Assignment: Module 6 Final Project Report

Nicholas Hooper; Amit Patel; Ganya Reddy; Trusha Sonawane; Yash Singh
09 DEC 2024
Northeastern University: College of Professional Studies

Introduction

In medical emergencies, the moments immediately following an incident are often the most critical, with the speed of action playing a significant role in determining the outcome. Unfortunately, many communities lack effective systems to bridge this crucial gap before professional help arrives. To address this challenge, we are developing the Community Health Support and Emergency Response App—a platform that connects trained local volunteers with individuals in urgent need of assistance. This app empowers communities to act swiftly by allowing users to report emergencies and notifying certified volunteers nearby who can provide aid until professional responders are on the scene.

Our app is specifically designed for situations where rapid intervention is essential, such as cardiac arrests, accidents, or other medical crises. By leveraging real-time location tracking and instant notifications, the platform ensures seamless coordination and communication between those in need and nearby responders. It serves both general users and trained volunteers, providing a free and accessible way to report emergencies while offering a premium subscription for organizations and advanced users to access additional features like detailed analytics.

Imagine this: Someone you love is in immediate danger. Every second counts. Wouldn't it be incredible if help was just a tap away?

That's the vision behind our Community Health Support and Emergency Response App. In too many communities, there's a critical gap between when an emergency strikes and when professional help arrives. This precious time can mean the difference between life and death. Our app is designed to bridge that gap by connecting people who need help with trained volunteers in their area.

We're passionate about creating a tool that empowers communities to respond to emergencies effectively. By leveraging the skills and compassion of local volunteers, we can save lives and build stronger, more resilient neighbourhoods.

This project holds deep significance for us as it combines the power of technology with the strength of community networks to address a life-saving need. By tapping into the skills and availability of local volunteers, our app enhances a community's ability to respond effectively to emergencies. We believe this initiative not only has the potential to save lives but also fosters a sense of empowerment and connection among individuals, helping to build stronger and more resilient communities.

Application Overview

The Community Health Support and Emergency Response App is created to address the critical moments during medical emergencies when immediate action can save lives. The app connects people in need of urgent help with trained local volunteers who can step in and provide assistance until professional responders arrive. By using real-time location tracking and instant

notifications, the app ensures a quick and effective response to emergencies such as cardiac arrests, accidents, or other life-threatening situations.

Designed with accessibility and ease of use in mind, the app serves both everyday users and certified volunteers. It offers essential features like reporting emergencies and notifying nearby volunteers at no cost, ensuring everyone can benefit. For organizations and advanced users, premium options with additional tools like detailed analytics are available through a subscription.

This app goes beyond being a simple tool—it's a way to empower communities and strengthen their ability to respond to emergencies. By utilizing the skills and availability of local volunteers, it aims to save lives, build stronger connections within communities, and create a more resilient society.

Business Analysis

User Personas

1. **Patients**

Patients are those people in medical emergencies who need the help of the app to reach out for immediate assistance from trained volunteers until the professional medical responders arrive. Their goals are to ensure timely assistance and reduce response times during critical situations. Patients use the app to:

- Report emergencies by inputting key details such as symptoms, location, and type of help required.
- Track status updates, like the time of arrival of an assigned volunteer and the progress of assistance.
- Communicate directly with the assigned volunteer or dispatch center in case of changes or emergencies.

2. **Volunteers**

Volunteers are trained responders who provide the initial aid during any medical emergency. Volunteers attempt to stabilize the patient and provide basic medical support. Their goals are to ensure timely assistance and reduce response times during critical situations. Volunteers use the app to:

- Receive push notifications of emergencies in their geographic area.
- Review case details, including patient symptoms, priority level, and location before accepting assignments.
- Update case progress, including arrival time, assistance provided, and case resolution status.
- Log any observations or additional details for future use by medical facilities.

3. **Facilities**

Facilities include hospitals, clinics, and emergency medical centers that are responsible for providing advanced medical care. The aim is to manage resources efficiently and coordinate with volunteers and dispatch centers for smooth patient handovers. Facilities use the app to:

- Monitor in real time the incoming cases, including patients' information, location, and expected arrival time.
- Allocate resources like beds, medical staff, and equipment based on the priority of the cases and volunteer inputs.
- Communicate with dispatch and volunteers to optimize response times and patient care.
- Log case outcomes, including treatment provided, for future analytics and reporting.

Business Rules and Logic

The application implements some of the business rules to manage the workflow efficiently while maintaining data with accuracy and security as follows:

1. **Emergency Case Management:**
   - Patients can create an emergency case after authentication.
   - Each case is assigned a unique identifier that links the patient and volunteer and facility records.

2. **Volunteer Assignments:**
   - Notifications are sent to volunteers based on proximity to the location of the emergency.
   - Volunteers are filtered and matched to cases based on their skillsets and availability.
   - A case can only be accepted by one volunteer at a time to avoid conflicts.

3. **Facility Coordination:**
   - Facilities receive automatic notifications when a case escalates and requires their involvement.
   - Cases are prioritized based on the severity of patient symptoms, volunteer feedback, and estimated arrival times.
   - Facilities can update case outcomes and provide feedback for system improvements.

4. **Data Access and Security:**
   - Patients can view their own case details and perform updates on them.
   - Volunteers can view and update case information of the cases that are assigned to them.
   - Facilities have explorer access to case data related to their incoming patients.
   - All sensitive data is encrypted, and access is strictly according to policies.

SQL Support for Business Functionality

SQL is crucial to implement and manage the core functionalities of the application:

1. **Patient Functionality:**
   - SQL queries fetch and display patient-specific information, such as active case details, notifications, and status updates.
   - Stored procedures validate and store the emergency case data entered by the patients.

2. **Volunteer Functionality:**
   - SQL dynamically matches volunteers to cases based on location and skills using optimized query filters.
   - Logs of volunteer actions - arrival time and assistance provided - are stored for future analytics and reporting.

3. **Facility Functionality:**
   - SQL queries pull detailed information on incoming cases, linking patient, volunteer, and case tables.
   - Advanced SQL reports enable facilities to analyze trends and make appropriate resource allocations.

Table Design

The ERD is in Appendix 1 and is thoroughly explained in the architecture section of this report. For this section, the primary structures of the data tables are described, and the primary and secondary keys are illuminated.

Below are the 5 primary tables that are created by the different personas described in the previous section. The primary key is identified by a purple box and the secondary key is a green box.

facilities_column	cases_column	users_column	notification_column	volunteer_column
contact_email	case_id	email	case_id	availability_status
contact_name	emergency_location	first_name	instructions	certifications
contact_number	emergency_type	last_name	notification_id	email
facility_id	response_status	location	timestamp	first_name
location	response_time	medical_conditions	*NULL*	last_name
scope_of_practice	user_id	phone_number	*NULL*	location
NULL	volunteer_id	reporting_history	*NULL*	phone_number
NULL	*NULL*	user_id	*NULL*	volunteer_ID

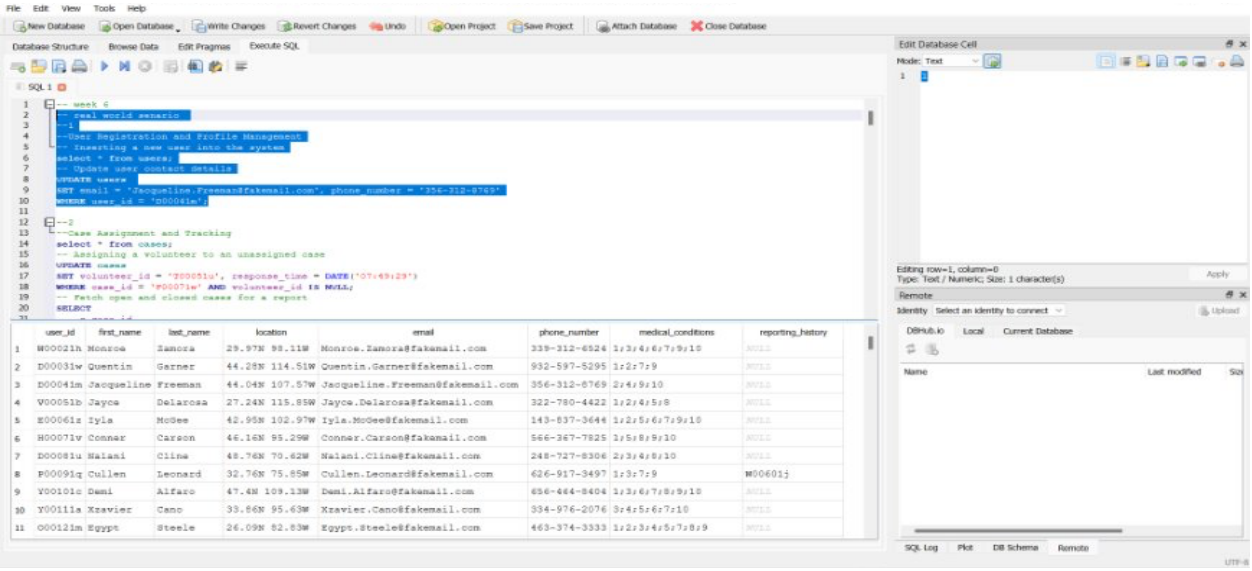
Emergency Event
case_id *[primary key]*
user_id *[foreign key]*
volunteer_id *[foreign key]*
facility_id *[foreign key]*
emergency_location
emergency_type
response_status
response_time

The important piece is how these tables are combined, which is actualized in the emergency event table. The ERD, in appendix 1, shows accurate connection points between entities illustrating one or many relationships. Overall, a single user has connections to many potential personas from the other categories. With this, dispatch amalgamates data and creates the emergency event table by synchronizing the foreign keys and creating a new primary key to track the case and associated data.

Data Implementation & Analytics

1. User Management and New Registrations

Keeping user profiles updated is crucial for managing both new and existing registrations effectively. This ensures that all user information is current and accurate, facilitating better communication and service delivery.



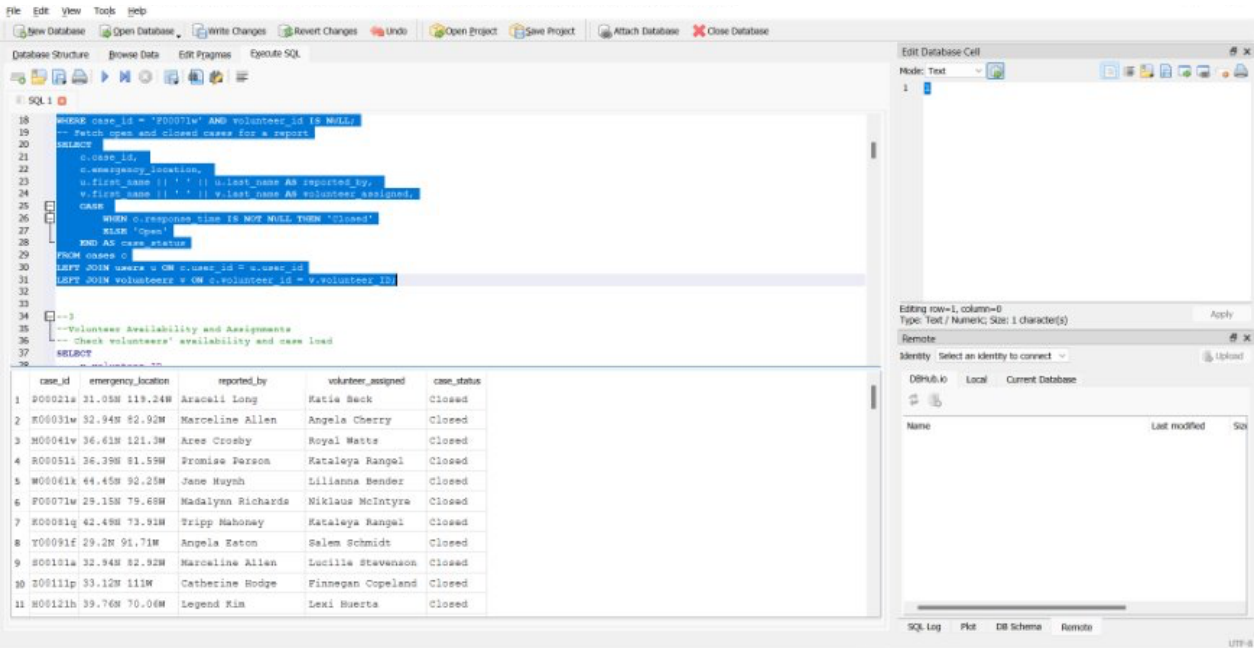
2. Case Management

Assigning the right volunteers to cases that require more attention:

In the emergency response system, volunteers are matched to cases based on their skills and experience. For instance, if a case involves a severe injury, the system assigns a volunteer with advanced medical training to ensure the best possible care. This targeted approach ensures that critical cases receive the attention they need.

Retrieving case history and past cases to identify trends and patterns:

The system maintains a comprehensive database of past emergency cases. By analyzing this data, it can identify recurring trends and patterns. For example, if a particular area frequently reports fire incidents, the system can alert authorities to investigate potential causes and implement preventive measures. This proactive approach helps in mitigating future emergencies and improving overall safety.



3. **Volunteer Management**

Register and Manage Volunteers:

Efficiently register new volunteers and maintain up-to-date profiles for existing ones. This ensures that all volunteer information is accurate and readily accessible.

Track Volunteer Availability and Workload:

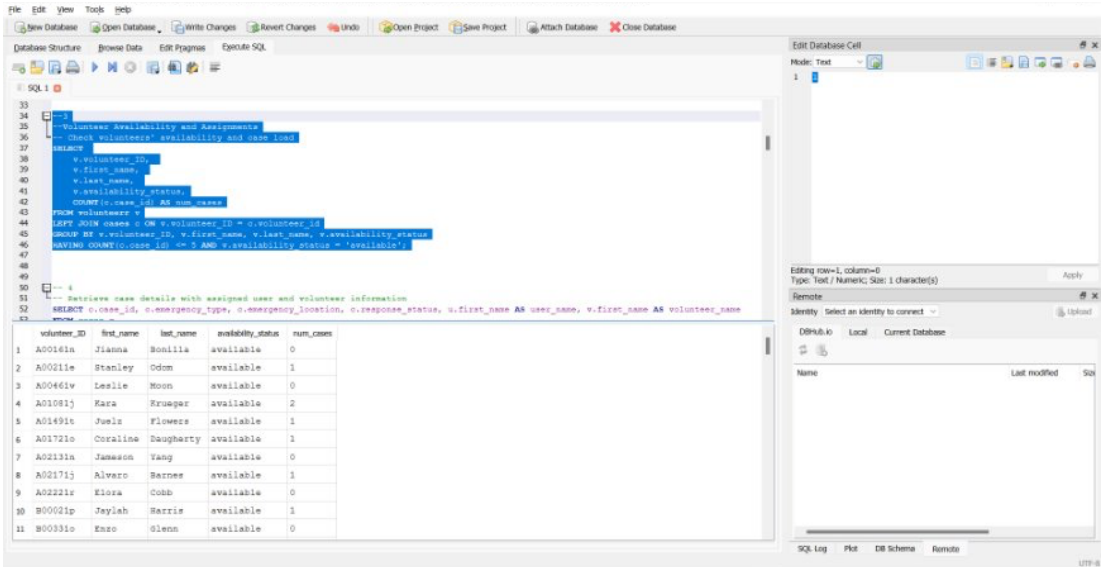
Monitor the availability and workload of volunteers to ensure balanced distribution of tasks. This helps prevent burnout and ensures that volunteers are assigned cases they can handle effectively.

Retrieve Volunteer Assignment History:

Access detailed records of past volunteer assignments to identify trends and patterns. This information can be used to improve future volunteer management and case assignments.

Addressing Volunteer Shortages:

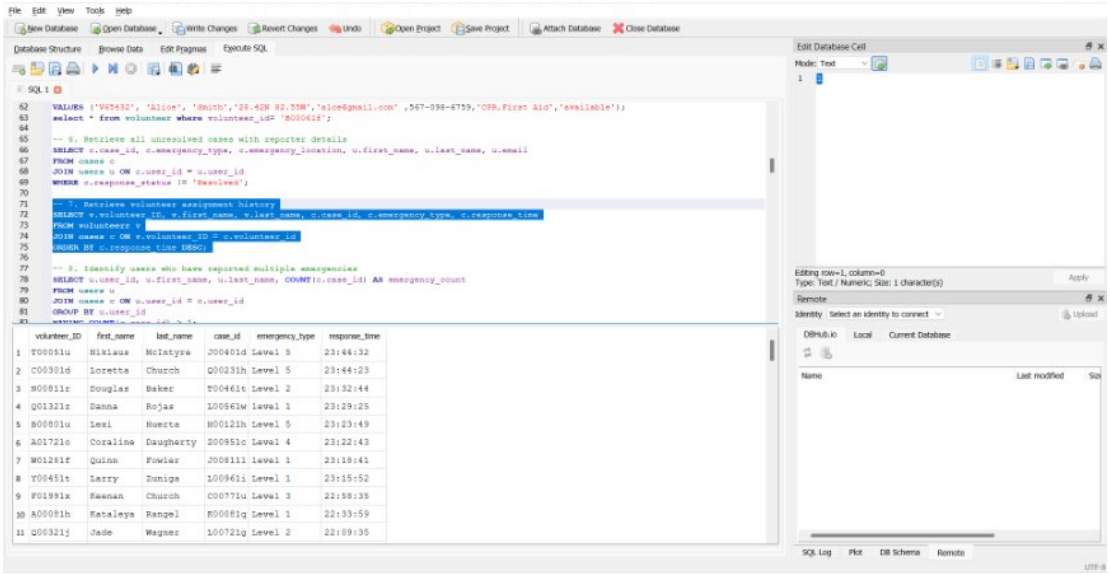
In some situations, the number of cases may exceed the number of available volunteers, leading to a shortage of assistance. This can overwhelm volunteers with the volume of cases they need to manage, potentially resulting in poor decision-making. To mitigate this, it's essential to have a robust system in place for tracking volunteer availability and workload, ensuring that volunteers are not overburdened.



4. **Emergency Analytics**

Emergency analytics involves the following steps: Identify trends and patterns in emergency reports; analyze volunteer performance and workload; analyze location-based emergency data.

Emergency analytics helps analyze and define trends and patterns during emergencies, as well as evaluate volunteer performance. This ensures that volunteers are assigned cases based on their skills and location. For example, consider an emergency notification system where volunteers are assigned randomly based on availability. If an inexperienced volunteer is assigned to a sensitive case requiring advanced first aid, it could lead to inadequate care. To avoid such scenarios, it's better to assign trained volunteers to cases that require more than basic first aid.



5. **Notification and Prioritization**

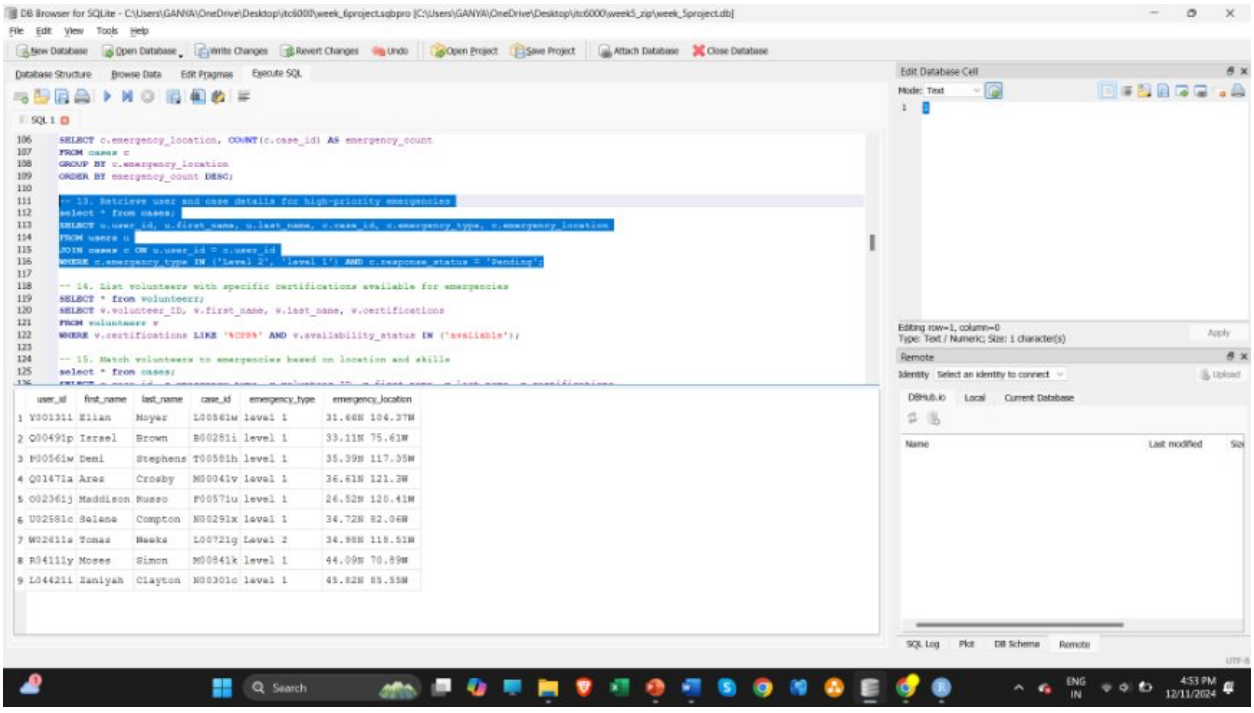
Send notifications to volunteers

Imagine a city's emergency response system that uses a app to notify volunteers. When an emergency occurs, the system sends instant notifications to all available volunteers in the vicinity. For instance, if there's a fire in a residential area, volunteers within a 5-mile radius receive an alert on their phones, detailing the location and nature of the emergency. This ensures that help is dispatched quickly and efficiently.

Prioritize high-priority emergency cases

The same emergency response system categorizes emergencies based on their severity. High-priority cases, such as life-threatening medical emergencies or large-scale disasters, are flagged for immediate attention. For example, if there's a severe car accident with multiple injuries, the system prioritizes this case over less critical incidents, like a minor injury. Volunteers with advanced medical training are notified first, ensuring that the most urgent cases receive the fastest and most appropriate response.

By implementing these features, the emergency response system ensures that volunteers are promptly informed and that critical cases are handled with the highest priority, ultimately improving the overall effectiveness of the emergency response



Security Concerns

Given the nature of health data, security and privacy will be paramount to the architecture.

Access to the primary secure cloud, which is the data lake and data warehouse, will be accessible in 3 levels: Admin, explorer, and read-only. Only dispatch will have admin rights, allowing them to see and manipulate everything within the cloud. The explorer level will allow users to access information that is connected to any relevant primary and secondary keys, e.g., medical staff at a receiving hospital can see information related to historical case_id on an incoming patient based on their user_id key. Finally, read-only allows an entity to only see the information on its primary key, e.g., the data they uploaded themselves.

Access to the secondary cloud is severely restricted. It will house the most sensitive data, data that needs to make it to its destination un-interrupted and un-manipulated. While access and security for the primary cloud are more about securing PPI, the secondary cloud is more concerned with data integrity. This data is critical to the application's business and action portion as it holds all the coordinating instructions between entities. This server will only allow administrative access, and this type of access will be severely restricted.

Architecture

Appendix 1 is the ERD and illustrates the totality of the system architecture. It is a star schema with protocols built in for security, governance, and vertical scalability. The communication hub is the dispatch center. Dispatch holds administrative rights to a primary secure cloud, which is where entity data is amalgamated, and new case files are created. Initially, each entity stores their data on a local device, like their cell phone, and then uploads it to the hosting center. However, before the data can populate the data lake, filters check for quality and governance.

Forcing each entity to interact with a small portion of the data will limit latency and variability issues. It also enables a clean local user interface with the host application handling the backend as a service, BaaS, requirements.

Overall, the dispatch center has many connections in and out, as it handles multiple entities each with its own data mart while also creating case files and coordinating message packages. Notice that the message packages are one and only one illuding to the strict security protocols surrounding these instructions. Also notice that many users can be associated with a single event, allowing for multi-patient events.

Hosting and Storage Requirements

The cloud hosting service fits into two places within the data flow architecture. The first, as previously described, is in the data lake. This will be a relational solution, hosting user, volunteer, and facility data. A relational storage solution fits this use case as the data is large, uniform, and static. The second cloud host will only manage coordinating messaging via a NoSQL solution as this data will be dynamic and communicate with a wide range of recipients and unknown hardware.

Separating the coordinating feature serves two primary functions: data latency and security. The primary data lake will not be inhibited by the changing communication structure and bespoke solutions to coordinating emergencies. The second aspect is the ability to add additional security protocols to all message packages to ensure sensitive data is kept secure and access is restricted.

Growth is an expected aspect of the application. For this use case, large files are not anticipated as a requirement; however, high volume is a potential concern. The afore-mentioned partitioning of locally created content separated by two clouds formatted to meet their unique needs should address both the vertical and horizontal growth in the data warehouse and data mart respectively.

The final storage consideration is indexing. The data is partitioned within the host cloud; however, the data lake is correlated with primary and secondary keys. Upon passing the filter into the data warehouse the keys will be assigned and indexed for faster reference by dispatch. As illustrated in appendix 1, each entity: user, volunteer, event, facility, and message have its own unique primary key, which acts as a secondary key across most tables.

Wrap Up and Future Considerations

The system is designed to streamline emergency response processes, improve efficiency, and provide valuable insights for better decision-making. By effectively managing users, volunteers, and cases, the system aims to enhance community safety and well-being. The core functionalities and objectives of the emergency response system. It emphasizes the importance of efficient user management, case tracking, volunteer coordination, and data-driven insights. By leveraging these features, the system can significantly contribute to timely and effective emergency response.

Executing this project has illustrated the importance of creating a sound data-flow prior to project implementation. Moreover, successfully communicating this plan, including storage, security, and bandwidth requirements to the greater team is critical in establishing a solid foundation for application performance.

References

Coronel, C., & Morris, S. (2016). *Database systems: Design, implementation, and management* (12th ed.). Cengage Learning.

Oracle. (n.d.). *What is a data warehouse?* Retrieved December 4, 2024, from https://www.oracle.com/database/what-is-a-data-warehouse/

Appendix

Appendix 1: