Civil Defense Command and Control Center

Group Members:

Nancy Akoum 202200112

Joy Andraos 202100325

Ahmad Kurdi 201901897

Ahmad Abou El Kheir 201906248

Submitted to Dr. Ramzi Haraty in partial fulfillment of the requirements for the course “CSC 490: Software Engineering.”

1/11/2023

Contents

I. Instructor’s Evaluation 3

II. Report License: 4

III. Preface 5

IV. Introduction: 6

V. Glossary: 7

VI. User Requirements Definition 9

VII. System Requirements Specification: 10

Functional Requirements: 10

Non-functional Requirements: 10

VIII-System Architecture: 12

X. System Models 18

XI. System Evolution 20

Appendix A: Hardware Requirements 22

Appendix B: Database Requirements 24

XIII. Index 25

I. Instructor's Evaluation

Dear Dr. Haraty,

Thank you for taking the time to fill out our evaluation feedback form, we take feedback extremely seriously. Feel free to write any comments you wish.

Grade:

PDF Report: 1 2 3 4 5

Likes:

-
-

Dislikes:

-
-

General Comments:

-
-

Future Recommendations:

-
-

Yours truly,

II. Report License:

III. Preface

This document is tailored to lay the foundations of the Civil Defense’s Command and Control Center system. It covers the overall functional and non-functional requirements from general to specific details, as well as the system's architecture to be developed, which needs mutual agreement from clients and producers. Another primary purpose is to offer software architects, developers, project managers, system administrators, emergency management professionals, and stakeholders associated with civil defense agencies the features of implementations along with its constraints, through all its production phases to the maintenance phase.

Version History:

Version 1.0 (Date: 22/10/2023):

This version of the document marks the beginning of the Civil Defense and Control Center project. The creation of this version stems from the project’s initiation, reflecting the team’s commitment to defining clear objectives and requirements for the system.

Version 1.1(Date: 1/11/2023):

In this version, modifications were made to the user and system requirements based on feedback received from Dr. Ramzi Haraty. Additionally, an index was added for better document navigation. The system architecture was detailed as well, outlining the components, modules, and their interactions.

IV. Introduction:

The Lebanese Civil Defense Command and Control Center continues to practice its operations in traditional ways. Therefore, the need for an efficient and reliable system to facilitate and modernize their working methods has become mandatory. Cypher Nova suggests a system that addresses the challenges faced by Civil Defense by providing user-friendly platforms that enable real-time incident monitoring, efficient resource allocation, and seamless communication among emergency response teams.

The system requested several features such as emergency notification from the call center, view emergency information (time, type, severity…), incident location tracking, vehicles in mission tracking, risk assessment, efficient resource allocation, account of volunteers on the mission, and post-mission report of the damage and operation details.

By using advanced technologies, the system ensures tracking of real-time information and facilitated communication which creates a dynamic and efficient approach to emergency management and helps minimize the impact on communities.

V. Glossary:

The glossary includes explanations of major technical and non-technical terms used.

1. Resource Allocation: The optimization of resources including equipment and supplies, based on real-time data and incident priorities.
2. Scalability: The ability of a system to handle an increasing amount of workload or user demands.
3. User Authentication: The process of verifying the identity of users accessing the system.
4. Consistency: It refers to the uniformity and correctness of data and processes in a system.
5. Redundancy: It refers to the duplication of critical components or functions within a system to ensure continued operation in the event of a failure or unexpected situation.
6. Real-time Tracking: is a method of monitoring and following something as it happens, providing immediate and up-to-the-minute information about its location, status, or progress, and is used in applications like GPS navigation.
7. Database: a structured collection of information or data that is organized and stored for easy retrieval and management used to store, manage, and retrieve data efficiently.
8. State Holder: the component responsible for maintaining and managing the current state or condition of a system, process, or object.
9. TLS 1.2 Protocol (Transport Layer Security) is a cryptographic protocol used for securing communication over computer networks. It ensures that data transmitted between two devices, like a web browser and a web server, remains confidential and cannot be easily intercepted or tampered with by unauthorized parties.
10. Backend: is the part of a software application that operates behind the scenes and is responsible for managing data, databases, and the overall logic of the application.
11. Frontend: the user interface or the visible part of a software application that users interact with directly including design, layout, and components that users see and interact with.
12. Machine Learning Models involve using algorithms and models to enable computers to learn from data and make predictions or decisions without being explicitly programmed.
13. WebSocket: communication protocol that enables real-time, full-duplex communication between a client and a server over a single TCP connection.
14. Node.js: open source, server-side JavaScript runtime environment that allows developers to build scalable and high-performance network applications.

15. Express.js: lightweight and flexible Node.js application framework that simplifies the process of building robust and scalable web applications and APIs.
16. SQL (Structured Query Language): domain-specific language used for managing and manipulating relational databases, allowing users to create, retrieve, update, and delete data.
17. HTTP (Hypertext Transfer Protocol): foundation of data communication allowing web browsers and servers to exchange text, images, multimedia, and other content over the internet.
18. HTTPS (Hypertext Transfer Protocol Secure): secure version of HTTP, that provides encrypted communication and data integrity between a user's web browser and the website they are visiting, ensuring secure transmission of sensitive information over the internet.

VI. User Requirements Definition

A Civil Defense and Control Center is required to manage several operations and emergencies:

1. The center shall provide real-time monitoring of incidents, enabling instant awareness of ongoing emergencies and their evolving status.
2. The center shall optimize resource allocation by analyzing available resources and matching them with the specific needs of each incident, ensuring a swift and effective response.
3. The center shall facilitate communication among emergency response teams, enabling quick exchange of information and coordination of efforts.
4. The center shall track the progress of ongoing incidents, providing detailed insights into the status of the response efforts.
5. The center shall generate post-mission reports detailing damage assessments and operational specifics.
6. The center shall be scalable to accommodate a growing volume of incidents and users without compromising performance.
7. The center shall have a backup and disaster recovery mechanism to prevent data loss in case of system failure

VII. System Requirements Specification:

Functional Requirements:

1. **Incident Monitoring and Management**

   A centralized dashboard shall display real-time data, including location, type, and severity.

   It shall allow authorized users to create, update, and close incidents with appropriate status codes (e.g., active, resolved, closed).

   It shall support the categorization of incidents based on predefined types (e.g., natural disasters, accidents, public emergencies).

2. **Resource Allocation**

   It shall maintain an updated database of available resources, including personnel, vehicles, and equipment.

   It shall enable automatic resource allocation based on incident type, location, and availability.

   In certain extraordinary situations, it shall allow manual intervention by authorized users to override automatic resource allocation.

3. **Communication**

   The center shall allow notifications and message sending to enhance communication between centers.

4. **Reporting and Analysis:**

   The center shall generate detailed post-mission reports, including incident timeline, action taken, resources utilized, and outcomes.

   It shall provide interactive data visualizations and geospatial maps to track the trucks-on-mission and the availability of resources.

   It shall offer predictive analytics features to forecast potential incidents based on historical data and external factors.

Non-functional Requirements:

1. **Security:**

   The center shall implement multi-factor authentication for user access to ensure authentication reliability.

   The center shall have role-based access control, allowing different levels of access based on user roles and responsibilities.

2. **Performance and Scalability:**

   The center shall achieve a response time of no more than 2 seconds for processing user requests, ensuring swift and efficient operations.

3. **Interfaces to other systems:**

   The center must integrate with external geospatial mapping services to visualize incident locations accurately.

   Interfaces with weather and environmental system provide real-time weather data, natural disaster predictions and environmental conditions.

   Interface with public communication platforms such as social media channels or official websites allows transparency and provides safety instructions during emergencies.

   By integrating with these systems, the Civil Defense and Control Center shall anticipate and prepare for incidents, improving overall emergency response strategies.

VIII-System Architecture:

Our Civil Defense and Control Center requires two main types of architecture: a layered architecture in addition to two client-server models. This is due to the need for a server to handle requests, and geospatial data and separate the different components (interface, backend, maps…) to facilitate maintenance and scalability. Therefore, this allows the testing of each component independently of others. Additionally, the layered pattern ensures security on the server-end level which can perform any required action without getting in the way of the user interface.

A. Layered Architecture

The layered architecture is divided into three main layers: the presentation layer, the application layer, and the data layer.

1. Presentation Layer

The web-based dashboard is the user interface for real-time incident monitoring and management. It provides authorized users with a visually appealing platform to monitor and manage incidents, and it can be accessed from any device (tablet, phone, pc...).

The structure and style of the dashboard shall be built using HTML for markup and CSS for styling.

React.js is employed for building dynamic and interactive elements of the dashboard. For real-time incident monitoring, WebSocket establishes a two-way communication channel between the server and the dashboard. This allows instant updates from the server to the dashboard without continuous polling. Authorized users can view a list of active incidents with details such as location, type, and severity.

Interactive maps can be integrated (using Google Map API) for visualizing incident location geospatially and tracking of trucks-on-mission, and available ones.

Users can update incident statuses, categorize incidents, and add comments for internal communication.

2. Application Layer

First, Node.js and Express.js are the technologies that shall be used for server-side operations.

To start with, user login is ensured in this layer who will receive the user credentials and check for their availability in the database.

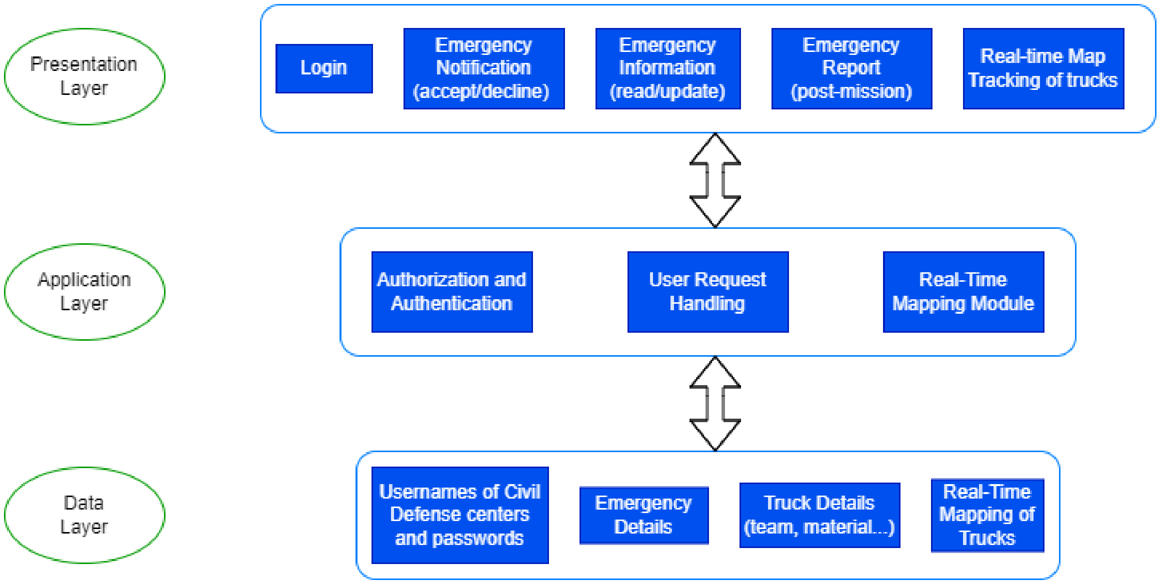
As for incident CRUD operations, users can CREATE by allowing new incidents with details like location, type, and severity, READ to view the information of emergencies, UPDATE by modifying incident details and status, and DELETE emergencies in case of cancelation. Additionally, they can close a case by providing the functionality to close an incident with the appropriate status code. All these operations are made available to the user by means of the application layer that executes the required actions on the database.

Second, the application layer is in charge of integrating geospatial maps to the interface using mapping libraries to visualize incidents' locations accurately and to track the vehicles in real time from the database. This will be done via Google Cloud services which allow such features such as Google Maps Javascript API Google Cloud Sup, and Google Cloud SQL for the database.

3. Data layer

This third and final layer covers the databases which are crucial to the web application. They are used to store data such as user credentials, emergency details (time, location, team, type, degree of severity…), and real-time location.

It is done with Google Cloud SQL and the data is stored on Google Cloud for increased reliability and is backed up frequently. When users make requests such as reading or updating information (from the presentation layer), the requests get sent to the application layer which handles them, and if it is successful, the data can be viewed or updated from the database (data layer).



B. Client-Server Architecture for Requests

To process user requests and authentication, a backend server is needed. After logging in, it checks the user's credentials to ensure they are valid before showing the requested page.

Then, the server will retrieve the required information from the cloud database such as emergency details (time, team, degree of severity, location…), notifications of new emergencies or a map with real-time tracking of trucks. Data can be modified by the users such as modifying/adding details of an emergency.

The communication between the server and the client is made encrypted and secure through the SSL certificate which ensures that no information transfer will occur unless a secure connection is established using HTTPS.

For the users and emergencies database, the following client-server diagram shows the architecture of the system:

C. Client-Server Architecture for Mapping

To provide real-time tracking, a WebSocket is required. It is bidirectional duplex protocol that is used for client-server communication where the connection between client and server remains “alive” until it is terminated on both ends by either the client or the server. The users need to know nothing about it. They can directly access the map and track in real-time the trucks and their availability thanks to the WebSocket.

For GPS databases, the following client-server diagram shows the architecture of the system:

IX. Subsystems

Our architecture implements many subsystems to manage the different types of operations. They are listed below.

1. Web Application Component

Its function is to provide a user-friendly and efficient interface to our users who are the centers of the Lebanese Civil Defense. It is the main interface of our website and it allows the users to manage and handle emergencies. It will mainly be coded with Hypertext Markup Language (HTML), Cascading, Style Sheets (CSS) and Javascript (JS).
Input: user queries
Output: processing of the queries

2. Google Cloud Services

The following Google Cloud services shall be used to facilitate vehicle tracking:

- Google Cloud Pub/Sub: to manage real-time tracking and message sending between the tracking device and the server
- Google Cloud Functions: functions allow to process the GPS data and store it in a database

3. Google Server

Our server-side web application shall be hosted on Google Cloud to receive and process emergency notifications, information and the GPS data transmitted by the tracking device. Therefore, it is in charge of all basic and advanced operations.

4. Google Cloud SQL

This database shall be used to store information on emergencies and real-time GPS data.
This cloud database will process all SQL queries that are generated on the server side. The SQL database component is connected to the Google Server and not directly to the “Web Application Component” for security purposes.
Input: SQL queries
Output: execution of queries (successfully or with error)

5. GPS (Global Positioning System)

Software-wise, the system shall use Google Maps Platform to display the Lebanese maps and geolocation data by using Google Maps JavaScript API, Geocoding API, and Directions API. Additionally, GPS tracking hardware shall be installed in the vehicles to collect and transmit GPS location data. We opted for the following device:
Smart Car GPS Tracker GT06 Protocol with Real Time Vehicle GPS Tracking Device Stop Engine G909

Here are its main characteristics:

- Positioning Mode: GSM, GPS, GPRS
- Function: Real-time location tracking

- Type: Wireless
- Placement: Dashboard
- Track View: PC, iOS APP, SMS Coordinate, Android APP
- Warranty: 1 Year
- Battery Life (Hour): <12 hours
- Brand Name: Trackerking
- Model Number: G909
- Use: Motorcycle, vehicle

X. System Models

Our system acts as the nerve center, managing responses, and resources, and facilitating communication between the call center and control centers. To effectively implement their work, system modeling is essential. This process breaks down the flow of a request in the civil defense system from initiation to resolution.

The flow of requests in the Civil Defense and Control System:

1. **Incident Initiation:**

    Input: Incident details, location, and severity

    Process: The system receives and validates the incident report for authenticity and completeness.

    Output: Validated incident report data ready to be analyzed.

2. **Incident Categorization and Prioritization:**

    Input: Validated incident report data

    Process: Control center Staff categorize the incident based on its type (fire, accident…) and prioritize it according to severity and urgency.

    Output: Categorized and Prioritized Incident Data ready to proceed.

3. **Resource Allocation and Response Planning:**

    Input: Categorized and prioritized data, emergency plans, available resources

    Process: Control Center Staff analyze the incident and allocate appropriate resources for response.

    Output: Instructions for emergency services generated and stored in the incident report data.

4. **Public Communication:**

    Input: Incident alerts, updates, and Safety instructions generated by Control Center Staff

    Process: Control Center Staff format the information received into a clear, concise, and informative message suitable for public consumption. If necessary, translate messages into multiple languages to ensure a broader reach within diverse communities.

    Output: Messages are sent out via the selected communication channels to the specified target audience. Acknowledgment of receipt may be tracked to assess the effectiveness of the communication.

5. **Monitoring and Coordination:**

Input: Ongoing incident Status updates, reports from emergency services

Process: Control Center Staff continuously monitor the situation, coordinate response efforts, and make real-time adjustments to resource allocation instructions sent back to emergency services.

Output: Updated incident status, and resource allocation instructions sent back to emergency services.

6. **Incident Resolution and Reporting:**

Input: Updated incident status indicating resolution

Process: Once the incident is resolved, Control Center Staff confirm the resolution, generate incident reports, and store them for future reference and analysis

Output: Incident resolution report stored in the historical record

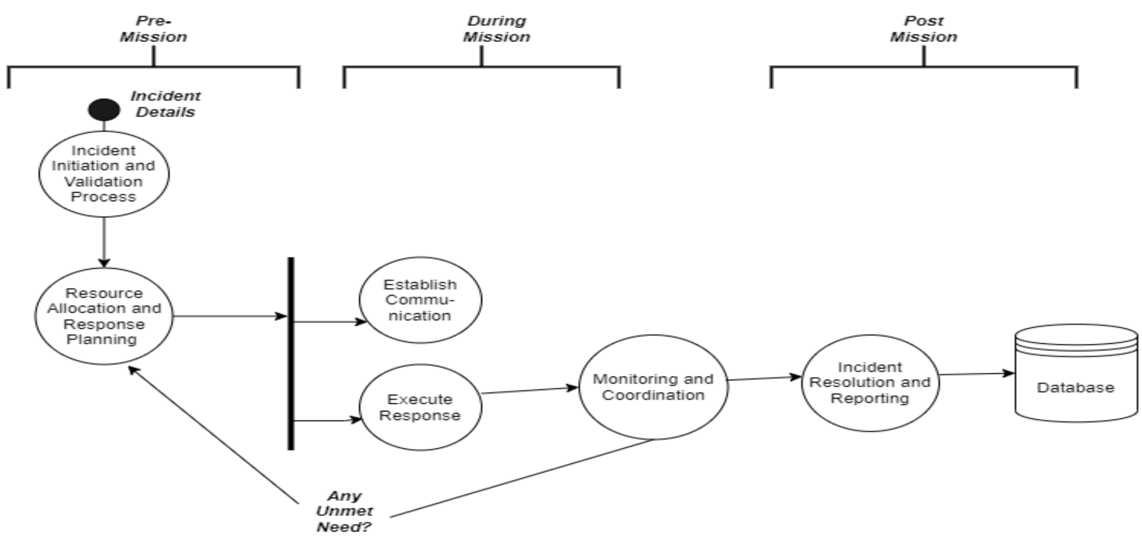


Figure 1.1: System Models

XI. System Evolution

The Civil Defense and Control Center should be designed with a forward-looking approach. To ensure its adaptability, it is essential to anticipate the potential changes in technology, user needs, and other factors that might influence the system's evolution. This section describes the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and technological advancement.

Fundamental Assumption:

1. **Emergency Readiness:** Our system will continue to handle a wide range of emergencies, including technological accidents, natural disasters, and crises. Emphasis will be placed on enhancing real-time coordination, resource management, and information dissemination during crises.
2. **Scalability:** The system, built on reusable technology like MongoDB, will remain adaptable to new hardware and software updates. Regular scalability assessments will be conducted to ensure seamless integration with evolving technologies.
3. **Privacy and Security:** The security of sensitive information will be maintained through robust protocols like TLS 1.2. Regular security audits and compliance checks will be performed to uphold the highest standards of data security and user privacy.
4. **Hardware Evolution:**
    - **4.1. Processing Power:** The system will leverage advancements on processing power for real-time data and decision-making abilities.
    - **4.2. Storage Capacity:** High-capacity storage devices will be integrated to accommodate extensive datasets, including historical records, multimedia files, and real-time information.
    - **4.3. Communication Interfaces:** Anticipate the emergence of 5G and beyond, enabling fast and low latency communication. These interfaces will ensure precise and smooth data transmission between the control center and field units.
5. **System Architecture:** Flexible system architecture will be maintained allowing the smooth integration of emerging hardware technologies without causing disruptions.
6. **Redundancy and Disaster Recovery:** Redundant data centers and cloud-based solutions will be implemented to ensure continuous system operation during hardware failure or natural disasters.
7. **Legal and Ethical Considerations:** The use of AI and drones will adhere to strict legal and ethical guidelines.

8. **Budget and Funding:** A sustainable budget and funding strategy will be developed to support hardware and software enhancement, training initiatives and ongoing system maintenance.
9. **Monitoring and Feedback:** Continuous monitoring mechanisms will be established to track system performance and user satisfaction. Feedback loops will be actively maintained allowing user to provide input for system improvements.

XII. Appendices

Appendix A: Hardware Requirements

This section outlines the optimal hardware configuration required for the Civil Defense and Control Center system to function effectively.

1. **Tablet Devices:**

   We opted for the M101Q8-ME10.1" Qualcomm® Snapdragon™ 660 Android Healthcare Rugged Tablet. It is the same tablet that is used in ambulances. It offers the following features:

   - Qualcomm® SDA660 Kryo 260 CPU + Octa-core up to 2.2GHz, Android Healthcare tablet
   - 10.1" (1920 x 1200) TFT LCD Display, Android Healthcare tablet
   - 3 GB RAM + 32 GB eMMC
   - Housing enhanced with antimicrobial properties
   - Lightweight, rugged, IP65 waterproof and dustproof
   - 13 MP Camera on the Rear Side (with LED auxiliary light, Auto Focus)
   - 8 MP Camera on the Front Side
   - Extended Battery Life without Sacrificing Performance
   - Real-Time Connectivity and Data Capture

Therefore, it is suitable for emergency vehicle applications.

2. **Call Center Computer**:

   It is not necessary to have very high-end computers. However, the following specifications are required to support Google Maps JavaScript API:

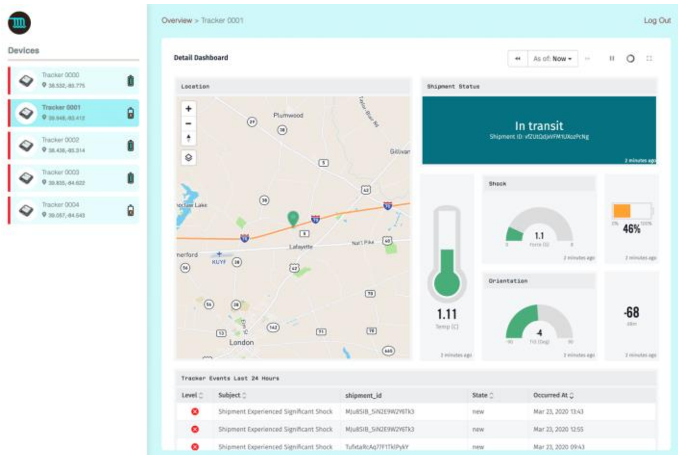
   - Operating System: most recent computers running Windows, macOS, or Linux can support the Google Maps JavaScript API (Application Programming Interface)
   - Web Browser: needs to be compatible with Google Maps such as Google Chrome, Mozilla Firefox, and Microsoft Edge
   - Internet Connection: a stable, secure, and fast internet connection is essential for loading maps through the API.
   - CPU: A computer with a dual-core or quad-core processor is typically sufficient for using the Google Maps API
   - Memory (RAM): 4GB of RAM or more is recommended for a smoother experience

- Graphics: a graphics card is not required but having one may improve the display of maps for 3D views and satellite imagery.
- Storage: a large amount of storage for the operating system and other software is recommended but the Google Maps API itself does not consume significant local storage.
- Monitor: a good-sized monitor with a decent resolution enables a flexible view and works with maps. A resolution of 1920x1080 (Full HD) or higher is ideal.

3. A specialized GPS in each truck

This can be done by using Specialized GPS Systems that are manufactured for emergency response and enable specific features such as real-time tracking and communication with dispatch centers.

The GPS will be linked to the maps available through our website in each center to view and track the trucks. The call center can view all the trucks on its map. We opted for Google Maps which uses JavaScript API and can integrate a GPS device to track the trucks of the Civil Defense.



4. **Internet Service:**

Fiber Optic is the fastest and most reliable option available. Here are some of its features:

- High Speed: fiber optic cables offer extremely high data transfer speeds, with the potential for multi-gigabit or terabit per second (Tbps) data transmission.
- Low Latency: fiber optics have extremely low latency, which means data can travel at the speed of light, making it suitable for real-time applications like real-time tracking.
- Long-Distance Transmission: fiber optic cables can transmit data over much longer distances without signal degradation compared to traditional copper cables.
- Immunity to Electromagnetic Interference: unlike copper cables, fiber optics are not susceptible to electromagnetic interference (EMI) or radio frequency interference (RFI), making them exceptionally reliable in environments with electromagnetic noise or interference.
- Security: fiber optic signals are difficult to intercept or tap into because they do not radiate electromagnetic signals making them more secure for data transmission.

Appendix B: Database Requirements

The database requirements listed in this appendix are intended for a later project phase as the system evolves and expands.

XIII. Index

A

Appendices, 16
- Database Requirements, 16
- Hardware Requirements, 16

D

Database
- Requirements, 18

G

Glossery, 16

H

Hardware
- Requirements, 16

I

Introduction, 16

P

Preface, 16

R

Report License, 16

S

System Architecture, 16

System Evolution, 16

System Models, 16

System
- Requirements, 16