PROG7312

POE

Lecturer:

Miss Nobubele

Nyiko Shabangu

ST10119175

NOVEMBER 18, 2024

Table of Contents

Report Project Completion Report

Project Overview

The project aimed to develop a Municipal Services Application designed to streamline the management of municipal services. This application allows users to request, track, and access municipal data efficiently. By leveraging robust data structures and algorithms, the application optimizes performance, ensures data consistency, and enhances overall user experience.

Challenges in Task 3

Task 3 focused on integrating service request management and issue reporting functionalities. Several challenges emerged during this phase:

Debugging and Background Processes

- Challenge: Managing background processes and displaying real-time results from long-running tasks added complexity, particularly when debugging these processes.
- Solution: We utilized specialized debugging tools and detailed logging mechanisms to track issues and ensure efficient error resolution.

Key Learnings and Insights

1. Programming Techniques
   - I gained hands-on experience in backend development using C# and the .NET Framework, while also enhancing my frontend skills with Windows Forms. Integrating these technologies was complex, but it provided valuable insight into delivering a seamless, responsive user experience.
2. Data Structures and Algorithms
   - I implemented advanced data structures like AVL Trees and Heaps, which significantly improved service request management and prioritization. Understanding their time and space complexities allowed us to optimize the system for large-scale data management.

Technology Recommendations

1. Machine Learning for Predictive Analytics
   - Recommendation: Integrating machine learning algorithms to predict service requests based on historical data could enhance the system's functionality.
   - Justification: Machine learning models can identify recurring service needs in specific areas, enabling the system to proactively notify users about likely issues and improve service efficiency and satisfaction.

Conclusion

The Municipal Services Application has evolved into a robust, user-friendly platform that enhances service delivery and optimizes user interactions. Future enhancements, including integrating cloud services, mobile support, and machine learning, would further enhance scalability and responsiveness.

Implemented Data Structures in "Service Request Status" Feature

The "Service Request Status" feature is essential for managing and tracking service requests. Various data structures were employed to ensure efficient management, fast access, and accurate tracking:

1. **Binary Trees**
   - Role: Binary trees are used for efficient searching and organizing of service requests.
   - Example: An AVL tree organizes service requests based on priority. New requests are inserted in O(log n) time, ensuring quick retrieval.
   - Time Complexity: O(log n) for insertion, searching, and deletion.
2. **AVL Trees**
   - Role: AVL trees maintain balance to guarantee that search, insertion, and deletion operations remain efficient.
   - Example: Real-time updates, such as status changes, are managed by the AVL tree, where each node represents a request sorted by timestamp or urgency.
   - Time Complexity: O(log n) for insertion and searching.
3. **Heaps (Priority Queue)**
   - Role: Heaps manage service requests based on priority.
   - Example: A min-heap ensures the highest-priority requests (e.g., urgent maintenance) are processed first. Insertion and deletion take O(log n) time, while accessing the highest priority is O(1).
   - Time Complexity: O(log n) for insertion and deletion, O(1) for accessing the root.
4. **Hash Tables (Dictionaries)**
   - Role: Hash tables provide fast lookups for service request statuses.
   - Example: A hash table stores the status of each request (e.g., "In Progress," "Completed"). Retrieving a request’s status by its ID takes O(1) time.
   - Time Complexity: O(1) for insertion and lookup.

These data structures, optimized for specific tasks, work together to enable efficient management of service requests, ensuring real-time updates, quick lookups, and balanced operations across large datasets.

Lecture Feedback

"The lecture focused too much on additional requirements when it could have been kept simple."

The lecture would have been more effective if it had emphasized the core tasks and simplicity. By concentrating on the essential concepts and avoiding unnecessary complexities, the learning experience could have been more focused and manageable. A clear emphasis on fundamental principles would have reinforced key ideas without overwhelming students with extra details. Simplifying the approach would have made the content more accessible, allowing students to grasp the material more easily.

Reference List

Chen, Y., Li, L., Li, W., Guo, Q., Du, Z. and Xu, Z. (2023). Fundamentals of neural networks. *Elsevier eBooks*, [online] pp.17–51. doi:https://doi.org/10.1016/b978-0-32-395399-3.00008-1.

DESCARTES, N. (2024). *Truly Understanding Neural Networks through their Implementation in C#*. [online] Codeproject.com. Available at: https://www.codeproject.com/Articles/5375908/Truly-Understanding-Neural-Networks-through-their) [Accessed 13 Oct. 2024].

Frogglew (2024). *What is Azure Machine Learning? - Azure Machine Learning*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2.

GeeksforGeeks (2022). *MVC Framework Introduction*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/mvc-framework-introduction/ [Accessed 28 Jun. 2024].

Klein, Prof. Richard (2014). *8 Binary Search Trees | Introduction to Data Structures & Algorithms*. [online] Wits.ac.za. Available at: https://courses.ms.wits.ac.za/~richard/IDSA-Notes-YouTube/bst.html [Accessed 14 Oct. 2024].

Malik, D.S. (2018). *C++ programming : program design including data structures*. 6th ed. Boston, Ma: Cengage Learning.

Marcin Jamro (2018). *C# data structures and algorithms : explore the possibilities of C# for developing a variety of efficient applications*. Birmingham Packt.

Synopsys.com. (2023). *What Is CI/CD and How Does It Work? | Synopsys*. [online] Available at: https://www.synopsys.com/glossary/what-is-cicd.html [Accessed 28 Jun. 2024].

www.w3schools.com. (n.d.). *DSA Tutorial*. [online] Available at: https://www.w3schools.com/dsa/index.php.