DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

CSCI 5408 Data Warehousing

Project Feasibility Study

Submitted By:

Group 14

Chetanpreet Singh Sachdeva (B00853545)

Anqi Chen (B00838586)

INTRODUCTION

The main task of this project is to build a simple database and its management system. A good DBMS should be able to perform certain functions: Data Storage, Data Retrieval, Data Analysis, manage Databases, Generate Logs, Concurrency control, etc.

CHALLENGES

1. Finding the Right Data Structure

We looked at different Data Structures based on their advantages and disadvantages. The main use cases helped us narrow down the scope of our research. As we are planning to work with Java, we have chosen Graphs and Array lists for several reasons:

- Graphs can represent Entities with their vertices and Relationships using their edges.
- Use different java classes to represent each entity, in which variables of that entity can act as attributes. Once we use a class object to represent one set of attributes (like a row of the table), we can use an array list of array list to create an entire table.
- Here the Outer Array list acts as a row number and the inner one holds the class object/entity for one member(row).

We can then use Graphs as a mean to represent the relationship between the different entities (Classes).

2. Identifying different Keys

Every Database holds multiple types of keys in each table. To solve this problem, we can implement certain algorithms that make sure unique values are present in certain attributes.

3. Converting SQL Queries

We know that the user-input will be in an SQL format and for our Database Management System (DBMS) to work with it, we would need some form of conversion.

As of our queries will be simple and straightforward, we can create direct methods for a particular set of queries. For example: For a simple select* form of query, we can create a method that can display the contents of the entire designated table.

4. Handling Different Datatypes

As known, a database will have multiple tables and no table would have a single datatype. So, to counter this issue we decided on using java classes as an entity.

Here, the class member variables would act as attributes. Now if we implement an array list of an object of such entity type, we can then handle multiple data types.

The only problem would be additional traversal might be required if we are working with a particular attribute.

5. Generating Logs

Logs will be generated with the project that'll document each action of a user. Logs are useful when it comes to debugging and testing.

Java offers us customization in terms of logs and hopefully, we'll be able to implement the different types of logs we're asked to create.

6. Handle Multi-User Request

A Database Management System (DBMS) should be prepared to face multiple users at once. The challenge we're facing is to implement Transaction Management Concepts in Java. We also need to implement a Data Dictionary which keeps checking on the authorities of each user.

Assumptions

1. We assume that the user will not use commands like join, group by, order by, and instead use simple commands to display data (Select), Insert data, update data, or Delete Data.
2. As of now, we're assuming that the program data will be stored locally on our machine.

ALGORITHMS

We've identified the need for a few Algorithms:

1. For checking if an attribute has a unique value in an Array list

a. Select the Array List.

b. Select the Attribute that you want to Check.

c. Traverse the Array list through the attribute of that entity.

d. If a duplicate entry is found, return fails.

e. If not, continue traversal.

f. Return Success if no duplicate entry is found.

2. Conversion of SQL query to method

g. Select the First word of Query. Assume it's select*. Store it temporarily.

h. Now select the word following 'from' so we know which table are we talking about. Store this too.

i. Now go to the method that displays all records and pass in the table name for reference.

j. Traverse the entire table and display records to the user.

FLOWCHART

The flowchart below (Figure 1) shows the sequential steps of a user interacting with our DBMS, which presents the workflow of complete processes.

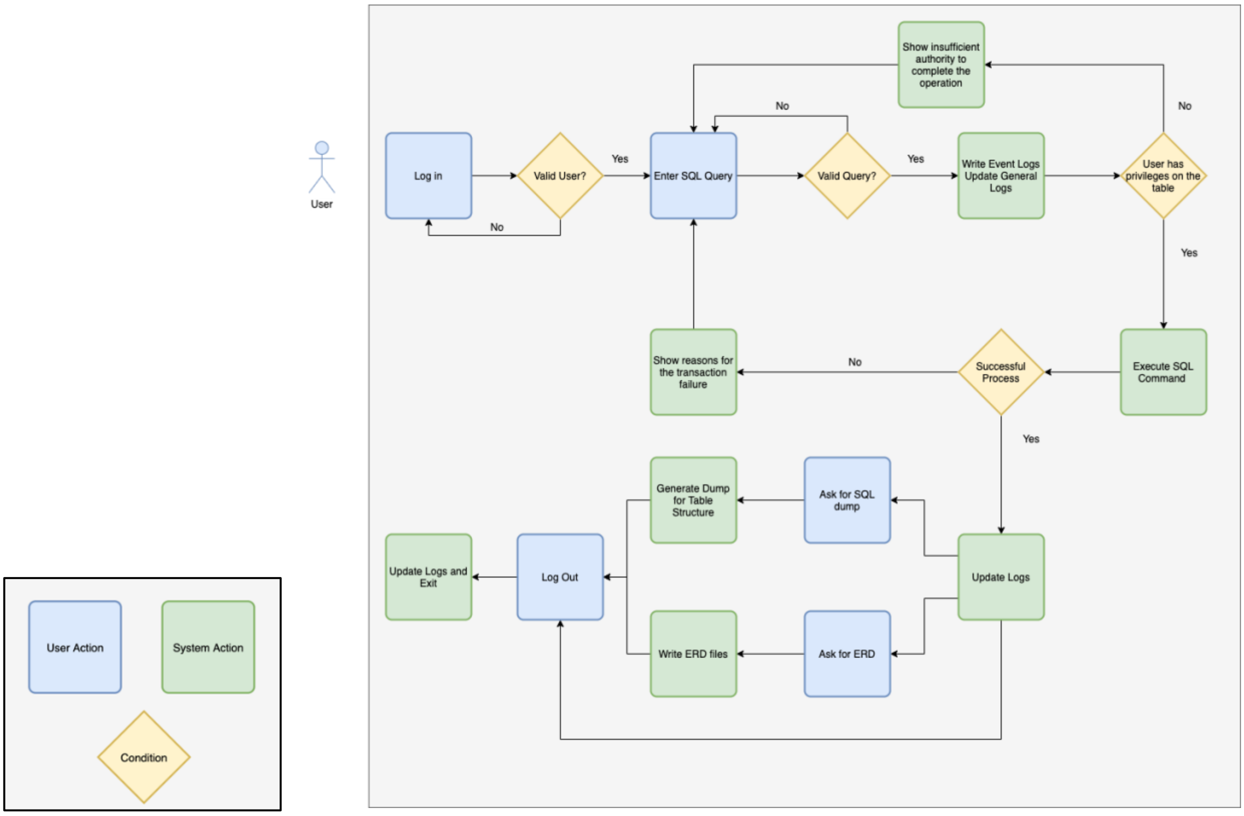
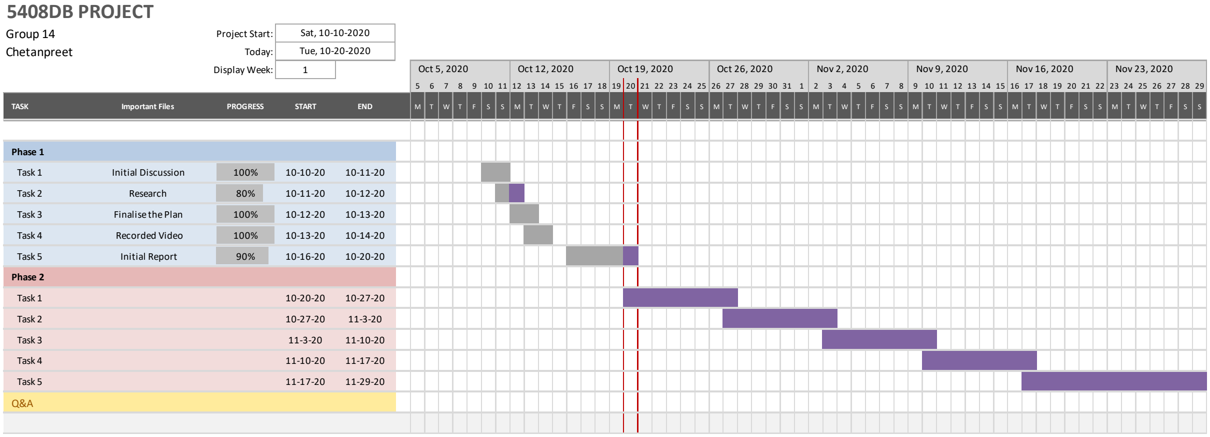


Figure 1 Flowchart of User Actions and System Responses were drawn using draw.io[1]

TIMELINE

We use MS Teams for collaboration. After several regular meetings, we determined the initial plan of the project and set a rough time plan, as shown below (Figure 2).



5408DB PROJECT

Group 14
Chetanpreet

Project Start: Sat, 10-10-2020
Today: Tue, 10-20-2020
Display Week: 1

TASK	Important Files	PROGRESS	START	END
Phase 1				
Task 1	Initial Discussion	100%	10-10-20	10-11-20
Task 2	Research	80%	10-11-20	10-12-20
Task 3	Finalise the Plan	100%	10-12-20	10-13-20
Task 4	Recorded Video	100%	10-13-20	10-14-20
Task 5	Initial Report	90%	10-16-20	10-20-20
Phase 2				
Task 1			10-20-20	10-27-20
Task 2			10-27-20	11-3-20
Task 3			11-3-20	11-10-20
Task 4			11-10-20	11-17-20
Task 5			11-17-20	11-29-20
Q&A				

Figure 2 Simple Gantt Chart to Show Project Timeline[2]

REFERENCE

[1] “diagrams.net.” https://app.diagrams.net/ [Accessed Oct. 20, 2020].

[2] “Simple Gantt Chart,” Vertex42.com. https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html [Accessed Oct. 14, 2020].