International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

DizBoard: A Monitoring Dashboard Using Python Library Streamlit

1. Hulawale Harshada, 1.Joshi Priyanka, 1. Kulkarni Sarthak. 2. Prof. Shinde Ashwini

[1]*Students, Department of Computer Engineering, Zeal College of engineering and research narhe Pune*

[2]*Assistant Professor, Department of Computer Engineering, Zeal College of engineering and research narhe Pune.*

***

Abstract: In the era of data-driven decision-making, monitoring dashboards have become indispensable tools across various industries, facilitating the real-time visualization of metrics and data analytics. "DizBoard" introduces an innovative approach to dashboard development, leveraging the Python library Streamlit, known for its simplicity and efficiency in creating interactive web applications. This paper presents the design, implementation, and functionality of DizBoard, a comprehensive monitoring dashboard aimed at providing users with intuitive access to real-time data and insights.

Keywords: Monitoring Dashboard, Streamlit, Python.

1. INTRODUCTION

In the era of rapid technological advancements, the need for efficient monitoring tools for IT systems has become paramount. To address this need, we present "Dizboard: A monitoring dashboard using the Python library Streamlit." Proposed system provides user centric application which simplifies monitoring of complex software ecosystems. This reduces the need of human effort and time to monitor web services. Due to the use of this application a non-technical person will also be able to monitor and get updates of different types of services. It provides a centralized platform for users to monitor the status of systems, add new environments, and receive notifications for any downtime. The dashboard leverages the simplicity of Streamlit to offer a seamless user experience with minimal coding effort. In this paper, we outline the design, features, and implementation details of Dizboard, emphasizing its usefulness in system administration and network monitoring tasks. We demonstrate how Streamlit's intuitive interface and integrated data visualization capabilities contribute to a dynamic and responsive monitoring experience. As this application can be used by any organization, here we demonstrated implementation of this dashboard in a computer lab at college campus to monitor web services.

1.1 PROBLEM STATEMENT

To address the challenges posed by the increasing complexity and scale of modern software applications, there is a critical need for a user-centric monitoring solution that simplifies the process of web services. Current tools often require extensive technical expertise, leading to inefficiencies in issue identification and resolution. Manual monitoring methods and reactive approaches to issue resolution result in extended downtimes and service disruptions. Additionally, the lack of accessible notification systems further compounds the problem. Therefore, the development of DizBoard, a Streamlit-based application, aims to provide a comprehensive and intuitive platform that empowers both technical and non-technical users to monitor application health effectively and proactively.

2. RELATED WORKS

Alessandro Tundo, Chiara Castelnovo et.al "Declarative Dashboard Generation" [2020], Complex software systems, known as systems of systems, demand adaptable monitoring solutions for effective control. Current dashboard systems are often challenging to configure and adjust as indicators change. This paper presents our initial work on an automated dashboard generation process that utilizes

meta-model layouts to create comprehensive dashboards from operator-selected indicators. Although emphasizing timely actions by operators, the abstract does not elaborate on the specific mechanisms or features in place to ensure the responsiveness of the generated dashboards for prompt decision-making.[9]

Jaimesolis-Martinez, Jordanpas Cual Espada et.al "UXJs: Tracking and Analyzing Web Usage Information With a Javascript Oriented Approach" [2020], UXJs offers a new approach to tracking and analyzing user behavior on websites. It focuses on comprehensive data collection, quantitative presentation, and automated statistical analysis, aiming to provide valuable insights to web developers for improving their websites' design and usability. Understanding user behavior within a website is crucial for assessing design, structure, and usability. While mentioning automatic statistical analysis, the abstract does not elaborate on the methodologies or algorithms used, leaving a gap in understanding the robustness and accuracy of the proposed approach.[5]

Bong-Hwan Oh, Serdar Vural et.al " A Lightweight Scheme of Active-Port-Aware Monitoring in Software-Defined Networks " [2021], APAM is proposed as a lightweight monitoring mechanism for SDN to address the challenges of heavy monitoring overhead in complex network environments. It achieves dynamic monitoring and adapts to changing network conditions, leading to improved efficiency and accuracy in network monitoring. Existing network monitoring approaches are often heavy, leading to significant monitoring overhead. This becomes problematic in software-based network systems with multiple networks and complex policies. The article introduces a lightweight monitoring mechanism called Active-port Aware Monitoring (APAM).[2]

Narongsak Sukma, Wasin Srisawat, et.al "An Analysis of Log Management Practices to reduce IT Operational Costs Using Big Data Analytics" [2019], The paper presents a solution that streamlines SOC operations by utilizing Big Data Analytics for log management. This results in significant cost savings, increased efficiency, and improved employee satisfaction, making it a promising approach for modern security operations. The claim of increased work flexibility and reduced stress lacks supporting details, making it difficult to assess the validity and long-term impact of these asserted benefits. This also lacks specifics on the components of the 60% cost reduction, making it challenging to evaluate the accuracy and sustainability of the claimed reduction.[8]

Mary Dempsey, Attracta Brennan et.al "A Review of the Most Significant Challenges Impacting Conventional Project Management Success" [2021], This research paper discusses the importance of understanding the success factors and challenges in conventional project management for organizations. The review process identified five significant challenges: communication, control, competence, culture, and complexity, with an emphasis on their interdependence. However, it acknowledges limitations due to database selection and search terms. It also suggests that combining other project management methods and recommendations can help address these challenges. The study may not cover the full spectrum of project management challenges, potentially overlooking important issues outside the selected scope. The research methodology, including chosen databases and publications, may introduce bias, limiting the diversity of perspectives on project management challenges.[6]

3. PROPOSED SYSTEM

The proposed system is called "DizBoard: a monitoring dashboard using python library Sreamlit"and is designed to streamline the monitoring of software environments, DizBoard offers a user-centric solution built on the Streamlit platform, tailored for use in educational institutions like college labs. Upon authentication, users gain access to a personalized dashboard displaying their name and key functionalities. The dashboard allows users to add new environments, such as specifying the number of systems or servers, and inputting relevant details like names and IP addresses. This information is securely stored in a database (e.g., MySQL) for subsequent monitoring and management tasks.

DizBoard is uniquely designed to cater to college lab settings, where it can monitor PC systems efficiently. Users can initiate the "Monitor Systems" feature to retrieve stored system information and perform real-time connectivity checks using tools like ping3. The dashboard showcases online systems as successful and identifies offline systems for immediate attention. Users can receive notifications via email or within the application regarding any system issues, ensuring proactive management.

The dashboard also offers an overview of the overall connection status, including the total number of systems, online/offline counts, and visual insights through graphical representations like pie charts. This comprehensive approach ensures efficient oversight of software ecosystems within college lab environments. By consolidating monitoring and management tasks into a unified interface, DizBoard optimizes system reliability and performance with minimal effort, enhancing operational efficiency for educational institutions.

3.1 IMPLEMENTATION

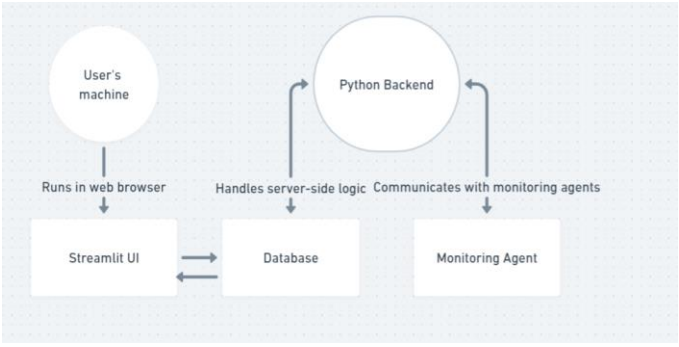


Fig 3.1.1System architecture

Fig 3.1.2Use case

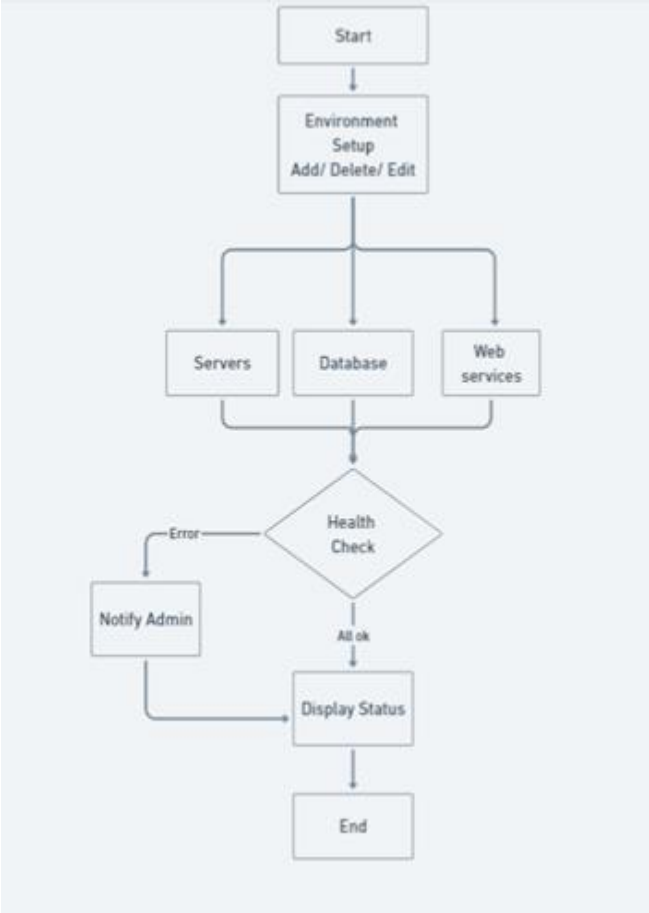


Fig 3.1.3Data flow diagram

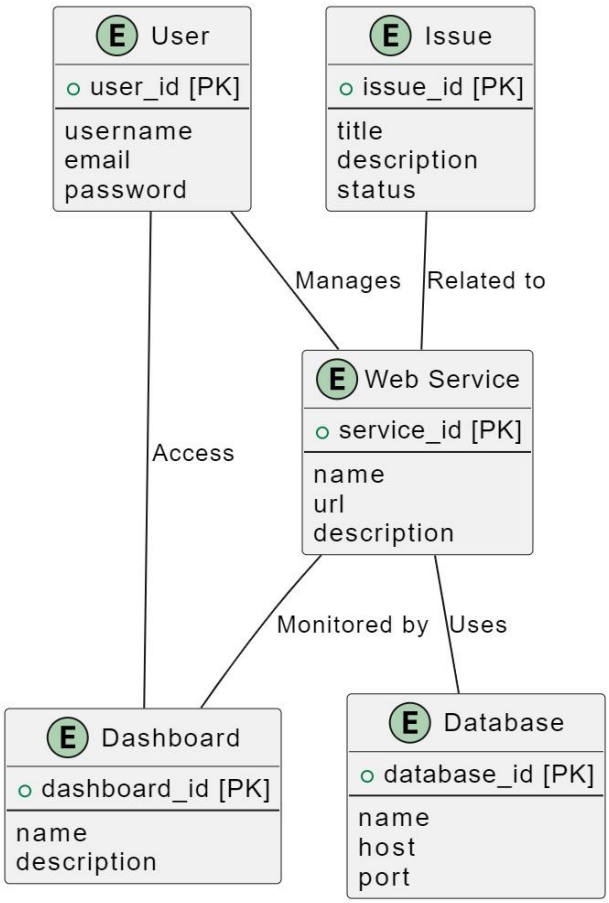


Fig 3.1.4 ER diagram

4. METHODOLOGY

The development and deployment of DizBoard follow a systematic methodology aimed at ensuring robustness, scalability, and user-friendliness. The methodology encompasses several key stages:

Requirements Gathering and Analysis: The methodology begins with comprehensive requirements gathering, where the specific needs and functionalities of the monitoring dashboard are identified. This includes understanding user roles, system monitoring requirements, database management needs, and notification preferences.

System Design: After gathering requirements, the system design phase entails architecting the application's structure, including database schema design, user interface layout, and backend functionalities. This phase involves defining the architecture, selecting appropriate technologies (Streamlit, MySQL, Python libraries), and designing data flow diagrams (DFDs) to visualize system processes.

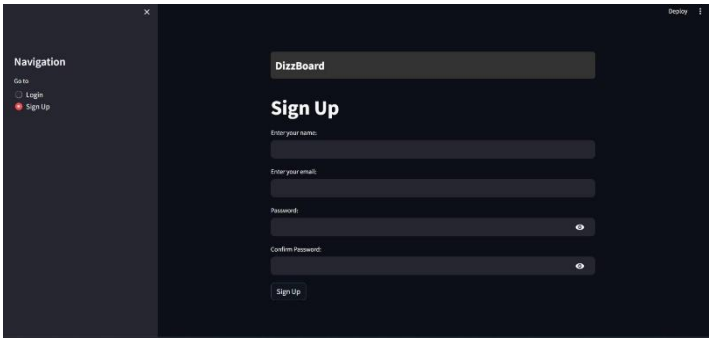
Implementation: The implementation phase involves translating the system design into executable code. This includes developing the user interface using Streamlit, setting up the MySQL database for data storage, integrating monitoring functionalities using Python libraries like ping3 for connectivity checks, and implementing email notification capabilities with smtplib.

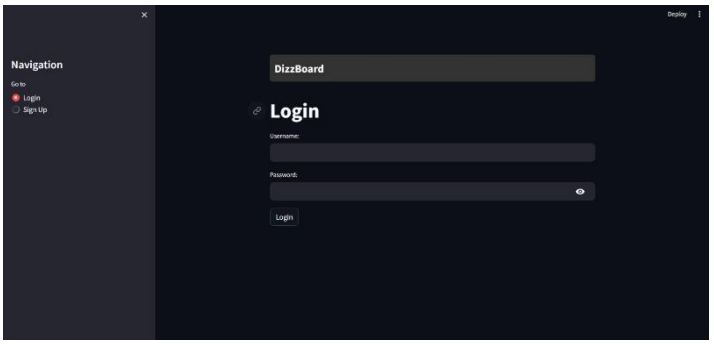
Testing: The testing phase focuses on validating the application's functionality and performance. This includes unit testing of individual components, integration testing to ensure seamless interaction between modules, and system testing to verify end-to-end functionality. Testing scenarios cover user authentication, system addition, monitoring checks, and email notifications.

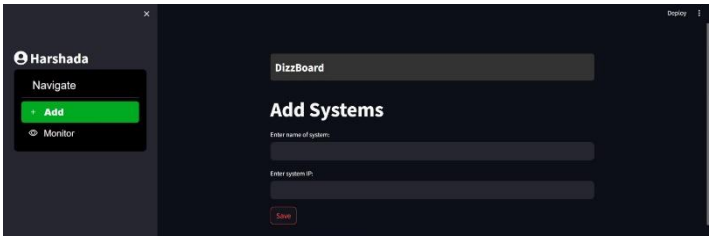
Deployment and User Training: Upon successful testing, the application is deployed in the target environment, such as a college lab or educational institution. User training sessions are conducted to familiarize stakeholders with DizBoard's features and functionalities, ensuring effective utilization and adoption.

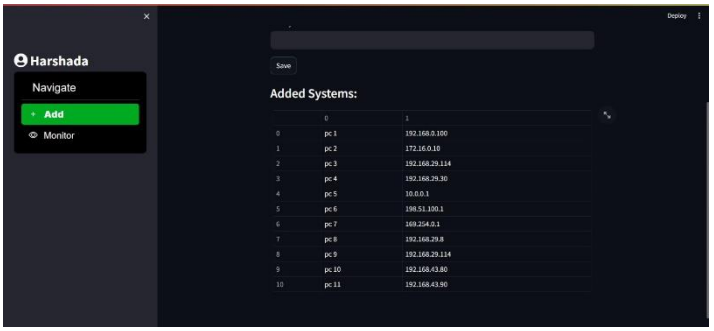
Maintenance and Iteration: Post-deployment, ongoing maintenance ensures the application remains operational and responsive to user needs. Feedback from users and stakeholders informs iterative improvements and feature enhancements, driving continuous refinement of DizBoard's capabilities

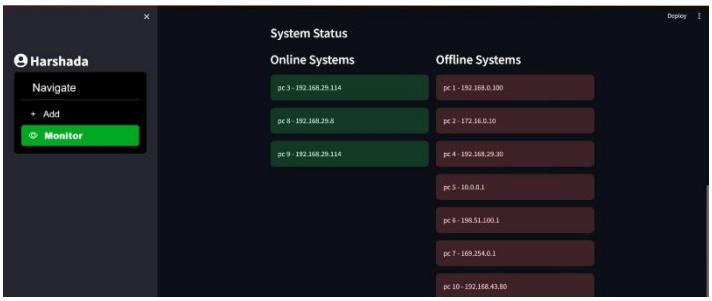
5. RESULT

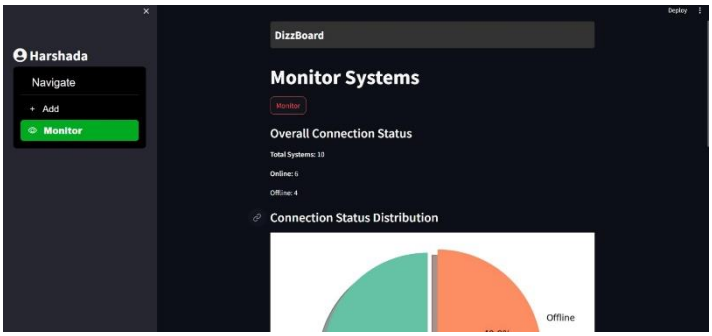


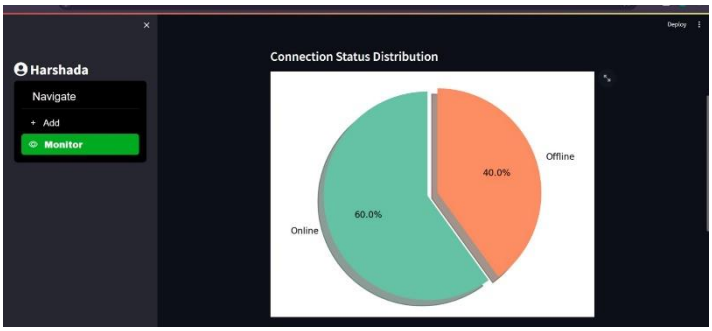


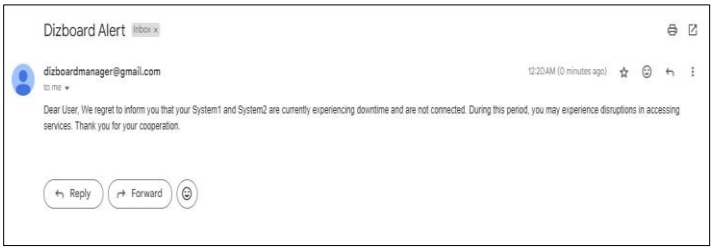












6. CONCLUSION

In this study, the proposed architecture of DizBoard, a monitoring dashboard utilizing Streamlit and MySQL, demonstrates significant advancements in software application monitoring. Designed for college lab environments, DizBoard provides intuitive system monitoring, database management, and email notifications, enhancing operational oversight and reliability. Its implementation in educational settings underscores its practical utility, streamlining IT resource management and fostering system efficiency. Future development will prioritize user feedback and technological advancements to ensure DizBoard remains a leading solution for software monitoring in educational and institutional contexts. This study highlights DizBoard's transformative impact on operational seamlessness and reliability in diverse computing environments.

7. REFERENCES

[1] Ali Ibrahim, Isa. (2015). Client/Server Interface Monitoring and Management. Volume-2. 2350- 0557.

[2] B. -H. Oh, S. Vural, et al., "A Lightweight Scheme of Active-Port-Aware Monitoring in Software-Defined Networks," in IEEE Transactions on Network and Service Management, vol. 18, no. 3, pp. 2888-2901, Sept. 2021, doi: 10.1109/TNSM.2021.3066273.

[3] Cândido, Jeanderson & Aniche,et al. (2021). Log-based software monitoring: a systematic mapping study. PeerJ Computer Science. 7. e489. 10.7717/peerj-cs.489.

[4] Gaol, Ford Lumban, Santoso, et al. "Design and development of the application monitoring the use of server resources for server maintenance" Open Engineering, vol. 12, no. 1, 2022, pp. 524-538. https://doi.org/10.1515/eng-2022-0055.

[5] J. Solís-Martínez, J. P. Espada, et al., "UXJs: Tracking and Analyzing Web Usage Information With a Javascript Oriented Approach," in IEEE Access, vol. 8, pp. 43725-43735, 2020, doi: 10.1109/ACCESS.2020.2977879.

[6] M. Dempsey, A. Brennan, et al. "A Review of the Most Significant Challenges Impacting Conventional Project Management Success," in IEEE Engineering Management Review, vol. 50, no. 3, pp. 193-199, 1 thirdquarter,Sept. 2022, doi: 10.1109/EMR.2022.3187168.

[7] M. Zhang, P. Martin, et al. "Workload Management in Database Management System: A Taxonomy (Extended

Abstract)," 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 2018, pp. 1823-1824, doi: 10.1109/ICDE.2018.00269.

[8] N. Sukma, W. Srisawat, et al., "An Analysis of Log Management Practices to reduce IT Operational Costs Using Big Data Analytics," 2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Bangkok, Thailand, 2019, pp. 1-5, doi: 10.1109/TIMES-iCON47539.2019.9024400.

[9] Tundo, C. Castelnovo, et al. "Declarative Dashboard Generation," 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Coimbra, Portugal, 2020, pp. 215-218, doi: 10.1109/ISSREW51248.2020.00075.