

Connection status report generator

Pratyush Gupta¹, Somnath Banerjee², Debani Prasad Mishra³, Surender Reddy Salkuti⁴

^{1,2,3}Department of Electrical Engineering, IIIT Bhubaneswar, Bhubaneswar, India

⁴Department of Railroad and Electrical Engineering, Woosong University, Daejeon, Republic of Korea

Article Info

Article history:

Received Sep 20, 2020

Revised Mar 21, 2021

Accepted Mar 30, 2021

Keywords:

Connection

Interface

Java

Network

Report generator

Tracker

ABSTRACT

This paper presents “Connection Status Report Generator” which is an auto executable application and it generates a detailed textual and pictorial representation of the network connectivity status of a particular computer and sends the generated reports to the concerned party. The features developed in this paper aim to constantly monitor the network connectivity status as well as ease the troubleshooting process of finding the major cause of call-drops which is a popular problem in every industry. This paper is divided into three major sub-categories of real-time connection status tracker, report generator, and the image viewer interface. The proposed executable application is coded in Java and designed to run as a background application with minimal system prerequisites.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Surender Reddy Salkuti

Department of Railroad and Electrical Engineering

Woosong University

17-2, Jayang-Dong, Dong-Gu, Daejeon-34606, Republic of Korea

Email: surender@wsu.ac.kr

1. INTRODUCTION

The world is facing an unforeseen situation that demands us to understand and practice “work-from-home” more than ever before. The call-center employees are adversely affected by this method of work. Now and then several call-drop-related concerns/complaints are registered by employees and clients to their respective organizations which can indeed lead to a substandard customer experience. It is essential to find out the main cause of these call-drops, i.e., whether the problem lies with the individual’s network connectivity at the time of the call-drop or the problem lies with the server with which the employee was connected at the time of call-drop [1]. This paper helps in the troubleshooting process and finds out whether the problem lies with the employee’s network connection. The “Connection-Status-Report-Generator” is an auto-executable application that records connection status in real-time as well as saves the reports for future reference in both text and pictorial format. The paper is widely divided into three major parts namely real-time connection status tracker, report generator, and the image viewer interface [2].

This application is entirely coded in Java programming language and concepts of networking have also been applied [3]. The business logic and the design are very important for the end-users, choosing Java as a programming platform is a good option to enhance the source code and independence from system operating [4]. The use of various packages like java.util, java.io, java.time, java.net, javafx, java.nio, have been used in coding the proposed application [5], [6]. The image viewer interface was designed using javafx [7]. The interface for sending mail was designed using java swing [8]. Eclipse integrated development environment for java was used to code the application [9], [10].

Real-time connection status tracker uses the concepts of multi-threading which helps to monitor the connection status over a longer period of time which can be further modified to be developed as a dashboard which can be exported for instant troubleshooting [11]-[13]. Facilities like mailing the generated textual and pictorial report which uses Java mail API is developed [14], [15]. It also requires the simple mail transfer protocol (SMTP) server details of the client's computer [16]-[18]. This protects the authenticity of the generated report without any chances of tampering from the clients' side. It runs on Windows operating system with the most basic version of Java which enables to run the application on the majority of the systems [19].

One of the challenges of the application was to have minimum interference in the clients' activities and run in the background constantly generating reports and graphically tracking the connection status. This was achieved by the windows startup feature which automatically triggers the application on switching on the system. So the application can generate reports from switch on till switch off without any need for actions from the client's side. The paper is organized as follows: Section 2 describes the proposed research method. Section 3 presents the results and discussion. Lastly, the paper is summarized in Section 4.

2. PROPOSED RESEARCH METHOD

The entire application, i.e., connection status report generator has been sub-divided into three major subdivisions namely real-time connection status tracker, report generator, and image viewer interface [20], [21]. The flowchart of the basic design/algorithm of the real-time connection status tracker is depicted in Figure 1(a). The flowcharts of the basic design/algorithm of the report generator and image viewer interface sections are depicted in Figures 1(b) and 1(c), respectively. The real-time connection status tracker plots line graphs which depict the connected and disconnected status of clients' computers in real-time [22], [23]. The report generator generates read-only .txt containing the details of connection status and .png files containing a snapshot of plotted pie-charts to give a visual representation of the created text files. The image viewer interface displays all the generated .png files (snapshots of pie-charts) making them easily accessible for the client/user [24].

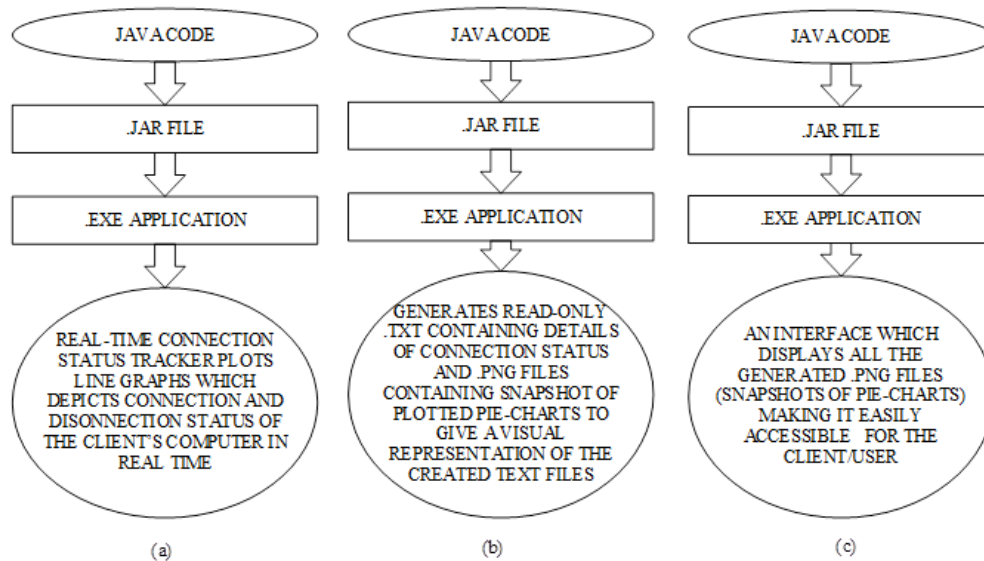


Figure 1. Flowcharts of (a) real-time connection status tracker, (b) report generator, (c) image viewer interface

3. RESULTS AND DISCUSSION

In this section, a brief explanation of all the features developed in the proposed application is described.

3.1. Real-time connection status tracker

It is an executable application that detects the connected and disconnected status every 5 seconds (which can be modified as per requirement) [25]. The application when minimized and allowed to run in

background can provide connection status data for a long interval of time by plotting line graphs. It can record only 2 states, i.e., connected and disconnected [26], and comes with a horizontal scroll bar to access data over a longer period of time as shown in Figure 2.

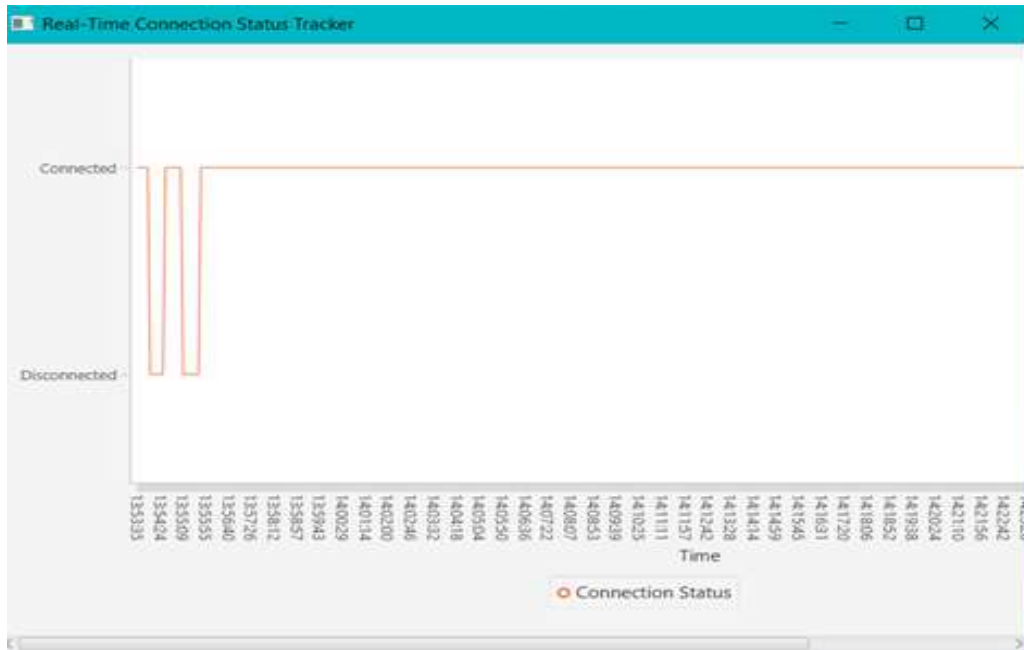


Figure 2. Real-time connection status tracker window

3.2. Records connection status in .txt format

3.2.1. Generation of read-only .txt files

The generated reports are read-only and have the date-time stamp as their name making them easily accessible from the project directory. Moreover, each file having a unique name ensures that the files are not overwritten and a new file is created whenever the .exe application is run [27]. This feature is depicted in Figure 3. The created text files occupy a few hundred KBs depending on the duration of the operation, hence many such files can be created without any storage issues.



Figure 3. Generation of read-only .txt files

3.2.2. IP address is recorded

The IP address of the client/employee is recorded in the first line of the report. If the first reports are sent by e-mail (or any other medium) to the concerned party [28] then it will be easy to distinguish the reports based on the IP-address and it is shown in Figure 4. It can be modified to display the personalized name of the respective system along with the IP Address.



Figure 4. The IP address is recorded

3.2.3. Report updated on a regular interval

The generated report as shown in Figure 5 records the data for every 5 seconds which can be modified as per the requirement [29]. The interval can be increased for reports of longer duration to make it more readable and reduce the memory occupied by each file.

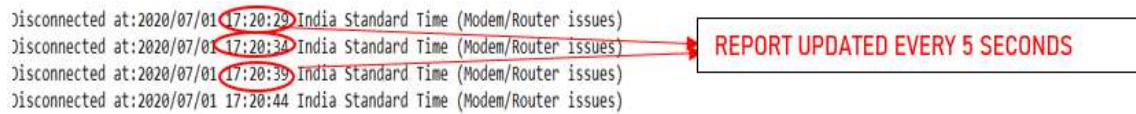


Figure 5. Report updated every 5 seconds

3.2.4. Works across different time zones

The report specifies the time zone of the client/employee’s computer which assures that the program can be executed all around the world, and a sample of this has been depicted in Figure 6. This depicts that the application can simultaneously monitor reports from different continents without any bewilderment.

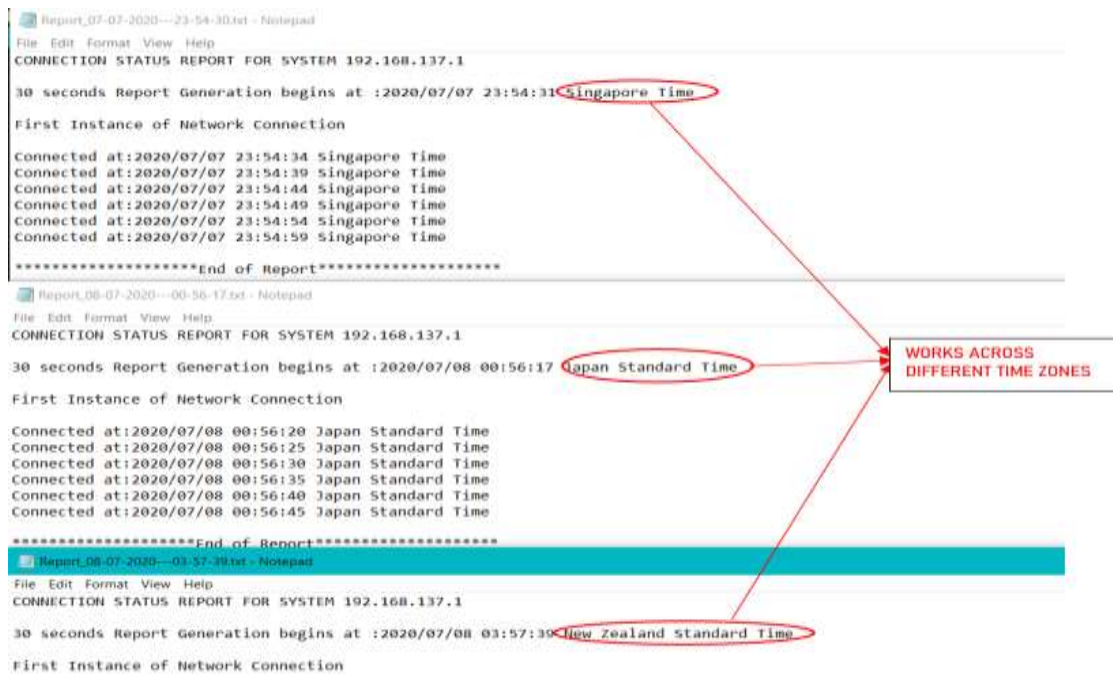


Figure 6. Works across different time zones

3.2.5. Detects sleep mode/system freeze

The report specifies the time and duration for which the system was in freeze mode (hang) or sleep mode [30]. Figure 7 shows one such instance.


```

Disconnected at:2020/07/01 17:19:51 India Standard Time (Modem/Router issues)
System In Sleep Mode/System Freeze at: 2020/07/01 17:19:51 India Standard Time for: 37 seconds
Disconnected at:2020/07/01 17:20:29 India Standard Time (Modem/Router issues)
    
```

Figure 7. Detects sleep mode/system freeze

3.2.6. Files can be sent via e-mail after the completion

The files can be sent via email (optional) on the completion of file creation depending on the availability of an internet connection. If the internet connection is unavailable at the time of file creation then the file keeps on updating its content until the internet connection is restored [31]. In Figure 8, the 1-minute u was supposed to get completed by 11:51:51 but didn't complete due to the unavailability of an internet connection. The moment the internet connection was restored the file got completed and could be sent via e-mail to the concerned party or developer [32]. An interface is developed using java swing at the launch of the application which takes consent from the respective user before sending the reports and provides an option to only generate files for personal analysis without sending them.

```

1 minute Report Generation begins at :2020/06/27 11:50:51 India Standard Time
First Instance of Network Connection
Connected at:2020/06/27 11:50:54 India Standard Time
Connected at:2020/06/27 11:51:03 India Standard Time

Connection Lost
Disconnected at:2020/06/27 11:51:05 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:10 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:15 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:20 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:25 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:30 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:35 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:40 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:45 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:50 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:51:55 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:00 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:05 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:10 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:15 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:20 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:25 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:30 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:35 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:41 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:46 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:51 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:52:56 India Standard Time (Modem/Router issues)
Disconnected at:2020/06/27 11:53:01 India Standard Time (Modem/Router issues)
Reconnected at :2020/06/27 11:53:09 India Standard Time
*****End of Report*****
    
```

File was supposed to be completed by 11:51:51

Since there was no internet connection at the stipulated time the report didn't complete

The report gets completed the moment internet connectivity is restored and the mail is sent instantly

Figure 8. Detects sleep mode/system freeze

3.2.7. Report distinguishes between modem/router and server issues

The application pings 'Google.com' on a regular interval and based on the response received concludes the connection status of the computer. If "Google.com" is facing a downtime then the application pings 10 other sites which have the least probability of facing an outage [33], if a response is received from any of those 10 sites but not from "Google" then disconnected due to Server Issues is recorded in the report otherwise Disconnected due to Modem/Router issues is recorded in the report.

3.3. Generation of .png and .txt files simultaneously

Png and .txt files are simultaneously created in the base directory as shown in Figure 9. The .png files created contain a snapshot of the pie chart which depicts the ratio of percentages of time when the computer was connected, disconnected to the internet due to modem/router issues or server down as shown in Figure 10. The pie chart also displays the time interval for which the pie chart has been plotted.

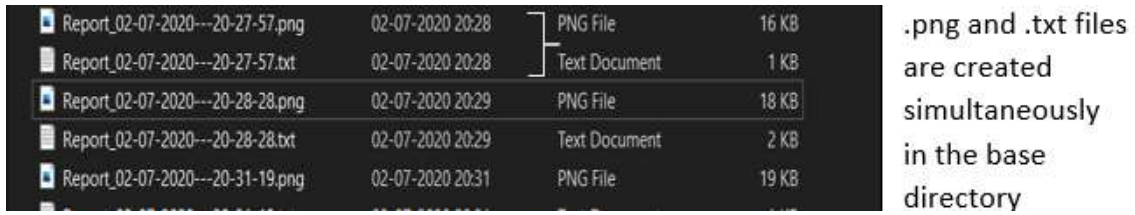


Figure 9. Generation of .png and .txt files simultaneously

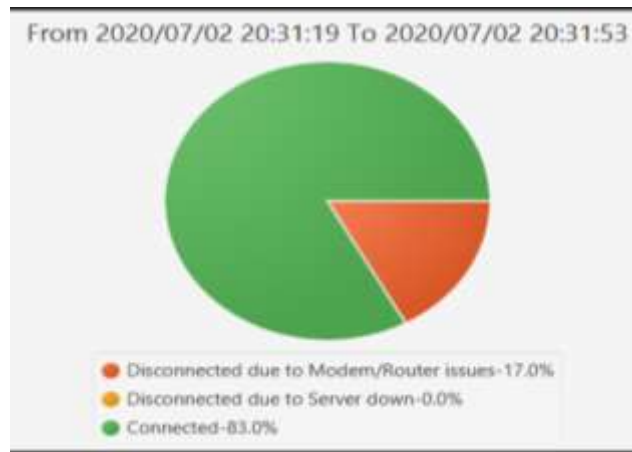


Figure 10. Files (.png) created contain snapshot of pie charts that are plotted

3.4. Generation of .png and .txt files simultaneously

The application offers a different choice of timings [34]. One such instance is shown in Figure 11 where five choices of timings are shown. It can be modified as per the requirement of the user.

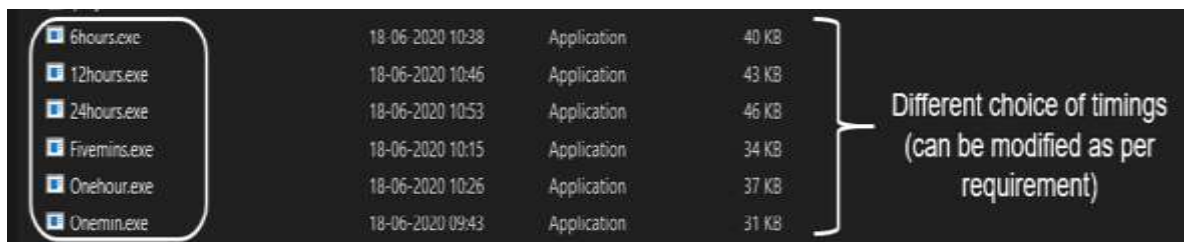


Figure 11. The different choice of timings

3.5. Deploying application as windows service

Windows service features enable an application to automatically get the compiled and executed on switching on the computer without manually executing it. The connection status report generator will continue to run in the background and will be executed automatically on startup and does not support any interface the application can be seen in the services window in Figure 12. The windows service feature does not support applications with the interface [35], so this feature cannot be used simultaneously with the mail interface.

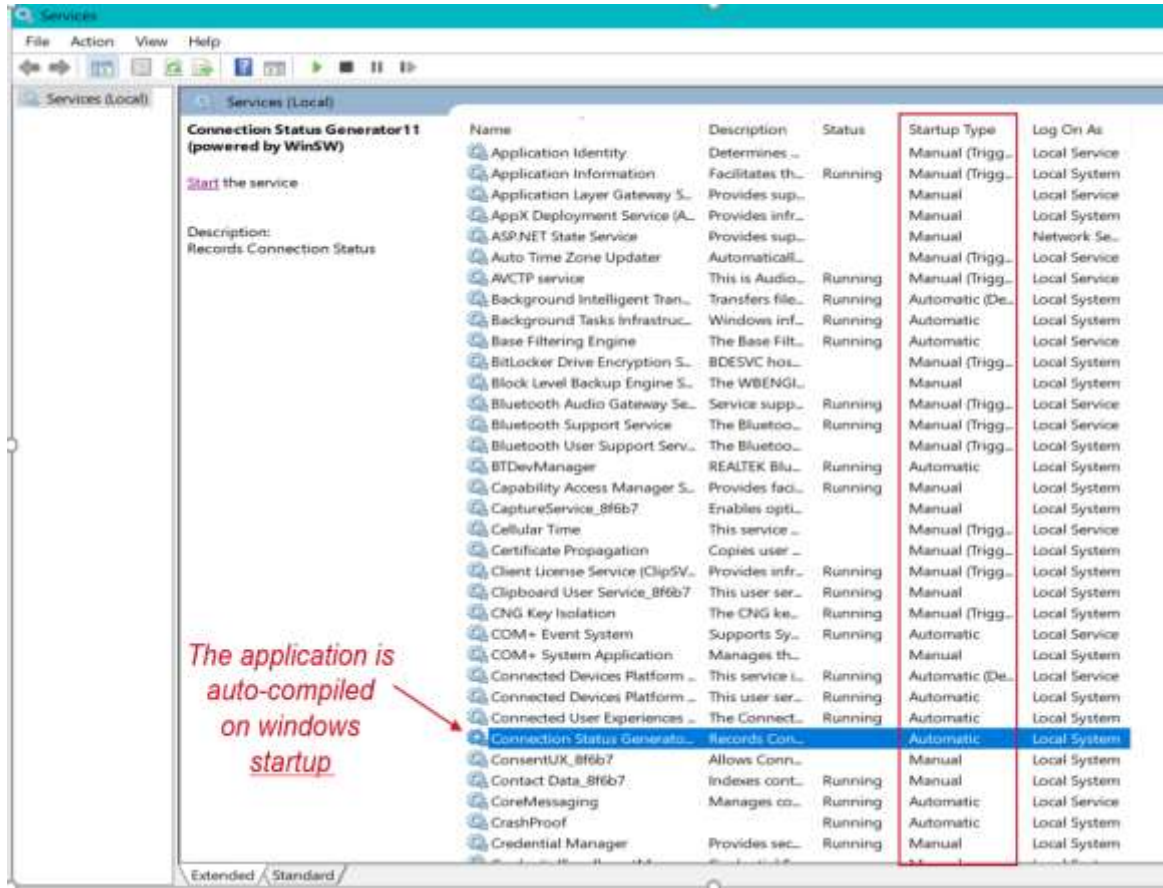


Figure 12. Sevcies window

3.6. Image viewer interface

To view all the .png files stored in the application directory an interface is created. It has two buttons 'browse' and 'load'. The load button loads all the images stored in the application directory and displays them in list view [36], whereas the browse button can manually select any folder in our local computer and display all the related .png files as shown in Figure 13.

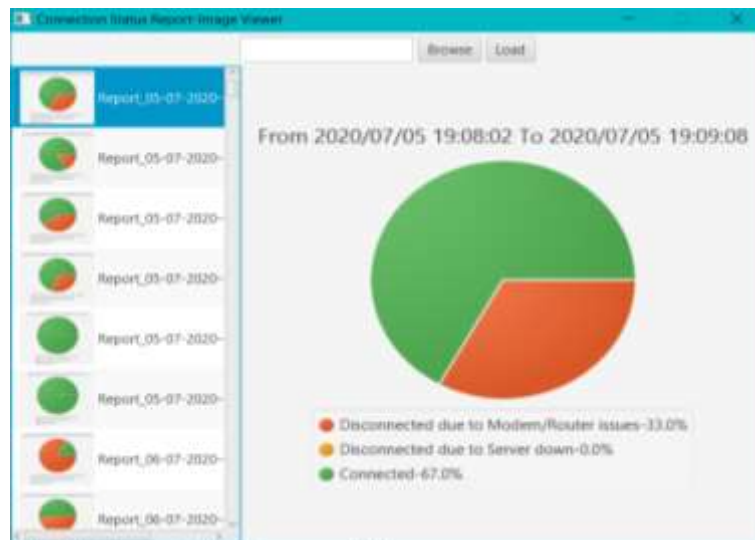


Figure 13. Image viewer interface

3.7. System prerequisites

- a) Windows operating system: This application is only supported on Windows operating system.
- b) Pre-installed Java: This application requires pre-installed java. Any minimum version of Java, i.e., JDK1.0.1 or above is required for the application to run on the client/employee's computer.

4. CONCLUSION

The unavailability of basic devices and equipment in households amidst the growing trend of “work-from-home” is a bothering scenario. Majority of the time, the cause of disruptions faced while working remains to be a mystery, i.e., whether the client's computer got momentarily disconnected to the network or the server to which it was communicating faced an outage. The ‘connection-status-report-generator’ can ease the troubleshooting process for not only the respective teams but also helps in individual assessment of the particular problem by the ‘real-time-connection-status-tracker’ feature. This application can prove to be a valuable asset for not only call-center employees but who face innumerable instances of abrupt call drops leading to substandard customer experience. This application can be upgraded in the future to a global dashboard where the real-time connection status tracker will provide the updates of all the concerned systems to a centralized system for monitoring and troubleshooting purpose. The detailed reports can also be referred to by the centralized system to ease the troubleshooting process.

ACKNOWLEDGEMENTS

This research work has been carried out based on the support of “Woosong University's Academic Research Funding - 2021”.

REFERENCES

- [1] M. Gallo, W.M. Hancock, “Fundamental Concepts of Computer Networks and Networking”, *Networking Explained (Second Edition)*, pp. 1-29, Dec. 2002, doi: 10.1016/B978-155558252-4/50027-4.
- [2] T. Singh, “A Survey on Java Programming Language and Methods of Improvisation”, *International Journal of Innovations & Advancement in Computer Science*, vol. 6, no. 12, pp. 158-162, Dec. 2017.
- [3] G. Singh and T. Singh, “Review M-Learning: Learning Environment Using Android Mobile Application”, *International Journal of Advanced Technology in Engineering and Science*, vol. 4, no. 1, pp. 225-227, May 2016.
- [4] T. Singh, “Review: A Study on Needs of Essentials Factors for E-Learning”, *International Journal of Advance Research in Science and Engineering*, vol. 6, no. 1, pp. 31-35, Jan. 2017.
- [5] T. Singh, “New Software Development Methodology for Student of Java Programming Language”, *International Journal of Computer and Communication Engineering*, vol. 2, no. 2, pp. 194-196, Mar. 2013, doi: 10.13140/2.1.3063.7765.
- [6] T. Singh, “New Learning Methodology for Student of Java Programming Language”, *International Journal of Engineering Research and Development*, vol. 3, no. 11, pp. 17-19, 2012, doi: 10.13140/2.1.3063.7765.
- [7] D. O. Simion, “Using Java in Business Applications”, *ECC'10: Proceedings of the 4th conference on European computing conference*, pp. 218–223, Apr. 2010.
- [8] R. Standtke and U. U. Nitsche, “Java nio framework Introducing a High-performance I/O Framework for Java”, *In Proceedings of the Third International Conference on Software and Data Technologies - PL/DPS/KE*, pp. 206-211, 2008, doi: 10.5220/0001901502060211.
- [9] D. Lea, “The java.util.concurrent synchronizer framework”, *Science of Computer Programming*, vol. 58, no. 3, pp. 293-39, Dec. 2005, doi.org/10.1016/j.scico.2005.03.007.
- [10] Cheshta and D. Sharma, “Developing applications using JavaFX”, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 6672-6676, Apr. 2017, doi: 10.15680/IJIRCCE.2017.0504014.
- [11] J. Verma, I.S. Behl, and A. Saini, “Research paper on swings”, *International Journal of Scientific Research and Education*, vol. 1, no. 7, pp. 173-186, 2013.
- [12] L. Bettini and P. Crescenzi, “Java-meets eclipse: An IDE for teaching Java following the object-later approach,” *10th International Joint Conference on Software Technologies*, Colmar, pp. 1-12, 2015.
- [13] G. C. Murphy, M. Kersten, L. Findlater, “How are Java software developers using the Elipse IDE?,” *IEEE Software*, vol. 23, no. 4, pp. 76-83, July-Aug. 2006, doi: 10.1109/MS.2006.105.
- [14] S. Moon and B. M. Chang, “A thread monitoring system for multithreaded Java programs”, *ACM Sigplan Notices*, vol. 41, no. 5, pp. 21-29, May 2006, doi: 10.1145/1149982.1149985.
- [15] N. Goel, V. Laxmi, and A. Saxena, “Handling Multithreading Approach Using Java”, *International Journal of Computer Science Trends and Technology*, vol. 3, no. 2, pp. 24-31, Mar-Apr 2015.
- [16] K. Panwar and D. Sharma, “Multithreading In Java,” *International Journal of Research*, vol. 1, no. 10, Nov. 2014.
- [17] G. Pandey and D. Dani, “Android Mobile Application Build on Eclipse”, *International Journal of Scientific and Research Publications*, vol. 4, no. 2, pp. 1595-1598, Feb. 2014.

- [18] I. Fadi, O. Olumide and I. Dumitrache, "On multi robot simulation: multi-threading approach for implementation", *GESJ: Computer Science and Telecommunications*, vol. 1, no. 30, pp. 38-50, 2011.
- [19] V. Sundaresan, D. Maier, P. Ramarao, M. Stoodley, "Experiences with multi-threading and dynamic class loading in a Java just-in-time compiler", *International Symposium on Code Generation and Optimization*, pp. 1-11, Apr. 2006, doi: 10.1109/CGO.2006.16.
- [20] A.M. Hassan, "JAVA and DART programming languages: Conceptual comparison," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 17, no. 2, pp. 845-849, Feb. 2020, doi: 10.11591/ijeecs.v17.i2.pp845-849.
- [21] M. N. F. Jamaluddin, A. Ismail, A. A. Rashid, T. T. O. Takleh, "Performance comparison of Java based parallel programming models," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 16, no. 3, pp. 1577-1583, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1577-1583.
- [22] F. Lv, X. Xie, C. Z. X. Cheng, "Research and Development of E-mail Program Based on Java", *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, Sept. 2009, doi: 10.1109/ICASID.2009.5276955.
- [23] S. Ramadass, H. A. Bazar, O. A. Abouabdalla, and A. Manasrah, "Active e-mail system SMTP protocol monitoring algorithm", *2nd IEEE International Conference Broadband Network & Multimedia Technology*, Nov. 2009, doi: 10.1109/ICBNMT.2009.5348490.
- [24] Y. J. Kim, H. K. Kwak, K. S. Chung, "A Distributed Architecture Based SMTP Server for Large Email Service", *The KIPS Transactions Part C*, vol. 16c, no. 5, pp. 597-604, Oct. 2009, doi: 10.3745/KIPSTC.2009.16C.5.597.
- [25] J. Wang, Y. Hu," *Ninth International Symposium on Computers and Communications*, vol. 1, Jan. 2004, doi: 10.1109/ISCC.2004.1358381.
- [26] P. Rajeswaria and P. Anbalagan, "Design and deployment of android based mobile application for performance analysis of micro strip patch antenna", *Microprocessors and Microsystems*, vol. 77, Sept. 2020, doi: 10.1109/ISCC.2004.1358381.
- [27] D. E. Yawson and F. A. Yamoah, "Gender variability in E-learning utility essentials: Evidence from a multi-generational higher education cohort," *Computers in Human Behavior*, vol. 114, Jan. 2021, doi: 10.1016/j.chb.2020.106558.
- [28] L. Ardito, R. Coppola, G. Malnati, M. Torchiano, "Effectiveness of Kotlin vs. Java in android app development tasks", *Information and Software Technology*, vol. 127, Nov. 2020, doi: 10.1016/j.infsof.2020.106374.
- [29] N. L. Mingoc and E. L. R. Sala, "Design and Development of Learn Your Way Out: A Gamified Content for Basic Java Computer Programming", *Procedia Computer Science*, vol. 161, pp. 1011-1018, 2019, doi: 10.1016/j.procs.2019.11.211.
- [30] R. Z. Cabada, M. L. B. Estrada, F. G. Hernández, R. O. Bustillos, C. A. R. García, "An affective and Web 3.0-based learning environment for a programming language", *Telematics and Informatics*, vol. 35, no. 3, pp. 611-628, Jun, doi: 10.1016/j.tele.2017.03.005. 2018.
- [31] Obinna Okoyeigbo, Edevbie Agboje, Evioghene Omuabor, Uyi Aiyudubie Samson, and Abidemi Orimogunje, "Design and implementation of ajava based virtual laboratory for data communication simulation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no.6, pp. 5883-5890, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5883-5890.
- [32] G. L. Taboada, S. Ramos, R. R. Expósito, J. Touriño, R. Doallo, "Java in the High Performance Computing arena: Research, practice and experience," *Science of Computer Programming*, vol. 78, no. 5, pp. 425-444, May 2013, doi: 10.1016/j.scico.2011.06.002.
- [33] G. L. Taboada, J. Touriño, R. Doallo, "Java Fast Sockets: Enabling high-speed Java communications on high performance clusters," *Computer Communications*, vol. 31, no. 17, pp. 4049-4059, Nov. 2008, doi: 10.1016/j.comcom.2008.08.012.
- [34] D. Lea, "The java.util.concurrent synchronizer framework", *Science of Computer Programming*, vol. 58, no. 3, pp. 293-309, Dec. 2005, doi: 10.1016/j.scico.2005.03.007.
- [35] T. Yiyu, "A hardware-oriented object model for Java in an embedded processor", *Microprocessors and Microsystems*, vol. 63, pp. 85-97, Nov. 2018, doi: 10.1016/j.micpro.2018.08.007.
- [36] C. Chakraborty, S. S. Pathak, and S. Chakrabarti, "An O(n) Telephony Gateway Selection Methodology for IP-PSTN Packet Routing", *IEEE INDICON*, pp. 517-520, 2009, doi: 10.1109/INDCON.2009.5409344.