NATIONAL UNIVERSITY OF LESOTHO

CS4430

PRINCIPLES OF DISTRIBUTED DATABASE SYSTEMS

CLUSTERCORE

201902624 L LETSIE

201902131 M SEEQELA

201902703 F MOTSU

202004214 N TLALI

MMB SYSTEM DESIGN

1. SYSTEM ARCHITECTURE SCENARIO

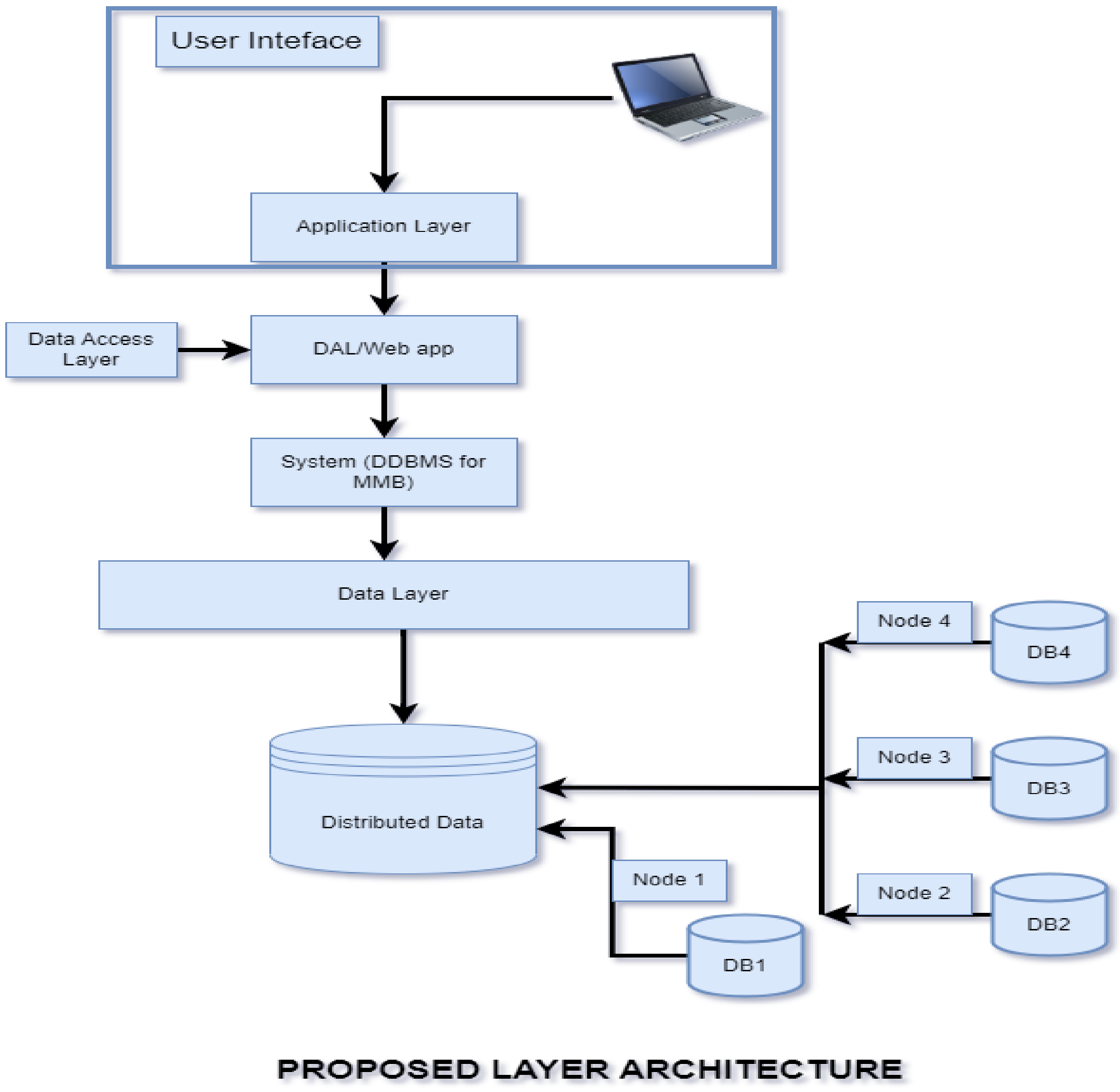


Figure 2: Proposed client/server layer architecture

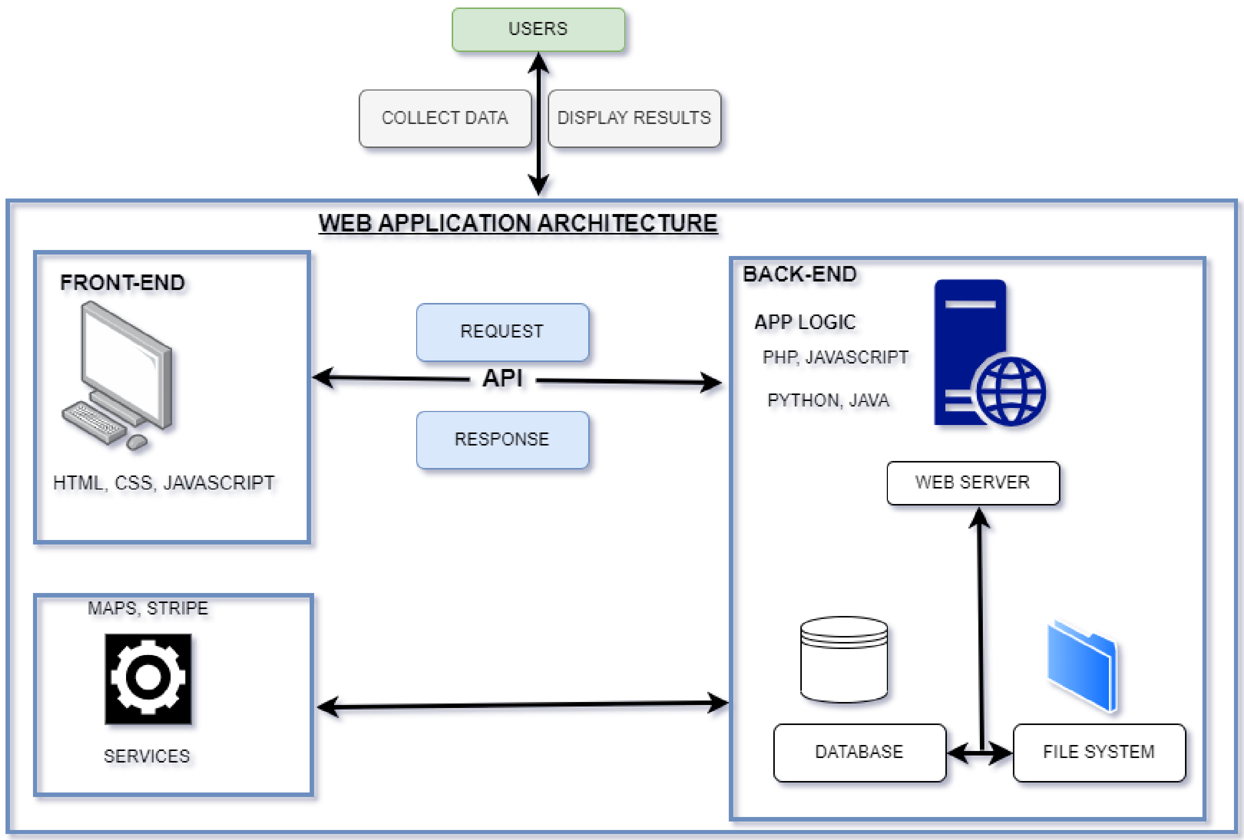
Hardware Architecture: We would adopt using a combination of on-premises and cloud resources. On-premises hardware can be used to run the database servers and application servers, while cloud resources can be used to provide additional scalability, redundancy, and disaster recovery. In this design, the on-premises hardware will consist of high-performance servers with ample storage capacity to store the database files, while the cloud resources will consist of virtual machines and cloud storage to provide additional capacity and redundancy.

Networking Architecture: A hybrid solution would be appropriate (combining local site networking and remote networking). The local site networking would consist of high-speed Ethernet switches and routers to provide high-speed connectivity between the servers and clients on the local network. Remote networking would consist of secure VPN connections to connect the remote sites to the main brewery distribution database system. This design will ensure that data can be transmitted securely between sites, and remote sites can access the centralized database system with ease.

Software Architecture: A combination of open-source and commercial software can be used to achieve a robust and scalable solution. The platform will consist of a Linux operating system, a MySQL database server, and Apache Tomcat application server. The database server will be responsible for storing and managing the data, while the application server will provide a platform for the development of web-based applications that will interact with the database. The software architecture will also incorporate middleware solutions for data integration between different applications and platforms.

The design choices made in this scenario are intended to provide a highly available, scalable, and robust system for a brewery distribution database system. The hybrid networking architecture will ensure that data can be transmitted securely and reliably between local and remote sites. The hardware architecture combines the benefits of on-premises hardware with the scalability and redundancy of cloud resources. The software architecture incorporates open-source solutions, which are cost-effective, scalable, and easy to customize, while also leveraging commercial software for enterprise-grade features and support.

2. EXPERIMENTAL SYSTEM ARCHITECTURE



Hardware Architecture: The system will consist of multiple servers, each running a distributed database management system (DBMS) and connected to storage devices. The servers will be high-performance machines with sufficient processing power, memory, and storage to handle the data and workload requirements.

Networking Architecture: The system will have a local site with a local area network (LAN) connecting the servers. The servers will communicate with each other through a high-speed LAN to ensure fast data transfer and processing. Additionally, a remote site will also be set up with a wide area network (WAN) connection to the local site.

Software Architecture: The platform will be built on a Linux operating system, which is known for its stability, security, and open-source nature. The application will use a distributed database management system like Apache Cassandra, which can handle large amounts of data, is highly available, and can scale horizontally. The application will also use microservices architecture to allow for modular development, scaling, and maintenance.

The distributed database system is chosen to ensure high availability and scalability of the system. Apache Cassandra is chosen due to its ability to handle large amounts of data and its distributed nature, allowing for easy scaling. Linux is chosen for its stability and security, and the microservices architecture is chosen for its flexibility and scalability**.**

3. DISTRIBUTED DB DESIGN

Google Cloud Storage (GCS): Google Cloud Storage (GCS) will be used as the storage solution for the distributed database. GCS provides scalable, durable, and highly available object storage that can handle large amounts of data. It also has built-in data redundancy and backup capabilities, ensuring data availability and integrity.

Data Partitioning: Data partitioning will be used to distribute the data across the nodes in the distributed database. Hash partitioning will be used to evenly distribute the data among the nodes based on a hash function applied to a specific data attribute. This method ensures that the data is distributed evenly and avoids data hotspots, where one node may become overloaded.

Data Allocation and Replication Scheme: The data will be allocated across multiple nodes in the distributed database, with each node responsible for a specific partition of the data. To ensure high availability and fault tolerance, data replication will be used. Each partition will have multiple replicas distributed across different nodes. A quorum-based replication scheme will be used, where a write operation is considered successful only after it is written to a specified number of replicas, ensuring data consistency and durability.

The following diagram illustrates the distributed database design for the brewery distribution system:

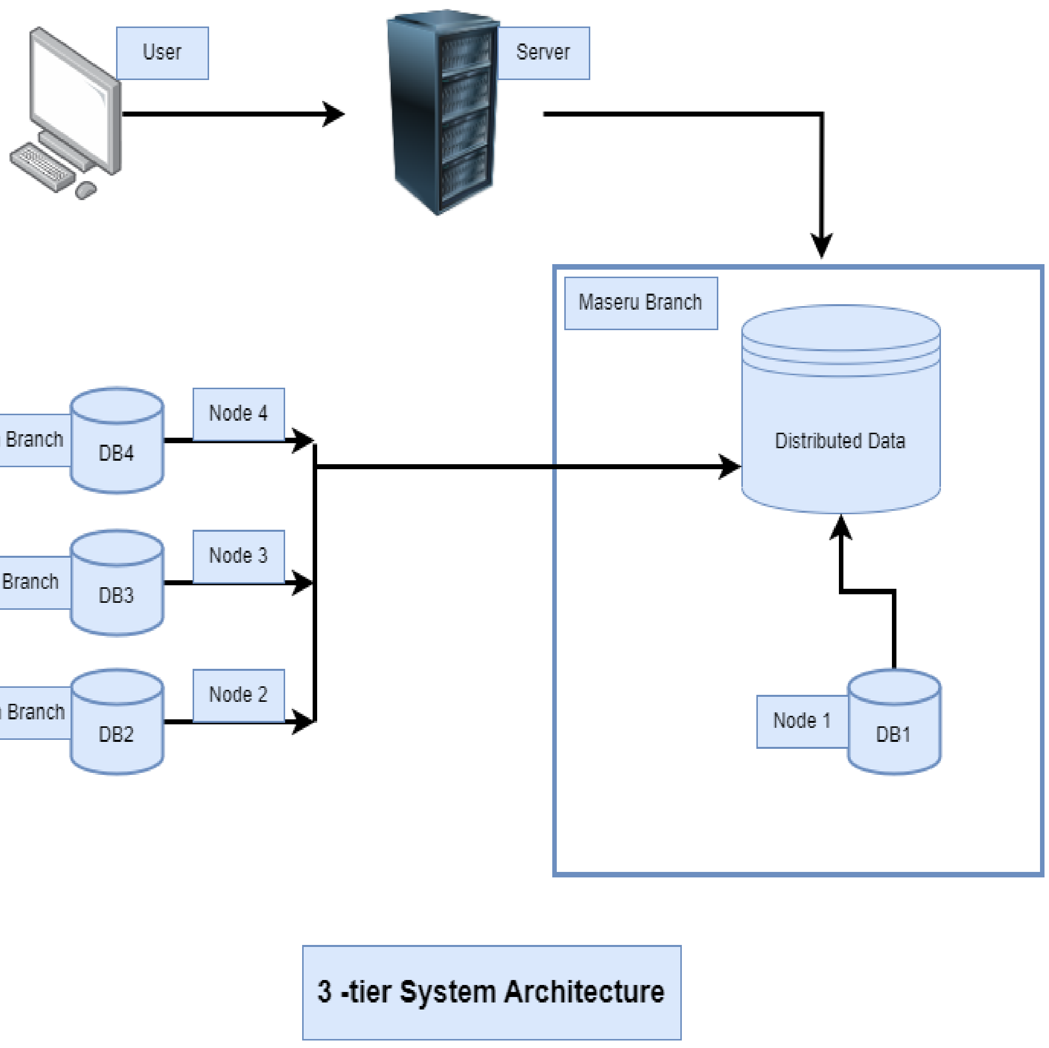


Figure 3: 3-tier system architecture

In this design, four nodes (1, 2, 3, and 4) are used to store and manage the data. The data is partitioned and distributed among the nodes, and each partition has multiple replicas distributed across the nodes. GCS is used as the storage solution, providing scalable and durable object storage.

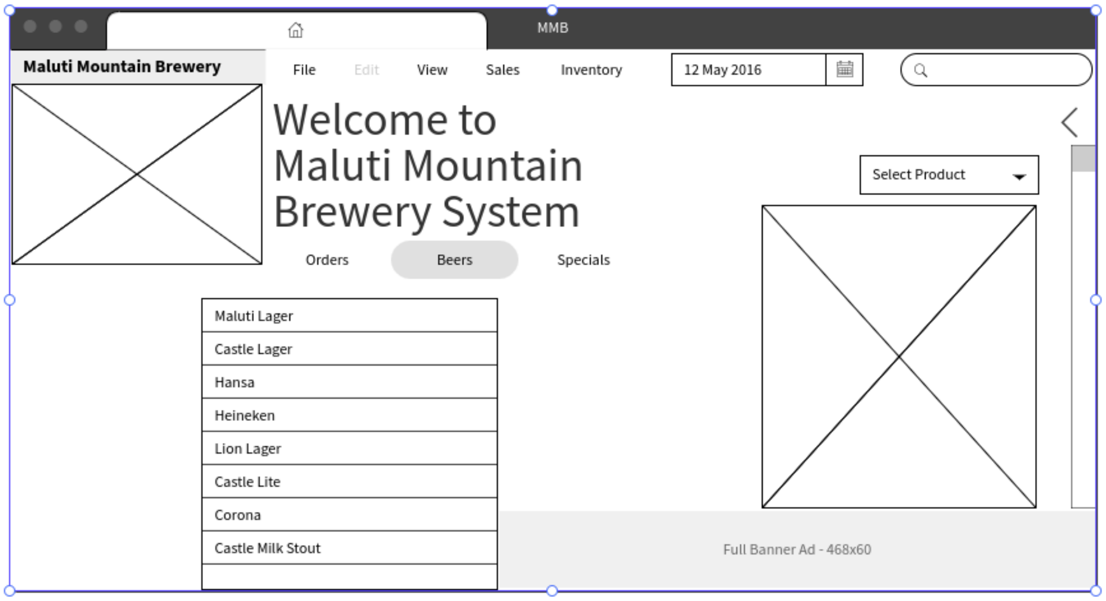
The chosen data partitioning scheme is hash partitioning, which ensures even distribution of data and avoids data hotspots. The chosen replication scheme is a quorum-based replication, which ensures high availability and fault tolerance.

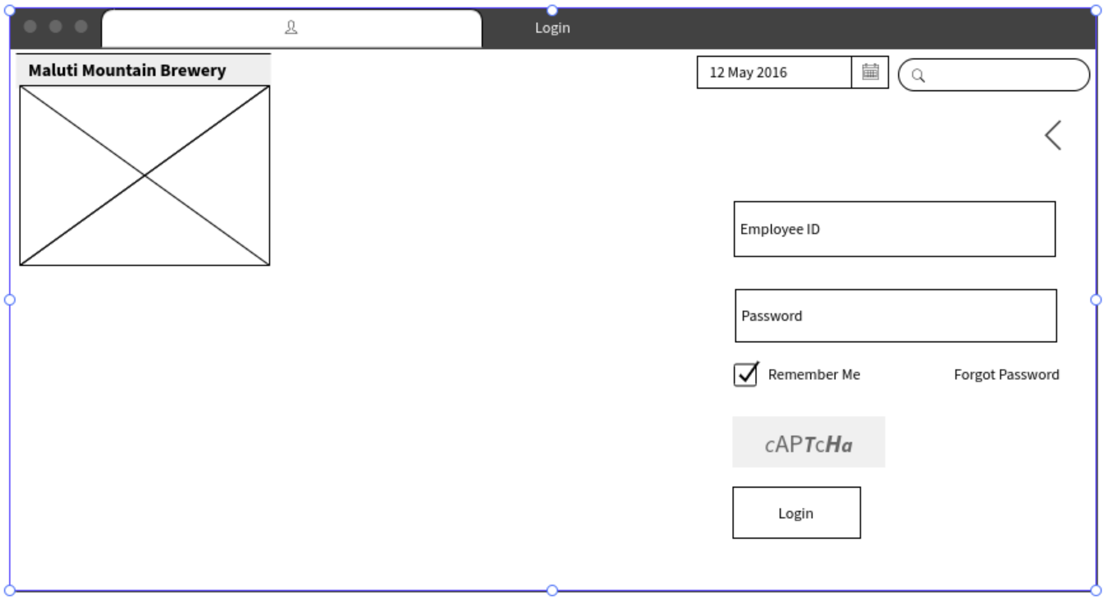
Distributed database design for the brewery distribution system is designed to provide high availability, fault tolerance, and scalability. By using GCS, data partitioning, and replication, this design ensures data availability, consistency, and durability.

4. UI/UX DESIGN

Wireframe Diagram: A wireframe diagram is a visual representation of the layout and functionality of a user interface design. The following wireframe diagram illustrates the design for the brewery distributed database system for distribution:

The following is the home page





The above is the login page for the system

The wireframe design follows a simple and intuitive layout, with a navigation bar, search bar, table of data, and footer bar. The navigation bar provides easy access to different sections of the application, such as inventory management, sales, and distribution. The search bar allows users to search for specific data, such as a particular product or customer. The table of data displays the relevant information in a clear and organized manner, allowing users to quickly analyze and manage the data. The footer bar provides additional information, such as contact information and legal notices.

In conclusion, the wireframe diagram for the brewery distributed database system for distribution is designed to provide a simple and intuitive user interface, allowing users to easily navigate, search, and manage the data.

5. TEST PLAN

The following outlines the tests that will be run at the end of the implementation to evaluate the quality of the brewery distributed database system for distribution:

- ***Functionality Testing***
  - Test the functionality of each feature, such as inventory management, sales, and distribution.
  - Verify that each feature works as expected, and that data is accurately stored and retrieved from the database.
  - Ensure that the system meets the functional requirements outlined in the project scope.

- ***Performance Testing***
  - Test the performance of the system under different loads, such as peak hours or high traffic.
  - Monitor the response time of the system and ensure that it is within acceptable limits.
  - Test the system's scalability by increasing the number of nodes in the distributed database.

- ***Security Testing***
  - Test the security of the system by attempting to hack into the system or exploit vulnerabilities.
  - Verify that data is securely stored and transmitted, and that user authentication and access control are properly implemented.
  - Ensure that the system meets security requirements outlined in the project scope.

- ***Usability Testing***
  - Test the usability of the system by observing users performing tasks, such as adding new products or managing inventory.
  - Evaluate the ease of use of the interface, and ensure that the system is intuitive and easy to navigate.
  - Obtain feedback from users on the design and functionality of the system.

- ***Integration Testing***
    - Test the integration of different components of the system, such as the front-end and back-end.
    - Ensure that the different components work seamlessly together, and that data is accurately transmitted between them.
    - Verify that the system meets integration requirements outlined in the project scope.

Test Plan Summary Table:

The following table summarizes the tests that will be run to evaluate the quality of the brewery distributed database system for distribution:

Test Type	***Description***
Functionality	Test the functionality of each feature
Performance	Test the performance of the system under different loads
Security	Test the security of the system
Usability	Test the usability of the system
Integration	Test the integration of different components of the system

The test plan for the brewery distributed database system for distribution outlines the tests that will be run to evaluate the quality of the system. These tests cover various aspects of the system, including functionality, performance, security, usability, and integration. By testing the system in these areas, the quality of the implementation can be evaluated and improved.

Bibliography

Food and beverage. (2023). Retrieved 2023, from Rockwell Automation: https://www.rockwellautomation.com/en-us/industries/food-beverage/brewery-production-automation.html

Barton, D. (2021). Opening a Microbrewery. *From Brewmaster to Profitable Brewery Operations Management*.

BEGG, C. E., & Connoly, T. M. (2014). Database Systems, A Practical Approach to Design, Implementation, and Managemen. *Introduction to Distributed Database Management Systems*.

Burns, B. (2018). *Designing Distributed Systems.* Sebastopol: O'Reilly Media, Inc.

Fadoua, H., & Grissa , T. A. (n.d.). Intelligent Implementation Processor Design for Oracle. *Distributed Databases System*, 12.

Fogel, S. (2010). *Oracle Database Administrator's Guide.* Oracle.

Howley, E., Jim , D., & Hongliang , L. (2014). A Coevolutionary Approach to the Beer Game. *Individual Versus Group Rationality*, 19.

Jayashree, J. (2013). Distributed Database Management System and Query Processing. *International Journal of Computer Science And Technology*, 5.

M, L., & R, E. (2012). On the Move to Meaningful Internet Systems. *Information Infrastructures for Utilities Management in the Brewing Industr*.

MAGAZINE, A. O. (2022). *Brewering at its peak(Maluti Mountain Brewery).* Maseru Lesotho: AFRICA OUTLOOK MAGAZINE.

Melanie. (2023). The Beer Supply Chain in 2023 and Beyond. *unleashed inventory management software*.

Muriu, S. (2019). *studocu.* Retrieved from https://www.studocu.com/row/document/chuka-university/computer-science/database-project-proposal/5736303

Ozsu, T. M., & Valduries, P. (2011). *Principle of Distributed Database Systems.* London: Pearson Edufcation, Inc.

Winnie, N. (2011). *A public monitoring and evaluation web-application for the breweries.* Kampala: Kampala International University.