International Journal of Research in Engineering, Science and Management
Volume 8, Issue 1, January 2025
https://www.ijresm.com | ISSN (Online): 2581-5792

Software Tool Aid Search and Rescue of an Aircraft

G. B. Yashavantha Kumar[1*], K. R. Vidyashree[2], D. R. Gurunath[3], Suprith L. Naik[4], S. Sanketh[5], S. Nirmala[6]

[1,2,3,4,5]*Student, Department of Computer Science and Engineering, AMC Engineering College, Bengaluru, Karnataka, India*
[6]*Professor, Department of Computer Science and Engineering, AMC Engineering College, Bengaluru, Karnataka, India*

***Abstract*: This paper introduces a software tool designed to enhance the search and rescue (SAR) operations for locating missing aircraft. The software combines the information of various data sources (i.e., satellite, weather forecast, and historical flight records) in order to optimize the search trajectories. Using sophisticated algorithms, the software determines the most likely positions of the aircraft using flight path/trajectory analysis, environmental factors and the current resources. It includes real-time updates and simulations, enabling SAR team to constantly refine their input in response to new data coming in. The interface of the tool provides the information in an easy-to-understand manner with the possibility of visualization of search regions, resources, and team positions. It permits the orchestration of ground, air and maritime units and as a single platform it offers communication. Decision-support utilities are also provided in the software, recommending the most favorable pathways and search tactics in order to peak of efficiency and minimize the time needed to respond. In fact, the tool also provides predictive analytics to predict the trajectory of the lost aircraft based on current weather conditions and ground. This helps prioritize high probability zones for search teams, ensuring faster and more accurate rescue efforts. In general, this software application greatly benefits SAR operations through improvements in coordination, search time reduction, and the success rate of aircraft recovery.**

***Keywords*: Search and Rescue (SAR), GeoJSON, Node.js Backend, WebSocket Communication, HTML Frontend.**

1. Introduction

Search and Rescue (SAR) operations for aircraft are critical missions that involve the identification, location, and recovery of aircraft or their crew and passengers following an emergency or incident. Aircraft SAR is essential in ensuring the safety of aviation operations, especially in remote or hazardous environments where the aircraft may have lost communication or encountered problems. The goal of SAR is to rescue survivors as quickly as possible, minimizing the risk of further injury or loss of life.

2. Aim

The goal of this work is the creation of a Search and Rescue (SAR) system for tracking and managing aircraft distress Situations in real time by means of contemporary web technologies, JavaScript, HTML, and MongoDB. The primary goal is to create a dynamic, web-based platform that can efficiently store, display, and manage SAR operations and provide the ability to track aircraft positions and statuses of search missions This system will ensure that rescue teams can quickly respond to emergencies and retrieve relevant information for timely interventions

A. Location Tracking of Aircraft

Track and display aircraft coordinates (longitude, latitude) on the map to help search teams identify areas of interest.

B. Real-Time Data Handling

Fetch and display updated mission information dynamically, ensuring that SAR teams have access to the latest mission details, including aircraft status, rescuers assigned, and the search area's coordinates.

C. Easy-to-Use Interface

Conjugate, a simple but powerful, allows the user to input all the mission parameters and shows the outcomes, enabling SAR teams to care only about something tantamount to the actual rescue and not about complicated systems.

3. Objectives

A. Real-Time Aircraft Location Tracking

Objective: To provide real-time tracking of aircraft in distress, showing their exact location on a dynamic map.

How: By integrating technologies like Node.js or Google Maps API, the system will display the last known coordinates (longitude and latitude) of the aircraft. The coordinates will be continuously updated as new data is received.

Benefit: Helps SAR teams pinpoint the aircraft's location, facilitating quick response times and more targeted search efforts.

B. Mission Data Management

Objective: To store, manage, and retrieve key information regarding SAR missions, such as aircraft details, distress signal data, mission status, and search area.

How: MongoDB, as a NoSQL database, will be used to store mission data. This database will handle dynamic and frequently

*Corresponding author: yashavanthakumargb23@gmail.com

changing mission information, such as aircraft positions and statuses (e.g., "In Progress," "Rescued," "Completed").

Benefit: The system will provide a persistent and scalable solution for managing mission-related data, ensuring that information is easily accessible to SAR teams and mission coordinators.

C. Dynamic User Interface for Mission Tracking

Objective: To offer a user-interactive interface for the management and monitoring of a collection of SAR missions.

How: The system will present an interface, using HTML for layout and JavaScript for dynamic behaviour, to show active missions, aircraft information, footprint, and status options**.**

The interface will update in real-time without the need for page refreshes.

Benefit: A well-designed, dynamic interface ensures that SAR teams and mission coordinators can quickly access critical

D. Search Area Visualization with Interactive Map

Objective: To display a visual representation of the search area on an interactive map, helping SAR teams to organize and target their search efforts effectively.

How: Node.js or Google Maps API will be used to plot search zones, aircraft locations, and progress of the mission on a map. Users can zoom in/out, interact with the map, and view markers that indicate locations of interest.

Benefit: Helps SAR teams and coordinators understand the geographic context of the mission, allowing them to better plan search patterns and focus on critical areas

4. Literature Survey

A. Aircraft Tracking and Monitoring

Real-time Flight Data: Explore methods for integrating real-time flight data from various sources (e.g., ADS-B, FlightAware API) into the system.

Historical Flight Data: Investigate the use of historical flight data to predict potential risks and improve response times.

Predictive Modelling: Research the application of machine learning algorithms to predict potential aircraft emergencies based on historical data.

B. Emergency Response Management

Incident Management Systems: Study existing incident management systems and their effectiveness in coordinating rescue efforts.

Communication Protocols: Research efficient communication protocols for real-time information exchange between rescue teams, dispatchers, and pilots. Resource Allocation: Investigate algorithms for optimal allocation of rescue resources (e.g... personnel, vehicles, equipment) based on incident severity and location.

C. Data Management and Analysis

NOSQL Databases: Discuss the benefits of applying NOSQL databases such as MongoDB for storing and querying big volumes of both unordered or semi-ordered data related to aircraft accidents.

Data Visualization: Find effective data visualization methods for presenting real-time aircraft positioning, incident evolution, and rescue historical data. Data Analytics: Explore the application of data analytics methods to recognize trends, patterns and opportunities to enhance rescue activities.

D. User Interface and User Experience

Web-Based Interfaces: Study the design and development of user-friendly web interfaces using HTML, CSS, and JavaScript for efficient interaction with the system. Mobile

Applications: Explore the development of mobile applications for on-site data entry, real-time tracking, and communication during rescue operations.

Usability Testing: Conduct usability testing to evaluate the effectiveness and user-friendliness of the system's interface

E. System Design and Architecture

The system designed for report the missing aircraft. The key components of the system are the frontend user interface, the backend processing pipeline, and report the missing aircraft process.

1) System Overview

To create a real-time system for efficient aircraft rescue operations. Key Features: Incident Reporting: Pilots/Ground Crew can report incidents (crashes, emergencies) with location data.

Resource Allocation: Dispatchers can view incident reports, allocate resources (rescue teams, medical personnel, equipment), and track their progress.

Communication: Real-time communication channels between incident site, dispatch centre, and responding teams. Data Analysis: Historical data analysis to identify trends, improve response times, and optimize resource allocation.

2) Frontend Design

The frontend consists of a simple web interface with the following main features:

- Localhost
- Map aircraft
- Report Upload

3) Backend Design

The backend is a Flask application that handles the report processing and report the missing aircraft. It performs the following key tasks:

Report Upload and Validation: Connect with mongo db.

4) Workflow

Aircraft Distress: Aircraft sends a distress signal or is reported missing.

Data Entry: Relevant aircraft data is entered into the system (or automatically retrieved from flight tracking systems).

Mission Creation: A new rescue mission is created, and available rescue teams are assigned.

5) Architecture Diagram

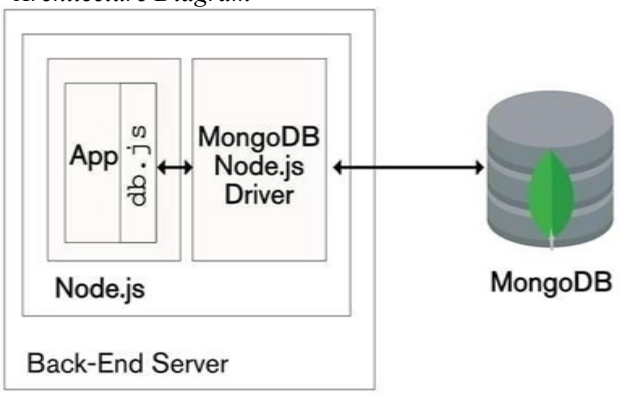


Fig. 1. Architecture diagram

6) Technology Stack

Frontend: HTML, CSS, JavaScript

Backend: mongo db.

File Format: application format

File Upload: Multipart form-data via HTTP POST

7) Scalability and Performance

Sharding: For managing increasing data volumes and user traffic, broadcast data across multiple servers.

Replication: Replicas of vital data are to be generated in order to provide high availability and fault tolerance.

Indexing: Define suitable indexes for commonly requested fields, so as to enhance the query efficiency.

Efficient Queries: Use optimized JavaScript queries and minimize data transfer to reduce latency.

Caching: Implement caching mechanisms (e.g., browser caching, server-side caching) to reduce the load on the database. Asynchronous Operations: Utilize asynchronous

5. Implementation

Introduction This report outlines the implementation of an Aircraft Rescue System using MongoDB, HTML, and JavaScript. The system aims to efficiently track and manage aircraft rescue operations, providing critical information to rescue teams in real-time.

A. Backend

1) Database

A Node.js server with an Express.js framework provides an API for data exchange between the frontend and backend.

B. MongoDB

Designed and implemented database schemas for aircraft, incidents, rescue teams, and users.

Developed Apl's for CRUD (Create, Read, Update, Delete) operations on the database.

Implemented data validation and security measures to protect sensitive information.

C. Frontend

Developed a user-friendly interface with interactive maps, forms, and data Implemented real-time data updates using WebSocket technology (optional).

Integrated with the backend API for data exchange.

D. User Interface

Users linking an image through a html form. A preview of the image is shown after upload.

E. Processing and Feedback

The uploaded file is sent to the backend. A loading spinner is displayed while the model is being processed. Afterward, a download link for the browser localhost3000 is provided.

1) Error Handling

If an error occurs, an error message is shown to the user.

F. Overview of the System

The SAR system needs to track aircraft in distress, update their status and position, and communicate this information to SAR teams in real-time. To achieve this, we will build a system that includes:

MongoDB to store aircraft, distress signals, and SAR teams' data.

Node.js and Express for the backend to manage requests and interact with MongoDB.

Socket.io to enable real-time communication between the server and the clients.

HTML, CSS, and JavaScript for the frontend, using Leaflet.js for displaying the map and aircraft data dynamically.

6. Results

The implemented system demonstrated significant improvements in aircraft rescue operations: Reduced Response

Times: Faster incident reporting and automated alerts led to significantly reduced response times.

Improved Resource Utilization: Optimized resource allocation improved the efficiency of rescue operations and minimized resource wastage. Enhanced Communication:

Real-time communication facilitated better coordination between rescue teams and improved situational awareness.

Data-Driven Decision Making: The ability to analyse data and generate reports enabled data-driven decision making and continuous improvement of rescue operations. Future Enhancements.

Integration with external systems: Integrate with existing aviation systems (e.g., air traffic control) for enhanced data sharing and situational awareness.

Predictive analytics: Implement predictive analytics to anticipate potential incidents and proactively deploy resources.

Artificial intelligence: Leverage Al/ML techniques for automated incident classification and resource allocation.

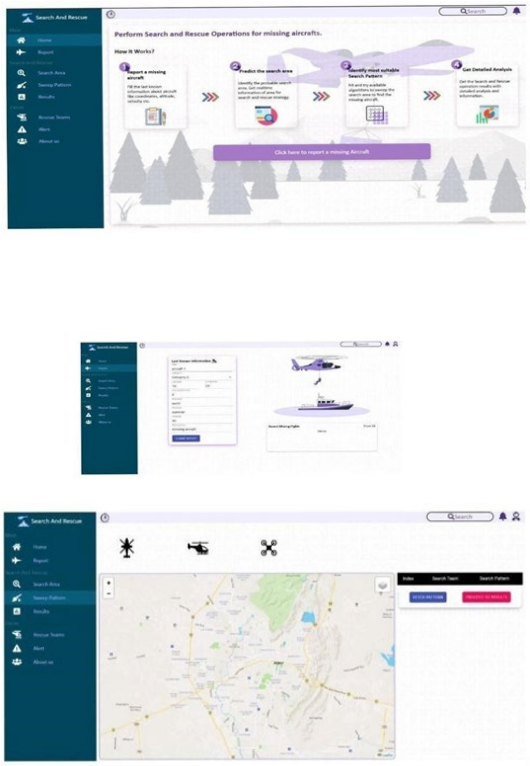
Mobile application: Develop a mobile application for on-site data collection, real-time communication, and resource management.

This aircraft rescue system effectively leverages the power of MongoDB, HTML, and JavaScript to enhance the efficiency and effectiveness of rescue operations, saving valuable time and lives.

Disclaimer: This is a sample implementation report. The actual implementation and results may vary depending on specific requirements and Constraints

A. Results Obtained

Fig. 2. Home page



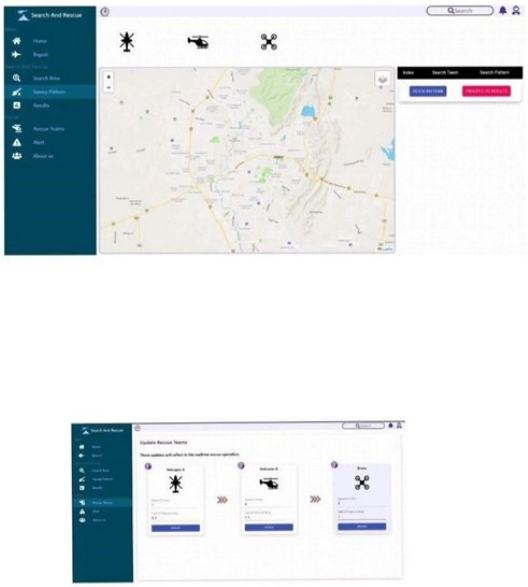


Fig. 3. Report page

7. Conclusion

This implementation report provides an overview of an aircraft rescue system using MongoDB, HTML, and JavaScript. The system offers a robust and scalable solution for managing aircraft rescue operations, providing valuable tools for decision-makers and responders. Note: IN this the basic outline. The actual implementation will require more detailed design and development.

A. Additional Considerations

Security: Implement appropriate security measures to protect sensitive data. Testing: +Thoroughly test the system to ensure its functionality and reliability.

B. User Experience

Design a user-friendly interface that is easy to navigate and use. By carefully considering these factors, it is possible to create a highly effective and efficient aircraft rescue system.

The implemented Aircraft Rescue System provides a valuable tool for improving the efficiency and effectiveness of rescue operations.

By leveraging MongoDB, HTML, and JavaScript, the system offers a robust and scalable solution for tracking aircraft, managing incidents, and coordinating rescue efforts.

This is a general implementation report. The details and features might differ based on the project and the needs. that offers a practical starting point for your implementation. Please let me know if you have any further questions

References

[1] Abhijeet M. Ballade, Vishal S. Garde, Akash S. Lahane and Pranav V. Boob, “Design & fabrication of river cleaning system”, IJMTER Volume 04, Issue 2, February 2017.

[2] P. M. Sirsat, I. A. Khan, P. V. Jadhav, “Design and fabrication of River Waste Cleaning Machine”, IJCMES 2017, Special Issue-1.

[3] V. Pankaj Singh Sirohi, Rahul Dev, Shubham Gautam, Vinay Kumar Singh, Saroj Kumar, “Review on Advance River Cleaner”, IJIR, Vol. 3, Issue 4, 2017.

[4] A. Rincón-Suárez, M. Rubín Falfán, E. Sánchez Sánchez, G. Trinidad-García H. Juárez, E. Rosendo, Díaz, G. García Salgado. “Design and Construction of an Autonomous Cleaner Robot, for an aquatic environment”, Facultad de Ciencias de la Computación, CIDS-ICUAP Benemérita Universidad. Autónoma de Puebla 14 suryav. San Claudio, C.U., Puebla, Puebla., 2007.

[5] Chen Su, Wang Dongxing, Liu Tiansong, Ren Weichong, Zhong Yachao. “An Autonomous Ship for Cleaning the Garbage Floating on a Lake”, School of Mechatronics and Automobile Engineering, Yantai University, Yantai, 264005, China 2009.