtained.

5.2.1 GitHub Project Selection Criteria

The developer selected the projects from GitHub by using these five criteria.

1. The software projects must be in the Java language
2. The software projects must not have been forked from anywhere
3. The number of contributors must be more than one
4. The number of watchers must be one
5. The latest update must have occurred after May 2019

There were many reasons why these criteria had to be applied in order to obtain software projects for evaluation on JARVAN. First, the software projects had to be in the Java language because JARVAN currently supports only software projects presented in the Java language. Hence, if the software projects used for evaluation are not supported by Java, there will be no result reported. Another reason provided for the application of the criteria that the software project must not have been forked from anywhere was because the developers wanted to ensure that the software projects were originally created by the owners of the software projects. Next, the number of contributors had to be more than one as the developers wanted to get software projects which were real software projects, and not tutorial projects or student assignments. They believed that real software projects have many contributors, so they only looked for software projects that had more than one contributor to them. Subsequently, the number of watchers had to be one since the developers found that the majority of software projects on GitHub had the number of watchers at one, and they wanted to obtain software projects that were common and not famous ones as the famous one may not have the issues of being copied codes from Stack Overflow. Hence, they applied this criterion during the selection. Finally, the software project must have been updated after May 2019, for the developers wanted to obtain software projects which were still existent as the software projects that were updated not more than two years ago had a high possibility of still being current. Hence, this criterion was applied.

5.2.2 Projects Sampling and Statistics

During the project selection, a total of 437 projects were found that passed all of the criteria. A hundred projects were randomly sampled from the pool of 437 to be analyzed projects. The developers believed that the analysis of a hundred projects would offer a big enough representation of all the software projects that the developers focused on. Furthermore, the developers wanted to manually validate results of JARVAN, so a number of hundred software projects made it feasible for them to manually investigate the results.

The developers did not consider the project sizes when they randomly selected the projects. Thus, it is possible that the project sizes may affect the evaluation results. A full list of software projects used for this evaluation including their statistics can be found in Appendix A and B. The next section will discuss the evaluation process more.

5.3 Search Precision Evaluation

This section explains how a search precision evaluation was performed and discusses the results. The reason why the search precision evaluation was needed was mainly because the developers wanted to ensure that JARVAN could perform well by giving accurate and precise results.

5.3.1 Methodology

After the developers selected a hundred projects and analyzed all of them on JARVAN, the three developers conducted a manual validation of the results reported by JARVAN. The developers performed the manual validation individually without discussing the results with each other. They considered all the hundred projects by using their own opinion to judge whether the source codes were cloned from Stack Overflow or not. The validation results were separated into three groups which consisted of the source code being really cloned group, the not being cloned group, and the boilerplate code group. After the developers had gathered their own results, they organized a meeting to compare the results. When they got conflicting results, they defended their view and tried to resolve the conflicts by finding an agreement.

Manual Validation

52 conflicts out of 193 came out of the manual validation which in other words accounted for 27 per- cent. The main conflict which the developers noticed was that the source code was auto- generated by IDE. Therefore, the assumption was that that

Table 5.2: Conflict results from the manual validation

Cloned	Boilerplate	Not Cloned	Total
109	76	8	193

source code was either a clone or a boilerplate code, while ultimately led to disagreements among the team of developers. After the meeting, the developers resolved 51 of the conflicts, and one conflict was resolved after a discussion with the project advisor. The manual validation results from each developer and the final conclusions can be found in Appendix C.

5.3.2 Results and Discussion

When looking at the results that the developers obtained from the experiment, they found that there were 193 methods were detected by JARVAN across the hundred projects. As shown in Table 5.2, they determined that 109 methods were cloned, 8 methods were not cloned, and 76 methods were boilerplates. The precision with regards to boilerplate codes being determined to be a cloned code was 0.96. Without considering the boilerplate code, the precision was 0.56. Although the precision, in this case, is relatively low, it is only the case when the boilerplate code can be obviously identified, which may not be the case in practice. In conclusion, the developers found that the JARVAN system had a sufficiently high precision rate when reporting on reused code snippets from Stack Overflow.

5.4 Search Performance Evaluation

This section delves into the search performance of the JARVAN system by focusing on the execution speed. The developers created a search performance evaluation experiment to confirm that the JARVAN system could analyze a software project by using a minimal and proper amount of time.

5.4.1 Methodology

To evaluate the search performance, the developers ran hundred projects on the JARVAN system and then recorded the execution time to conduct the cloning and analyzing phases for the execution time. They executed the projects three times and recorded

Table 5.3: Cloning time for hundred projects by the JARVAN system in seconds

Mean	Median	Maximum	Minimum
6.65	3.09	71.05	2.06

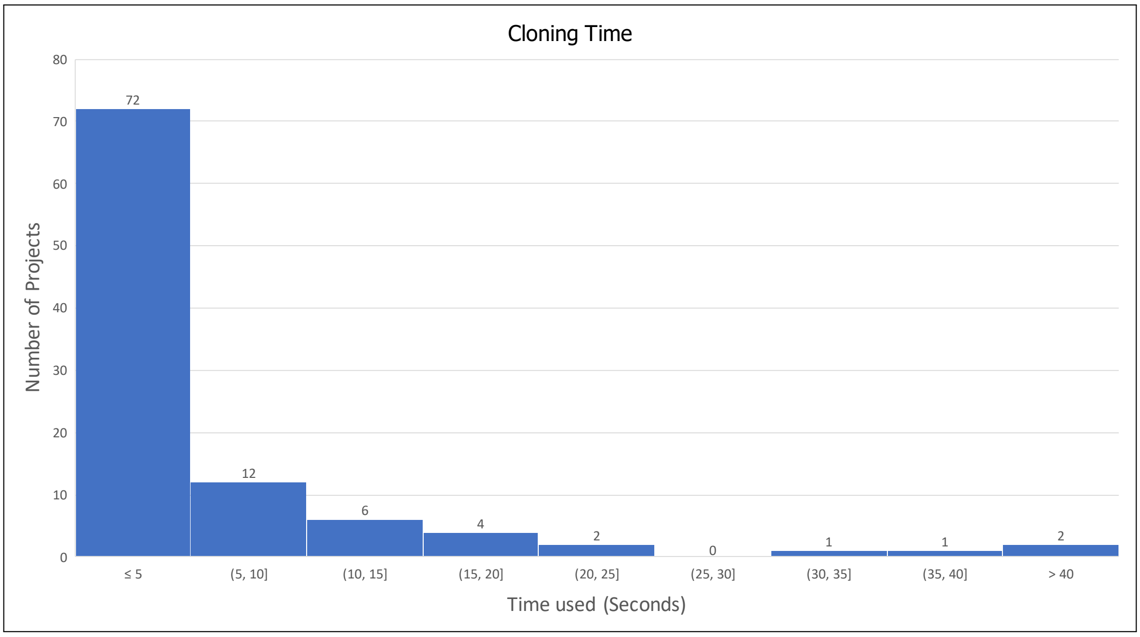


Figure 5.1: The histogram chart showing the frequency of cloning time used

the time it took for the execution of each phase. After that, they calculated the average time used for the execution. The developers separated the time into two phases because they wanted to emphasize the time used for each phase. The cloning time was not actually relying on the JARVAN system, but on the internet speed and the project size, so they wanted to focus more on the analyzing phase. Therefore, they separated the execution time into the cloning and analyzing phases.

The developers have also added a the boiler-plate code filtering during the search which was not available in Siamese. Thus, the evaluation of the time it took to conduct a search was not exactly the same as when the original Siamese system was used for evaluation. The developers did not evaluate the indexing time because it happened only a few times when the database needed to be updated, so it did not directly impact the users.

Table 5.4: Analysis time for hundred projects by the JARVAN system in seconds

Mean	Median	Maximum	Minimum
33.80	12.29	507.46	2.02

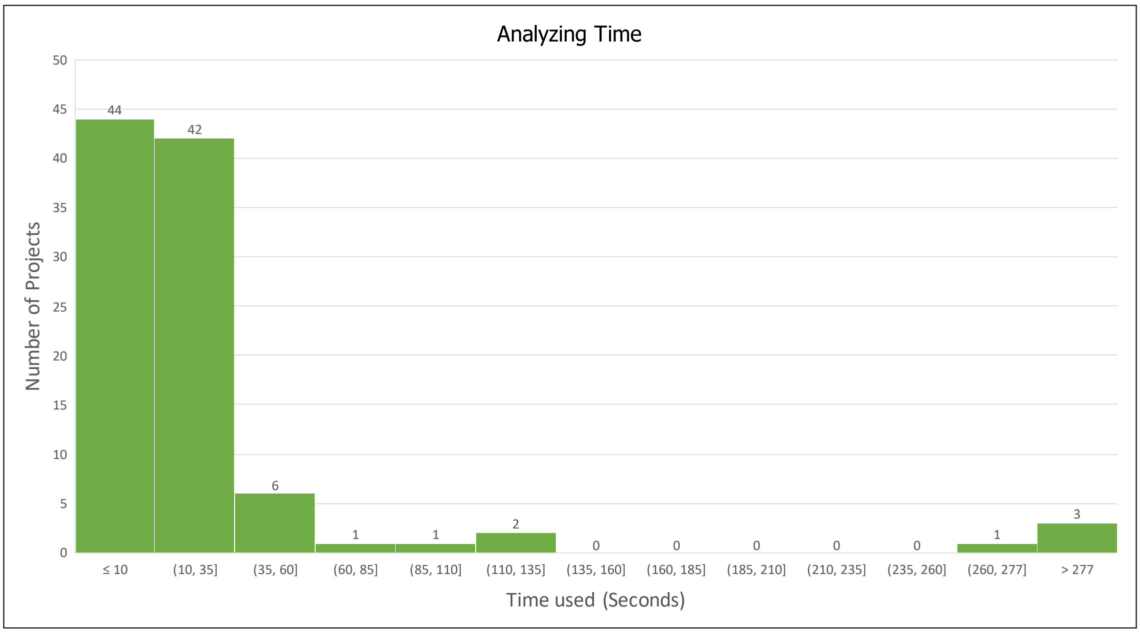


Figure 5.2: The histogram chart showing the frequency of analyzing time used

5.4.2 Results and Discussion

With the regards to the cloning phase, there were 72 projects for which the time to clone them was less than 5 seconds per project, while 12 projects used 5 to 10 seconds per project, 6 projects used 10 to 15 seconds per project, and 2 projects used 20 to 25 seconds per project. Figure 5.1 shows the time it took to clone for hundreds of projects through a histogram chart. Based on the statistics shown in Table 5.3, the average cloning time that the JARVAN system needed was approximately 6 seconds with a median of time of about 3 seconds per project. This implied that JARVAN was fast at cloning. However, project number 62 named ‘open-huirong’ recorded the highest cloning time with roughly 71 seconds. This could mean that the project size affects the amount of time used for cloning. A full list of the cloning time for all projects can be seen in Appendix D.

When looking at the analysis phase, 44 projects used less than 10 seconds per project to conduct an analysis, while 42 projects used 10 to 35 seconds per project, and 6 projects used 35 to 60 seconds per projects. There were 8 projects that used more than a minute per project for analysis. Hence, the majority of the projects spent an average of less than 30 seconds on analyzing. Figure 5.2 shows the histogram chart for the time spent on analysis for each of the hundred projects. It can be concluded that JARVAN is fast when performing the analysis phase. As shown in Table 5.4, the average time spent on analyzing is roughly 33 seconds with a median time of approximately 12 seconds. Nevertheless, project number 48 named 'JwRalph_Seo' used the highest amount of time to conduct an analysis with more than 500 seconds. The developers examined this occurrence meticulously and found that it took so long because the project had the size of 2,845 files and there were 6,105 methods to be analyzed. This can indicate that a huge size of a software project can significantly affect the performance of JARVAN during the analysis phase. Additionally, the developers also investigated the speed when conducting an analysis on JARVAN per method. They found that the average time spend on analyzing per method was 0.389 seconds with a median of 0.191 seconds. A full list for the analysis time of all projects can be seen in Appendix E.

5.5 User Evaluation

This section looks more closely into the user evaluation. The developers conducted interviews with participants to evaluate the JARVAN system. This section includes the evaluation methodology, background information regarding the participants, and a result and discussion section, which will be explained in greater detail.

5.5.1 Methodology

This subsection describes the methodology of how the developers conducted the interviews with participants. It also includes how the participants were selected to participate in the interviews and evaluation. This subsection will explain them more thoroughly.

Participant Selection

There were several criteria that were applied when selecting a group of participants for the user evaluation. The developers focused on people who were working in the software development industry with at least one year of experience of developing a software project and familiarity with the Java environment. The participants were recruited from the social media platform, Facebook, where people were asked to volunteer to participate in the user evaluation. Eventually, based on the criteria for participants who would be allowed to take part in the user evaluation, there were five participants who passed the criteria that the developers had set. Once they were recruited, the developers scheduled an interview with the participants based on their availability. The interview took approximately 20-30 minutes for each participant. During the interview, the participants were asked about their background, asked to do the test, and asked for suggestions. The full background details of each participant will be clarified in the participant background subsection.

The Test Case Creation

In order for the participants to effectively evaluate the JARVAN system, the developers created test case question sets for all participants. The developers also set up a test environment by using JARVAN as a platform for the testing. The developers created three test cases based on the possible outcomes of a JARVAN search. The first possible outcome is that JARVAN finds code clones, but they do not violate the software license. The second possible outcome is that JARVAN finds code clones, which violated the software license. The last possible outcome is that JARVAN cannot find any code clones. The purpose of this test is to evaluate that the participants are able to understand the possible results of JARVAN, and whether they know the meaning of what JARVAN has reported to them correctly. While the participants are using the JARVAN system, there is a list of directions and questions given to them to follow and answer. A full list of these instructions and questions can be seen in Appendix I. The questions relate to the user interface of JARVAN. The results of the test will be discussed in the result and discussion subsection.

After Test Survey

Once the participants finished the test cases, they were asked to take a survey to find out their views about the JARVAN system. The questionnaire was provided in a scale format to determine the satisfaction of the participants towards the JARVAN system. The questions focused on what the participants thought about JARVAN's capacity to locate code clones, its usefulness and how it assisted them. The participants were also asked to give suggestions on how to improve the JARVAN system. However, some participants mentioned them already during the interview, so they did not include their views again in the survey. The complete answers to the survey for each participant can be found in Appendix H.

5.5.2 Participant Background

With regards to the users who participated in the evaluation section, all participants had around 1- 5 years' experience of using JAVA language and more than two years' experience of working in the IT development field. However, three out of five participants were not aware what a code clone was while two of them were aware that code clones could cause problems in software projects. When looking at the background experience of all the participants, all of them used to copy source codes from the internet, especially from Stack Overflow. However, only one of them knew that code snippets from Stack Overflow fell under CC BY-SA 4.0 while others were unaware of that the following parts will explain the background knowledge of each participant more thoroughly. The full responses from each participant can be seen in Appendix F.

- Participant 1: Android Developer for Via Group (Thailand) Co., Ltd. (2 years)

  Based on the interview held with him, he has knowledge of code clones and is aware that code clones can present problems in software projects. He strongly agrees that the developer should make a reference to the owner of the code when they copy the code from the internet.

- Participant 2: Office 365 Consumption Hero for Microsoft Thailand (2 years)

  Based on the interview held with him, he was unaware what code clones were and did not think that code clones were a problem in software projects. In addition, he knew that code snippets from Stack Overflow fell under the CC BY-SA 4.0 license and he thought it was natural that when developers copy codes from the internet, they should make a reference to the owner of the code.

- Participant 3: Implementer for CPFIT (2 years)

  Based on the interview held with him, she did not know what a code clone was did, nor did she know that code snippets from Stack Overflow fell under the CC BY-SA 4.0 license. She strongly believes that code clones create problems in software projects. She agrees that when developers copy codse from the internet, they should make a reference to the owner of the code. However, she strongly disagrees that developers should not copy codes from the internet. Additionally,s he disagrees that code clones should be eliminated from software projects.

- Participant 4: Software Support and Maintenance for AppMan Co., Ltd. (2 years)

  Based on the interview held with him, he understands what code clones are, but he thinks code clones are not the problem in software projects. Therefore, he disagrees that code clones can cause problems in software projects and that code clones should be eliminated from software projects. Furthermore, he knew that code snippets from Stack Overflow fell under the CC BY-SA 4.0 license, and he agrees that when developers copy codes from the internet, they should make a reference to the owner of the code.

- Participant 5: Technology Developer for SCG (2 years)

  Based on the interview held with him, he did know what code clones were, but did not know that code snippets from Stack Overflow fell under the CC BY-SA 4.0 license, so he did not believe that code clones were the problem in software projects. However, he agreed that developers should not copy codes from the internet and when developers copy codes from the internet, they should make a reference to the owner of the code.

5.5.3 Results and Discussion

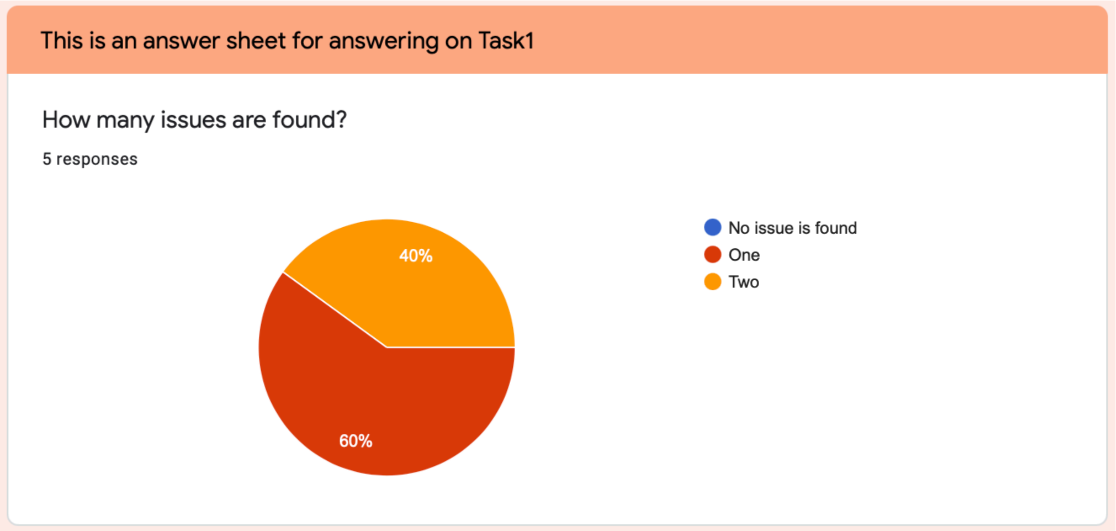
This subsection discusses the evaluation test results for each participant during the evaluation. In addition, it also includes suggestions to improve the JARVAN system offered by the participants.

Test Results

The full results for each participant can be seen in Appendix G. When looking at the test cases given to the participants, the first question focuses on the repository page. It asks *how many repositories are shown on the repository page*. The correct answer was 3 repositories. All participants could answer this question correctly. The next question asked *which of the shown repositories is/are private repository*. The correct answer is Project number 3. All participants could answer this question correctly.

The questions, that followed after the first one, were based on the test case. The questions were applied to all test cases in a similar way. The questions entailed: *how many issues were found?, how many clones were found?*,and *which is the software license violated CC BY-SA 4.0 license in this repository?*. A total of 3 test cases were provided to all participants during the evaluation.

With regards to the answers to the questions for the first test case, one issue could be found. When looking for code clones, two could be found. Lastly, no software license could be found that violated the CC BY-SA 4.0 license in this repository. All participants could answer most of the questions correctly. Figure 5.3 shows the result for the question *How many issues were found?* for test case 1. There were two participants who could not answer this question correctly. *how many issues are found*.



This is an answer sheet for answering on Task1

How many issues are found?

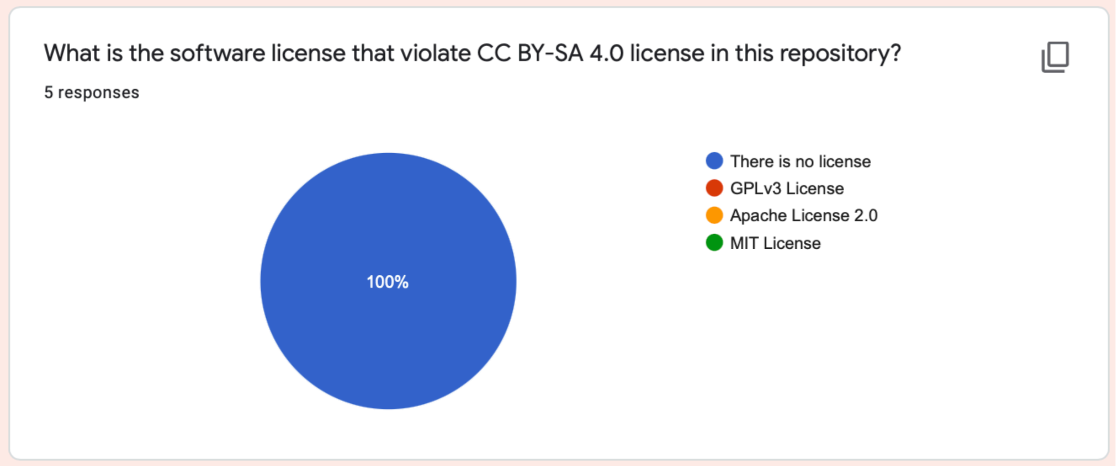
5 responses

Figure 5.3: The result for the question *How many issues were found?* for test case 1

Figure 5.4 illustrates the result for from the second test case. With regards to the answers to the questions for the second test case, no issue was found. When looking for code clones, JARVAN could not find any code clones. Lastly, no software license could be found that violated the CC BY-SA 4.0 license in this repository. All participants could answer all of the questions correctly.

Figure 5.5 shows the result for the last test case. With regards to the answers to the questions for the last test case, two issues could be found. When looking for code clones, two code clones were found. Moreover, Apache License 2.0 was found which was in violation of the CC BY-SA 4.0 license in this repository. All participants could answer all of the questions correctly.

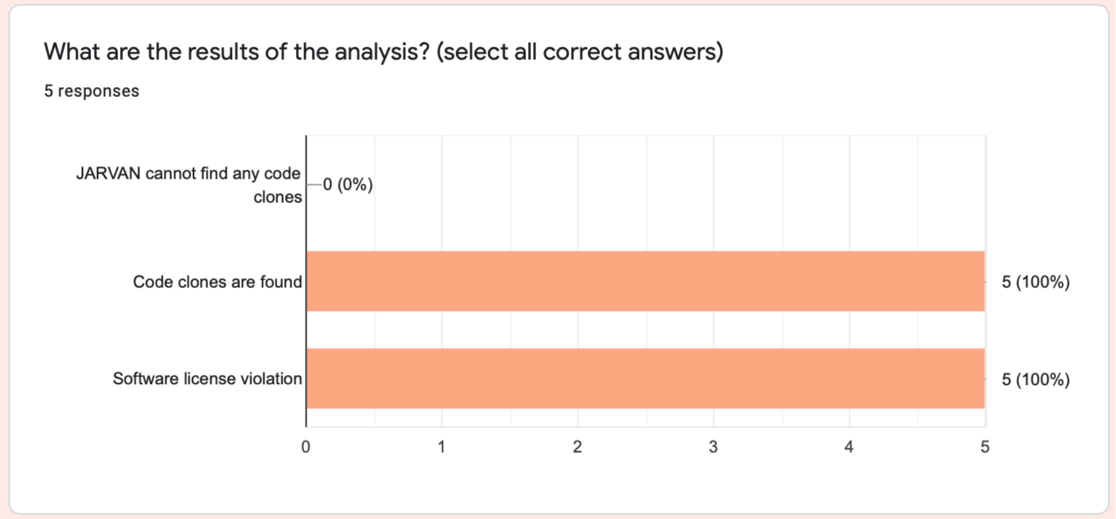
To conclude, all of the participants could answer the questions correctly. Nevertheless, for test case 1, some participants could not answer the question of how many issues were found correctly. The developers believe that this was because the user interface of JARVAN might have caused a misunderstanding for the participants. However, with regards to the other test cases, all participants could answer all the questions correctly without any mistake. Therefore, it should be noted that sometimes when reporting how many issues were found, JARVAN might cause a misunderstanding among the users.



What is the software license that violate CC BY-SA 4.0 license in this repository?

5 responses

Figure 5.4: The result for the question *What is the software license that violates the CC BY-SA 4.0 license in this repository?* for test case 2



What are the results of the analysis? (select all correct answers)

5 responses

Figure 5.5: The result for the question *What are the results of the analysis?* for test case 3

Participant Survey Results and Suggestions

After the interview, the developers asked the participants to conduct a survey and give suggestions to improve the JARVAN system. The survey look into the participants' opinions toward the JARVAN system. The answers to the questions are based on a Richter scale and, with the answers ranging from *1 - the very least* to *5 - very much*. The questions asked were *How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects, How much do you think JARVAN helped you to make a references to a code copied from Stack Overflow* and *how much do you think JARVAN helped you to be more aware when reusing a code from Stack Overflow*. The full responses to the survey from all the participants on the survey can be seen in Appendix H.

When looking at the provided feedback, all participants greatly believe that JARVAN can help them locate codes that have been copied from Stack Overflow into their projects. Figure 5.6 shows the participants' opinions for the *how much do you think JARVAN can help you located a code that has been copied from Stack Overflow in your projects? Question*. They believed that JARVAN could help them a lot to make a reference to the code copied from Stack Overflow. Figure 5.7 represents the participants' opinions for the *How much do you think JARVAN helped you to make a references to the a code copied from Stack Overflow? question*. Lastly, three out of the five participants thought that JARVAN could help them be more aware when reusing codes from Stack Overflow. One respondent believed that it may or may not help, and another person did not believe that it gave much help. The developers asked the participants who did not think that JARVAN could help them to be more aware when reusing a code from Stack Overflow for more details. They responded that they did not think that JARVAN could help them to be more aware because they still did not see how code clones could cause problems in software projects. Thus, even though JARVAN reported issues to them regarding code cloning, they still do not take serious precautions. Figure 5.8 depicts the participants' opinions for the *How much do you think JARVAN helped you to be more aware when reusing a code from Stack Overflow? question*.

Lastly, the participants were also asked to give some suggestions in order to im-

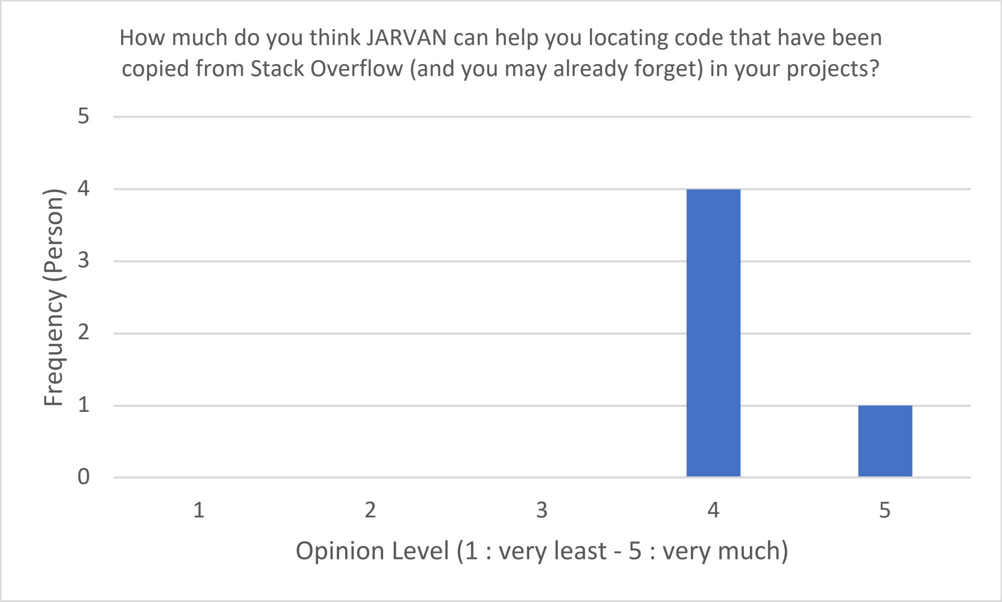


Figure 5.6: Participants' answer to the question *how much do you think JARVAN can help you locate a code that has been copied from Stack Overflow in your projects?*

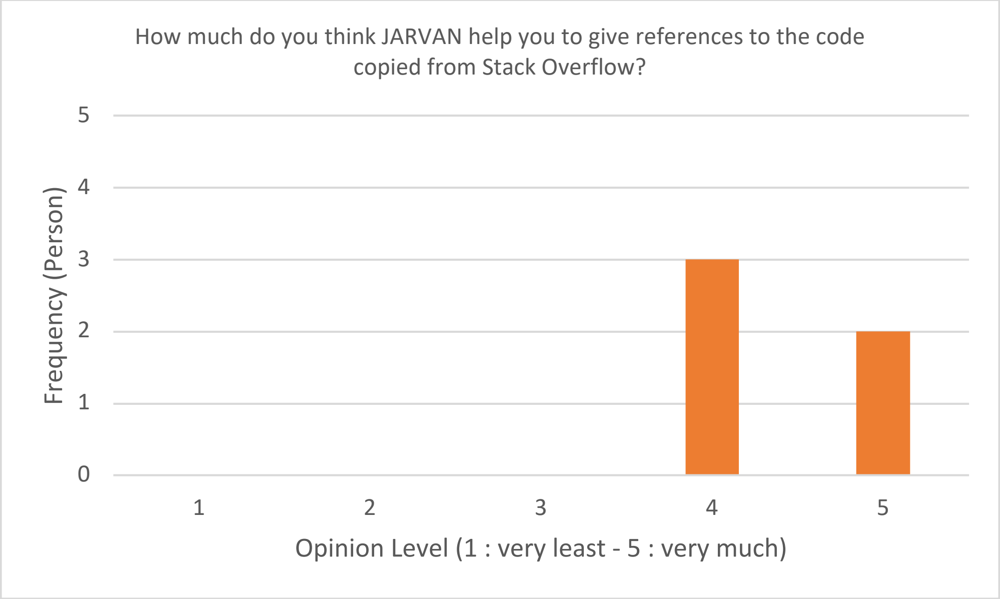


Figure 5.7: Participants' answers to the question *how much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?*

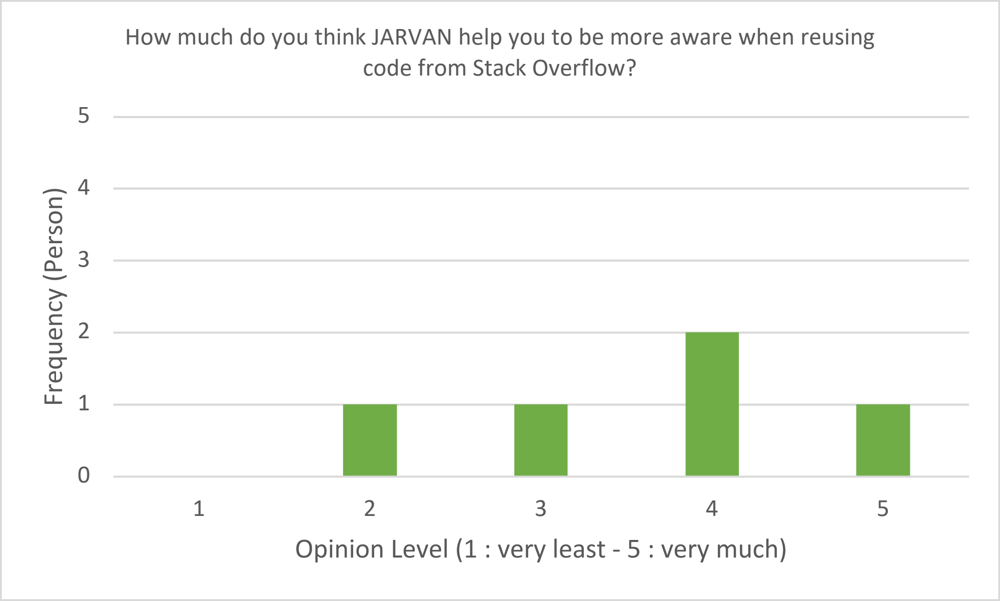


Figure 5.8: Participants' answers to the question *how much do you think JARVAN helps you to be more aware when reusing code from Stack Overflow?*

prove the JARVAN system. Participants suggested that it would be better if JARVAN could support a code clone search on other sources rather than only on Stack Overflow. They also suggest that it could be better if JARVAN could keep records of the searches conducted by each user because when the users want to do research on the same software project, they can see the differences and improvements. In addition, they mentioned that it would be better if JARVAN could report issues on GitHub after certain issues were found by JARVAN as it could help users to easily see the code clones in their repository on GitHub and remind them rather than just seeing the result on the JARVAN result page. The participants stated that it might be better if JARVAN could show the similarities between the code on Stack Overflow and the user's code as well. It was suggested that this could be done by adding a hidden toggle menu to on each code clone that was found so that users could expand the toggle menu and see more details about the code. In addition, the participants also said that it would be better if in the future JARVAN could support more programming languages instead of just Java.

There were also suggestions for JARVAN to be improved in terms of its user interface and design. First, the participants suggested that on the results page, when users click the back button, it should be able to go back to the repository page, and not the login page. They also said that the tables represented in the system were too eye-catching and that it would be better if the table colors were softer. They also mentioned that it would be better if some pieces of text on each page could be highlighted to show that they were important pieces of information for the users. Currently, the text is believed to be too plain, so users do not understand what they should focus on. Lastly, they said that some buttons were too big, and that it would be better if those button sizes were reduced a little.

CHAPTER 6
CONCLUSION

This chapter summarizes and discusses the content of this project. The topics include the conclusion, problems and limitations faced when conducting the research, and any future work that could be done based on this project.

6.1 Conclusion

The main objective of the project is to allow the Java developers to check the reused source code from Stack Overflow. The developers of this project therefore created a tool called 'JARVAN' which was integrated with the Siamese Code Clones Search Tool and used data from Stack Exchange and SOTorrent. To provide flexibility and simplicity for the users, JARVAN was developed as a web application. Therefore, the users are able to use the system on any web browser on their own machine without software installation. Moreover, JARVAN connects with the GitHub account of the users to make it convenient for the users to analyze their software projects without the need to download their projects and analyze them locally. JARVAN was created to detect source codes that comes from Stack Overflow and notify the users of them via a report. The reporting system can notify two issues to the users. First, the system is able to inform the users when a source code was reused from Stack Overflow answers through a report, as the users are given a list of suspected files and methods that JARVAN detected. For example, the users can see when a certain source code has had an update and the users can also see the latest version on Stack Overflow. The second issue is the potential license conflict of the user's repositories. JARVAN can inform the users that their repositories may cause a license conflict violation when they reuse codes from Stack Overflow because they do not follow the guidelines of the CC BY-SA 4.0 license adopted by Stack Overflow. The JARVAN system had a precision score of 98 percent. With regards to its performance, JARVAN needs an average time of 0.389 seconds per method to search for clones. The developers strongly believe that JARVAN can help Java developers, who generally implement Java software projects, avoid reusing source codes from Stack Overflow answers without giving attribution to the source and violating the software license.

6.2 Problems and Limitations

At the start id the implementation, the developers planned to implement the JARVAN system on Spring Boot to tightly integrate it with the Siamese code clone search tool which was written in Java. However, there were several technical difficulties. As a result, the developers selected to use Node.js which it used to call Siamese by using shell commands. This was the most flexible solution for them to implement the web application of the JARVAN system at that time. Secondly, JARVAN was only able to search for a reused source code in the Java language.

Siamese was originally was designed to mainly detect code clones in Java, so the developers did not focus on enhancing Siamese so that it would be able to detect code clones in other languages. Therefore, JARVAN currently only supports the Java language. Thirdly, JARVAN can search for software projects stored on GitHub repositories only. Moreover, because of the convenience of the GitHub API provided by GitHub, the developers did not need to implement an internal system to handle file uploading. Nevertheless, some users may not store their software projects on GitHub, so it might be slightly inconvenient for them to use JARVAN. Lastly, at the moment, JARVAN cannot serve several requests concurrently as it can only handle a single request at a time.

6.3 Future Work

For future work, it would be better if JARVAN could display code snippets on Stack Overflow and the user's repository on the results so that the user can clearly compare and understand the similarities and differences between the source codes. It would increase the convenience for the user as then user would not need to switch pages back and forth between Stack Overflow and their GitHub repository to make a comparison. Additionally, JARVAN can be improved further to help developers by creating GitHub pull requests or issues. This will help developers receive a fix right away into their repository. Lastly, JARVAN can be enhanced to be able to handle concurrent requests by integrating Spring Boot with JARVAN since Spring Boot can manage a pool of connections and handle the distribution of entity managers. Consequently,it will automatically be able to embed a web container, such as Tomcat, to handle requests simultaneously just like common web containers.

APPENDIX A
LIST OF GITHUB PROJECTS USED IN THE EVALUATION

Project No.	Project Name	URL
1	AES-Message-Encryption-Decryption-With-Java-version-2	https://api.github.com/repos/tarik064/AES-Message-Encryption-Decryption-With-Java-version-2
2	alcina	https://api.github.com/repos/nevella/alcina
3	android-booking-app	https://api.github.com/repos/gorold/android-booking-app
4	android-client	https://api.github.com/repos/OPENCBS/android-client
5	ASE	https://api.github.com/repos/dja12123/ASE
6	avalon	https://api.github.com/repos/jinatonic/avalon
7	BibliotecaPOO	https://api.github.com/repos/VNeres/BibliotecaPOO
8	biotea-rdfization	https://api.github.com/repos/biotea/biotea-rdfization
9	bitsandbolts-checkmate	https://api.github.com/repos/swbest/bitsandbolts-checkmate
10	BridgeJavaSDK	https://api.github.com/repos/Sage-Bionetworks/BridgeJavaSDK
11	Capstone	https://api.github.com/repos/zacharynoel/Capstone
12	Checkers	https://api.github.com/repos/marcus433/Checkers
13	ChessGame	https://api.github.com/repos/Vadman97/ChessGame
14	cilicili-parent	https://api.github.com/repos/living2room/cilicili-parent
15	colims	https://api.github.com/repos/compomics/colims
16	ColorReport	https://api.github.com/repos/alclabs/ColorReport
17	CSYE7374_FinalProject	https://api.github.com/repos/brahmbhattspandan/CSYE7374_FinalProject
18	DayMoon	https://api.github.com/repos/deskmel/DayMoon
19	dietplanner	https://api.github.com/repos/code2rise/dietplanner
20	distfork-plugin	https://api.github.com/repos/jenkinsci/distfork-plugin
21	DIY	https://api.github.com/repos/BrainGoodbye/DIY
22	donut-maven-plugin	https://api.github.com/repos/DonutReport/donut-maven-plugin
23	ece466_2015	https://api.github.com/repos/wlz1028/ece466_2015
24	Envanter	https://api.github.com/repos/hakanozer/Envanter
25	estatePageScanner1	https://api.github.com/repos/Dusanstancik/estatePageScanner1
26	ets-kml22	https://api.github.com/repos/opengeospatial/ets-kml22
27	FlappyBird	https://api.github.com/repos/kazemicode/FlappyBird
28	fluxoAges	https://api.github.com/repos/agespucrs/fluxoAges
29	freelib-utils	https://api.github.com/repos/ksclarke/freelib-utils
30	FriendLines	https://api.github.com/repos/josuecm13/FriendLines
31	ftc5159-18	https://api.github.com/repos/richard808/ftc5159-18
32	gcp-java-endpoints	https://api.github.com/repos/Mg30/TinyPetition
33	GestorHorarios	https://api.github.com/repos/theIker/GestorHorarios
34	Glide	https://api.github.com/repos/liangdrew/Glide
35	graviton-worker-base-java	https://api.github.com/repos/libgraviton/graviton-worker-base-java
36	guardian-lite	https://api.github.com/repos/tonytw1/guardian-lite
37	Guesstimation	https://api.github.com/repos/drewsher96/Guesstimation
38	Hidden-Hills	https://api.github.com/repos/ecoatelant/Hidden-Hills
39	htl_schach_3b	https://api.github.com/repos/albertgreinoecker/htl_schach_3b
40	hub-sonarqube	https://api.github.com/repos/blackducksoftware/hub-sonarqube
41	HydrationApp	https://api.github.com/repos/nicPorcu/HydrationApp
42	jasa_smartlife_wg	https://api.github.com/repos/Seagull8491/jasa_emotion_wg
43	Java-EscribirEnArchivoDeTexto	https://api.github.com/repos/PipoLucido/Java-EscribirEnArchivoDeTexto
44	JavaFX---Map-Project	https://api.github.com/repos/kusoggakik/FantasyX
45	Java_Scuola	https://api.github.com/repos/EnricoRuggieroB/Java_Scuola
46	jazz-plugin-maven-archetype	https://api.github.com/repos/jazz-community/jazz-plugin-maven-archetype
47	jebu-core	https://api.github.com/repos/mikrosimage/jebu-core
48	JwRalph_Seo	https://api.github.com/repos/jungwonrs/JwRalph_Seo
49	kieker-monitoring	https://api.github.com/repos/zhang0908/kieker-monitoring
50	LarkServer	https://api.github.com/repos/hollykunge/LarkServer

Project No.	Project Name	URL
51	Merchants	https://api.github.com/repos/BrianLa0616/Merchants
52	midas-scheduling-plugin	https://api.github.com/repos/blakematis/midas-scheduling-plugin
53	MobileUSOZ	https://api.github.com/repos/piotrekopyd/MobileUSOZ
54	mule-module-cors	https://api.github.com/repos/mulesoft/mule-module-cors
55	myCINE	https://api.github.com/repos/socoolheeya/myCINE
56	News	https://api.github.com/repos/ZZWR1/News
57	nuxeo-csv	https://api.github.com/repos/nuxeo/nuxeo-csv
58	nuxeo-shell	https://api.github.com/repos/nuxeo/nuxeo-shell
59	ObServe	https://api.github.com/repos/Updownquark/ObServe
60	OfferNoProblem	https://api.github.com/repos/lengku8e/OfferNoProblem
61	online_program	https://api.github.com/repos/qiaofenlin/online_program
62	open-huirong	https://api.github.com/repos/pipipapi/open-huirong
63	oscm-interfaces	https://api.github.com/repos/servicecatalog/oscm-interfaces
64	pipeline-build-utils	https://api.github.com/repos/daisy/pipeline-build-utils
65	pistach.io	https://api.github.com/repos/pranavbudhwant/pistach.io
66	PlaylistGenerator	https://api.github.com/repos/StrictTangent/PlaylistGenerator
67	plugin-sharesite	https://api.github.com/repos/ArneBab/plugin-sharesite
68	prisoners-dilemma	https://api.github.com/repos/pcrglennon/prisoners-dilemma
69	Programacion-3	https://api.github.com/repos/man88GG/Programacion-3
70	projectbueno	https://api.github.com/repos/bgroman/projectbueno
71	Quiet	https://api.github.com/repos/vickeee97/Quiet
72	refactoring-toy-example	https://api.github.com/repos/danilofes/refactoring-toy-example
73	ReportSystemServer	https://api.github.com/repos/giraffeman123/ReportSystemServer
74	rtp	https://api.github.com/repos/hearmfield/rtp
75	Sadis	https://api.github.com/repos/igorsouzacarvalho88/Sadis
76	scriptiveunit	https://api.github.com/repos/dakusui/scriptiveunit
77	Semester-4	https://api.github.com/repos/henry-dv/Semester-4
78	sensorDemo	https://api.github.com/repos/Scion01/sensorDemo
79	service-web	https://api.github.com/repos/nus-ncl/service-web
80	silq	https://api.github.com/repos/silq/silq
81	SimGen	https://api.github.com/repos/PasternakMichal/SimGen
82	simple-mvn-project	https://api.github.com/repos/dhinojosa/simple-mvn-project
83	SMS	https://api.github.com/repos/soosaisjc/SMS
84	SRTI	https://api.github.com/repos/hlynka-a/SRTI
85	Stock_App_GRP2	https://api.github.com/repos/Zaheenie/Stock_App_GRP2
86	Student_Hub_FYP	https://api.github.com/repos/M-Asad-Chattha/Student_Hub_FYP
87	symphony-rest-tools	https://api.github.com/repos/symphonyoss/symphony-rest-tools
88	Tabla_periodica	https://api.github.com/repos/JHPAT100/Tabla_periodica
89	tacs-tp-2019c1	https://api.github.com/repos/coderfernando/tacs-tp-2019c1
90	TestingLabII	https://api.github.com/repos/Florsalcedowd/TestingLabII
91	Time	https://api.github.com/repos/Hippocampome-Org/Time
93	UML_Editor	https://api.github.com/repos/Connodore/UML_Editor
94	userMicroserviceTFM	https://api.github.com/repos/BrunoML1991/userMicroserviceTFM
95	videocalling	https://api.github.com/repos/imshaz/videocalling
96	Virusito	https://api.github.com/repos/r7perezyera/Virusito
97	WhatDidYouMean	https://api.github.com/repos/intiv/WhatDidYouMean
98	workspace_deluxe	https://api.github.com/repos/kbase/workspace_deluxe
99	xld-credential-on-host-plugin	https://api.github.com/repos/xebialabs-community/xld-credential-on-host-plugin
100	zoophy-services	https://api.github.com/repos/ZooPhy/zoophy-services

APPENDIX B
THE INFORMATION REGARDING PROJECTS USED IN THE EVALUATION

Project No.	Project Name	No. of Files	Processed Methods	All Methods	Size(MB)
1	AES-Message-Encryption-Decryption-With-Java-version-2	3	2	3	0.006
2	alcina	2,419	5,284	31,038	43.306
3	android-booking-app	12	24	126	1.057
4	android-client	110	66	317	1.108
5	ASE	103	172	509	11.143
6	avalon	48	75	240	10.608
7	BibliotecaPOO	17	49	348	1.908
8	biotea-rdfization	166	70	5,083	6.983
9	bitsandbolts-checkmate	15	28	68	0.252
10	BridgeJavaSDK	47	100	280	5.153
11	Capstone	43	100	487	5.455
12	Checkers	29	34	183	4.493
13	ChessGame	20	69	173	2.357
14	cilicili-parent	211	182	931	44.983
15	colims	569	753	3,567	247.317
16	ColorReport	5	9	22	0.296
17	CSYE7374_FinalProject	43	30	192	144.141
18	DayMoon	110	218	925	140.628
19	dietplanner	38	81	249	1.372
20	distfork-plugin	8	13	61	0.072
21	DIY	31	46	180	0.363
22	donut-maven-plugin	3	2	15	0.054
23	ece466_2015	97	101	269	125.884
24	Envanter	43	167	601	12.573
25	estatePageScanner1	26	52	211	1.804
26	ets-kml22	52	175	303	1.503
27	FlappyBird	3	0	3	189.815
28	fluxoAges	70	105	273	7.904
29	freelib-utils	59	261	628	1.401
30	FriendLines	52	101	397	0.915
31	ftc5159-18	21	81	114	0.068
32	gcp-java-endpoints	8	7	46	0.930
33	GestorHorarios	31	49	228	11.517
34	Glide	14	23	114	30.815
35	graviton-worker-base-java	150	200	785	0.674
36	guardian-lite	106	125	687	3.492
37	Guesstimation	9	13	35	0.238
38	Hidden-Hills	39	87	360	7.317
39	htl_schach_3b	27	8	127	0.333
40	hub-sonarqube	33	50	230	0.621
41	HydrationApp	43	156	460	6.931
42	jasa_smartlife_wg	184	116	691	80.667
43	Java-EscribirEnArchivoDeTexto	10	5	40	0.115
44	JavaFX---Map-Project	9	16	63	2.961
45	Java_Scuola	17	7	53	0.036
46	jazz-plugin-maven-archetype	5	5	5	0.111

Note: Processed method is the number of methods that was processed by JARVAN

Project No.	Project Name	No. of Files	Processed Methods	All Methods	Size(MB)
47	jebu-core	123	6	2,957	0.443
48	JwRalph_Seo	2,845	6,105	18,679	112.533
49	kieker-monitoring	262	363	1,189	3.117
50	LarkServer	500	431	2577	1.907
51	Merchants	23	39	244	5.838
52	midas-scheduling-plugin	71	76	727	1.368
53	MobileUSOZ	63	155	529	11.504
54	mule-module-cors	12	34	93	0.125
55	myCINE	93	142	392	12.336
56	News	3	0	3	0.122
57	nuxeo-csv	0	0	0	0.694
58	nuxeo-shell	232	246	1,412	1.759
59	ObServe	98	553	3,453	12.817
60	OfferNoProblem	3	0	3	1.036
61	online_program	80	115	442	16.100
62	open-huirong	24	68	193	957.825
63	oscm-interfaces	566	69	3,976	65.677
64	pipeline-build-utils	28	49	187	0.572
65	pistach.io	25	53	210	34.688
66	PlaylistGenerator	19	35	114	41.553
67	plugin-sharesite	142	207	1,101	0.445
68	prisoners-dilemma	14	45	103	1.132
69	Programacion-3	34	106	328	1.311
70	projectbueno	22	23	96	0.066
71	Quiet	94	117	539	0.179
72	refactoring-toy-example	18	4	51	0.384
73	ReportSystemServer	25	51	208	12.665
74	rtp	5	11	46	6.028
75	Sadis	339	2,282	8,641	13.353
76	scriptiveunit	128	197	1,140	4.830
77	Semester-4	17	14	63	0.057
78	sensorDemo	11	19	81	0.296
79	service-web	130	312	1,441	54.984
80	silq	153	150	519	54.154
81	SimGen	165	1,345	3,335	106.242
82	simple-mvn-project	23	11	36	42.216
83	SMS	2	1	5	0.009
84	SRTI	54	268	600	437.455
85	Stock_App_GRP2	7	16	55	0.182
86	Student_Hub_FYP	66	168	618	8.458
87	symphony-rest-tools	106	138	691	4.420
88	Tabla_periodica	9	1	34	58.029
89	tacs-tp-2019c1	56	21	368	15.849
90	TestingLabII	36	21	116	2.200
91	Time	181	692	1,933	29.288
92	uade-ad-restapi	6	18	51	0.071
93	UML_Editor	18	16	177	0.187
94	userMicroserviceTFM	44	19	185	0.208
95	videocalling	6	30	174	0.568
96	Virusito	38	100	315	62.242
97	WhatDidYouMean	7	9	37	1.953
98	workspace_deluxe	344	1,814	5,386	32.318
99	xld-credential-on-host-plugin	4	3	13	0.357
100	zoophy-services	95	169	473	6.431

Note: Processed method is the number of methods that was processed by JARVAN

APPENDIX C
MANUAL VALIDATION RESULTS

Project No.	Project Name	Method Name	Phattharapong	Panaya	Kanika	Conclusion
1	AES-Message-Encryption-Decryption-With-Java-version-2	encrypt	1	1	1	1
		decrypt	1	1	1	1
2	alcina	base64Append	1	1	1	1
		base64Value	1	1	1	1
		longFromBase6	1	1	1	1
		longToBase64	1	1	1	1
		longFromBase64	1	1	1	1
		base64Value	1	1	1	1
		toBase64	1	1	1	1
		base64Append	1	1	1	1
		removeIf	1	1	1	1
		longFromBase64	1	1	1	1
		base64Append	1	1	1	1
		base64Value	1	1	1	1
		toBase64	1	1	1	1
		register	1	1	1	1
		registerAll	1	1	1	1
		WatchDir	1	1	1	1
		hashCode	0	b	1	b
		prettyPrintWithDOM3LS	1	1	1	1
		stripNonValidXMLCharacters	1	b	1	1
		convertToHex	1	b	1	1
		disableSslValidation	1	1	1	1
		forHTMLTextFlow	1	1	1	1
		splitQuery	1	1	1	1
		expandAll	1	1	1	1
		toBase64	1	1	1	1
		base64Append	1	1	1	1
		showStack	1	1	1	1
		showStack	1	1	1	1
		getString	1	b	1	1
		base64Append	1	1	1	1
		base64Value	1	1	1	1
		longFromBase64	1	1	1	1
		longToBase64	1	1	1	1
3	android-booking-app	onCreateDialog	1	b	1	1
7	BibliotecaPOO	main	b	1	1	b
		main	b	1	1	b
		main	b	1	1	b
8	biotea-rdfization	main	1	1	1	1
9	bitsandbolts-checkmate	getDeviceLocation	1	1	1	1
		getLocationPermission	1	b	1	1
		onRequestPermissionsResult	1	b	1	1
		getDeviceLocation	1	1	1	1
		onComplete	1	1	1	1
		getLocationPermission	1	b	1	1
		onRequestPermissionsResult	1	1	1	1

Note: Not Clone = 0, Clone = 1, Boilerplate = b

Project No.	Project Name	Method Name	Phattharapong	Panaya	Kanika	Conclusion
11	Capstone	onClick	0	0	0	0
		mayRequestContacts	1	1	1	1
		onPreferenceChange	1	b	1	1
		onBackPressed	0	0	0	0
		onActivityResult	0	0	1	0
		onCreate	b	0	1	b
		onOptionsItemSelected	b	b	1	b
17	CSYE7374_FinalProject	home	1	1	1	1
		main	b	b	1	b
18	DayMoon	verifyStoragePermissions	1	1	b	1
		bytesToHexString	1	b	1	1
19	dietplanner	onAttach	b	b	1	b
		onFling	b	b	1	b
		mapProperties	1	1	1	1
		decodeFile	1	1	1	1
24	Envanter	MD5	1	1	1	1
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		MD5	1	1	1	1
		main	b	b	b	b
		MD5	1	1	1	1
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		MD5	1	1	1	1
		main	b	b	b	b
		main	b	b	b	b
25	estatePageScanner1	createWorker	1	1	1	1
		createWorker	1	1	1	1
		createWorker	1	1	1	1
28	fluxoAges	generateCsvFile	1	1	1	1
30	FriendLines	onOptionsItemSelected	b	b	1	b
		onOptionsItemSelected	b	b	b	b
		onOptionsItemSelected	b	b	b	b
41	HydrationApp	onCreateView	0	b	1	b
		onOptionsItemSelected	b	b	b	b
		createNotificationChannel	b	b	b	b
42	jasa_smartlife_wg	onDestroy	b	b	b	b
		onNavigationItemSelected	b	b	b	b

Note: Not Clone = 0, Clone = 1, Boilerplate = b

Project No.	Project Name	Method Name	Phattharapong	Panaya	Kanika	Conclusion
48	JwRalph_Seo	hexToByteArray	1,b	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
		hexToByteArray	1	1	1	1
		byteArrayToHex	1	1	1	1
50	LarkServer	test	0	0	0	0
		deserialize	1	1	b	1
		getClientIp	1	1	0	1
52	midas-scheduling-plugin	fillIntChars	1	1	1	1
53	MobileUSOZ	setupNavigation	1	1	b	1
		onActivityResult	b	b	1	b
		handleSignInResult	b	b	b	b
		onComplete	b	b	0	b
59	ObServe	remove	b	1	1	1
61	online_program	main	0	0	0	0
		contextLoads	0	0	0	0
63	oscm-interfaces	handleMessage	1	1	1	1
65	pistach.io	loadImagefromGallery	b	b	1	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
		onAttach	b	b	b	b
66	PlaylistGenerator	onCreate	b	b	b	b
68	prisoners-dilemma	printDataToFile	0	1	1	1

Note: Not Clone = 0, Clone = 1, Boilerplate = b

Project No.	Project Name	Method Name	Phattharapong	Panaya	Kanika	Conclusion
69	Programacion-3	main	b	b	1	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		main	b	b	b	b
		initComponents	b	b	1	b
		main	b	b	1	b
		main	b	b	1	b
		initComponents	b	b	1	b
		main	b	b	1	b
71	Quiet	main	b	1	1	b
		main	b	1	1	b
73	ReportSystemServer	write	b	1	1	1
75	Sadis	fastTimestampCreate	1	1	1	1
		setObject	1	1	1	1
78	sensorDemo	onAttach	b	1	1	b
		onAttach	b	1	1	b
		onAttach	b	1	1	b
79	service-web	encrypt	1	1	1	1
80	silq	emailTemplateResolver	b	1	1	1
		isBlank	1	b	1	1
		getCurrentUser	1	1	1	1
		computeSignature	1	b	1	1
84	SRTI	DragIcon	1	1	1	1
		CubicCurveDemo	1	1	1	1
		RootLayout	1	1	1	1
86	Student_Hub_FYP	onClick	0	0	0	0
		onPreferenceChange	0	1	1	1
		onCreate	0	0	b	0
		onDataChange	1	0	1	1
		onCreateOptionsMenu	b	0	b	b
88	Tabla_periodica	onAttach	b	b	b	b
91	Time	numberToString	1	1	1	1
		writeValue	1	1	1	1
		shuffleArray	0	1	1	1
		isNumeric	1	1	1	1
		isNumeric	1	1	1	1
92	uade-ad-restapi	home	b	b	b	b
98	workspace_deluxe	equals	b	b	b	b
		equals	b	b	b	b
		hashCode	1	b	1	b
		getResourceListing	b	b	b	b
		getResourceListing	b	b	b	b
		getResourceListing	b	b	b	b
		getResourceListing	b	b	b	b
		getResourceListing	b	b	b	b
		getResourceListing	b	b	b	b
100	zoophy-services	sortByValue	b	1	1	b
		sortCountryMap	1	1	1	1

Note: Not Clone = 0, Clone = 1, Boilerplate = b

APPENDIX D
PROJECT CLONING TIME

Project No.	Project Name	1st Time (sec.)	2nd Time (sec.)	3rd Time (sec.)	SUM (sec.)	AVG (sec.)
1	AES-Message-Encryption-Decryption-With-Java-version-2	2.134	2.037	2.454	6.624	2.208
2	alcina	9.945	10.251	13.966	34.162	11.387
3	android-booking-app	2.536	2.598	3.165	8.299	2.766
4	android-client	2.196	2.185	2.151	6.532	2.177
5	ASE	12.666	3.382	3.457	19.506	6.502
6	avalon	4.453	3.499	3.314	11.266	3.755
7	BibliotecaPOO	2.269	2.235	3.149	7.653	2.551
8	biotea-rdfization	3.048	10.190	2.936	16.173	5.391
9	bitsandbolts-checkmate	2.912	2.133	2.097	7.143	2.381
10	BridgeJavaSDK	2.919	3.583	3.202	9.704	3.235
11	Capstone	3.178	3.421	2.573	9.172	3.057
12	Checkers	2.613	3.554	2.840	9.006	3.002
13	ChessGame	9.391	3.002	2.240	14.632	4.877
14	cilicili-parent	7.168	17.929	7.616	32.712	10.904
15	colims	30.776	40.740	31.321	102.837	34.279
16	ColorReport	2.517	2.060	2.124	6.701	2.234
17	CSYE7374_FinalProject	34.240	17.322	17.095	68.657	22.886
18	DayMoon	18.308	16.170	17.300	51.778	17.259
19	dietplanner	2.326	3.320	3.198	8.844	2.948
20	distfork-plugin	2.145	2.241	2.122	6.507	2.169
21	DIY	2.802	2.728	2.272	7.802	2.601
22	donut-maven-plugin	2.086	2.364	2.073	6.523	2.174
23	ece466_2015	19.249	15.373	15.106	49.727	16.576
24	Envanter	3.547	4.165	9.195	16.907	5.636
25	estatePageScanner1	3.196	2.515	8.628	14.338	4.779
26	ets-kml22	2.366	2.180	2.368	6.915	2.305
27	FlappyBird	27.599	20.724	20.450	68.772	22.924
28	fluxoAges	3.560	3.531	3.676	10.767	3.589
29	freelib-utils	3.019	2.419	2.205	7.644	2.548
30	FriendLines	2.869	2.320	2.124	7.313	2.438
31	ftc5159-18	2.545	2.046	1.992	6.583	2.194
32	gcp-java-endpoints	4.442	2.176	2.164	8.782	2.927
33	GestorHorarios	3.707	4.417	3.803	11.927	3.976
34	Glide	5.910	5.630	5.311	16.851	5.617
35	graviton-worker-base-java	2.845	4.542	2.643	10.030	3.343
36	guardian-lite	2.395	2.795	3.230	8.420	2.807
37	Guesstimation	2.533	2.130	2.461	7.124	2.375
38	Hidden-Hills	3.465	2.894	2.978	9.337	3.112
39	htl_schach_3b	2.147	3.444	2.875	8.466	2.822
40	hub-sonarqube	2.780	2.521	2.173	7.473	2.491
41	HydrationApp	2.784	3.146	2.835	8.765	2.922
42	jasa_smartlife_wg	19.134	14.878	13.418	47.430	15.810
43	Java-EscribirEnArchivoDeTexto	2.805	2.141	2.098	7.044	2.348
44	JavaFX---Map-Project	3.615	4.022	4.885	12.522	4.174
45	Java_Scuola	2.078	2.060	2.049	6.187	2.062
46	jazz-plugin-maven-archetype	2.084	2.113	2.113	6.309	2.103
47	jebu-core	2.207	2.968	3.066	8.241	2.747
48	JwRalph_Seo	28.675	40.385	36.925	105.985	35.328
49	kieker-monitoring	4.870	3.069	2.801	10.740	3.580
50	LarkServer	2.507	3.807	2.623	8.937	2.979

Project No.	Project Name	1st Time (sec.)	2nd Time (sec.)	3rd Time (sec.)	SUM (sec.)	AVG (sec.)
51	Merchants	3.361	2.922	2.698	8.981	2.994
52	midas-scheduling-plugin	2.190	2.171	2.785	7.146	2.382
53	MobileUSOZ	6.457	6.702	7.770	20.928	6.976
54	mule-module-cors	2.213	2.907	2.845	7.966	2.655
55	myCINE	3.969	4.126	3.316	11.410	3.803
56	News	2.058	2.369	2.001	6.427	2.142
57	nuxeo-csv	2.922	2.626	2.223	7.771	2.590
58	nuxeo-shell	4.138	2.800	3.005	9.943	3.314
59	ObServe	4.193	4.594	4.242	13.029	4.343
60	OfferNoProblem	2.796	2.351	3.113	8.260	2.753
61	online_program	5.626	4.047	3.731	13.405	4.468
62	open-huirong	74.246	66.626	72.268	213.141	71.047
63	oscm-interfaces	11.659	10.425	12.080	34.164	11.388
64	pipeline-build-utils	3.292	3.867	2.216	9.375	3.125
65	pistach.io	5.975	6.620	12.232	24.827	8.276
66	PlaylistGenerator	6.634	7.086	8.139	21.859	.286
67	plugin-sharesite	.2.269	2.317	2.359	6.945	2.315
68	prisoners-dilemma	2.749	2.921	2.144	7.814	2.605
69	Programacion-3	2.186	2.356	2.426	6.968	2.323
70	projectbueno	2.170	2.253	2.330	6.753	2.251
71	Quiet	2.163	2.789	2.122	7.074	2.358
72	refactoring-toy-example	2.129	2.182	2.068	6.378	2.126
73	ReportSystemServer	5.350	7.327	5.019	17.696	5.899
74	rtp	3.686	3.150	2.582	9.418	3.139
75	Sadis	4.469	3.845	3.650	11.965	3.988
76	scriptiveunit	2.817	3.584	2.872	9.273	3.091
77	Semester-4	2.127	2.105	2.018	6.250	2.083
78	sensorDemo	2.063	2.138	2.098	6.300	2.100
79	service-web	9.490	8.944	18.788	37.222	12.407
80	silq	9.925	9.858	11.743	31.526	10.509
81	SimGen	13.764	22.670	13.420	49.854	16.618
82	simple-mvn-project	7.912	6.947	6.620	21.478	7.159
83	SMS	2.178	2.061	2.109	6.348	2.116
84	SRTI	53.939	52.948	51.005	157.891	52.630
85	Stock_App_GRP2	2.433	2.102	2.151	6.686	2.229
86	Student_Hub_FYP	3.963	2.981	3.016	9.961	3.320
87	symphony-rest-tools	3.173	3.506	3.190	9.869	3.290
88	Tabla_periodica	8.192	8.439	8.344	24.975	8.325
89	tacs-tp-2019c1	5.398	3.621	4.061	13.080	4.360
90	TestingLabII	2.585	2.247	3.215	8.047	2.682
91	Time	15.230	6.042	5.114	26.386	8.795
92	uade-ad-restapi	2.130	2.378	2.649	6.858	2.286
93	UML_Editor	2.229	2.318	2.147	6.694	2.231
94	userMicroserviceTFM	2.904	2.153	2.159	7.217	2.106
95	videocalling	2.290	2.227	2.360	6.878	2.293
96	Virusito	9.718	10.894	9.298	29.911	9.970
97	WhatDidYouMean	3.328	2.939	3.132	9.409	3.136
98	workspace_deluxe	17.896	8.353	8.273	34.522	11.507
99	xld-credential-on-host-plugin	2.171	2.108	2.116	6.395	2.132
100	zoophy-services	3.165	2.716	3.372	9.252	3.084

APPENDIX E
PROJECT ANALYSIS TIME

Project No.	Project Name	1st Time (sec.)	2nd Time (sec.)	3rd Time (sec.)	SUM (sec.)	AVG (sec.)
1	AES-Message-Encryption-Decryption-With-Java-version-2	4.173	3.700	4.112	11.985	3.995
2	alcina	487.964	487.011	491.783	1466.758	488.919
3	android-booking-app	10.763	10.302	10.704	31.769	10.590
4	android-client	12.579	12.995	12.673	38.247	12.749
5	ASE	22.051	22.033	22.162	66.246	22.082
6	avalon	13.839	14.307	13.954	42.100	14.033
7	BibliotecaPOO	14.144	14.055	13.861	42.060	14.020
8	biotea-rdfization	29.105	29.078	28.611	86.795	28.932
9	bitsandbolts-checkmate	7.746	7.297	7.529	22.573	7.524
10	BridgeJavaSDK	15.102	15.115	15.090	45.307	15.102
11	Capstone	18.538	18.408	18.659	55.605	18.535
12	Checkers	8.806	9.142	9.391	27.338	9.113
13	ChessGame	12.575	12.699	12.686	37.960	12.653
14	cilicili-parent	27.503	26.958	27.726	82.187	27.396
15	colims	130.945	129.922	131.454	392.321	130.774
16	ColorReport	4.647	5.235	5.427	15.309	5.103
17	CSYE7374_FinalProject	8.040	8.470	9.002	25.512	8.504
18	DayMoon	34.740	35.694	33.974	104.408	34.803
19	dietplanner	15.129	14.410	14.739	44.279	14.760
20	distfork-plugin	5.075	5.372	5.056	15.503	5.168
21	DIY	10.491	10.211	10.080	30.783	10.261
22	donut-maven-plugin	3.640	3.697	3.598	10.935	3.645
23	ece466_2015	18.544	18.989	18.374	55.907	18.636
24	Envanter	40.997	40.517	39.875	121.389	40.463
25	estatePageScanner1	9.741	10.421	9.701	29.862	9.954
26	ets-kml22	24.673	24.683	23.614	72.970	24.323
27	FlappyBird	2.331	2.424	2.339	7.095	2.365
28	fluxoAges	17.637	17.488	17.640	52.765	17.588
29	freelib-utils	25.547	26.043	25.306	76.895	25.632
30	FriendLines	17.539	16.534	16.962	51.035	17.012
31	ftc5159-18	16.157	16.347	15.512	48.016	16.005
32	gcp-java-endpoints	4.837	4.852	4.369	14.058	4.686
33	GestorHorarios	8.137	8.192	8.106	24.435	8.145
34	Glide	6.850	7.009	7.199	21.057	7.019
35	graviton-worker-base-java	21.821	21.726	21.927	65.475	21.825
36	guardian-lite	18.538	18.503	18.592	55.632	18.544
37	Guesstimation	6.828	6.196	6.888	19.911	6.637
38	Hidden-Hills	13.363	13.469	13.260	40.093	13.364
39	htl_schach_3b	4.779	4.359	4.704	13.843	4.614
40	hub-sonarqube	9.507	8.822	9.592	27.921	9.307
41	HydrationApp	21.621	21.615	21.805	65.041	21.680
42	jasa_smartlife_wg	20.129	20.703	20.745	61.577	20.526
43	Java-EscribirEnArchivoDeTexto	4.469	4.517	4.540	13.527	4.509
44	JavaFX---Map-Project	5.781	5.220	5.283	16.284	5.428
45	Java_Scuola	6.014	5.510	6.094	17.619	5.873
46	jazz-plugin-maven-archetype	3.625	3.490	3.547	10.662	3.554
47	jebu-core	6.253	5.976	6.121	18.351	6.117
48	JwRalph_Seo	507.745	508.518	506.113	1522.376	507.459
49	kieker-monitoring	48.454	47.429	48.050	143.934	47.978
50	LarkServer	53.107	54.215	52.763	160.086	53.362

Project No.	Project Name	1st Time (sec.)	2nd Time (sec.)	3rd Time (sec.)	SUM (sec.)	AVG (sec.)
51	Merchants	11.127	11.072	11.038	33.237	11.079
52	midas-scheduling-plugin	11.763	11.783	11.847	35.393	11.798
53	MobileUSOZ	21.299	21.479	21.895	64.673	21.558
54	mule-module-cors	7.067	7.400	7.225	21.693	7.231
55	myCINE	21.285	20.376	21.048	62.709	20.903
56	News	2.475	2.487	2.431	7.393	2.464
57	nuxeo-csv	2.049	1.978	2.029	6.056	2.019
58	nuxeo-shell	30.775	30.828	31.189	92.791	30.930
59	ObServe	75.899	74.559	75.032	225.490	75.163
60	OfferNoProblem	2.483	2.473	2.402	7.357	2.452
61	online_program	17.557	16.937	17.469	51.963	17.321
62	open-huirong	11.278	11.112	10.970	33.360	11.120
63	oscm-interfaces	11.653	12.135	11.962	35.750	11.917
64	pipeline-build-utils	14.956	14.665	14.696	44.317	14.772
65	pistach.io	15.602	14.811	15.520	45.933	15.311
66	PlaylistGenerator	8.229	8.446	8.392	25.066	8.355
67	plugin-sharesite	29.439	29.656	29.456	88.551	29.517
68	prisoners-dilemma	9.216	9.209	9.207	27.633	9.211
69	Programacion-3	23.903	24.181	24.272	72.356	24.119
70	projectbueno	5.703	5.681	5.311	16.696	5.565
71	Quiet	16.943	16.927	16.702	50.571	16.857
72	refactoring-toy-example	3.530	3.797	3.637	10.964	3.655
73	ReportSystemServer	8.947	8.834	8.862	26.643	8.881
74	rtp	5.408	5.553	5.240	16.201	5.400
75	Sadis	331.529	329.285	330.334	991.149	330.383
76	scriptiveunit	22.825	22.562	23.051	68.438	22.813
77	Semester-4	5.712	5.717	5.916	17.346	5.782
78	sensorDemo	6.365	7.184	7.040	20.589	6.863
79	service-web	52.644	52.342	52.106	157.093	52.364
80	silq	19.633	19.468	19.962	59.064	19.688
81	SimGen	128.297	128.707	125.646	382.651	127.550
82	simple-mvn-project	4.802	4.795	4.699	14.296	4.765
83	SMS	3.412	3.320	3.376	10.108	3.369
84	SRTI	58.306	58.096	57.869	174.270	58.090
85	Stock_App_GRP2	6.789	7.825	7.537	22.150	7.383
86	Student_Hub_FYP	30.119	30.814	30.461	91.394	30.465
87	symphony-rest-tools	19.350	19.799	19.241	58.390	19.463
88	Tabla_periodica	5.381	5.406	5.395	16.182	5.394
89	tacs-tp-2019c1	7.204	7.216	7.148	21.568	7.189
90	TestingLabII	5.819	5.633	5.875	17.327	5.776
91	Time	95.215	94.374	95.170	284.759	94.920
92	uade-ad-restapi	6.460	6.646	5.978	19.084	6.361
93	UML_Editor	7.365	7.300	7.093	21.758	7.253
94	userMicroserviceTFM	5.925	5.907	5.445	17.277	5.759
95	videocalling	8.869	9.231	9.535	27.636	9.212
96	Virusito	19.005	19.147	18.982	57.134	19.045
97	WhatDidYouMean	5.894	6.450	6.347	18.692	6.231
98	workspace_deluxe	274.531	276.157	276.663	827.351	275.784
99	xld-credential-on-host-plugin	4.044	4.123	3.800	11.967	3.989
100	zoophy-services	35.214	34.986	35.233	105.433	35.144

APPENDIX F
PARTICIPANT BACKGROUND

Participant #1	Where do you currently work? Please specify a company name.	Via Group (Thailand) Co., Ltd.
	What is your position?	Android Developer
	How long have you been working since you have graduated? (i.e., 1 year 2months)	2 years
	What are the programming languages that you have used for work? (i.e., Java, Python, C)	Kotlin, Java, Python
	How long have you been programming with Java language? (i.e., 3 years 3 months)	5 years
	Do you know what code clones are?	Yes
	Do you think that code clones are problems in software projects?	Yes
	Have you ever copied some code from the internet (i.e., Stack Overflow)?	Yes
	Do you know that code snippets from Stack Overflow are under CC BY-SA 4.0 license?	No
	Based on the statements given, please indicate your agreement on this statement. [Code clones can cause problems in software projects.]	Agree
	Based on the statements given, please indicate your agreement with this statement. [Code clones should be eliminated from software projects.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [Developers should not copy code from the internet.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [When developers copy code from the internet, they should give a reference to the owner of the code.]	Strongly Agree

Participant #2	Where do you currently work? Please specify a company name.	Microsoft Thailand
	What is your position?	Office 365 Consumption Hero
	How long have you been working since you have graduated? (i.e., 1 year 2months)	1 year 7 months
	What are the programming languages that you have used for work? (i.e., Java, Python, C)	JavaScript, C#, Python, Power Fx
	How long have you been programming with Java language? (i.e., 3 years 3 months)	1 year
	Do you know what code clones are?	No
	Do you think that code clones are problems in software projects?	No
	Have you ever copied some code from the internet (i.e., Stack Overflow)?	Yes
	Do you know that code snippets from Stack Overflow are under CC BY-SA 4.0 license?	Yes
	Based on the statements given, please indicate your agreement on this statement. [Code clones can cause problems in software projects.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [Code clones should be eliminated from software projects.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [Developers should not copy code from the internet.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [When developers copy code from the internet, they should give a reference to the owner of the code.]	Neutral

Participant #3	Where do you currently work? Please specify a company name.	CPFIT
	What is your position?	Implementer
	How long have you been working since you have graduated? (i.e., 1 year 2months)	2 years
	What are the programming languages that you have used for work? (i.e., Java, Python, C)	Python
	How long have you been programming with Java language? (i.e., 3 years 3 months)	1 years
	Do you know what code clones are?	No
	Do you think that code clones are problems in software projects?	Yes
	Have you ever copied some code from the internet (i.e., Stack Overflow)?	Yes
	Do you know that code snippets from Stack Overflow are under CC BY-SA 4.0 license?	No
	Based on the statements given, please indicate your agreement on this statement. [Code clones can cause problems in software projects.]	Neutral
	Based on the statements given, please indicate your agreement with this statement. [Code clones should be eliminated from software projects.]	Disagree
	Based on the statements given, please indicate your agreement with this statement. [Developers should not copy code from the internet.]	Strongly Disagree
	Based on the statements given, please indicate your agreement with this statement. [When developers copy code from the internet, they should give a reference to the owner of the code.]	Agree

Participant #4	Where do you currently work? Please specify a company name.	AppMan Co., Ltd.
	What is your position?	Softre Support and Maintenance
	How long have you been working since you have graduated? (i.e., 1 year 2months)	2 years
	What are the programming languages that you have used for work? (i.e., Java, Python, C)	Python, JavaScript, C#
	How long have you been programming with Java language? (i.e., 3 years 3 months)	3 years
	Do you know what code clones are?	Yes
	Do you think that code clones are problems in software projects?	No
	Have you ever copied some code from the internet (i.e., Stack Overflow)?	Yes
	Do you know that code snippets from Stack Overflow are under CC BY-SA 4.0 license?	No
	Based on the statements given, please indicate your agreement on this statement. [Code clones can cause problems in software projects.]	Disagree
	Based on the statements given, please indicate your agreement with this statement. [Code clones should be eliminated from software projects.]	Disagree
	Based on the statements given, please indicate your agreement with this statement. [Developers should not copy code from the internet.]	Strongly Disagree
	Based on the statements given, please indicate your agreement with this statement. [When developers copy code from the internet, they should give a reference to the owner of the code.]	Agree

Participant #5	Where do you currently work? Please specify a company name.	SCG
	What is your position?	Technology Developer
	How long have you been working since you have graduated? (i.e., 1 year 2months)	2 years
	What are the programming languages that you have used for work? (i.e., Java, Python, C)	Javascripts
	How long have you been programming with Java language? (i.e., 3 years 3 months)	1 years
	Do you know what code clones are?	No
	Do you think that code clones are problems in software projects?	No
	Have you ever copied some code from the internet (i.e., Stack Overflow)?	Yes
	Do you know that code snippets from Stack Overflow are under CC BY-SA 4.0 license?	No
	Based on the statements given, please indicate your agreement on this statement. [Code clones can cause problems in software projects.]	Agree
	Based on the statements given, please indicate your agreement with this statement. [Code clones should be eliminated from software projects.]	Agree
	Based on the statements given, please indicate your agreement with this statement. [Developers should not copy codes from the internet.]	Agree
	Based on the statements given, please indicate your agreement with this statement. [When developers copy code from the internet, they should give a reference to the owner of the code.]	Agree

APPENDIX G
TEST QUESTIONS

Participant #1	How many repositories are shown?	3
	Which repository is/are private repository(s)?	Project3
Test case 1	How many issues were found?	One
	What are the results of the analysis? (select all correct answers)?	Code clones are found
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 2	How many issues were found?	No issue is found
	What are the results of the analysis? (select all correct answers)?	JARVAN cannot find any code clones
	How many clones were found?	0
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 3	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found;Software license violation
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	Apache License 2.0

Participant #2	How many repositories are shown?	3
	Which repository is/are private repository(s)?	Project3
Test case 1	How many issues were found?	One
	What are the results of the analysis? (select all correct answers)?	Code clones are found
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 2	How many issues were found?	No issue is found
	What are the results of the analysis? (select all correct answers)?	JARVAN cannot find any code clones
	How many clones were found?	0
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 3	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found;Software license violation
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	Apache License 2.0

Participant #3	How many repositories are shown?	3
	Which repository is/are private repository(s)?	Project3
Test case 1	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 2	How many issues were found?	No issue is found
	What are the results of the analysis? (select all correct answers)?	JARVAN cannot find any code clones
	How many clones were found?	0
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 3	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found;Software license violation
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	Apache License 2.0

Participant #4	How many repositories are shown?	3
	Which repository is/are private repository(s)?	Project3
Test case 1	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 2	How many issues were found?	No issue is found
	What are the results of the analysis? (select all correct answers)?	JARVAN cannot find any code clones
	How many clones were found?	0
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 3	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found;Software license violation
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	Apache License 2.0

Participant #5	How many repositories are shown?	3
	Which repository is/are private repository(s)?	Project3
Test case 1	How many issues were found?	One
	What are the results of the analysis? (select all correct answers)?	Code clones are found
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 2	How many issues were found?	No issue is found
	What are the results of the analysis? (select all correct answers)?	JARVAN cannot find any code clones
	How many clones were found?	0
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	There is no license
Test case 3	How many issues were found?	Two
	What are the results of the analysis? (select all correct answers)?	Code clones are found;Software license violation
	How many clones were found?	2
	What is the software license that violates the CC BY-SA 4.0 license in this repository?	Apache License 2.0

APPENDIX H
USER EXPERIENCE

Participant #1	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to be more aware when reusing a code from Stack Overflow?]	2
	Any comments or suggestions to improve JARVAN?	-Create JARVAN as plugins for other software development tools. For example, running JARVAN using GitHub action to review the code clone coverage, or running JARVAN on IDEs to Create developer code clone awareness. -Create tutorials on software License topics and what should or should not be done.

Participant #2	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to be more aware when reusing a code from Stack Overflow?]	3
	Any comments or suggestions to improve JARVAN?	

Participant #3	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?]	5
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to be more aware when reusing a code from Stack Overflow?]	4
	Any comments or suggestions to improve JARVAN?	

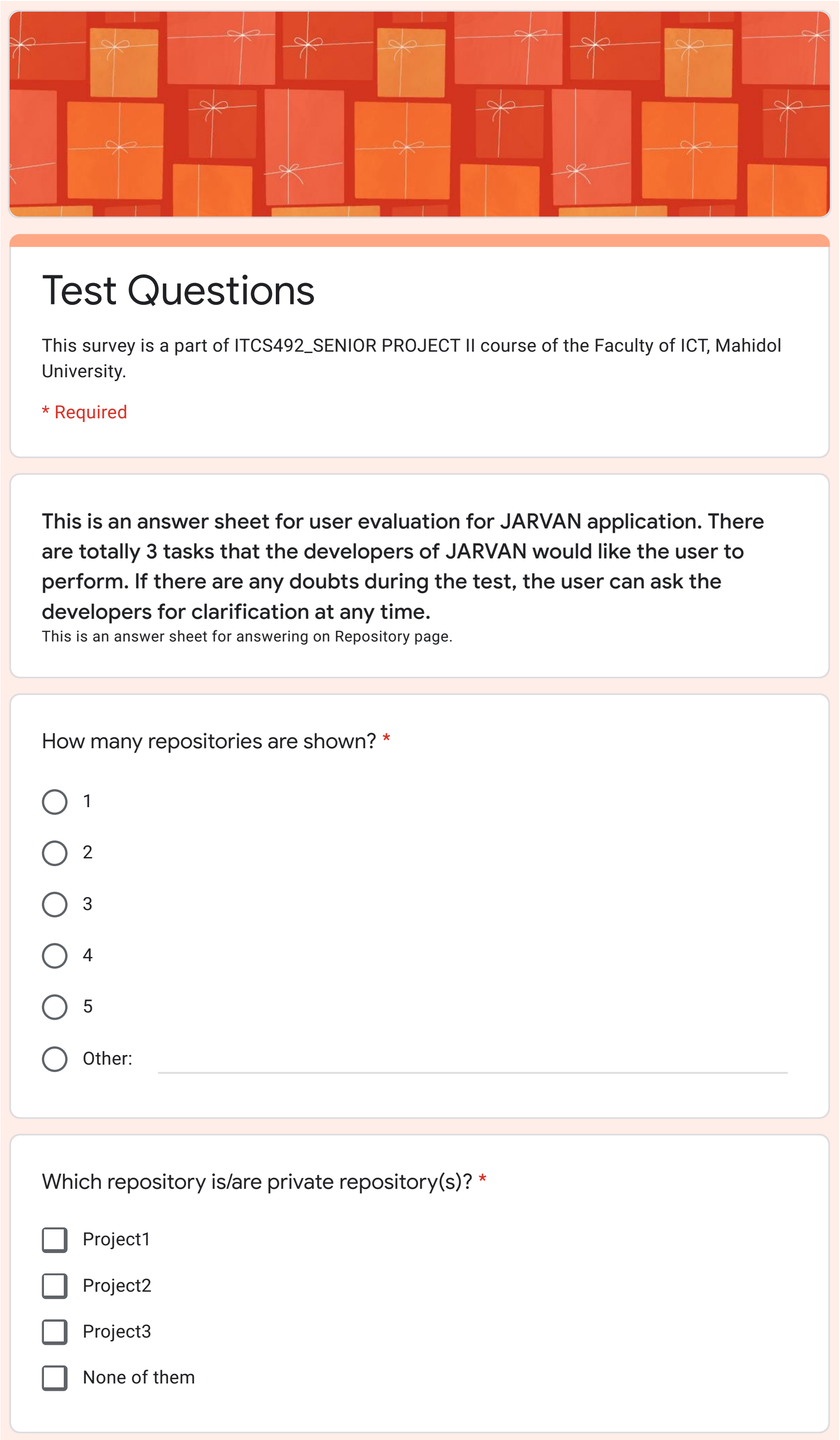
Note: 1=very least, 5=very much

Participant #4	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects?]	5
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?]	5
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to be more aware when reusing a code from Stack Overflow?]	5
	Any comments or suggestions to improve JARVAN?	- Help text suggestions how to use it - Colors in table (Pastel) - Help text/link beside cc by-sa 4.0 to help the user know more about this - Highlight in Red the License Name for user to notice it

Participant #5	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN can help you locate a code that has been copied from Stack Overflow (and you may already forget) in your projects?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to make a references to a code copied from Stack Overflow?]	4
	Based on the questions given, please indicate your opinion on those questions. (1 very least - 5 very much) [How much do you think JARVAN helps you to be more aware when reusing a code from Stack Overflow?]	4
	Any comments or suggestions to improve JARVAN?	

Note: 1=very least, 5=very much

APPENDIX I
TEST QUESTIONS



Test Questions

This survey is a part of ITCS492_SENIOR PROJECT II course of the Faculty of ICT, Mahidol University.

* Required

This is an answer sheet for user evaluation for JARVAN application. There are totally 3 tasks that the developers of JARVAN would like the user to perform. If there are any doubts during the test, the user can ask the developers for clarification at any time.

This is an answer sheet for answering on Repository page.

How many repositories are shown? *

○ 1

○ 2

○ 3

○ 4

○ 5

○ Other: ____________________

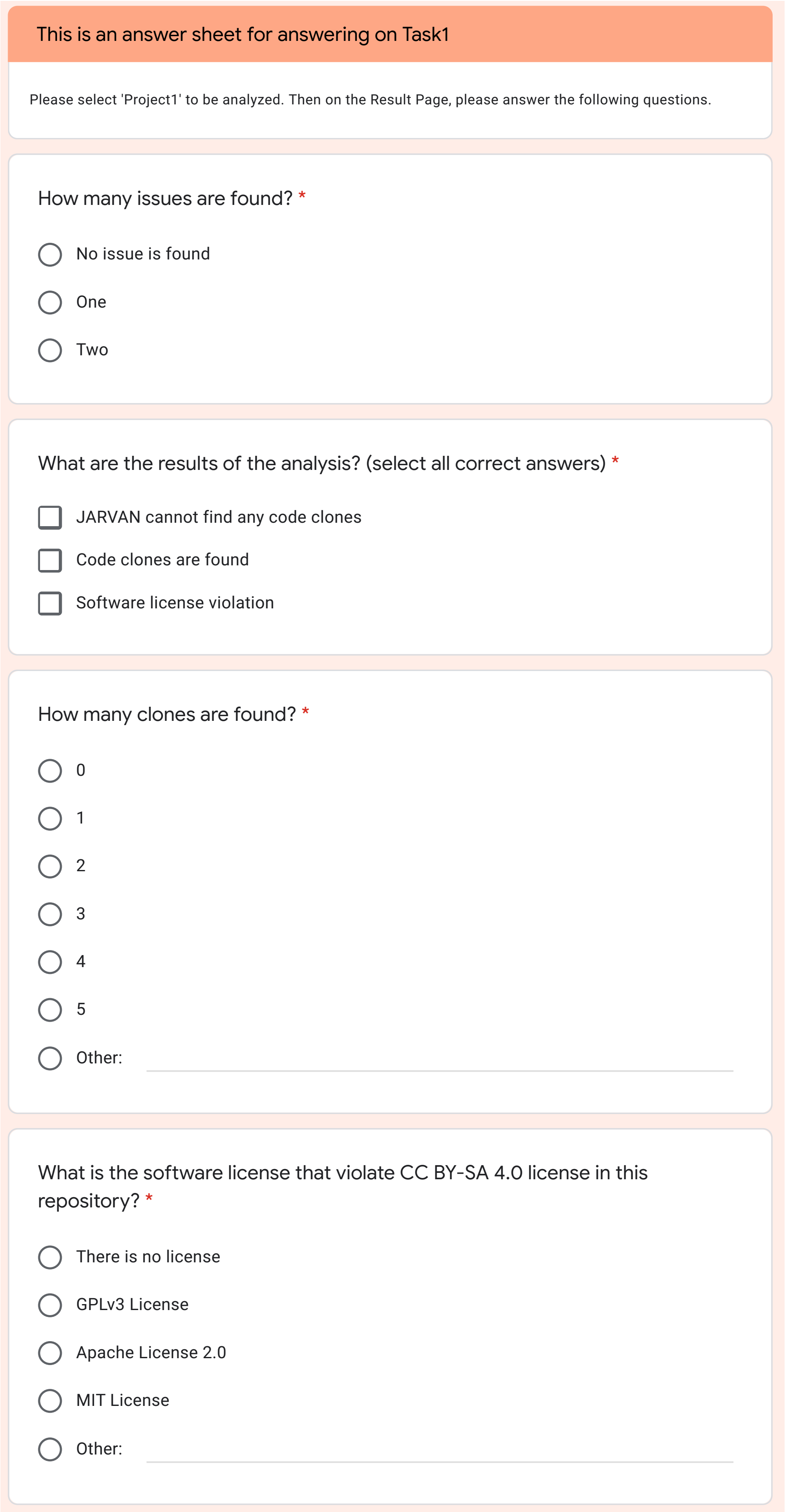
Which repository is/are private repository(s)? *

□ Project1

□ Project2

□ Project3

□ None of them



This is an answer sheet for answering on Task1

Please select 'Project1' to be analyzed. Then on the Result Page, please answer the following questions.

How many issues are found? *

○ No issue is found

○ One

○ Two

What are the results of the analysis? (select all correct answers) *

□ JARVAN cannot find any code clones

□ Code clones are found

□ Software license violation

How many clones are found? *

○ 0

○ 1

○ 2

○ 3

○ 4

○ 5

○ Other: ____________________

What is the software license that violate CC BY-SA 4.0 license in this repository? *

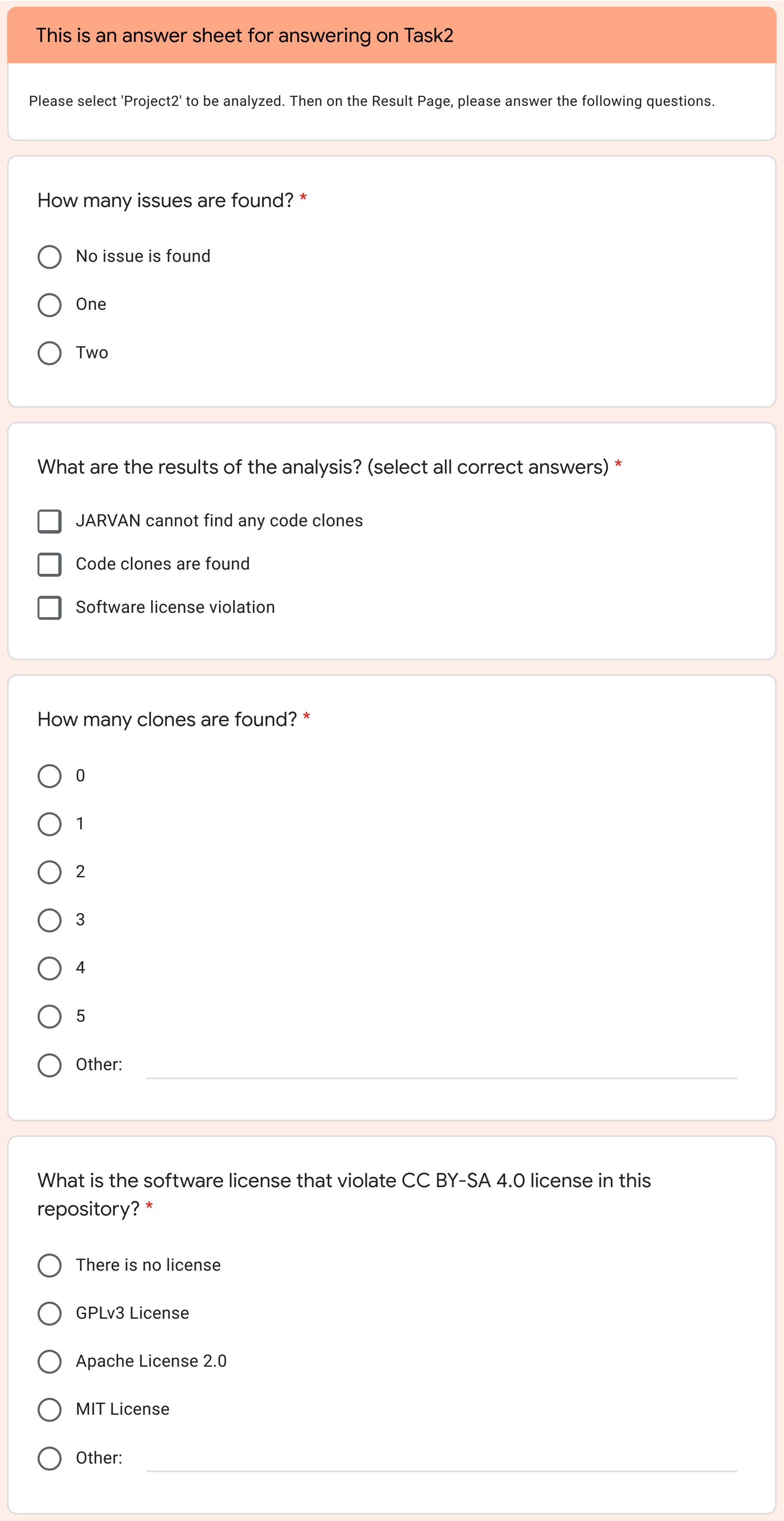
○ There is no license

○ GPLv3 License

○ Apache License 2.0

○ MIT License

○ Other: ____________________



This is an answer sheet for answering on Task2

Please select 'Project2' to be analyzed. Then on the Result Page, please answer the following questions.

How many issues are found? *

○ No issue is found

○ One

○ Two

What are the results of the analysis? (select all correct answers) *

□ JARVAN cannot find any code clones

□ Code clones are found

□ Software license violation

How many clones are found? *

○ 0

○ 1

○ 2

○ 3

○ 4

○ 5

○ Other: ____________

What is the software license that violate CC BY-SA 4.0 license in this repository? *

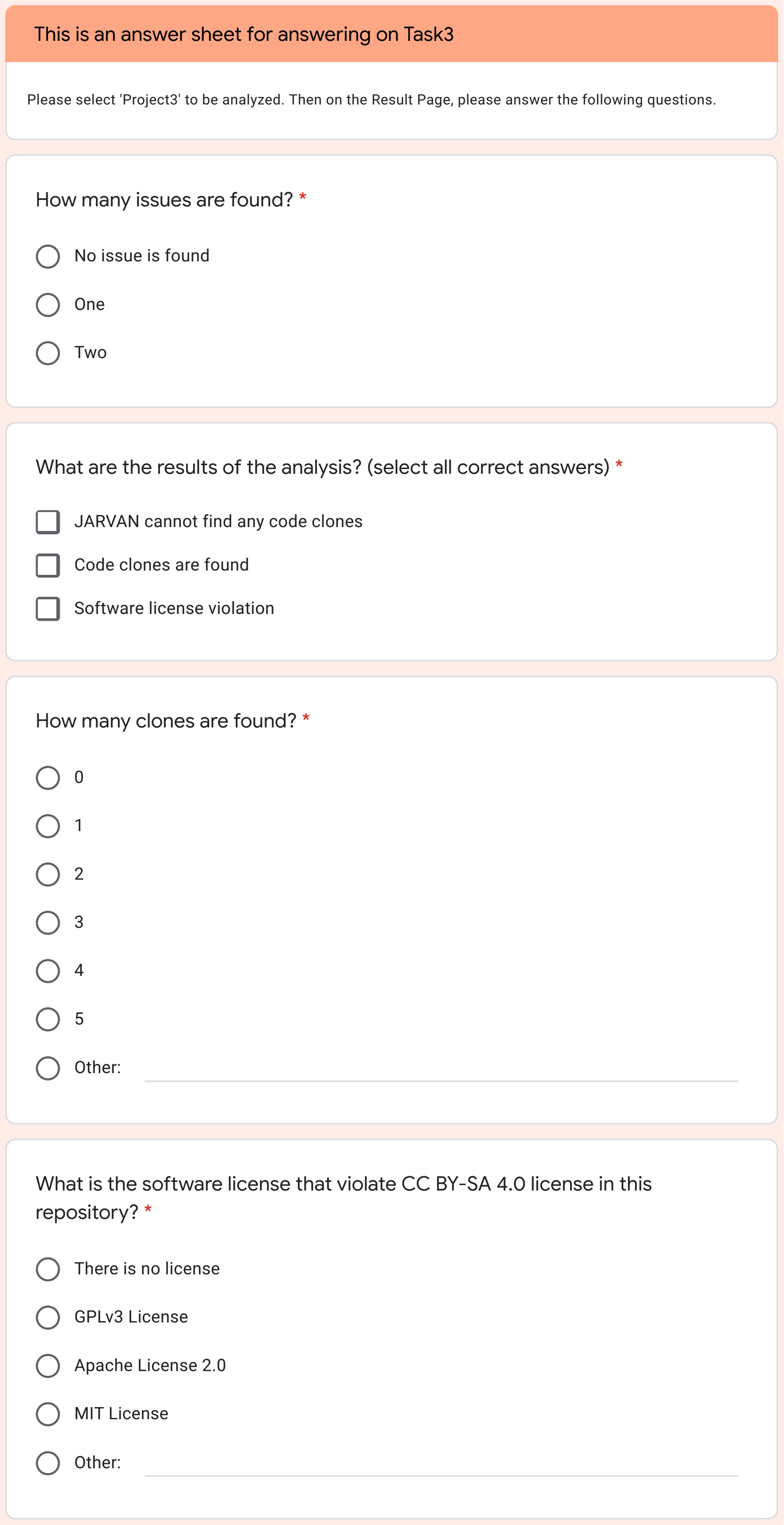
○ There is no license

○ GPLv3 License

○ Apache License 2.0

○ MIT License

○ Other: ____________



This is an answer sheet for answering on Task3

Please select 'Project3' to be analyzed. Then on the Result Page, please answer the following questions.

How many issues are found? *

○ No issue is found

○ One

○ Two

What are the results of the analysis? (select all correct answers) *

□ JARVAN cannot find any code clones

□ Code clones are found

□ Software license violation

How many clones are found? *

○ 0

○ 1

○ 2

○ 3

○ 4

○ 5

○ Other: ____________________

What is the software license that violate CC BY-SA 4.0 license in this repository? *

○ There is no license

○ GPLv3 License

○ Apache License 2.0

○ MIT License

○ Other: ____________________

REFERENCES

[1] Roy CK., Cordy JR., Koschke R., “Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach”, Sci Comput Program. May 2009;74(7):470–495, [Online]. Available: https://doi.org/10.1016/j.scico.2009.02.007.

[2] “Stack Overflow - Where Developers Learn, Share, & Build Careers”; [cited 17 November 2020], [Online]. Available: https://stackoverflow.com.

[3] “Stack Overflow company page - learn about stack overflow”; [cited 17 November 2020], [Online]. Available: https://stackoverflow.com/company.

[4] Ragkhitwetsagul C., Krinke J., Paixao M., Bianco G., Oliveto R., “Toxic Code Snippets on Stack Overflow”, IEEE Transactions on Software Engineering. 2019;p. 1–1.

[5] Yang D., Martins P., Saini V., Lopes C., “Stack Overflow in Github: Any Snippets There?”, In: Proceedings of the 14th International Conference on Mining Software Repositories. MSR ’17. IEEE Press; 2017. p. 280–290, [Online]. Available: https://doi.org/10.1109/MSR.2017.13.

[6] Zhang H., Wang S., Chen THP., Zou Y., Hassan AE., “An Empirical Study of Obsolete Answers on Stack Overflow”, IEEE Transactions on Software Engineering. 2019;p. 1–1, [Online]. Available: http://dx.doi.org/10.1109/TSE.2019.2906315.

[7] Baltes S., Dumani L., Treude C., Diehl S., “SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts”, In: Proceedings of the 15th International Conference on Mining Software Repositories. MSR ’18. New York, NY, USA: Association for Computing Machinery; 2018. p. 319–330, [Online]. Available: https://doi.org/10.1145/3196398.3196430.

[8] German DM., Di Penta M., Gueheneuc Y., Antoniol G., “Code siblings: Technical and legal implications of copying code between applications”, In: 2009 6th IEEE International Working Conference on Mining Software Repositories; 2009. p. 81–90.

[9] Ragkhitwetsagul C., Krinke J., “Siamese: Scalable and Incremental Code Clone Search via Multiple Code Representations”, Empirical Softw Engg. Aug. 2019;24(4):2236–2284, [Online]. Available: https://doi.org/10.1007/s10664-019-09697-7.

[10] “Building OAuth Apps”; November 2020 [cited 19 November 2020], [Online]. Available: https:// docs.github.com/ en/ free-pro-team@latest/ developers/ apps/building-oauth-apps.

[11] Wang H., “Inverted Index”; July 2020 [cited 1 July 2020], [Online]. Available: https:// pdfs.semanticscholar.org/ 01bd/ 5d205eec5172f92207a82ecf8ecacc0eeafc.pdf.

[12] “Stack Exchange Data Dump”; [cited 2 December 2020], [Online]. Available: https://archive.org/details/stackexchange.

[13] Sajnani H., Saini V., Svajlenko J., Roy CK., Lopes CV., “SourcererCC: Scaling Code Clone Detection to Big-Code”, In: Proceedings of the 38th International Conference on Software Engineering - ICSE ’16; 2016. p. 1157–1168.

BIOGRAPHIES

NAME	Mr. Phattharapong Poolthong
DATE OF BIRTH	27 November 1996
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Taweethapisek School, 2015: High School Diploma Mahidol University, 2019: Bachelor of Science (ICT)

NAME	Miss Panaya Sirilertworakul
DATE OF BIRTH	2 December 1998
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Mattayom Watnairong English Program School, 2017: High School Diploma Mahidol University, 2019: Bachelor of Science (ICT)

NAME	Miss Kanika Wonwien
DATE OF BIRTH	28 August 1998
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Sarasas Witead Bangbon School, 2017: High School Diploma Mahidol University, 2019: Bachelor of Science (ICT)