**LOG FILE ANALYZER**

Project report submitted in partial fulfillment of the requirement for the Internship

By

**Manish (768940)**

**Vikas (768994)**

Under the supervision of

**Dr. Pankaj Anadure**

To



**Cognizant Technology Solutions India Pvt Ltd , Manyata Tech Park, Nagawara, Bengaluru, Karnataka 560045**

**CERTIFICATE**

**Candidate’s Declaration**

We hereby declare that the work presented in this report entitled **“**Log File Analyzer**”** in partial fulfillment of the requirements for the award of the internshipis an authentic record of my own work carried out over a period from Feb 2019 to April 2019 under the supervision of Mr. Pankaj Anadure**.**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Manish (768940) Vikas (768994)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Pankaj Anadure

Dated :

**ACKNOWLEDGEMENT**

Learning through the project under the guidance of our esteemed mentor Mr. Pankaj Anadure, whose expertise knowledge in the domain of Frontend and Backend Development, not only cleared all our ambiguities but also generated a high level of interest and gusto in the subject. We are truly grateful for his guidance and support throughout the project. We would also like to thank our SME for allocation of the project as well as its resources.

The prospect of working in a group with high level of accountability fostered a spirit of teamwork and created a feeling of oneness which thus, expanded our ken, motivated us to perform to the level best of our ability and create a report of the highest quality.

To do the best quality work, with utmost sincerity and precision has been our constant endeavor.

Date: Manish (768940)

Vikas (768994)

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Title** | **Page No** |
| 1 | Introduction | 1 |
| 1.1 | Problem Statement | 1 |
| 1.2 | Objective | 1 |
| 1.3 | Methodology | 2 - 3 |
| 2. | Literature Survey | 4 |
| 2.1 | Chapter Summary | 16 |
| 3. | System Development | 16 |
| 3.1 | Tools | 16 |
| 3.1.1 | Anaconda | 16 |
| 3.1.2 | Jupyter Notebook | 16 |
| 3.1.3 | Python | 16 |
| 3.2 | Libraries | 17 |
| 3.2.1 | Numpy | 17 |
| 3.2.2 | Pandas | 18 |
| 3.2.3 | Matplotlib | 18 |
| 3.2.4 | Scikit-Learn | 19 |
| 4. | Data Analysis & Algorithm | 20 |
| 4.1 | Data Assembly | 20 |
| 4.2 | Data Preprocessing | 23 |
| 4.2.1 | Data Cleaning & Featurization | 23 |
| 4.2.2 | Missing Values | 24 |
| 4.2.3 | Irrelevant Data | 24 |
| 4.2.4 | Discretization | 24 |
| 4.2.5 | Balancing Age & Gender Group | 25 |
| 4.3 | Algorithm(Methods) | 26 |
| 4.3.1 | Support Vector Machines | 26 |
| 4.3.2 | Logistic Regression | 26 |
| 4.3.3 | Artificial Neural Network | 26 |
| 4.3.4 | Random Forest | 27 |
| 4.3.5 | Naïve Bayes | 27 |
| 4.3.6 | Decision Tree | 27 |
| 4.4 | Evaluation Criteria | 28 |
| 4.4.1 | Precision | 28 |
| 4.4.2 | Recall | 28 |
| 4.4.3 | Precision-Recall Curve | 29 |
| 5. | Test Plan |  |
| 6. | Result & Performance Analysis | 30 |
| 6.1 | Analysis of Classifier | 30 |
| 6.2 | Cost Analysis | 30 |
| 6.3 | Algorithm Analysis | 31 - 32 |
| 7. | Conclusion | 33 - 34 |
| 7.1 | Summary | 34 |

**LIST OF ABBREVATIONS**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **ABBREVIATION** | **FULLFORM** |
| 1. | AHRC | Agency for Healthcare Research and Quality |
| 2. | ADA | American Diabetes Association |
| 3. | DSME | Diabetes Self Management Education |
| 4. | RAD | Rapid Application Development |
| 5. | ICD9 | International Statistical Code of Diseases and Related Health Problems |
| 6. | SVM | Support Vector Machines |
| 7. | ANN | Artificial Neural Network |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Page** |
| 1. | Features List and their descriptions with % of missing values | 17 |
| 2. | % of Americans with diabetes by age factor | 19 |
| 3. | Number of different tests to be performed | 22 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG. NO.** | **DESCRIPTION** | **PAGE** |
| 1. | Overview of The Methodology | 2 |
| 2. | Recategorizing Readmission Feature | 23 |
| 3. | Balancing Age Group | 25 |
| 4. | Cleaning Gender Group | 25 |
| 5. | Implementation of Random Forest | 31 |
| 6. | Implementation of Logistic Regression | 32 |

**ABSTRACT**

Log analysis is the process of turning your log files into data and then making intelligent decisions based on that data. In software testing, the analysis of logs is designed to check and monitor the application work. Also logs help to fix some errors. Logs are the text lines which contains systematic information about service work and actions such as: IP-address, dates, time, and viewed sites, potential buyers domains, status code, and the size of the answer, the general info about query, loading time, user-agent, etc. of particular operations that occur in different environments - application itself or the system that runs an app. Logs include the messages of several types: informational, warning and error.

Logs help to specify who uses an application and how often it is used. Moreover, testers can define whether the session was successful, and to detect possible mismatches or mistakes a users of the app can face. The log analysis helps to design new scenarios for further web or mobile testing.

**Chapter – 1**

**INTRODUCTION**

In computer log management, log analysis (or system and network log analysis) is an art and science seeking to make sense out of computer-generated records (also called log or audit trail records). The process of creating such records is called data logging.

Logs are emitted by network devices, operating systems, applications and all manner of intelligent or programmable device. A stream of messages in time-sequence often comprise a log. Logs may be directed to files and stored on disk, or directed as a network stream to a log collector.

Log messages must usually be interpreted with respect to the internal state of its source (e.g., application) and announce security-relevant or operations-relevant events (e.g., a user login, or a systems error).

Logs are often created by software developers to aid in the debugging of the operation of an application or understanding how users are interacting with a system, such as search engine. The syntax and semantics of data within log messages are usually application or vendor-specific. Terminology may also vary; for example, the authentication of a user to an application may be described as a login, a logon, a user connection or authentication event. Hence, log analysis must interpret messages within the context of an application, vendor, system or configuration in order to make useful comparisons to messages from different log sources.

Log message format or content may not always be fully documented. A task of the log analyst is to induce the system to emit the full range of messages in order to understand the complete domain from which the messages must be interpreted.

* 1. **PROBLEM STATEMENT**

This is the project has its motivation from the log files which are cumbersome to read. Log Files contains huge amount of useful information about which must be taken care. This project is created for debugging of the operation of the application or understanding how users are interacting with a system.

* 1. **OBJECTIVE**

The essential goal of the venture is:

1. To extract the raw data and convert that data into useful or readable form.
2. To analyze the data and perform certain actions like searching, sorting, etc.
3. To visualize different parameters of the data to get some results.
   1. **METHODOLOGY**

**Figure 1: Overview of the methodology**

A diagram of our strategy is given in Figure 1. We first preprocess the raw data into some useful information. Utilizing this preprocessed information, we perform certain actions on that information like searching particular rows across the data, perform sorting among different parameters, also try to visualize the effect of different parameters.

**Chapter - 2**

**LITERATURE SURVEY**

**2.1 Chapter Summary**

Log file analysis has many applications, such as site security. In terms of search engine optimization, the process usually involves downloading the file from your server and importing it into a log file analysis tool, where all the information about every “hit” on the site (whether bot or human) can be analyzed to inform SEO decisions and learn about previously unknown issues.

Log file analysis is an arduous process that frequently results in the discovery of critical technical problems that could be found no other way. Log files contain incredibly accurate data that allow a brand to better understand how search engines are crawling their site and the kind of information they are finding.

Log file data includes a record of the URL/resource that was requested, action taken, time and date, IP of the machine it originated from, user agent/browser type, and other pieces of information.

**Chapter – 3**

**SYSTEM DEVELOPMENT**

**3.1 TOOLS & FRAMEWORKS**

**3.1.1 Visual Studio Code**

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences.

Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine.

Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse.

It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

**3.1.2 ANGULAR**

Angular (commonly referred to as "Angular 2+" or "Angular v2 and above") is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS.

Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.

**Why Angular ?**

* Angular presents you not only the tools but also design patterns to build your project in a maintainable way. When an Angular application is crafted properly, you don’t end up with a tangle of classes and methods that are hard to modify and even harder to test. The code is structured conveniently and you won’t need to spend much time in order to understand what is going on.
* It’s JavaScript, but better. Angular is built with TypeScript, which in turn relies on JS ES6. You don’t need to learn a totally new language, but you still receive features like static typing, interfaces, classes, namespaces, decorators etc.
* No need to reinvent the bicycle. With Angular, you already have lots of tools to start crafting the application right away. You have directives to give HTML elements dynamic behavior. You can power up the forms using Form Control and introduce various validation rules. You may easily send asynchronous HTTP requests of various types. You can set up routing with little hassle. And there are many more goodies that Angular can offer us!
* Components are decoupled. Angular strived to remove tight coupling between various components of the application. Injection happens in NodeJS-style and you may replace various components with ease.
* All DOM manipulation happens where it should happen. With Angular, you don’t tightly couple presentation and the application’s logic making your markup much cleaner and simpler.
* Testing is at the heart. Angular is meant to be thoroughly tested and it supports both unit and end-to-end testing with tools like Jasmine and Protractor.
* Angular is mobile and desktop-ready, meaning you have one framework for multiple platforms.
* Angular is actively maintained and has a large community and ecosystem. You can find lots of materials on this framework as well as many useful third-party tools.

**3.1.3 TYPESCRIPT**

TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language.

TypeScript is designed for development of large applications and trans compiles to JavaScript. As TypeScript is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs.

TypeScript lets you write JavaScript the way you really want to. TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. TypeScript is pure object oriented with classes, interfaces and statically typed like C# or Java. The popular JavaScript framework Angular 2.0 is written in TypeScript. Mastering TypeScript can help programmers to write object-oriented programs and have them compiled to JavaScript, both on client-side and server-side (Node.js) execution.

**3.1.4 PYTHON**

Python is very simple programming language. It is very easy to learn Python. The concept of objects and classes can be used in the Python very easily, so that’s why it is called object-oriented programming language. Python source code is converted to byte code when we compile the source code. After that this byte code is executed by the interpretor. So this language is also known as the Interpreted language. Python has facility of in built high level data structures which makes this language very easy to learn.

**3.1.5 NODE JS**

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser.

Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

**3.1.5 FLASK**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

**3.2 LIBRARIES**

**3.2.1 NUMPY**

Numpy is a Library which provide the facility of easy manipulation of multi-dimensional array in Python. Without this library the facility of use of multidimensional array is little bit difficult. By importing this package, multidimensional array can be manipulated easily. Also, other high dimensional operation of mathematics can be used with the help of this package.

**3.2.2 PANDAS**

Data analysis can be done very easily with the help of Python. Python provides multiple packages and libraries which provides user to different type of facilities. Pandas is one of them. This library is used for data analysis. With the help of this library user can optimized performance.

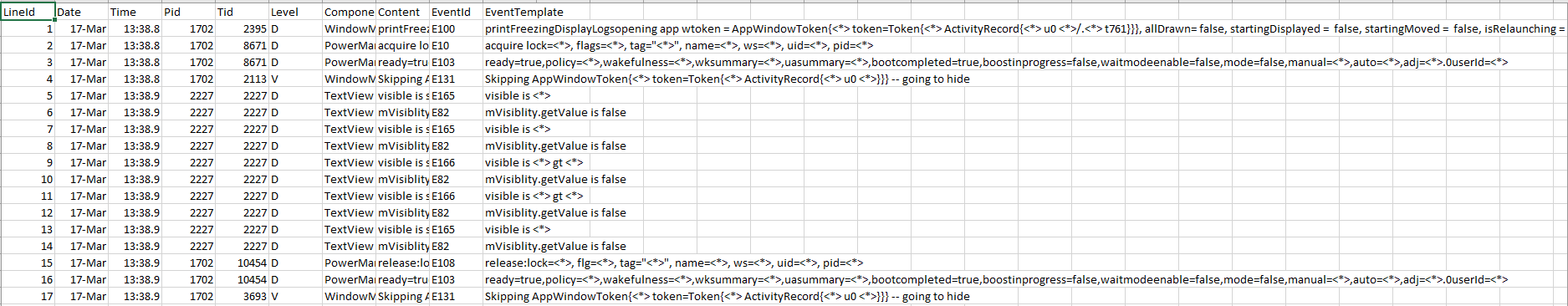
Data can be analyzing in Pandas with: - series and data frames. Series means 1-D array in Pandas. With the help of Series user can store any datatype. DataFrames means rows and columns.so Data frame is 2-D array.

**3.2.3 MATPLOTLIB**

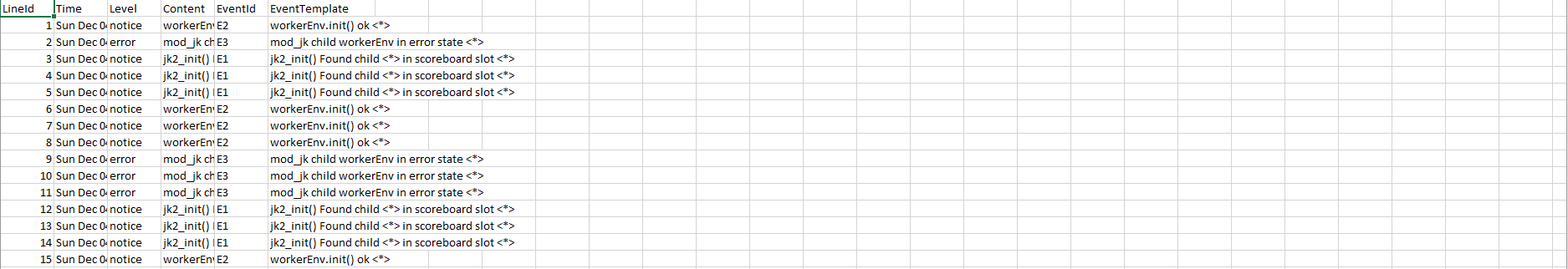
Matplotlib is one of the most important library in Python. This library generally used by the user for creation of graphs. Data can be visualized very easily with the help of this library. This library supports users to create different type of graphs such as- Histogram, Power spectra, Error charts etc.

**3.3 LOG FILES**

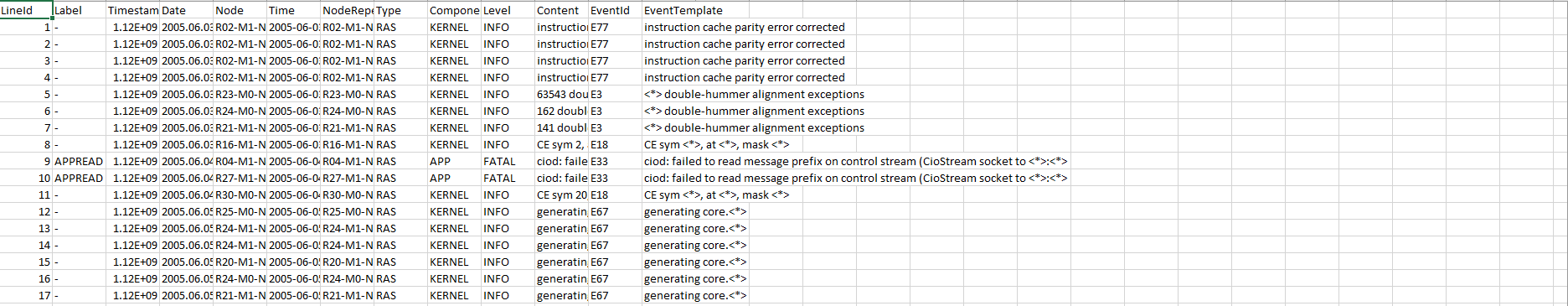
* Android



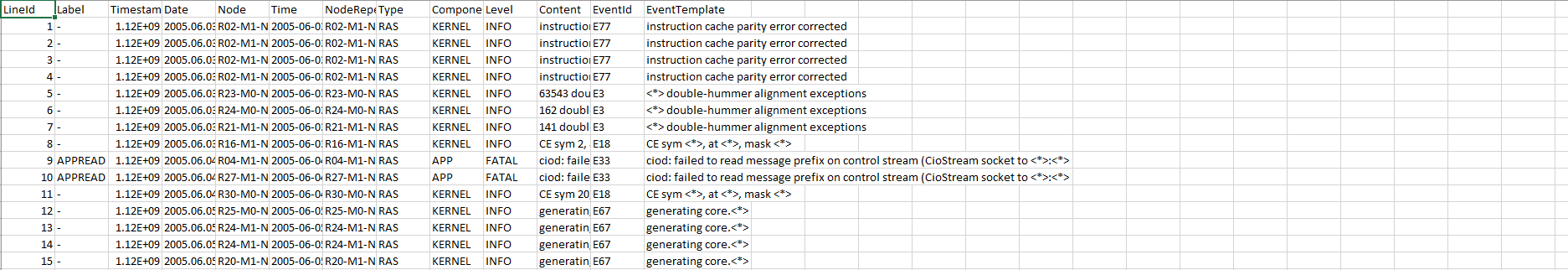
* [Apache log](https://docs.bmc.com/docs/ars81/arsignald-log-603818415.html)



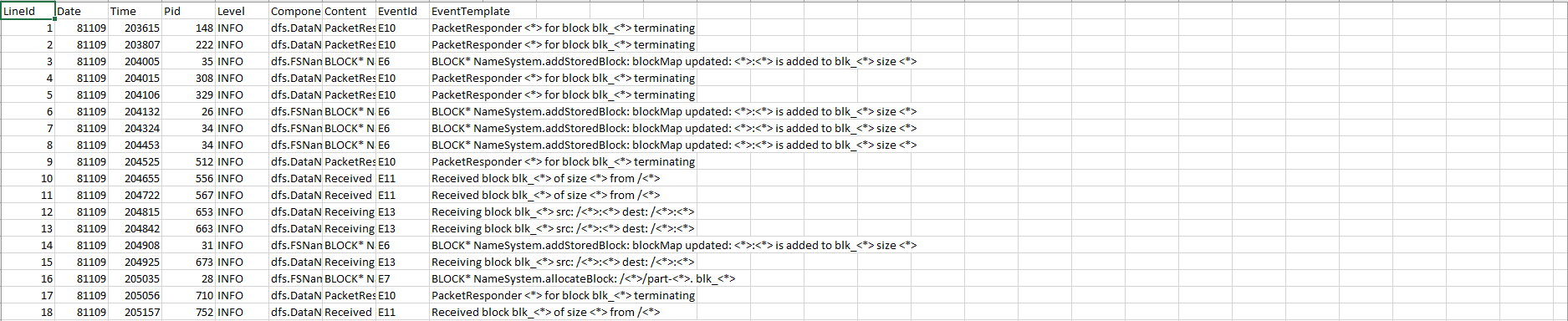
* [BGL log](https://docs.bmc.com/docs/ars81/api-log-225969057.html)



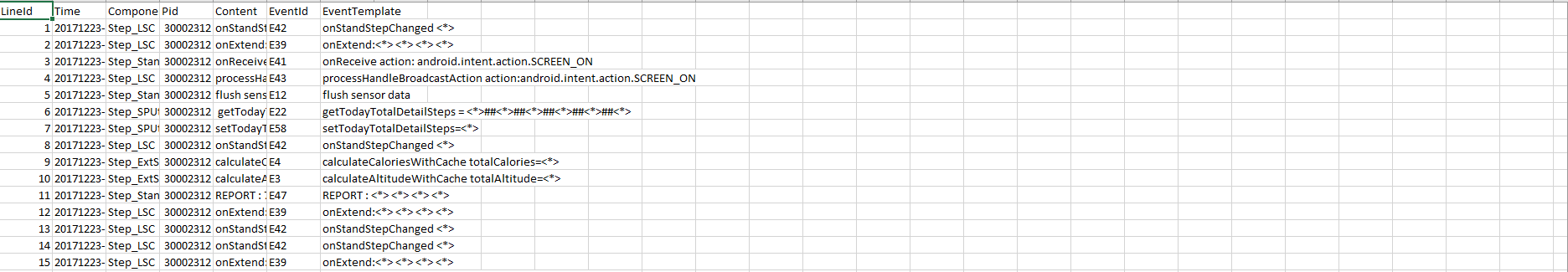
* [Hadoop log](https://docs.bmc.com/docs/ars81/escalation-log-225968918.html)



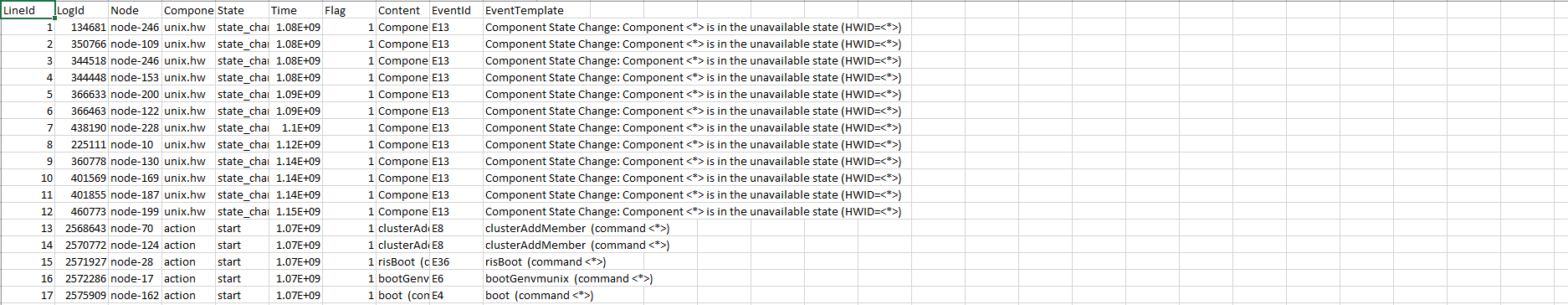
* [HDFS log](https://docs.bmc.com/docs/ars81/filter-log-225968913.html)



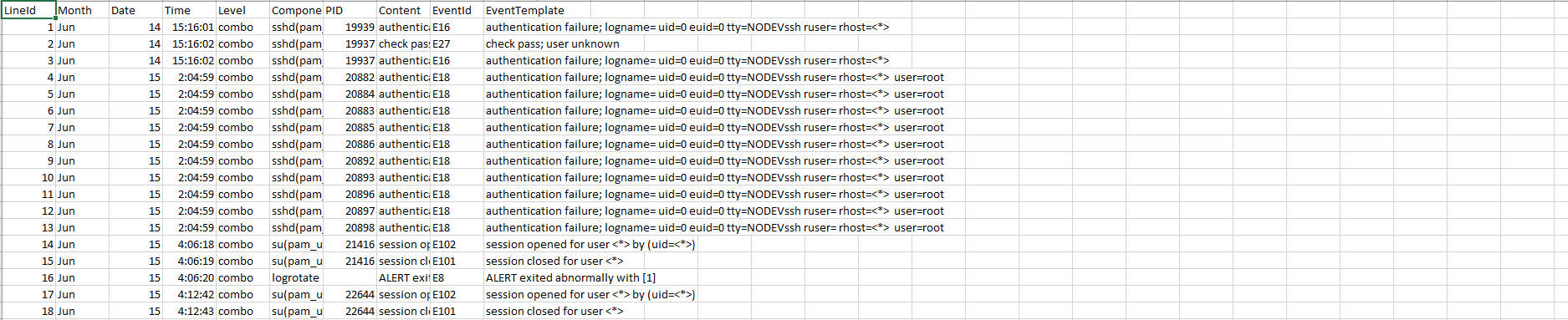
* Application log



* [HPC log](https://docs.bmc.com/docs/ars81/plug-in-server-log-225969061.html)



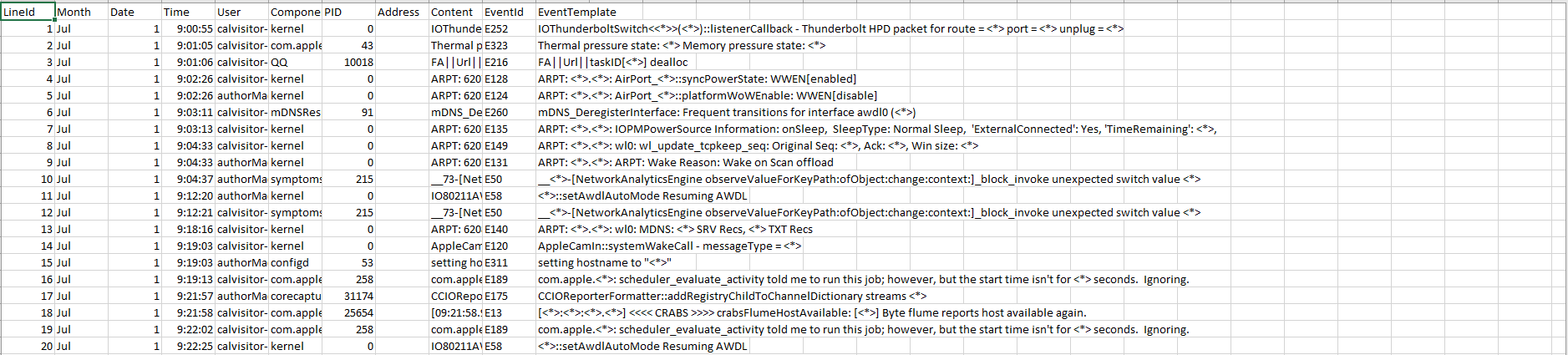
* [Linux log](https://docs.bmc.com/docs/ars81/server-group-log-225969056.html)



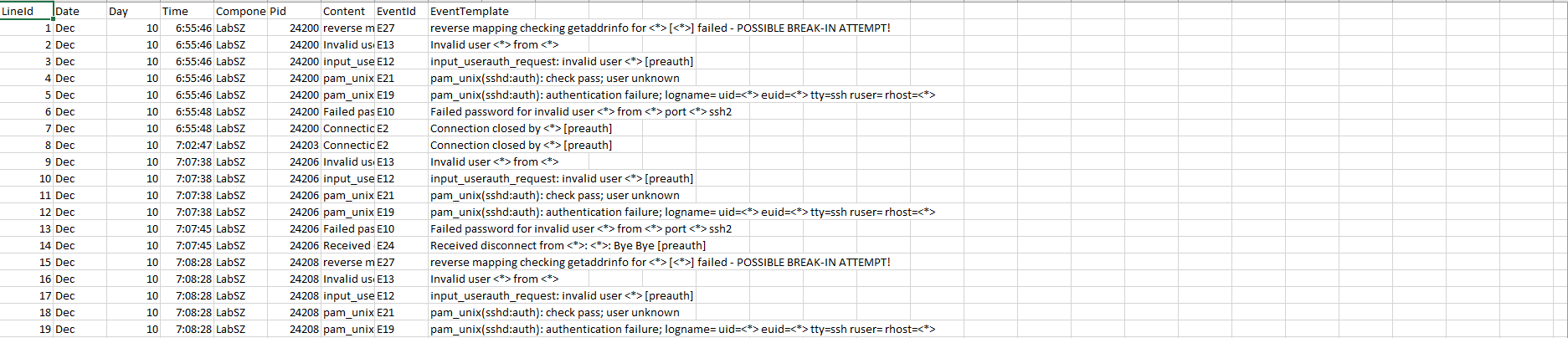
* [SQL log](https://docs.bmc.com/docs/ars81/sql-log-225970761.html)



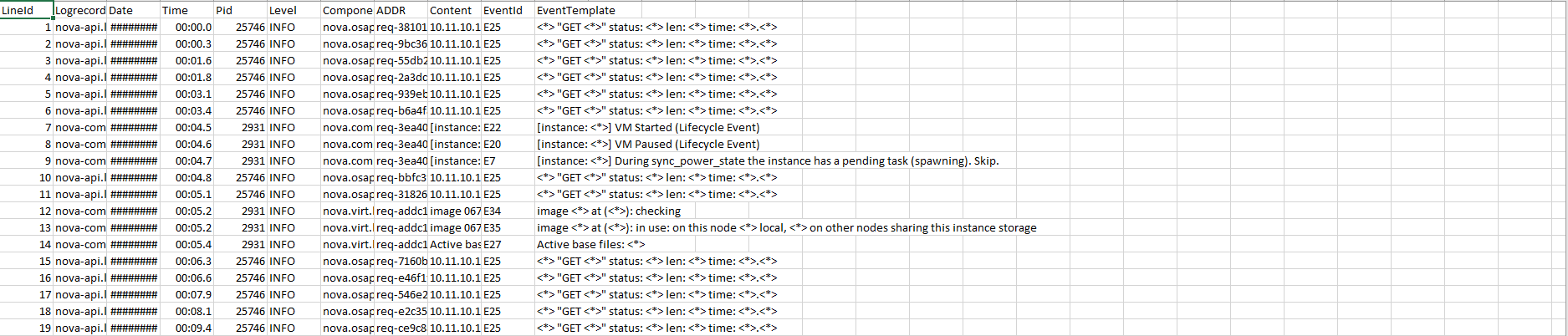
* [MAC log](https://docs.bmc.com/docs/ars81/thread-log-225969036.html)



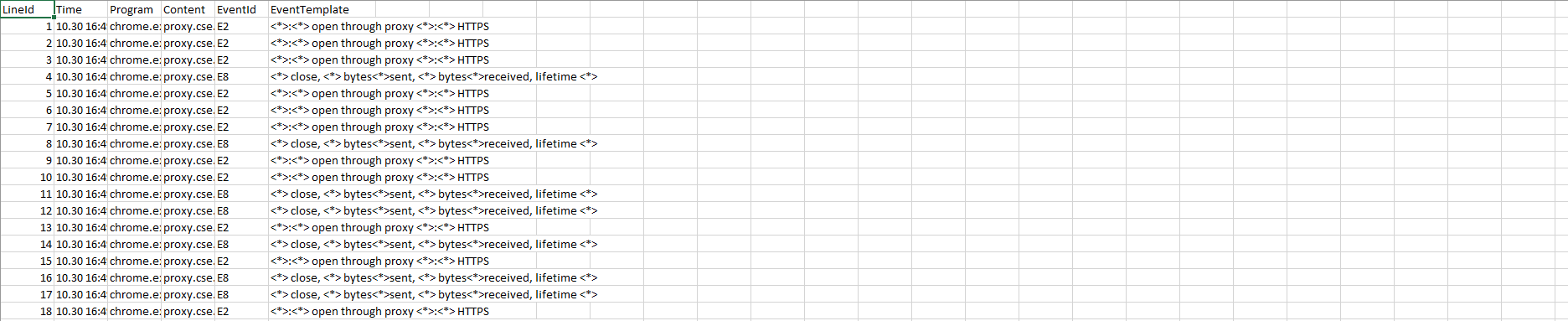
* [OpenSSH log](https://docs.bmc.com/docs/ars81/user-log-225969037.html)



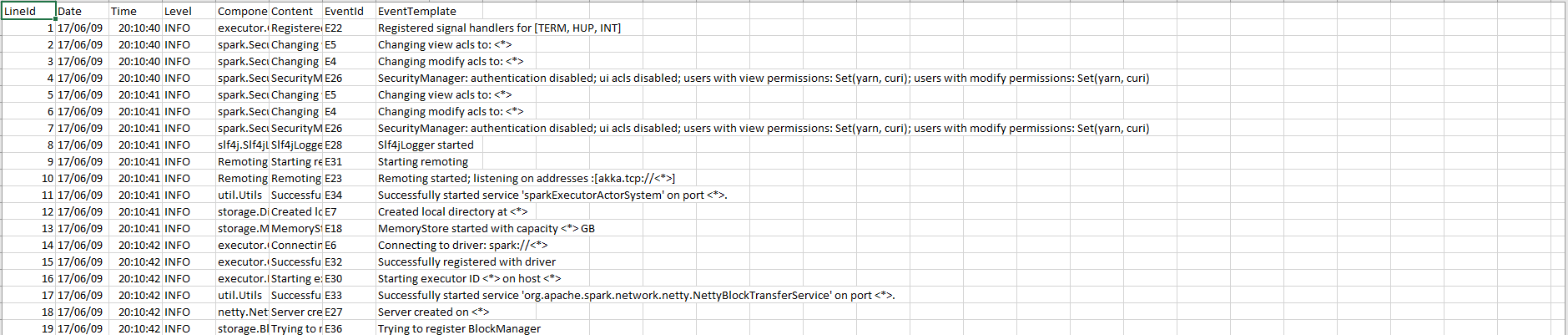
* OpenStack log



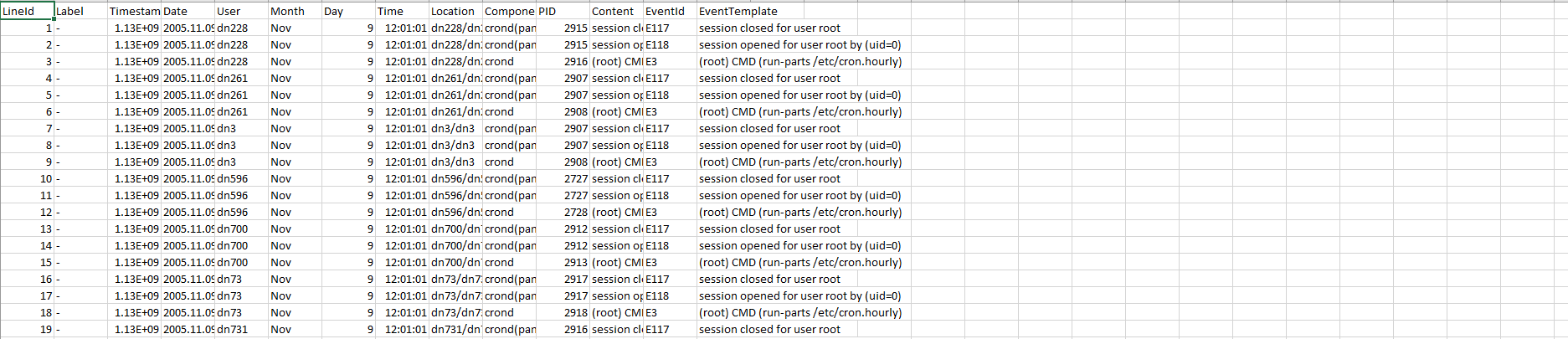
* Proxifier



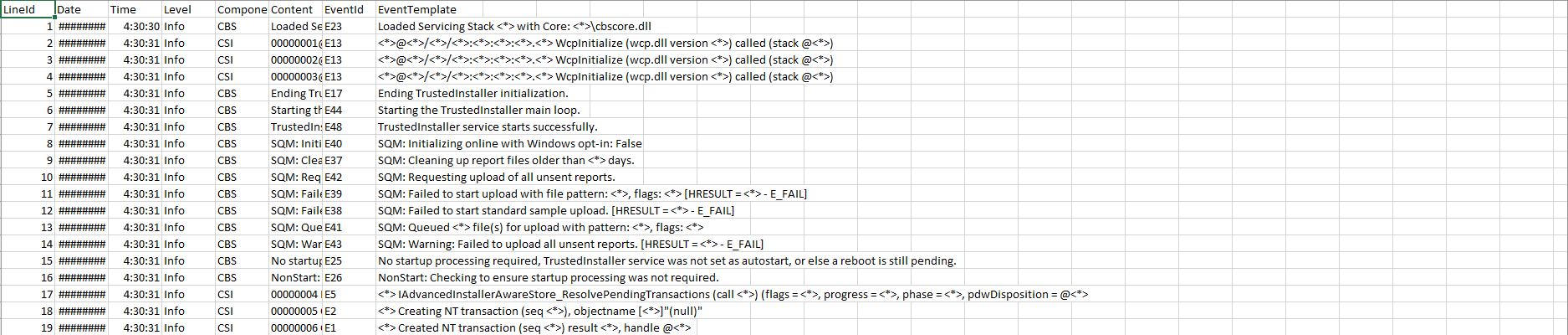
* Spark



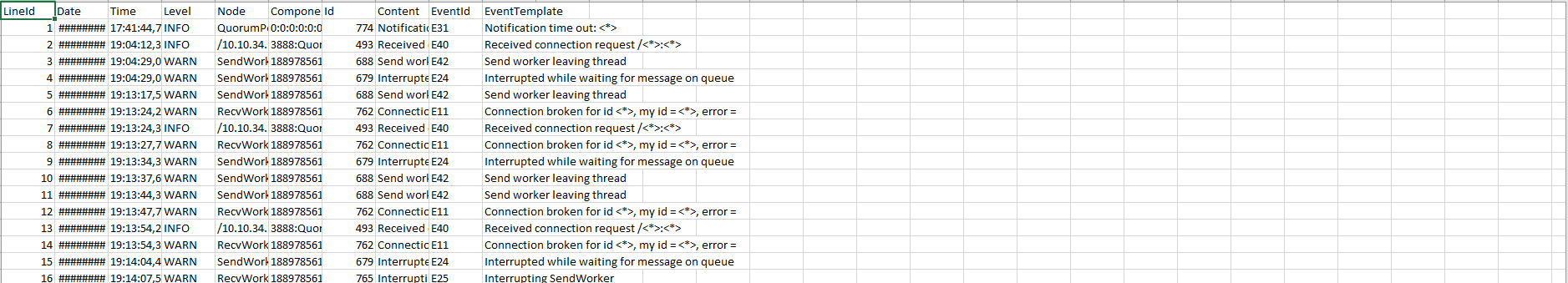
* ThunderBird



* Windows



ZooKeeper



**Chapter - 4**

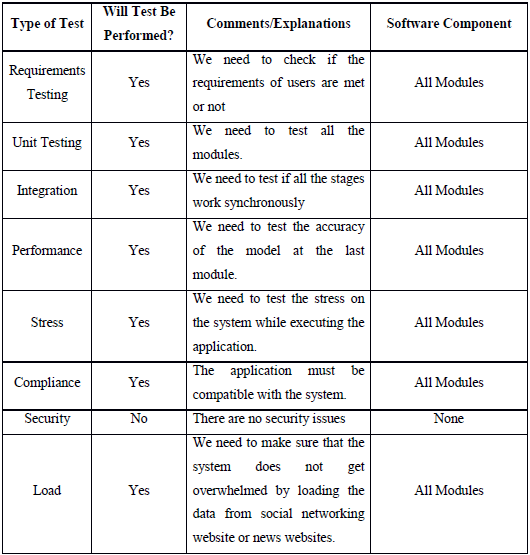
**DATA ANALYSIS**

**4.1 DATA ANALYSIS**

**Chapter - 5**

**TEST PLAN**

**Table 3: Number of different tests to be performed**



**Chapter – 6**

**RESULT & PERFORMANCE ANALYSIS**

**Chapter – 7**

**CONCLUSION**

**References**

[1] M. D. Silverstein, H. Qin, S. Q. Mercer, J. Fong, and Z. Haydar. Risk factors for 30 day hospital readmission in patients 65 years of age.Proceedings (Baylor University. Medical Center), 21(4):363, 2008.

[2] E. Eby, C. Hardwick, M. Yu, S. Gelwicks, K. Deschamps, J. Xie, and T. George. Predictors of 30 day hospital readmission in patients with type 2 diabetes: a retrospective, case-control, database study. Current Medical Research & Opinion, 31(1):107–114, 2014.

[3] H. J. Jiang, D. Stryer, B. Friedman, and R. Andrews. Multiple hospitalizations for patients with diabetes. Diabetes care, 26(5):1421–1426, 2003.

[4] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. BioMed research international, 2014, 2014.

[5] A. Hosseinzadeh, M. T. Izadi, A. Verma, D. Precup, and D. L. Buckeridge. Assessing the predictability of hospital readmission using machine learning. In H. Muoz-Avila and D. J. Stracuzzi, editors, IAAI. AAAI, 2013. 978-1-57735-615-8.