

**Detection of Energy Consumption and Performance of
Third-party Applications using Machine Learning and Artificial
Intelligence**

22_23-J 87

Project Proposal Report

Thiranya M.A.R.

IT19129440

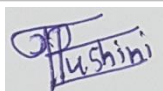
B.S c. (Hons) Degree in Information Technology
Department of Information Technology

Sri Lanka Institute of Information Technology

June 2022

Declaration

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Thiranya M.A.R	IT19129440	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Date:13/10/2022

Supervisor: Ms. Sanjeevi Chandrasiri ([See Appendix for Acknowledgment](#))

Co-Supervisor: Ms. Madhuka Nadeeshani ([See Appendix for Acknowledgment](#))

Abstract

Huge volumes of data are moved from one party to another in the modern world. Technology is consistently evolving at a quick pace, from the end user to the developer. The "bad" side of technology also continues to advance, which could be a crucial element in terms of security and effectiveness. As a result, before labeling a piece of technology as bad, a user is encouraged to use it, and mistakes are inevitable after the first few uses. As a result, an anomaly detection and prevention service could prohibit a third-party application or piece of software from damaging the user's PC even before it tries to install itself. The third-party application's security measures, potential harm to the device in terms of energy consumption, and whether the specifications are met will all be revealed by this technology, which will also offer information about it. This Non-Specific Technology Risk Calculator (NSTRC) is a tool that enables a user to assess the security performance and energy consumption of a third-party program before installation. It is based on artificial intelligence and uses particular parameters and modules, such as machine learning. The mechanism will also be able to offer the user a thorough examination of the application prior to installation and give the user the option to comprehend the current technological procedure in the detection of energy consumption and performance of the third-party application from the viewpoints of both the user and the developer. The program will compare the energy usage and performance of the user's device in this section prior to the installation of any third-party applications.

Keywords: Detection energy consumption, Detection performance level, Machine learning

Table of contents

Declaration	i
Abstract	ii
Table of contents	iii
List of figures	iv
List of tables	iv
List of abbreviations	iv
1. Introduction	1
1.1 Background & Literature survey	2
1.2 Research Gap	3
1.3 Research Problem	4
2. Objectives	5
2.1 Main Objectives	5
2.2 Specific Objectives	5
3. Methodology	5
3.1 System Diagram	7
3.2 Connectivity	7
3.3 Technologies to be used	8
3.4 Project Requirements	8
3.5 Work Breakdown Structure	10
3.6 Gantt chart	10
4. Commercialization	11
5. Reference list	12
6. Appendices	12

List of figures

Figure 3. 1: Compare with similar kind of third party applications	7
Figure 3.1. 1: Compare with user device and third-party applications	7
Figure 3.1. 2: High-level Architecture Diagram	8
Figure 3.5. 1: Work Breakdown Structure	10
Figure 3.6. 1: Gantt chart	10
Figure 5. 1: NSTRC Logo	12

List of tables

Table 1.2. 1: Research Comparison	4
-----------------------------------	---

List of abbreviations

NSTRC	Non-Specific Technology Risk Calculator
ML	Machine

1. Introduction

The majority of users install third-party software or mobile apps before testing the performance or how their energy consumption would change after doing so on their mobile devices or desktop computers. Most of the time, users skim the description provided by the vendor of the program or mobile application. But what if there was a method to assess the impact of third-party applications on our devices both before and during use? The Non-Specific Technology Risk Calculator (NSTRC) helps users make the necessary comparisons between third-party applications and user devices as well as between one third-party application and another of the same kind, which is really helpful for users to determine which application will best suit their needs.

1. Prior to installation, compare the user device's performance and energy usage with the third-party application.

- Before installing a third-party application, the NSTRC utility can analyze the user device's performance and energy consumption level. It then generates a report outlining the potential performance and energy consumption problems. The installation procedure may be continued if the user is okay with the identified harm, and it may be stopped if the user is not.

2. While utilizing the application, compare the user device's performance and energy usage with the third-party application.

- Users of the third-party application that is integrated with this NSTRC tool can receive warnings from the NSTRC tool about the current performance and energy consumption levels.

3. To determine which third-party program best fits user needs while causing the least damage to user devices, compare two or more applications that match the same user requirements.

- By using this NSTRC tool, users can compare their chosen application to two others that were created to address similar user needs. They can then determine which applications are compatible with their devices' operating systems.

1.1 Background & Literature survey

1.2

Today's world is adjusting to time and technology. We must always be technologically more advanced than others if we want to compete in this world that is changing so quickly. The pinnacle of modern technology is the rapid development of new software. Therefore, before installing any third-party software that will satisfy their needs, an user or software developer first evaluates the app's performance on their computer or mobile device. Our research project is consequently immediately connected to both the hardware device of the developer and the user whenever a user installs any third-party software.

We suggest a "Non-Specific Technology Risk Calculator," a framework, desktop application, and mobile application to accomplish the goals and offer a remedy to the current situation. We provide software that, when installed on a machine, would satisfy the needs of third-party software (desktop PC). Additionally, suggest using a mobile application to compare the client's mobile requirements with those of third-party applications in order to learn how those parameters impact performance, RAM, and CPU capacity. The device specs of the customers' and third-party software applications' mobile and desktop applications will be detected and compared by our proposed application. And our suggested application will permit the installation to proceed if the third-party program's specifications match the mobile/desktop standard. Otherwise, if they don't, our advised mobile application will notify the customer of the damage, at which point they may decide whether to continue with the installation or cancel it. If the software overheats or becomes too RAM-loaded, the customer should receive a notification via our suggested mobile application.

Advantages of the proposed system for users:

1. Before installing a type of software, users can assess how well it works on their own computers to develop an early opinion about the installation.
2. The system can identify the risks and drawbacks that the software may bring about after being installed on the user's computer, minimizing the chance of being impacted.

1.2 Research Gap

Some third-party applications are exclusively suitable for gaming [1], and there are some studies and tools available to test UI compatibility [3]. Some third-party programs fail to present warnings and instructions before installing themselves on users' devices, posing a safety risk and possibly reducing the speed with which users' devices operate. There are some language-dependent performance tools, but it will be difficult to test them with different languages.

Research Comparison

Research/ Product	UI based compatibility testing	Compare with multiple applications	Works with different developing languages	Compare with measurement variables (CPU, RAM, ROM)	Display warning messages	Check for security purpose
Research [1]	✗	✗	✗	✗	✗	✗
Research [2]	✗	✗	✗	✗	✗	✗
Mimic UI compatibility testing app	✓	✗	✗	✗	✗	✗
NSTRC	✓	✓	✓	✓	✓	✓

Table 1.2. 1: Research Comparison

1.3 Research Problem

In today's world, where things are changing so quickly, every second counts. Before attempting to install a program, a user may wonder about its effectiveness and safety. Performance is one of the most important factors that a software developer takes into consideration when creating or creating software. If we could see how the third-party program, we are about to install will balance the performance and capacity of our device/machine, it would be more advantageous for a user, a developer, or anybody else who utilizes the software. There are times when a user may regret installing a problematic third-party program that not only makes their smartphone run worse but also jeopardizes their device's security and privacy.

The following is a list of issues with performance and energy use that can arise after installing third-party programs on any device:

- Any PC that has third-party software installed does not perform well and will not support anything until the particular software is uninstalled.
- After the third-party program is installed, the PC's performance and task completion times increase, which affects the efficiency of other software.
- The user may not be aware of potential security and privacy threats from malware and virus attacks originating from the third-party application.
- The lack of performance of device machines caused by insufficient disk space or RAM capacity causes stress for software engineers and other team members, increasing time consumption for everyone on the team.

We suggest a "Non-Specific Technology Risk Calculator," a framework, desktop application, and mobile application to accomplish the goals and offer a remedy to the current situation. We provide software that, when installed on a machine, would satisfy the needs of third-party software (desktop PC). Additionally, suggest using a mobile application to compare the client's mobile requirements with those of third-party applications in order to learn how those parameters impact performance, RAM, and CPU capacity. The device specs of the users' and third-party software applications' mobile and desktop applications will be detected and compared by our proposed application. And our suggested application will permit the installation to proceed if the third-party program's specifications match the mobile/desktop standard. Otherwise, if they don't, our advised mobile application will notify the customer of the damage, at which point they may decide whether to continue with the installation or cancel it. If the software overheats or becomes too RAM-loaded, the customer should receive a notification via our suggested mobile application.

2. Objectives

2.1 Main Objectives

Use the comparison mechanism to identify and contrast the energy usage and performance of the user's device and third-party applications. The task is to gather specifications-related data and to present a report with a comparison analysis. Implementation will be carried out to monitor energy usage and performance and present a warning message to the user, requesting authorization to continue with installation despite any observed downsides. The program will also offer fresh measurements for the user's device's stable performance.

2.2 Specific Objectives

- Comparison of the performance and energy consumption between third-party app and user device.
- Generation of the report related to the comparison of the above.
- Detect energy consumption and display warning and show instructions to get under control of that situation.

2. Methodology

- The proposed NSTRC tool has the ability to do the comparison of the performance and energy consumption between third-party app and user device, generate reports related to the comparison of the third-party app and user device, detect energy consumption and display warning and show instructions to get under control of that situation and compare same kind of third party applications in order select the best suit third party application for the user which fulfill the user requirements effectively.

This is how NSTRC tool is going to carry out its functions:

- Take the user device measurement variables (CPU, RAM, ROM) which is stored in the database collected using a trained model.
- Compare the user device's measurement variables with the third-party application to find out the performance level and energy consumption to decide whether to install the third-party application or not.

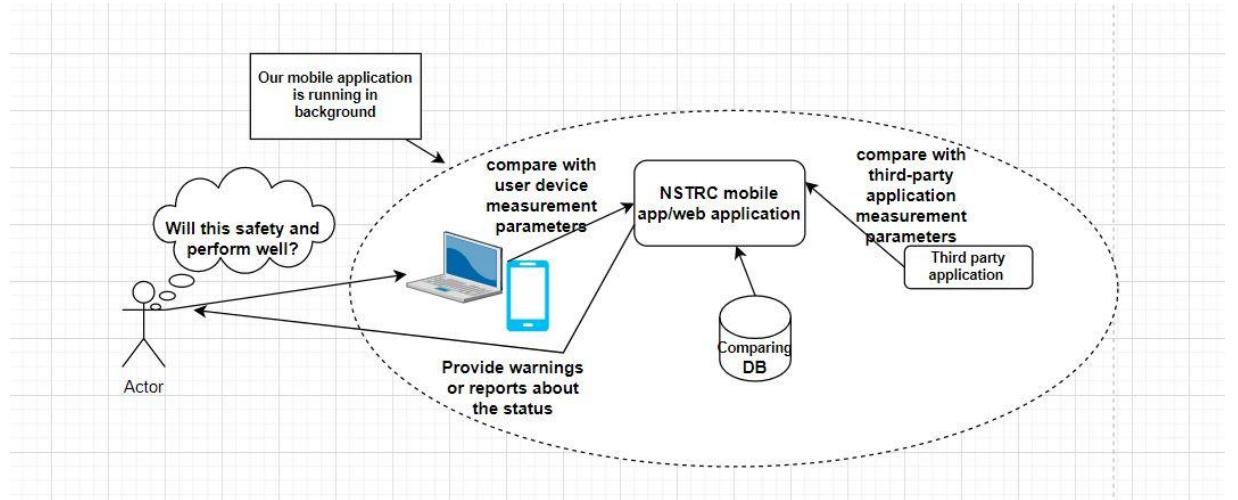


Figure 3. 1: Compare with user device and third-party applications

- Provide report containing warnings and instruction about the potential damage could happen to the user device and safety of the third-party application after a proper comparison done by using our NSTRC tool.
- Provide warnings messages about the energy consumption level and the safety level while the user uses the third-party application with the help of using our NSTRC tool.

- Compare among third party applications which is developed to fulfil similar kind of user requirements with the user device measurement variables stored in a separate database to find out which suits best for the user with less harm.

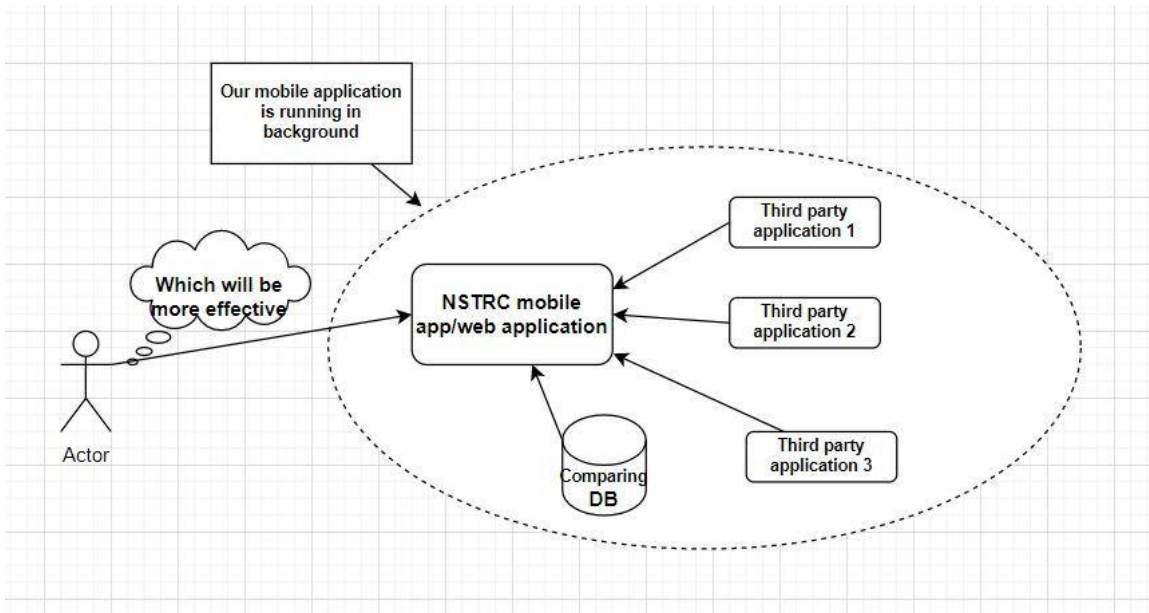


Figure 3. 1.1: Compare with similar kind of third-party applications

3.1 System Diagram

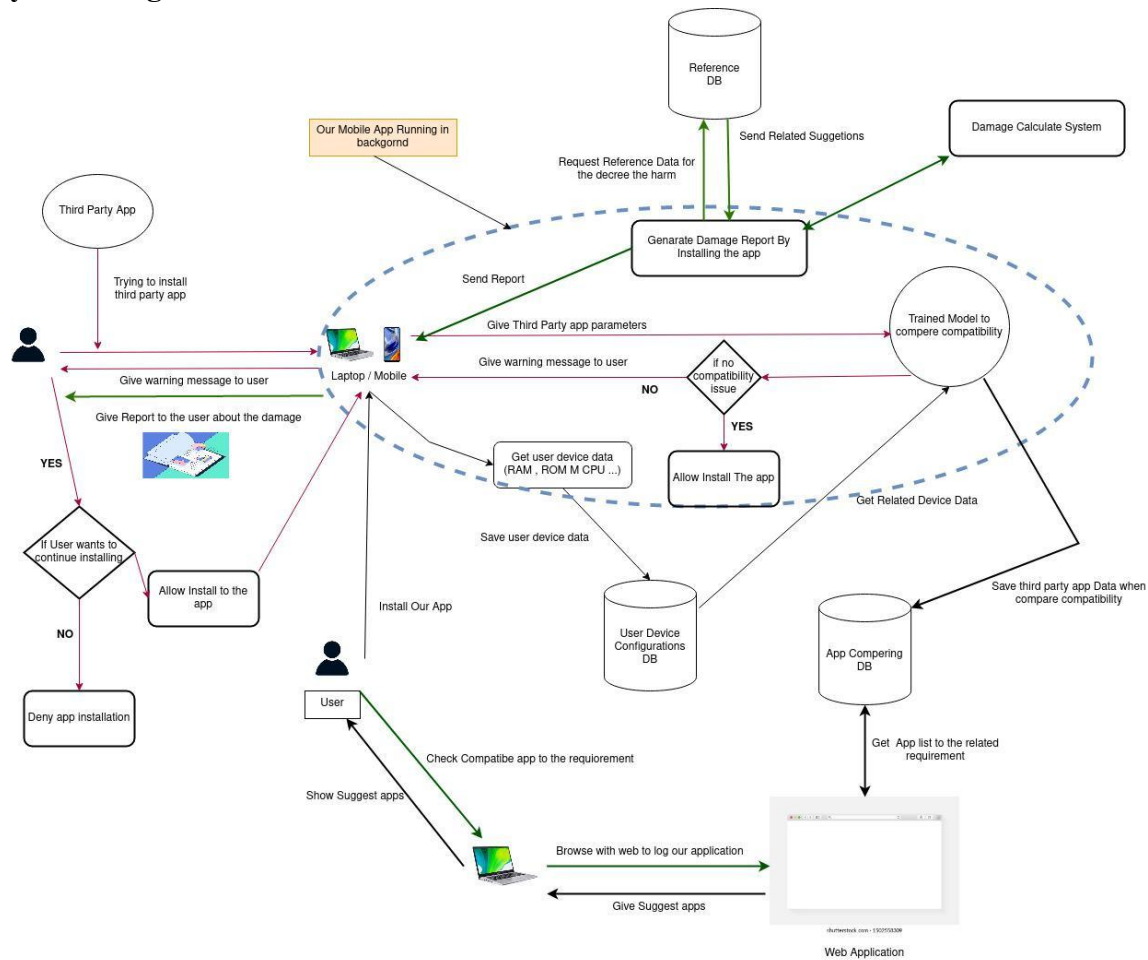


Figure 3.1.2: High-level Architecture Diagram

3.2 Connectivity

Choose the application from a third party the user wants to install. The NSTRC program will then examine the energy usage, performance level, and any other system requirements of the third-party application. After that, the NSTRC tool compares the measurement variables stored in the database that were gathered using the trained model with the third-party application and gives the user a report to show the warnings and the potential effects to the user's device if user install the third-party application. The installation procedure can be continued or cancelled depending on whether the user wishes to install despite the warnings

and unsafe surroundings. Additionally, while using any third-party programs, users can occasionally receive alerts and warnings concerning their energy consumption. The NSTRC tool can also be used if a user wishes to evaluate other third-party programs that were created to satisfy similar user requirements. Finally, users can examine the third-party application's safety and are informed of it as well.

3.3 Technologies to be used

The implementation of this proposed system is currently scheduled to use current trending technologies. The implementation of our proposed system will be divided into smaller sections, each of which will be developed separately. Once the smaller sections have been completed, the system will proceed to the testing environment process.

For the detection of specification for performance and compatibility check, Artificial Intelligence will be used as a mechanism to perform the detection process with calculated algorithms fit for its purpose. Python and Pycharm will be used to create algorithms using Machine and Deep Learning, hence to train a pre-model for user device configurations.

GitLab will be used as a version control and collaboration, and our desktop/mobile application will consist of the backend (Python and Pycharm) and the frontend interface (ReactJS). The database will be Mongo DB which is an ideal storage medium for the user device and third-party application configurations, and Visual Studio Code will be used as it is the perfect IDE for web application development and cloud hosting development.

3.4 Project Requirements

- Functional Requirements
 - Identified the user device performance level
 - Identify the third-party application performance selected by the user with user requirements
 - Comparison part using collected performance levels to identify the harm to the user devices.

- Comparison part two or more third party applications' performance levels which is going to fulfill similar kind of user requirements to identify the harm to the user devices and select which is best.
 - Provide useful clear report about the harm that could affects if user install the third-party application.
 - Provide warning messages as alerts while using the third-party application with the measured energy consumption calculated by the NSTRC tool.
- User Requirements
 - Users need to identify which third party application will fulfill their user requirements with less harm to their user devices.
 - Users need to identify which third party application will fulfill their user requirements with less amount of task completion time with high performance level before install any third-party application.
 - Users need to compare similar kind of third-party applications to find out which fulfill their user requirements with great experience and efficacy.
 - Users need to identify the potential security and privacy threats could cause after installing the third-party application before installing it to their user devices.
 - Users need clear guidance and details about comparison with third party applications and user devices performance levels and energy consumption levels in order to take right decisions.
 - Non-functional requirements
 - Compatible with any smart device
 - Processing speed should be high
 - Data should be Reliable
 - Available
 - Accurate
 - Interfaces should be in an attractive view for users

3.5 Work Breakdown Structure

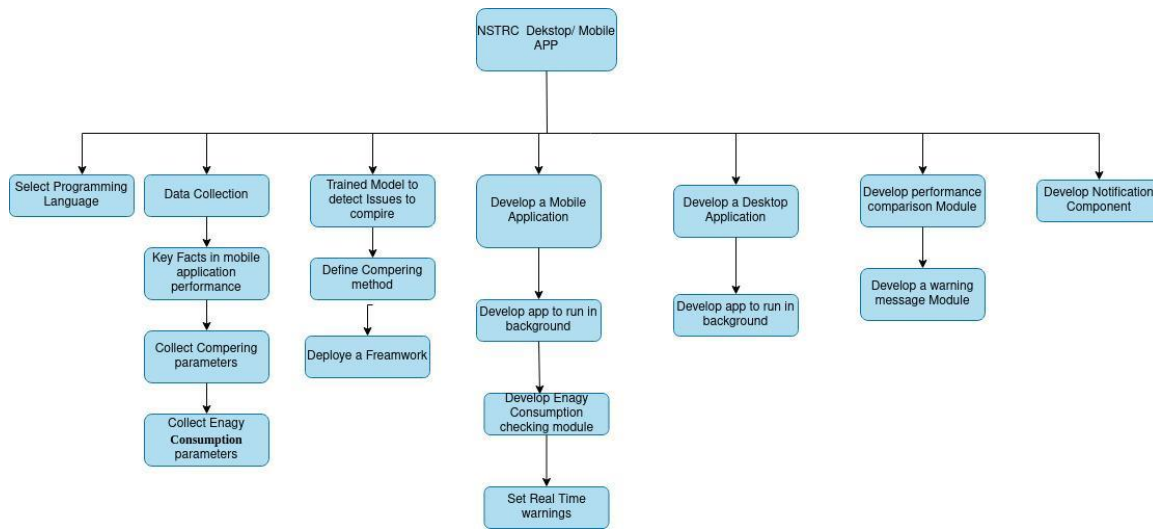


Figure 3.5. 1: Work Breakdown Structure

3.6 Gantt chart



Figure 3.6. 1: Gantt chart

3. Commercialization

The NSTRC application is going to use different types of users: Mobile application users, desktop application users, and web application developers. The application is to prevent harm to the user device or the web applications deployed on servers. The application will be released version-wise in the first step of the framework part and will be released for the most popular programming language according to proven statistical information.

In the current times, many user devices are struggling because of installed and invalid compatible third-party apps. Therefore, the NSTRC application will be able to help many users to protect their devices. Currently, there are many intern and trainee developers initiating their career into the technology field. As a result, they will need to be supported to write code in the proper way for proper functioning. Our tool will be able to provide them with the necessary convenience and application in order to carry out the functionalities. When the first release is available, we will make the entire application available for free open source. As time moves forward, new releases with versions will be implemented with better features and trusted resources.

As a desktop mobile application, we need to get closer to the user in order to reach our target. GIGABIT is a short and memorable name, and we designed a logo that is also simple and recognizable. The version is also embedded into the logo to make users aware of the version they are using in, and update it whenever a new release has been made.




Figure 7.1 NSTRC Logo

4. Reference list

- [1] Systemrequirementslab.com. 2022. *Requirements Test*. [online] Available at: <<https://www.systemrequirementslab.com/cyri>> [Accessed 14 October 2022].
- [2] Zhenyu Liu, Yun Hu and Lizhi Cai, 2014. Research on software security and compatibility test for mobile application. *Fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014)*,.
- [3] Ki, T., Park, C., Dantu, K., Ko, S. and Ziarek, L., 2019. Mimic: UI Compatibility Testing System for Android Apps. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*,.
- [4] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F. and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, [online] 287, p.116601. doi:10.1016/j.apenergy.2021.116601.
- [5] Eid, S., Makady, S. and Ismail, M. (2020). Detecting software performance problems using source code analysis techniques. *Egyptian Informatics Journal*, 21(4), pp.219–229. doi:10.1016/j.eij.2020.02.002.
- [6] Hrci Marketing. (n.d.). Top 5 Common Performance Problems. [online] Available at: <https://www.hrci.org/community/blogs-and-announcements/hr-leads-business-blog/hr-leads-business/2022/02/14/top-5-common-performance-problems> [Accessed 15 Jul. 2022].

5. Appendices

Appendix 1 - Acknowledgement from Supervisor, Ms. Sanjeevi Chandrasiri

**Sanji Chandrasiri**
to me, Madhuka, Thiranya, Jayasingha, Rajapaksha ▾

4:15 PM (0 minutes ago) ☆ ↶ ⋮

Dear Team,

I acknowledge the project proposal documents.

BR

Sanjeevi Chandrasiri
M.Sc.(IT)(SLIIT), B.Sc.(Hons)(IT)(SLIIT), MIEEE, MBCS, MIET
Senior Lecturer - Department of Information Technology
Year 1 Course Leader
Faculty of Computing
SLIIT

Appendix 2 - Acknowledgement from Supervisor, Ms. Madhuka Nadeeshani

**Madhuka Nadeeshani** <madhuka.n@slit.lk>

To: Rahman S.H it19189086; Rajapaksha R.M it19156316; Thiranya M. A. R. it19129440 +1 other
Cc: Sanji Chandrasiri

Fri 10/14/2022 4:34 PM

[EXTERNAL EMAIL] This email has been received from an external source – please review before actioning, clicking on links, or opening attachments.

Dear Team,

I endorse the project proposal documents.

Best Regards,
Madhuka

⋮