**USE CASE STUDY REPORT  
Group No.: 35  
Student Names:** Aishwarya Belakavadi Subrahmanya & Krithika Annaswamy Kannan

**Executive Summary**

The core objective of this project centered around creating an effective volunteer matchmaking platform that connects individuals offering their skills with organizations seeking specific expertise. Recognizing the inefficiencies in the traditional volunteer recruitment process, the emphasis was on developing a robust database using MongoDB, optimizing the matching process, and addressing the challenges encountered by both volunteers and organizations.

The platform aimed to streamline the process of finding suitable volunteers for organizations and vice versa. Leveraging MongoDB's flexible schema and document-based structure, the database was designed to minimize data duplication, enhance search functionalities, and facilitate efficient reuse of volunteer skills. This approach significantly reduced the time and resources required for volunteer matching.

The database architecture was shaped by input from volunteers and organizations, incorporating skill requirements, preferences, and organizational needs. Through Entity-Relationship (EER) and Unified Modeling Language (UML) diagrams, the conceptual model was transformed into a MongoDB-compatible schema, ensuring data integrity, and enabling agile querying.

In addition to the relational database, the project explored the feasibility of leveraging MongoDB's NoSQL capabilities. By utilizing MongoDB, queries were formulated to efficiently retrieve and manipulate data, showcasing the potential of NoSQL databases for optimizing volunteer-organization matches.

The system demonstrated success in facilitating efficient volunteer-organization pairings. Leveraging Python, the database showcased powerful analytics capabilities, providing valuable insights into volunteer trends, skill utilization, and organizational requirements. These analytics proved beneficial in identifying and deploying skilled volunteers to address urgent needs.

Leveraging continuous data analysis and improvement strategies, the volunteer matchmaking platform holds the potential for ongoing enhancements. By harnessing the insights gleaned from the analytics capabilities embedded within the system, iterative improvements can be made to the matching algorithms, volunteer engagement strategies, and organizational outreach methods. This iterative approach, guided by data-driven decision-making, offers the opportunity to refine and optimize the volunteer-organization pairing process continually.

1. **INTRODUCTION**

Volunteer matchmaking embodies a dynamic process of connecting individuals willing to contribute their skills and time with organizations seeking those specific talents to fulfill their missions. At its core, effective volunteer matchmaking relies on efficient data management strategies. By structuring and organizing information effectively, databases equipped with relational models become instrumental in facilitating seamless connections between volunteers and organizations.

Relational models, a key component of data management, provide a structured framework to represent relationships between different entities involved in volunteer matchmaking. These models delineate the attributes and interactions between volunteers, their skill sets, and the diverse needs of organizations. Through tables, keys, and relationships, a relational model organizes data in a manner that allows for precise querying and matching.

For instance, consider a relational database where volunteers' skills, availability, and preferences are stored as attributes within designated tables. Concurrently, organizations' requirements, projects, and expected skill sets are represented in separate tables. By establishing relationships between these entities based on skill compatibility and other criteria, relational models enable efficient querying, allowing the system to identify suitable matches between volunteers and organizations.

The strength of relational models lies in their ability to handle complex queries and ensure data integrity. Using SQL (Structured Query Language) in conjunction with these models enables sophisticated querying capabilities, empowering the system to retrieve precise information matching volunteer skills with organizational needs. This integration streamlines the matchmaking process, facilitating timely and accurate connections between volunteers and organizations.

Using a relational model within data management strategies for volunteer matchmaking enhances the system's capacity to efficiently pair volunteers' skills with the specific requirements of organizations. The structured approach of relational databases optimizes the matching process, fostering meaningful collaborations and maximizing the impact of volunteer efforts within diverse organizational settings.

1. **Conceptual Data Modeling**

**1. EER Diagram**

A black background with white squares

Description automatically generated

**2. UML Diagram**

A diagram of a company

Description automatically generated

1. **Mapping Conceptual Model to Relational Model**

**Primary Key- Underlined Foreign Key- Italicized**

**USER (ADDRESS, USER\_ID, USERNAME, EMAIL, PH\_NO)**

Primary key: User\_id  
Foreign key: NA

**VOLUNTEER ( VOL\_ID, AVAILABILITY)**

Primary key: Vol\_id  
Foreign key: NA

**SKILLS (SKILLS\_ID,SKILLS\_NAME, *PROFICIENCY\_LEVEL*)**

Primary key: Skills\_id  
 Foreign key: Proficiency\_level(foreign key from proficiency table as prof\_id ; null is allowed

**ORGANISATION(ORG\_ID,ORG\_NAME,CONTACT)**

Primary key: Org\_id  
Foreign key: NA

**NOTIFICATION(NOTIF\_ID,NOTIF\_TEXT,PRIORITY)**

Primary key: Notif\_id  
Foreign key: NA

**FEEDBACK(FEED\_ID,FEED\_TEXT,RATING)**

Primary key: Feed\_id  
Foreign key: NA

**COMMUNICATION(MSG\_ID,*SENDER\_ID,RECEIVER*\_IDCONTENT,TIME\_STAMP)**

Primary key:msg\_id  
Foreign key: sender\_id, reciever\_id(foreign key to user id in user table) ; Null allowed

**APPLICATION (APPLICATION\_ID, APPLICATION\_DATE, STATUS, *USER\_ID, OPPORTUNITY\_ID*)**

Primary key: application\_id  
Foreign key: user\_id (foreign key to user id in user table); Null not allowed and opportunity\_id; Null not allowed.

**OPPORTUNITY(OPP\_ID,TITLE,DESCRIPTION,*ORG\_ID*,LOC ATION,TIME\_PERIOD,SKILLS\_REQUIRED)**

Primary key: opp\_id

Foreign Key: org\_id; Null not allowed.

**LOCATION (LOCATION\_ID, CITY, STATE,COUNTRY)**

Primary key: location\_id Foreign key: NA

**TIME\_PERIOD (TIME\_ID, START\_DATE, END\_DATE)**

Primary Key: time\_id  
Foreign key: NA

**VOLUNTEER\_HOURS (ENGAGEMENT\_ID, HOURS\_WORKED)**  
Primary Key: engagement\_id  
Foreign key: NA

**DATA\_ANALYSIS\_AND\_IMPROVEMENT (ANALYSIS\_ID, DATA\_SOURCES,A\_RESULT)**

Primary Key: Analysis\_id  
Foreign key: NA

**MATCH\_MAKING\_ALGORITHM(MM\_ID,ALGORITHM \_NAME,*RESULT*)**Primary Key: mm\_id  
Foreign key: result (foreign key of opp\_id from opportunity table ; Null is not allowed) page3image54045456page3image54045664page3image54045872

**PROFICIENCY (PROF\_ID, PROF\_TYPE, PROF\_SCALE)**

Primary Key: prof\_id  
Foreign key: NA

1. **Implementation of Relation Model via MySQL and NoSQL  
   MySQL Implementation**
2. **Query retrieves information about opportunities where there is a match in the MATCH\_MAKING\_ALGORITHM table.**

**A screenshot of a computer

Description automatically generated**

1. **Query that retrieves communication records along with the sender's and receiver's details for a specific application.**

A screenshot of a computer

Description automatically generated

1. **Query retrieves opportunities related to the programming skill.**

A screenshot of a computer

Description automatically generated

1. **Query to retrieve users who have not applied to opportunities in SUBURBIA.**

A screenshot of a computer

Description automatically generated

1. **Query that retrieves details about opportunities based on certain conditions like start\_date.**

A screenshot of a computer

Description automatically generated

1. **hours worked exceed the average hours worked across all engagements.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **This query retrieves the organization names and the count of applications for each organization.**

A screenshot of a computer

Description automatically generated

1. **Query to retrieve usernames whose application status is pending**

**NoSQL Implementation**

1. Find all the organizations that are based in Boston, MA

A computer screen shot

Description automatically generated

1. Calculate the average proficiency and count the number of volunteers for each proficiency type.

A screenshot of a computer

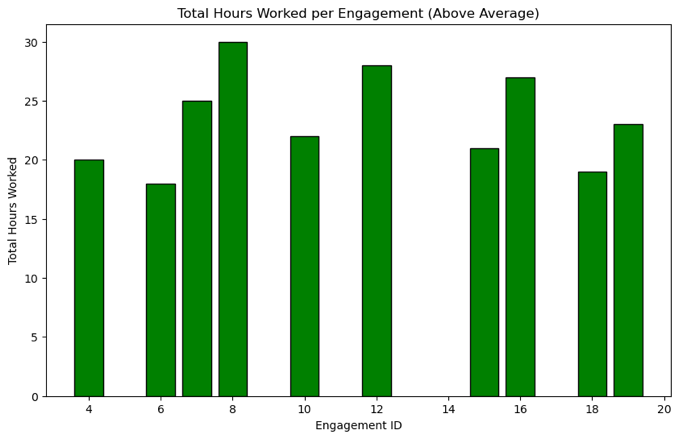
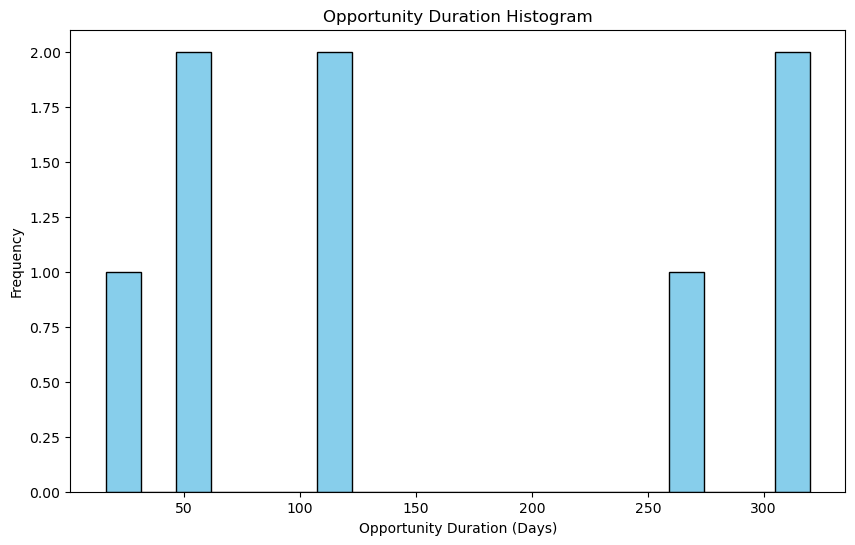
Description automatically generated

1. Count total feedback given by user and the average rating

A screenshot of a computer

Description automatically generated

**V. Database Access via Python**  
The database is accessed using Python and visualization of analyzed data is shown below.  
The connection of MySQL to Python is done using mysql.connecto



A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

**VII. Summary and Recommendation**

**The proposed relational database aims to simplify volunteer matchmaking by centralizing information, reducing redundant paperwork for organizations, and streamlining the connection process. A phased approach, starting with a user-friendly interface for volunteer registration and organization access, followed by pilot testing and iterative improvements based on feedback, is recommended for successful implementation.**

**To implement a phased approach starting with a user-friendly interface for volunteer and organization access, followed by pilot testing and iterative improvements based on feedback. Integrate features like background checks and activity tracking in subsequent phases. Collaborate closely with stakeholders for continuous refinement and ensure alignment with user needs for a successful implementation with enhanced efficiency in volunteer engagement.**