A PROJECT REPORT

On

**LOG SEGREGATOR AND ANALYSER FOR MANAGED INVESTMENTS**

Submitted in partial fulfillment of the

requirements for the degree of

**BACHELOR OF TECHNOLOGY**

In

**Information and Communication Technology**

Submitted by

**117014101 - Suruti.S**

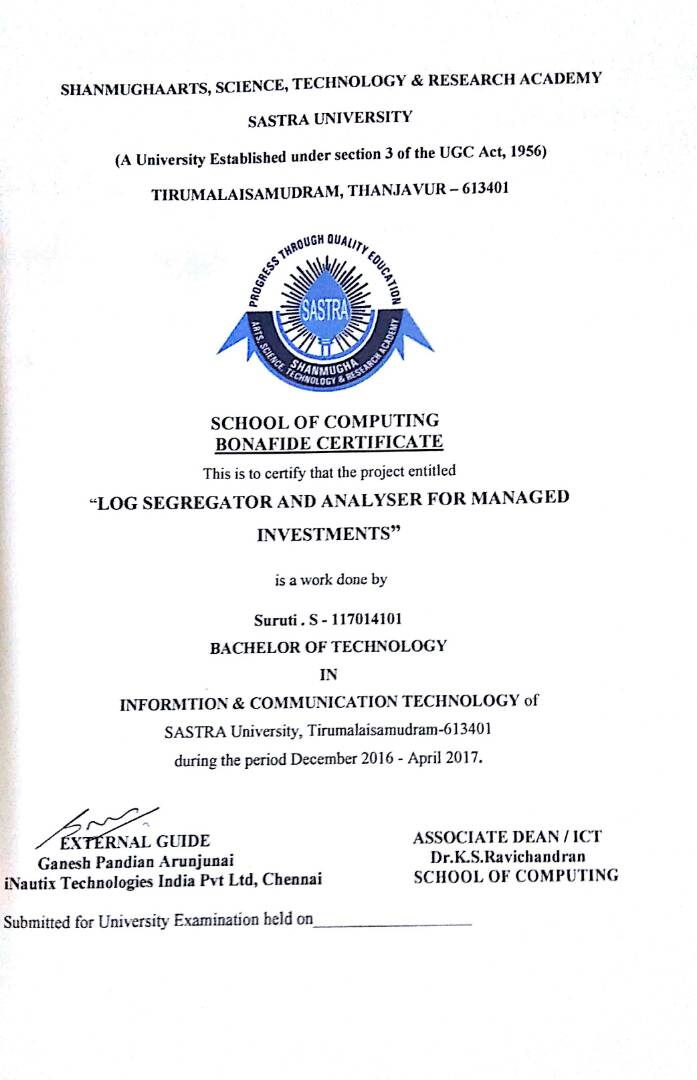
****

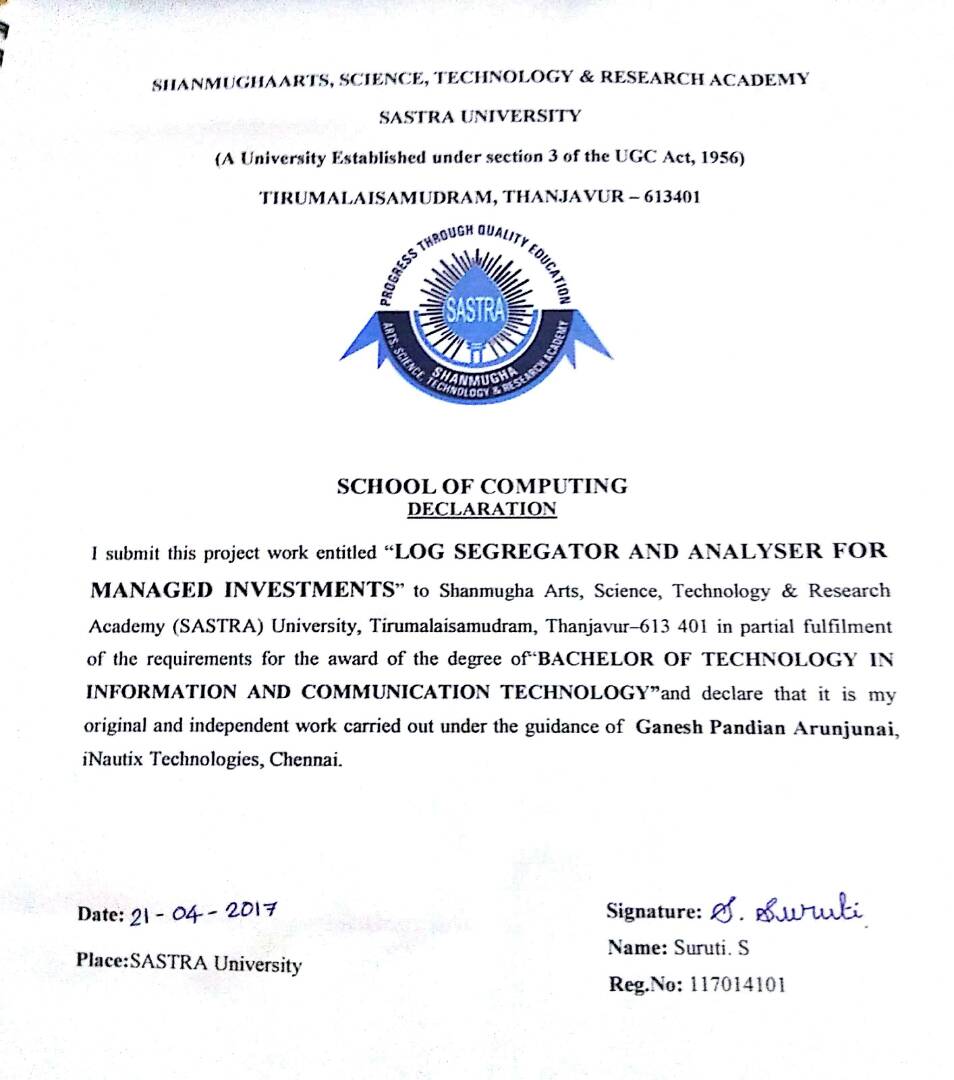
**Under the Guidance of**

**Ganesh Pandian Arunjunai**

**iNautix Technolgies ltd.**

**April 2017**





**Acknowledgement**

I would forever remain grateful to honorable **Prof. R. Sethuraman, Vice Chancellor** for his encouragement in my academic life at SASTRA University.

I wish to express my profound gratitude to **Dr. S. Vaidhyasubramaniam, Dean - Planning & Development, Dr. S. Swaminathan, Director CeNTAB** and **Prof. G. Bhalachandran, Registrar** for their overwhelming support provided during my course span in SASTRA University.

I am extremely thankful to **Dr. K.S. Ravichandran, Associate Dean, Department of Information and Communication Technology** for providing me the opportunity to do this project and for all the academic help extended in my project.

I sincerely express my gratitude to my manager **Mr. Padmaraj Ramachandran** and my team mentor **Mr.** **Ramasubramanian Balasubramanian, iNAUTIX TECHNOLOGIES INDIA PVT LTD** for their assistance and guidance for the successful implementation of project in a systematic and professional manner. I also express my gratitude to my mentor **Mr. Ganesh Pandian Arunjunai**,**, iNAUTIX TECHNOLOGIES INDIA PVT LTD** for always guiding me in the right direction to complete my project.

I also thank all the **Teaching and Non-teaching staff,** and those who have directly or indirectly helped me by extending their moral support and encouragement for completion of this project.

I thank God Almighty and my parents for helping me procure such a challenging and interesting project, and in completing the same in due course without much difficulty.

**SYNOPSIS**

Logging is very crucial for smooth deployment and maintenance of any application. Logging is a process in which all the events that occur during application execution are recorded, which will be of extreme use in troubleshooting obscure problems, debugging and its corresponding maintenance. Typical logs include warnings, errors, transactional changes, relay messages. Logs save us valuable hours. The size of the logger files grow exponentially for frequently used applications.

Most of the log analysers lack artificial intelligence and they are not customized to any organization. This necessitates for the development of a customized log analyser which makes the log file easily interpretable so as to extract the necessary information. Logging helps the application developers know the status of the requests which make their application more user friendly. By looking through the logs, the developers can easily troubleshoot issues, if any which in turn makes the life of the developers and support team easier.

This system helps us derive the necessary log files, which will aid in bug fixing and easy maintenance of any application. This shall be done even for legacy systems. It assists the developer in tracking the bugs, even after years of deployment. A particular or generalized event/ information can be searched to gain meaningful insights from it. Logs can be classified based on its severity and prompt action has to be taken accordingly. The activities of each user is recorded which helps in deriving usage analysis of the system by a user. This data can be exported in multiple formats and delivered to the support team for their reference.

**TABLE OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **FIG.NO** | **NAME OF THE FIGURE** | **PAGE NUMBER** |
| 1 | Fig 1.1 | Log Management Architecture | 5 |
| 2 | Fig 1.2 | Java Logging Framework | 6 |
| 3 | Fig 3.1 | Managing Dependencies | 9 |
| 4 | Fig 3.2 | Spring Framework Runtime Diagram | 10 |
| 5 | Fig 3.3 | Spring IoC Container Layout Diagram | 12 |
| 6 | Fig 3.4 | Aspect Oriented Programming Layout | 13 |
| 7 | Fig 3.5 | Spring MVC Flow Diagram | 14 |
| 8 | Fig 3.6 | jUnit Architecture | 15 |
| 9 | Fig 5.1 | Use Case Diagram | 18 |
| 10 | Fig 5.2 | Class Diagram | 19 |
| 11 | Fig 5.3 | Sequence Diagram | 20 |
| 12 | Fig 5.4 | Activity Diagram | 21 |
| 13 | Fig 7.1 | Search by Application | 27 |
| 14 | Fig 7.2 | Search Result | 28 |
| 15 | Fig 7.3 | Search by Severity | 29 |
| 16 | Fig 7.4 | Configuring a New Application | 30 |
| 17 | Fig 7.5 | Saved Searches | 31 |
| 18 | Fig 8.1 | Validation Testing | 32 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **TITLE** | **Page Number** |
| 1. | Introduction | 1 |
| 2. | Software Project Plan | 3 |
| 3. | Literature Review | 8 |
| 4. | Software Requirement Specification | 16 |
| 5. | System Analysis and Design | 18 |
| 6. | Coding | 23 |
| 7. | Output Snapshots | 27 |
| 8. | Testing | 32 |
| 9. | Implementation | 34 |
| 10. | Conclusion and Future Scope | 35 |
| 11. | References | 36 |