



**CSCI 5408 – Data Management, Warehousing, Analytics**

**Feasibility Study Report**

**D2\_DB**

**Group 12**

Guturu Rama Mohan Vishnu (B00871849)

Sharad Kumar (B00889585)

Kalpiti Machhi (B00911364)

Kavya Kasaraneni (B0088124)

Kishan Mahendrabhai Savaliya (B00896729)

## Introduction

The purpose of this project is to build a Distributed Data Management System which operates on two different Virtual Linux instances in GCP (Google Cloud Platform). This Distributed Data Management System should be able to store and handle the data efficiently without any data loss. As the storage is built-in, we must create a command-line interface along with backend DBMS. This system should be able to handle different SQL queries from two different users at the same time.

## Team Members & Their Strengths

Member	Strengths
Rama Mohan Vishnu Guturu	MySQL, Java, Data Structures
Kalpiti Machhi	Data Structure, Python, MySQL
Kishan Savaliya	Python, MySQL, Critical Thinking
Kavya Kasaraneni	MySQL, Java, Python
Sharad Kumar	MySQL, SQLite, Java, C#, Data Structures, Hands on experience on multiple projects

## Learning Involved

GCP:

- We will have to explore GCP to work on the project components which require remote operations on databases. Few initial learning involves creating virtual instances, hands on experience on cloud shell commands.

Java:

- As a team we must get used to advanced data structures and algorithms to work on the project. The implementations will be done in java language and would require proficiency in it as well. Major learning in Java language involves spring boot and socket programming

DBMS:

- We would have to keep learning DBMS deeply and must parallelly implement it in our project so that we can achieve our tasks. It is also important to understand the implementation and internal working of multiple databases.

## Challenges and Solutions

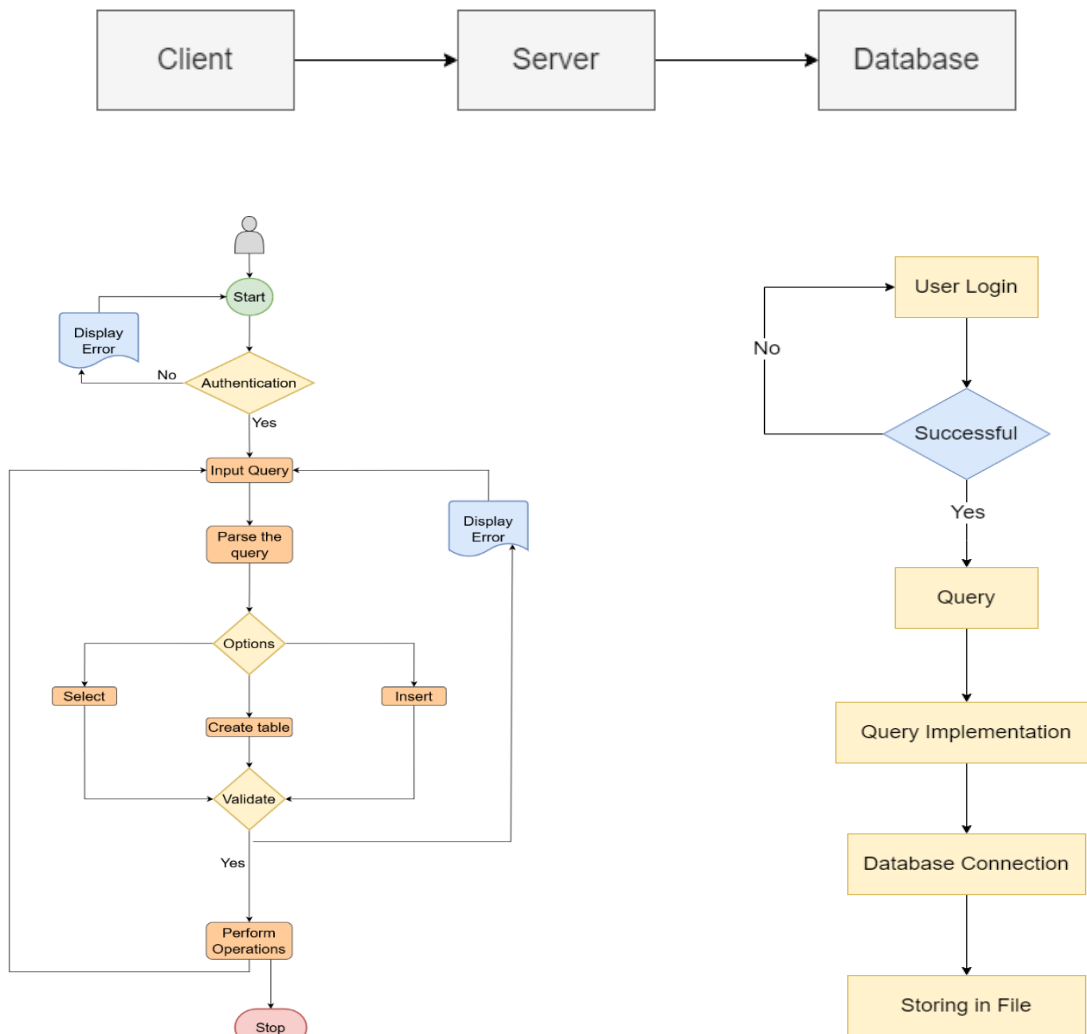
1. As not everyone is good with java in the team, as a group developing the project would take an ample amount of time: We have planned to have group sessions, one session per week which can help broaden everyone's knowledge.

2. Implementation of multiple databases: Each of us agreed to have parallel learning on different databases and have discussions in a team so that everyone can have a clear idea of each database.
3. Storage file format type: For now, our team has decided to go with .txt file as the data storage file type for our project.
4. Query Implementation: For query, we will use a customized “tag” to retrieve information from the database and store it into the .txt file.

## Architecture

Here we are using the 3-tier architecture where we are separating the user-interface and the database i.e., we have a server (GCP) which will be acting as a bridge between the client and database.

We are using it to improve the program-data dependency and support multiple views of data.



## Development Environment

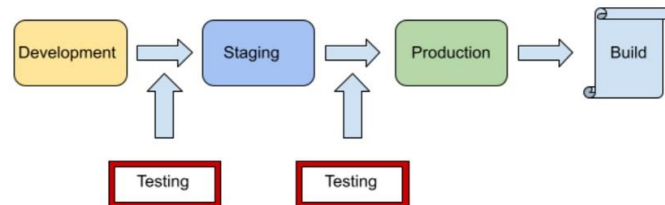
We are using IntelliJ IDE as our development environment, since it is very efficient which enables a very smooth development workflow experience, from design, implementation, building, deploying, testing, and debugging, to refactoring. The version of IntelliJ which we are going to use is 2021.3.1 which is the latest version. There is an alternative for this, is to use Visual Studio Code.

## Testing Strategy

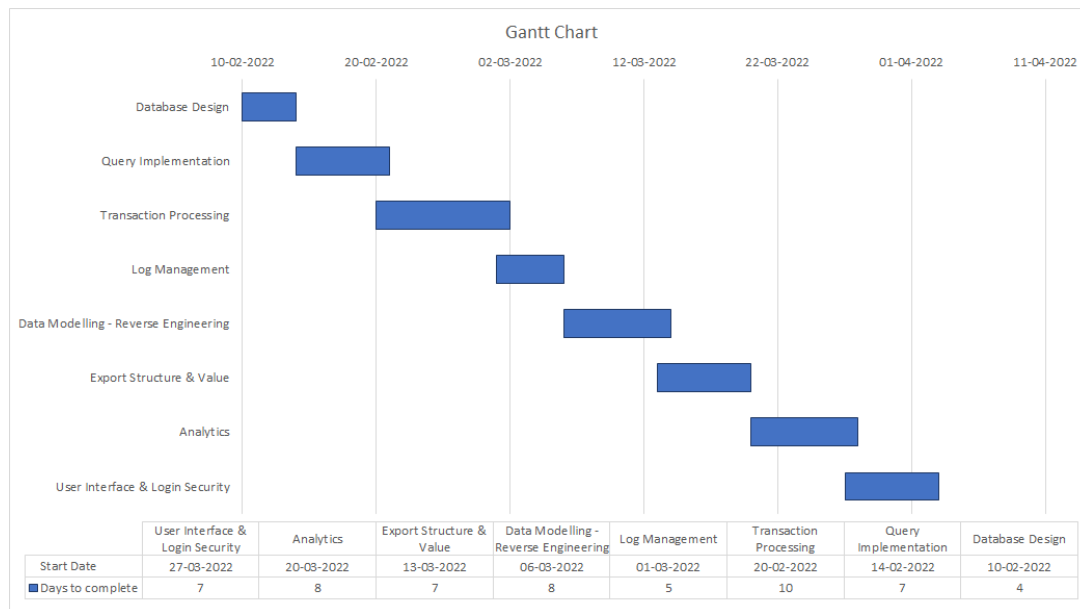
We will be following Test Driven Development (TDD) approach to develop the application. This process specifies that we write the test cases first and then begin the development for the tests written. At first, it is expected the tests will fail and then a new code is written to pass the test. This makes sure that the code is simple and bug-free.

We will be using the following types of tests:

- Unit Test: This test method will be used during the feature development. All team members will be writing the unit test cases for the features that they develop
- Integration Test: We will be performing an integration test when major feature modules will be merged to Staging Environment or Production Environment. Having an integration test will ensure that all the modules work fine independently and work as intended after integration with other modules.



## Development Phases



We believe the project is achievable in the given duration. It is divided into 8 development phases and the expected delivery of each phase is shown in the above gantt chart.

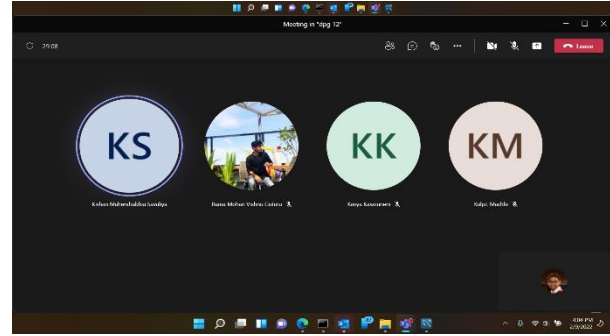
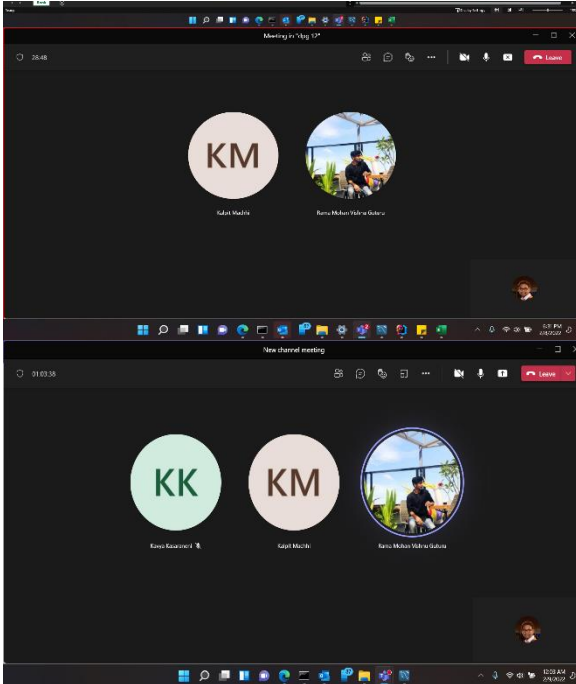
### Preliminary background research:

-- In order to get better understanding of Distributed database management systems, we referred some research papers. Those all papers provided the basic intuition of DDBMS designs and its implementation. Additionally, there were some issues mentioned in paper such as, transaction processing, handling multicopy updates, concurrency control and crash recovery, and also how to handle some of those issues.

-- There are a lot more advantages of distributed data management system. Database contains the definition of logical subnetworks. This also depicts the authorization model used to make logical subnetworks in distributed systems. By the use of distributed system, we can handle data from different geolocation and the main benefit of it is that, if there is a server crash at one location, it will not affect the data accessibility at another location.

### Meeting Logs

- Discussed about the common interests and strengths that will be provided in feasibility report.
- Updated document with names of team member and respective strengths
- Discussed about the various aspects of the requirements as suggested by Professor.
- Finalized the topics and content to be placed in feasibility report and updated the doc to present it



Meeting 1 (8/2/22)

Members : Kalpit,  
GSM Vishnu, Sharad

Meeting 2 (9/2/22)

Members : Kavya,  
Kalpit, GSM Vishnu,  
Sharad

Meeting 3 (9/2/22)

Members : Kavya,  
Kalpit, GSM Vishnu,  
Kishan, Sharad

## References:

- [1] "Database Architecture in DBMS: 1-Tier, 2-Tier and 3-Tier", *Guru99*, 2022. [Online]. Available: <https://www.guru99.com/dbms-architecture.html#:~:text=A%20Database%20Architecture%20is%20a,changed%2C%20replaced%2C%20and%20altered.> [Accessed: 7 February , 2022].
- [2] H. Stockinger, "Distributed Database Management Systems and the Data Grid," 2001 Eighteenth IEEE Symposium on Mass Storage Systems and Technologies, 2001, pp. 1-1, doi: 10.1109/MSS.2001.10003.
- [3] D. Cohen, "Database systems: Implementation of a distributed database management system to support logical subnetworks," in *The Bell System Technical Journal*, vol. 61, no. 9, pp. 2459-2474, Nov. 1982, doi: 10.1002/j.1538-7305.1982.tb03435.x.