Evaluation of an Integrated Disaster Response Support System Utilizing Information and Communication Technology

Yuta Seri[1], Tomoyuki Ishida[2]

[1]Graduate School of Engineering, Fukuoka Institute of Technology, Fukuoka, Japan

[2]Department of Information and Communication Engineering, Fukuoka Institute of Technology, Fukuoka, Japan

Received: November 15, 2025; Revised: December 30, 2025; Accepted: January 22, 2026; Published: February 05, 2026

Abstract

In this study, we developed an integrated disaster response support system to improve the efficiency of disaster response operations and integrate information management in local governments. In response to recent issues, e.g., a shortage of specialized staff and difficulties with knowledge transfer, the proposed system was designed to integrate functions, support disaster operations using the latest technologies, and eliminate the dependency on personal knowledge. The results of an effectiveness evaluation by local government employees demonstrated the system's potential to eliminate information fragmentation and support decision-making processes. In addition, a usability evaluation using the system usability scale rated it highly, confirming that even inexperienced staff can operate the proposed system intuitively. Furthermore, a chatbot equipped with retrieval-augmented generation functionality demonstrated high accuracy rates and efficiency in several information search tasks. The findings of this study indicate that the proposed system combines ease of use with advanced technology, thereby making it a useful platform that will contribute to improving the disaster response capabilities of local governments.

Keywords: Disaster Response Support, Disaster Prevention DX, Generative AI, Retrieval-Augmented Generation (RAG), Eliminating Knowledge Dependence, Intuitive UI/UX Design.

1 Introduction

Due to its geographical and topographical conditions, Japan is one of the world's most disaster-prone countries. According to the World Risk Report 2025 [1], Japan is ranked third in the world in terms of the "exposure" index, which measures a population's exposure to natural disasters. This index is calculated based on the frequency and intensity of hazards, e.g., earthquakes, tsunamis, and floods, and it indicates Japan's physical proximity to the threat of disasters. Thus, the Japanese government has made the development of effective disaster prevention measures a national priority.

However, structural issues have persisted in Japan's disaster response system for many years. For example, Iizuka [2] stated that many local governments cannot adequately share lessons learned from previous disasters within their organizations, resulting in "constantly repeating the same mistakes." The root of this problem is the personalization of disaster response work. As a result, tacit disaster response knowledge is concentrated in the hands of a few experienced local government employees, which makes it difficult to share and accumulate explicit knowledge throughout the organization.

Nishimura [3] conducted a survey targeting local government employees in prefectures and major cities, and the results demonstrated that only approximately 35% of employees have direct experience in disaster response work. This means that approximately 65% of employees lack sufficient practical experience and may be forced to make extremely difficult decisions in the event of an emergency. As a result, the transfer or retirement of disaster prevention personnel leads directly to a decline in the entire organization's disaster response capabilities; thus, many local governments are extremely vulnerable to risk.

Recently, "Disaster Prevention Digital Transformation" (DX) has been promoted to advance disaster prevention and mitigation using information and communications technology (ICT) and artificial intelligence (AI). However, while progress has been made in individual technologies and systems to address specific issues (e.g., safety confirmation and information transmission), attempts to integrate and deploys these technologies as practical platforms to support the entire diverse workflow at disaster sites have been insufficient. Even cutting-edge technologies may not be put to good use if they are incompatible with on-site operations and do not have a user interface/user experience (UI/UX) that is intuitive for staff with limited ICT literacy. To be truly valuable, disaster prevention DX must extend beyond simply introducing technology. It must also increase the overall response capabilities of the entire organization.

The remainder of this paper is organized as follows. Related work is described in Section 2, and the goal of the study is discussed in Section 3. The system configuration of the proposed integrated disaster response support system is described in Section 4, and the proposed system is described in detail in Section 5. The proposed integrated disaster response support system is evaluated in Section 6. Finally, the paper is concluded in Section 7.

2 Related Work

Recently, progress has been made domestically and internationally in the development of information sharing and decision-making support systems for disaster response. In particular, a comprehensive system that supports information visualization, automation, and efficiency is required to ensure rapid response while collaborating with diverse stakeholders, including disaster victims, local government employees, and volunteers. This section summarizes related studies that form the foundation for such systems.

In a study on matching disaster volunteers, Sekiguchi et al. [4] proposed an application to assist the operation of disaster volunteer centers to address the inefficiencies of conventional paper-based volunteer matching processes. This system allows disaster victims to easily request assistance through provisional and formal registration, thereby reducing the workload of volunteer centers. In addition, Kodama et al. [5] developed a support system for inexperienced volunteers to reduce the psychological barriers to volunteering. This system allows users to search for appropriate activities by entering their desired criteria, thereby encouraging them to participate in volunteer activities. Hijiri et al. [6] developed a system to support smooth communication between disaster victims, volunteers, and volunteer centers, effectively streamlining on-site coordination and activity reporting through chat-based information exchange. Furthermore, Huang et al. [7] developed a machine learning–based volunteer management platform for high school students to recommend suitable volunteer events. The findings of these previous studies contribute to the efficiency of volunteer activities and the appropriate distribution of assistance needs, and they form the basis for this study's automatic matching function between disaster victims and volunteers.

In terms of evacuation shelter management, Nakatani et al. [8] proposed a system to support the dispersed evacuation of evacuees by registering and visualizing the homes of relatives and friends other than designated evacuation shelters as potential evacuation sites. This prevents evacuation shelters from exceeding their capacity and supports safe and flexible evacuation. In addition, in light of the COVID-19 pandemic, Nakada et al. [9] developed a shelter navigation system that assesses the congestion status of evacuation shelters in real time and guides evacuees to the most appropriate evacuation destination. Akasaka et al. [10] proposed a disaster victim information management system that collects and shares disaster victim information even in situations where lifelines are cut off, and they are working to speed up disaster victim support at evacuation shelters.

Furthermore, to support the evacuation of people with special needs and guide people to welfare evacuation shelters, Shimoda [11] developed a system that utilizes a geographic information system. By sharing the support status of such individuals in real time, they aim to speed up information sharing processes. These studies aim to improve the efficiency of evacuation shelter management and are in line with the design guidelines for the evacuation shelter management functions developed in the current study.

The use of large language models (LLMs) and the retrieval-augmented generation (RAG) framework in disaster response is attracting attention. For example, Pujiono et al. [12] proposed a chatbot system for public service organizations that combines RAG and vector databases. In addition, Boné et al. [13] developed a Portuguese-speaking disaster support chatbot called DisBot to support first responders during disasters. This system employs dual intent entity Transformer architecture to accurately grasp the user’s intent, which enables smooth dialogue. Tsunoda et al. [14] developed a reporting support system that reduces reporting time and improves the efficiency of administrator work by introducing a chatbot into reporting tasks during corporate disaster response training. Xia et al. [15] proposed a typhoon disaster Q&A method that integrates RAG with the T5 LLM, and they reported improved information retrieval efficiency and answer quality. Furthermore, Urbanelli et al. [16] developed the ERMES chatbot for disaster management and risk reduction to enhance information exchange between citizens and disaster response agencies through real-time two-way communication. The findings of these studies demonstrate the potential of generative AI for decision-making support systems in disaster situations.

Research into the automatic analysis and sharing of disaster images is also progressing. For example, Sogi et al. [17] proposed a framework to instantly visualize disaster damage by combining image search using a visual language model with matching geographic information. Udo et al. [18] proposed a language bottleneck model that integrates image caption generation and classification. They reported that classification accuracy was improved considerably by using bootstrapping language-image pretraining (BLIP) and BLIP-2 methods, with particularly high performance demonstrated by the CLIP Interrogator (CLIP-I). In addition, Yasuda et al. [19] proposed a method to extract tags by generating disaster-specific descriptions using generative AI, demonstrating that disaster-specific expressions can be obtained by selecting prompts carefully. Furthermore, Klerings et al. [20] automatically captioned disaster site images collected through crowdsourcing and converted them into structured data. These studies are consistent with the design principles of our disaster image map visualization system and provide fundamental knowledge to support intuitive understanding of the damage situation.

Finally, in terms of visualization and decision-making support for disaster response tasks, Tillekaratne et al. [21] developed a dashboard that visualizes the Sri Lankan Disaster Situation Report to support disaster managers’ decision-making processes, and Hanashima et al. [22] proposed a dynamic decision-making support system that integrates natural observation data, e.g., precipitation, with social data, e.g., population and buildings, to predict the applicability of disaster relief methods. Furthermore, Li et al. [23] proposed the GeoGraphVis cyberinfrastructure, which employs knowledge graph technology to link and structure data from various domains with geospatial information. The findings of these studies demonstrate the importance of task management, decision-making support, and data storage in disaster response, and they form the theoretical foundation for our disaster response visualization and storage system.

3 Research Objectives

The goal of this study is to overcome the issues faced by the disaster information systems implemented by our research group in previous research and develop an integrated platform that provides comprehensive and sustainable support for disaster response operations, primarily involving local government employees. This study emphasizes the design of a system that is in line with on-site practical use and can be operated easily by anyone. The objects of this study are defined from the following three perspectives.

The first objective is to integrate various disaster prevention–related functions into a common platform. In a previous study, we developed various systems to address individual issues, e.g., evacuation shelter management and disaster information sharing. However, each system functioned independently; thus, information was fragmented, thereby making it difficult to grasp the overall disaster scenario. In the current study, we integrated these functions into a common web platform and designed a structure that can provide cross-sectional support for all phases of disaster response. Here, the goal is to develop a platform that allows diverse stakeholders, e.g., local government employees, volunteers, and disaster victims, to share information in the same environment and work collaboratively to facilitate effective and efficient responses.

The second objective is to reduce the burden on staff by automating part of disaster response work using emerging technologies, e.g., generative AI, image processing techniques, and RAG. Recently, advances in generative AI have made it possible to improve efficiency in various areas, including document summarization, decision-making support, and image recognition. In this study, we employ AI to automatically analyze and organize the vast amounts of image data and reports collected from disaster sites, thereby supporting information visualization and decision-making processes. In addition, we implement a chat system using RAG that allows staff to easily search and reference vast amounts of documents, e.g., disaster prevention plans. Through these technological approaches, we aim to automate various tasks, e.g., information organization and document referencing, which were previously performed manually, thereby increasingly the efficiency of disaster response efforts.

The third objective is to solve the issue of knowledge being personalized by designing a UI/UX that anyone can operate intuitively and promote the sharing and inheritance of knowledge within organizations. In many local governments, disaster prevention staff members have short tenures, which makes it difficult for them to accumulate experience due to individual dependency. To address this issue, it is essential to design a system that does not require specialized knowledge and implement a mechanism to store and share tasks and records across the entire organization. By eliminating complex operations and adopting a simple interface, this study implements a system environment that even inexperienced local government employees can use efficiently and effectively.

Taking these three points into consideration, this study aims to implement disaster prevention DX that combines technological sophistication with ease of use. The significance of this research lies in the presentation of a practical framework to promote the standardization of disaster prevention operations and knowledge sharing in local governments and sustainably improve organizational disaster response capabilities.

4 Configuration of Integrated Disaster Response Support System

The integrated disaster response support system developed in this study employs a four-layer architecture, including the user, presentation, application, and data layers, as shown in Figure 1. This architecture clearly separates the roles of each layer, which ensures the maintainability and scalability of the entire system and enables multiple disaster response support functions to operate in an integrated manner on a common platform.

[User layer]

The user layer consists of three main actors who use the system, i.e., local government employees, residents (disaster victims), and individual volunteers. Here, the local government employees are primarily responsible for disaster response tasks, managing evacuation shelter operations, and sharing information, and the residents use evacuation shelters and post damage information. The individual volunteers also play a role in supporting on-site support by registering relief activities and checking matching results.

[Presentation layer]

The presentation layer provides the web interfaces operated by the users. This layer comprises five subsystems, i.e., the evacuation shelter management, disaster response task visualization, RAG-based disaster knowledge sharing, disaster image sharing, and victim/volunteer automatic matching systems. Each subsystem has a UI customized to its specific purpose and user type, and each system is implemented based on Django's view [24] and template functions.

[Application layer]

The application layer handles the core processing logic for each subsystem. Based on Django's model and business logic, it comprises the following five logic layers.

1) Evacuation shelter management logic: handles shelter statistics management and resource estimation.

2) Task management and report generation logic: handles task registration, progress management, and report generation.

3) RAG integration processing logic: handles disaster plan search using generative AI.

4) Image analysis processing logic: generates image captions.

5) Matching processing logic: collates disaster victims with volunteer assistance requests.

These five logic layers are managed independently as Django applications interconnected through a common data layer.

[Data layer]

The data layer combines a relational database (MySQL) and a vector database to centrally store and manage the diverse data handled in disaster response scenarios. Here, the relational database stores structured data, e.g., evacuation shelter information, task records, and user information, and the vector database manages the document-embedded data utilized in the RAG system.

[External API]

The external API comprises natural language generation using the OpenAI API [25], image caption generation using BLIP-2 [26], and map drawing functions using Open Street Map [27], thereby enabling flexible integration with different external services.

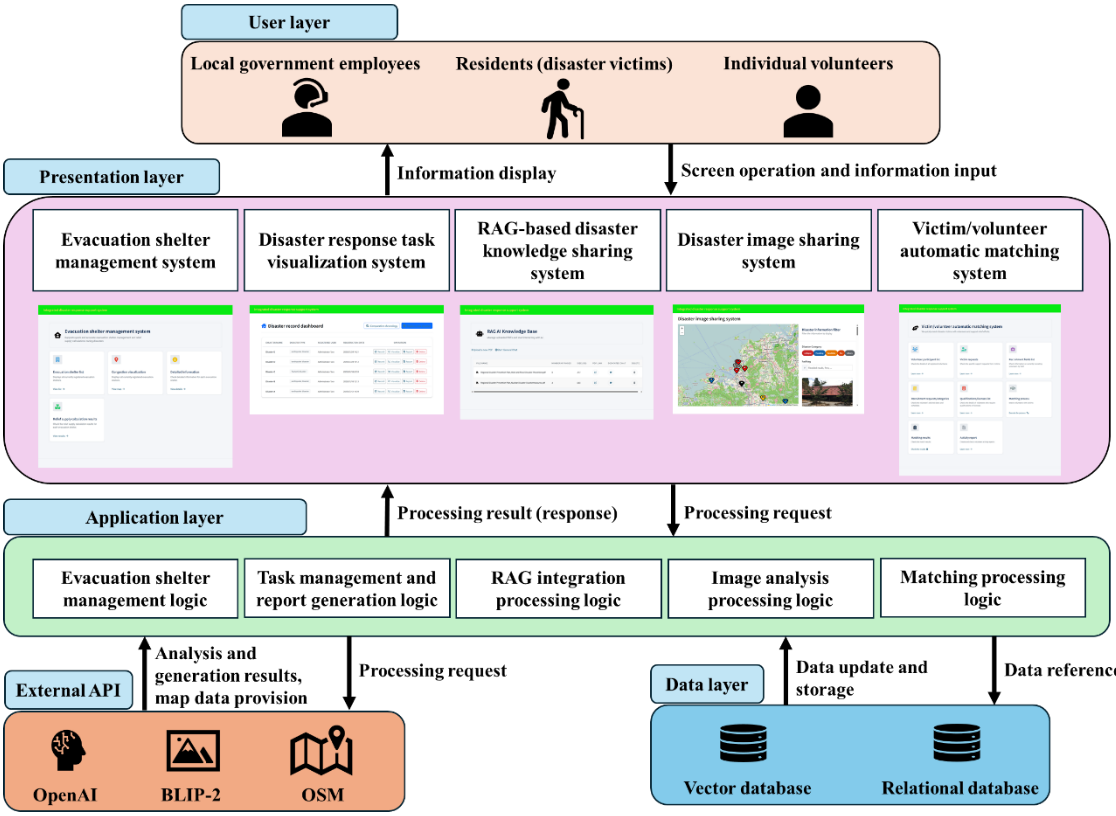


Figure 1. Configuration of proposed integrated disaster response support system

5 Integrated Disaster Response Support System

The five subsystems (i.e., the evacuation shelter management, disaster response task visualization, RAG-based disaster knowledge sharing, disaster image sharing system, and victim/volunteer automatic matching systems) integrated in the proposed system work collectively on an integrated database and a common web application platform to provide integrated support to collect, organize, analyze, and share the information required to support effective and efficient disaster response efforts. The specific functions and design of each subsystem are described in detail in the following sections.

5.1 Integrated Disaster Response Support System Home Screen

Figure 2 shows the home screen displayed to an administrator who logs in to the system. The integrated disaster response support system comprises five subsystems and functions as a common platform that satisfies the requirements of different users. The modules of each subsystem work together under a common authentication and authorization management platform to support disaster-related information sharing and decision-making processes in a central manner.

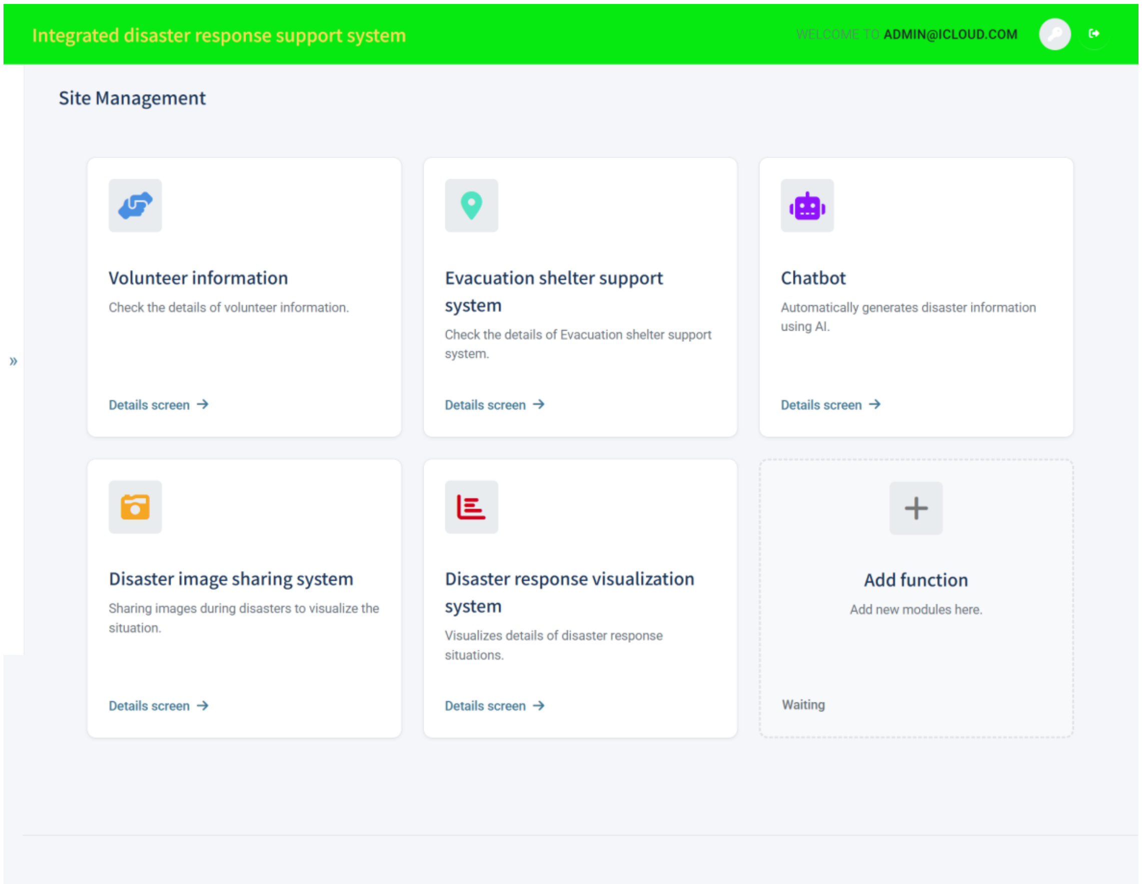


Figure 2. Home screen of integrated disaster response support system

5.2 Evacuation Shelter Management System

The staff at disaster response headquarters must be able to identify the number of evacuees at each evacuation shelter in real time, quickly confirm the evacuation shelter's congestion status, and estimate the required amount of relief supplies. However, currently, evacuee counts and supply requirements are frequently calculated on paper. Thus, time is spent consolidating the relevant information, which delays understanding the situation. Therefore, the proposed system visualizes the congestion status of each evacuation shelter in real time and provides a function to automatically estimate the required amount of relief supplies based on the number of evacuees at the evacuation shelter and their attributes. The home screen of the evacuation shelter management system is shown in Figure 3. The home screen includes four main function buttons, i.e., an Evacuation shelter list button for basic data management, e.g., adding and deleting evacuation shelters, a Congestion visualization button (to display a visualization of each evacuation shelter's congestion status), a Detailed information button to display detailed statistics, e.g., the gender ratio, age group, and population trends at each evacuation shelter, and a Relief supply calculation results button, which automatically calculates the required amount of relief supplies based on the attributes of the evacuees.

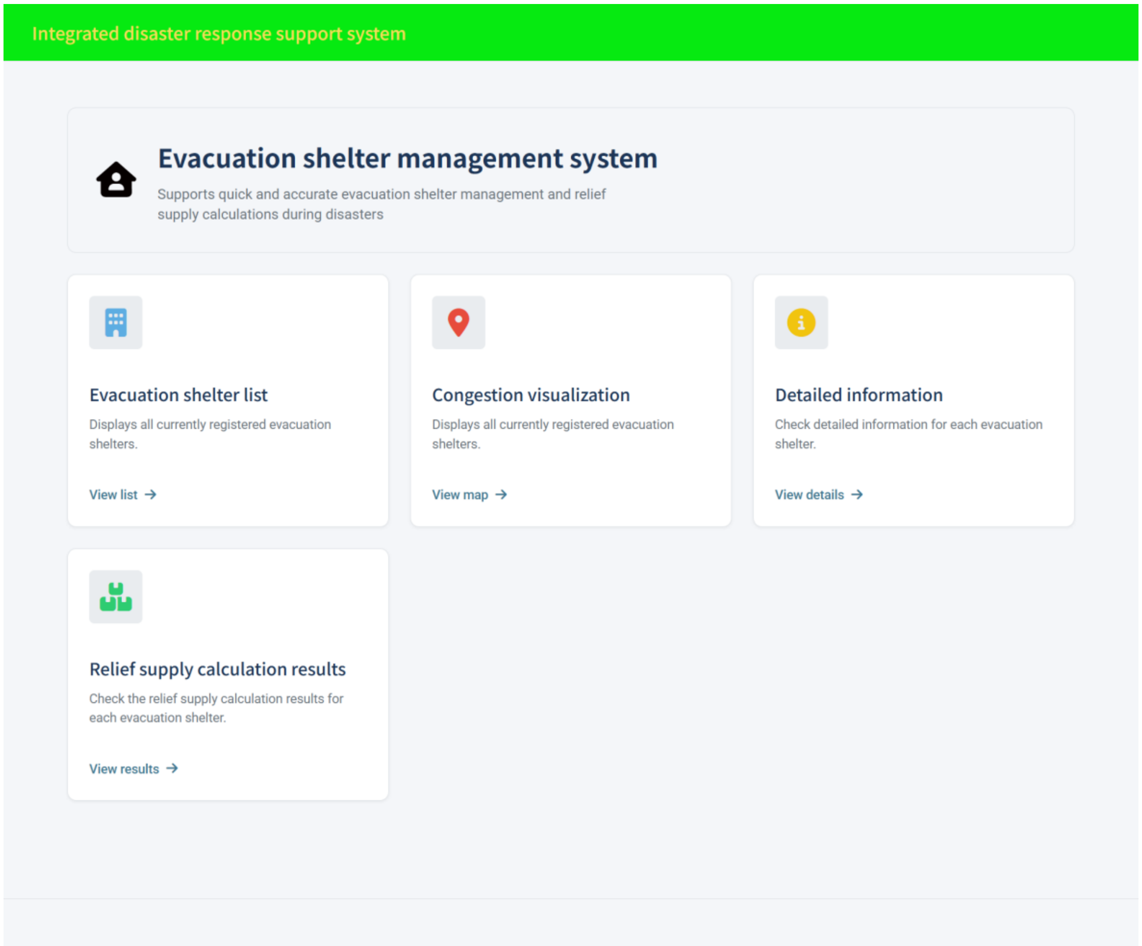


Figure 3. Home screen of evacuation shelter management system

Figure 4 shows the evacuation shelter list screen, which displays evacuation shelter information in a list format, allowing the users to check basic information, e.g., the evacuation shelter ID, name, postal code, address, and capacity. In addition, users can add and delete evacuation shelters, and update existing data from this screen.

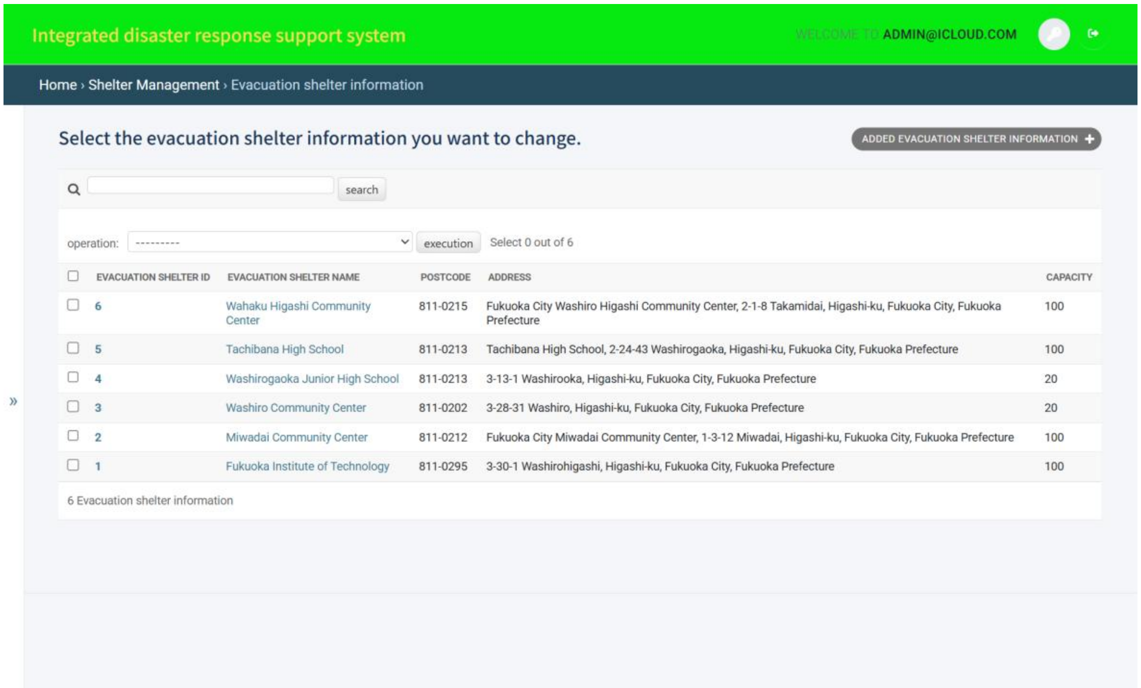


Figure 4. Evacuation shelter list screen

Figure 5 shows the congestion status visualization screen for each shelter. As can be seen, this screen is configured in a two-pane format, with a map area showing geographic information on the left and a sidebar area displaying detailed information about each shelter on the right. Here, the Leaflet [28] map drawing library is employed, and a marker is displayed at the location of each evacuation shelter. The congestion status is visualized using a quantitative threshold. On the server side, the congestion rate $R_i = (P_i/C_i) \times 100$ is calculated from the capacity C_i of shelter i and the current total number of evacuees P_i. In addition, logic is implemented to determine the icon image to display based on the following five-level threshold.

1) $0\% < R_i \leq 20\%$ (Available)

2) $20\% < R_i \leq 40\%$

3) $40\% < R_i \leq 60\%$

4) $60\% < R_i \leq 80\%$

5) $80\% < R_i$ (Nearly full/full)

We implemented a statistics dashboard that allows the users to quickly compare and check the latest status of all evacuation shelters (Figure 6). The UI for this screen utilizes a card-type layout that is easy to read. Here, each card displays basic information about the evacuation shelter and the current total number of evacuees in large characters. Regarding age structure, logic is implemented on the server side to automatically calculate the evacuee's age from their date of birth and the current date, and then the number of evacuees is categorized and aggregated into 11 categories, from infants to those in their 90s and older, in the form of a bar graph, which ensures that the system always accurately reflects the latest demographic trends.

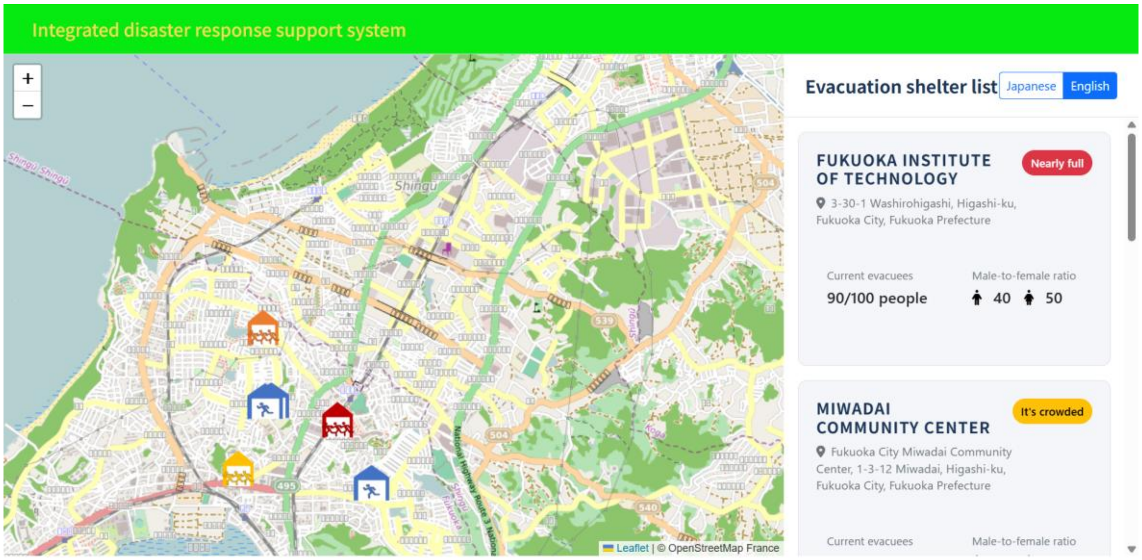


Figure 5. Congestion status visualization screen

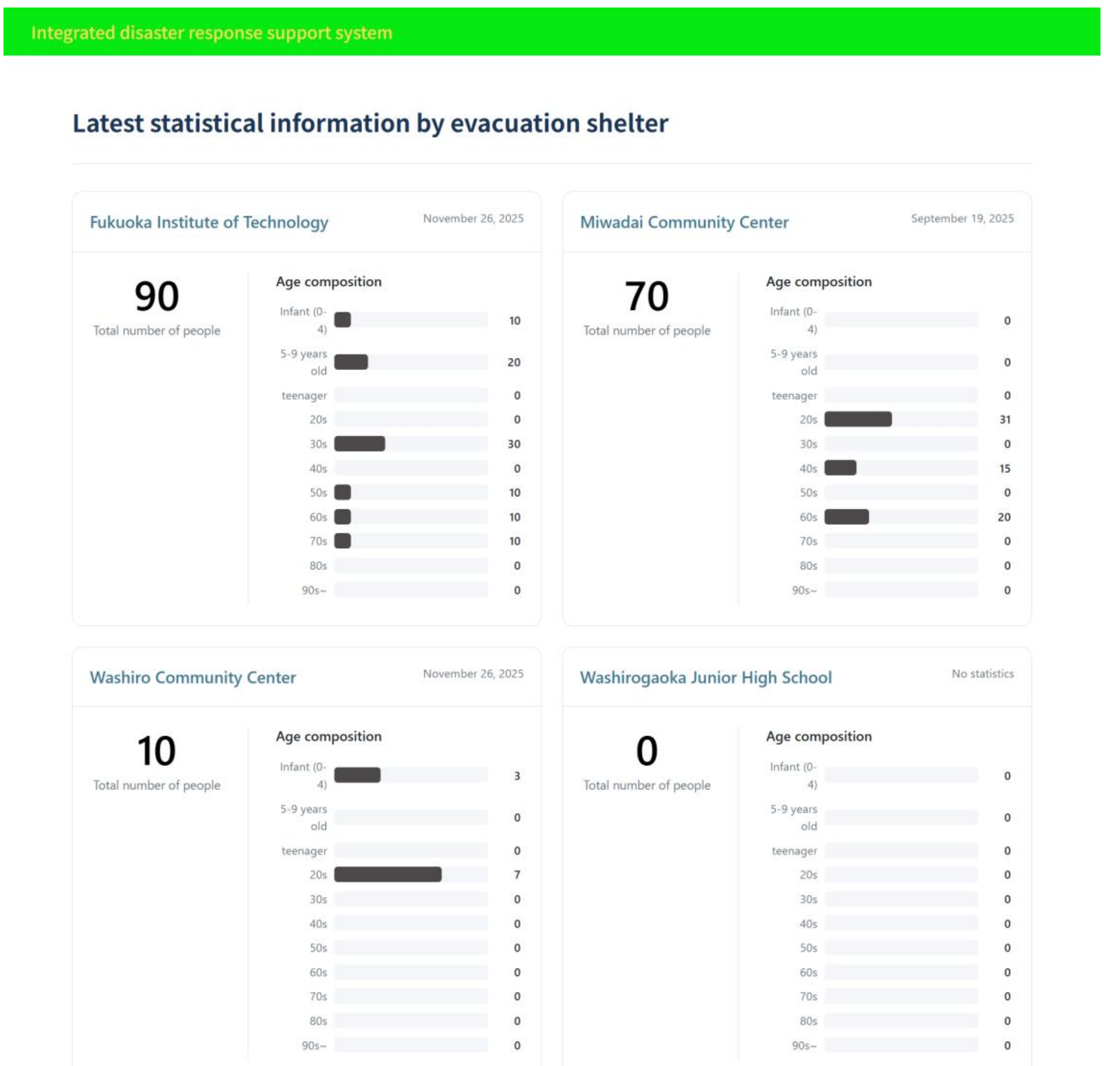


Figure 6. Statistics dashboard for each evacuation shelter

When a user selects a shelter on the statistical information dashboard, the screen transitions to a dashboard screen for comprehensive management of detailed evacuation shelter information (Figure 7). Here, the upper and middle sections of the screen are equipped with visualization functions to support decision-making processes. Specifically, the JavaScript Chart.js library is employed to dynamically plot the gender ratio (as a pie chart), age distribution (as a bar graph), and the trend in the number of evacuees (as a line graph). At the bottom of the screen, the list section prioritizes usability to manage the entry and exit behavior of evacuees. Furthermore, detailed information about each evacuee (e.g., photo, name, damage to house, etc.) is displayed in a table format. Using Bootstrap's [29] tab component, users can smoothly switch between lists of currently resident evacuees and those who have left the evacuation shelter.

Figure 7. Evacuation shelter details dashboard

Figure 8 shows the relief supplies automatic calculation screen for each evacuation shelter. This screen utilizes a matrix format with the evacuation shelters and relief supply items on the vertical and horizontal axes, respectively. A unique feature of this screen is the application of conditional formatting (heat mapping) based on threshold judgment to the calculated supply quantity. When rendering within the template, the following judgment logic is applied to the calculated quantity Q, and the corresponding CSS class is assigned dynamically.

- $Q > 200$: Level-high (red, warning)
- $50 < Q \leq 200$: Level-medium (orange, caution)
- $0 < Q \leq 50$: Level-low (yellow)
- $Q = 0$: Level-zero (grayed out)

This allows administrators to instantly and visually identify darker cells (shelters with high demand for supplies) without having to examine the numbers in great detail. The items included were based on the eight basic push-type relief supplies defined by the Cabinet Office, i.e., food, blankets, portable toilets, adult diapers, toilet paper, sanitary products, infant formula or liquid milk, and infant and child diapers [29].

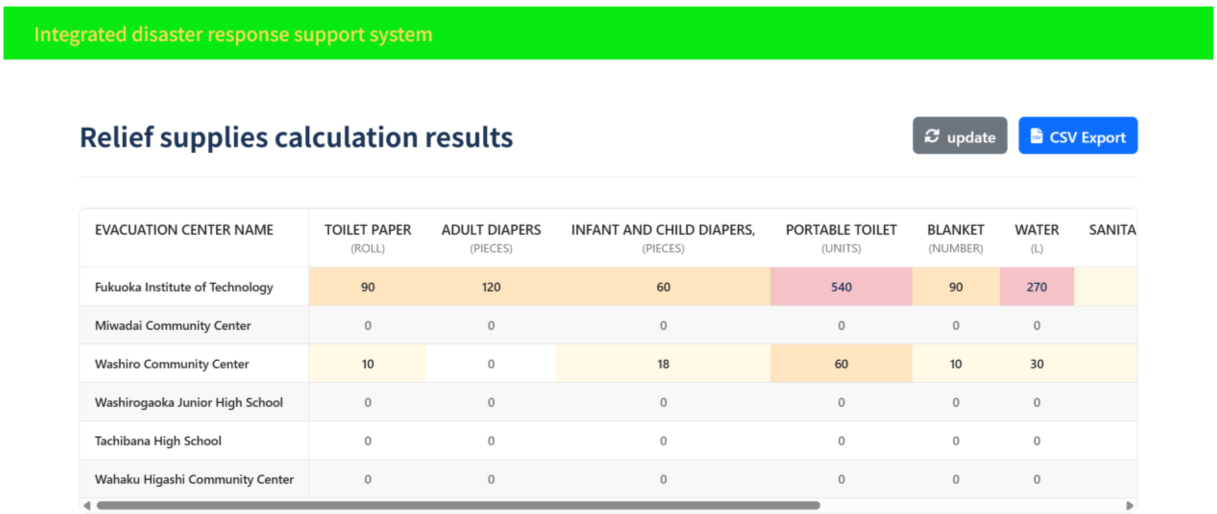


Figure 8. Relief supplies automatic calculation screen

The sequence diagram in Figure 9 shows the main operational flow when local government employees use the proposed system. The system communicates between the client and server via the HTTP protocol, and each function is provided using the following steps. First, for basic operations, e.g., the evacuation shelter list screen, requests involving create, read, update, and delete operations on the database are processed. Then, for congestion visualization, marker information corresponding to the congestion level is received from the server, and map tiles are obtained from the external OpenStreetMap API. These are then overlaid on the browser to create a visualization. In addition, for the relief supplies calculation, the server receives a request, extracts data on the number of evacuees and age composition from the database, performs calculations based on defined units, and returns the results to the client in tabular format or as CSV data.

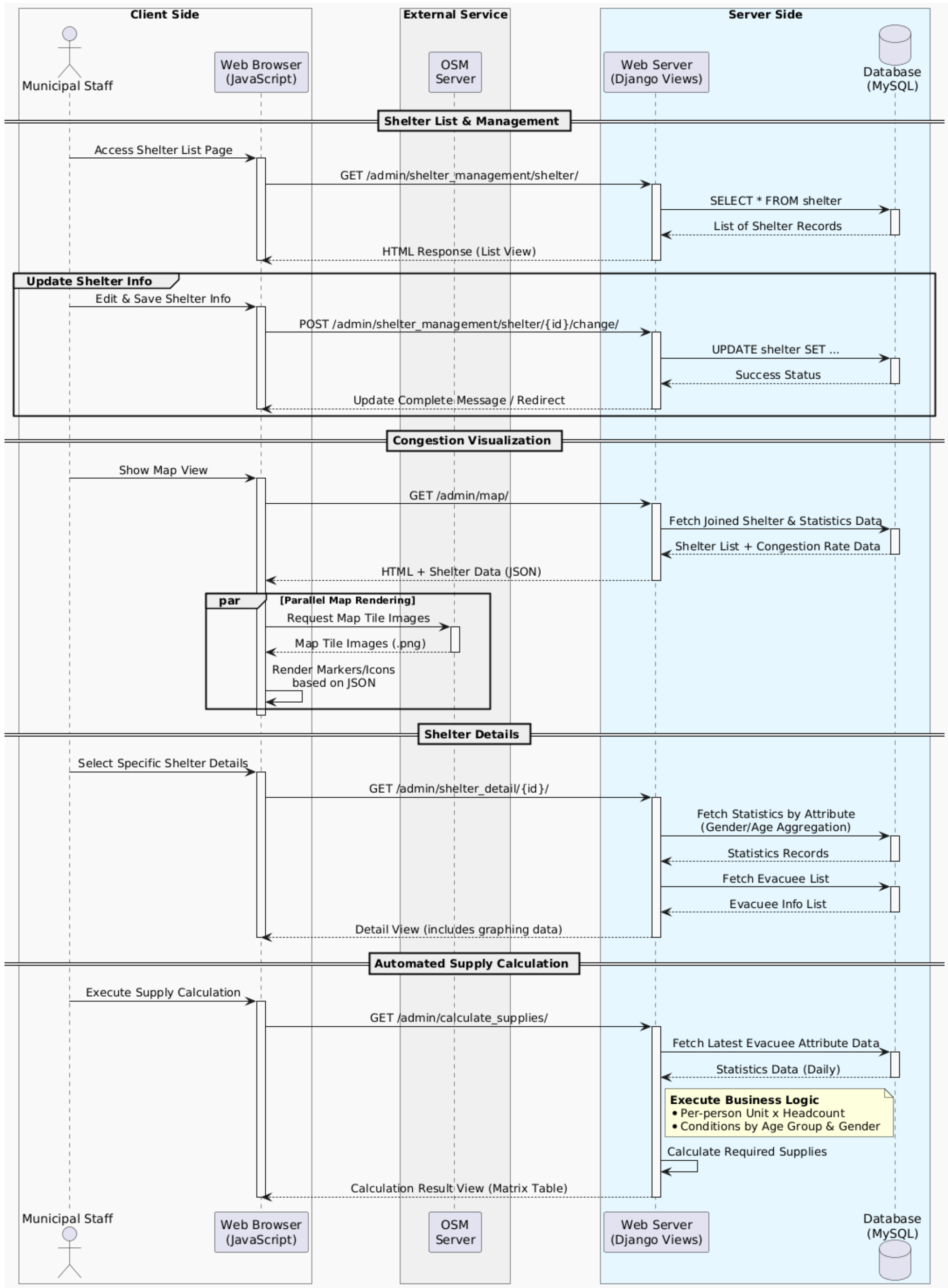


Figure 9. Evacuation shelter management system sequence diagram

5.3 Disaster Response Task Visualization System

The goal of the proposed system is to standardize disaster response operations and support the decision-making processes of local government employees. In addition, it seeks to improve the crisis management capabilities of the entire organization by reducing the risk of task omissions due to personalized knowledge and improving unified situational awareness among employees. The system's main functions include registering disaster occurrence information and listing, editing, and managing response tasks associated with each disaster. Here, each task has attributes, e.g., disaster phase (prevention, reconstruction, etc.), disaster category (e.g., shelter operation, infrastructure response, etc.), and importance (e.g., high, medium, and low). A chronology view (timeline) based on these attributes makes it possible to visualize the overall disaster response scenario. Furthermore, the system automatically generates disaster response reports in Word format (.docx) using the OpenAI API based on task records entered by employees.

The home screen (Figure 10), which serves as the system's starting point, utilizes a dashboard format that consolidates access to key functions for intuitive user operation. Here, Bootstrap 5 was adopted as the UI framework to ensure the visibility and operability of the disaster information list. The screen layout includes global action buttons at the top for "register new disaster" and "comparative chronology," and a list of registered disasters is displayed in the center. Each disaster record has operation buttons for Record, Visualize, Report, and Delete.

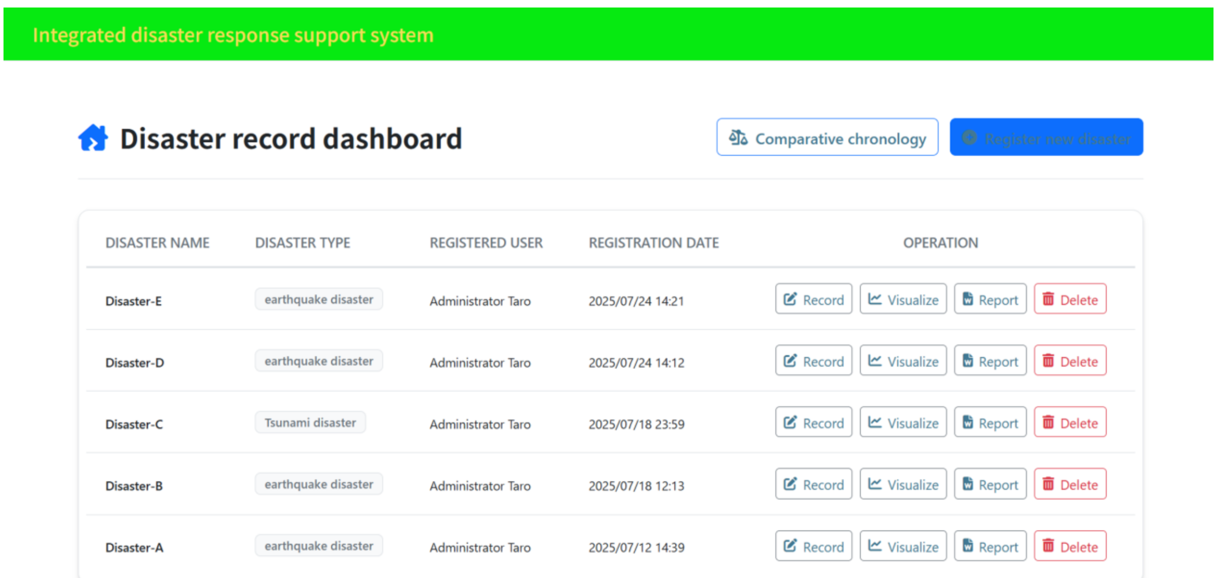


Figure 10. Home screen of disaster response task visualization system

Figures 11 and 12 show the interfaces used to analyze the accumulated response records from multiple perspectives. Here, we employ a method to switch between two different presentation layers for a single data set, i.e., the chronology dashboard views. The chronology view (Figure 11) is a timeline-style view that arranges all tasks in chronological order. By applying dynamic class manipulation using JavaScript to the DOM elements of each task, we autonomously generate a highly visible layout in which tasks are arranged alternately left and right. In addition, the importance of each task (high, medium, or low) is reflected in the border and icon colors using CSS variables, which allows users to visually identify priorities. The dashboard view (Figure 12) arranges each task as a card-type component in a grid, emphasizing status aggregation. In addition, this screen implements real-time filtering on the client side. Here, when the "disaster phase," "importance," "category," and "date" conditions are changed, a JavaScript function scans the data attributes of all DOM elements and instantly controls the display property of elements that do not match the conditions. This makes it possible to quickly extract and analyze only the necessary information from a large number of tasks without having to issue additional requests to the server.

This system also implements a disaster response task comparison analysis screen (Figure 13) to utilize past disaster response records as organizational knowledge. Here, in the backend processing, records corresponding to two user-selected disaster IDs are retrieved individually from the database. The retrieved dataset is then rendered as independent left and right DOM trees using a template engine. A flexbox is employed for the screen layout, displaying two timelines side by side as equal areas. Each column can be scrolled independently; however, a fixed header is applied to display the disaster name, which ensures that the user always knows which disaster is being viewed, even when scrolling through a large number of records.

To maximize the system's practicality, we implemented an AI-based automatic report generation function (Figure 14) that combines the inference capabilities of an LLM with rule-based document generation processing. In the backend processing, all task records associated with the selected disaster are concatenated as chronological text and sent to the OpenAI API. From a prompt engineering perspective, we enforce the output of structured data according to a JSON schema rather than simply requesting a summary, which suppresses output fluctuations specific to the LLM and ensures a stable, standardized format that can be processed by downstream programs. In the document generation phase, the Python Python-docx library is employed to dynamically map the acquired JSON data into a Word document (.docx).

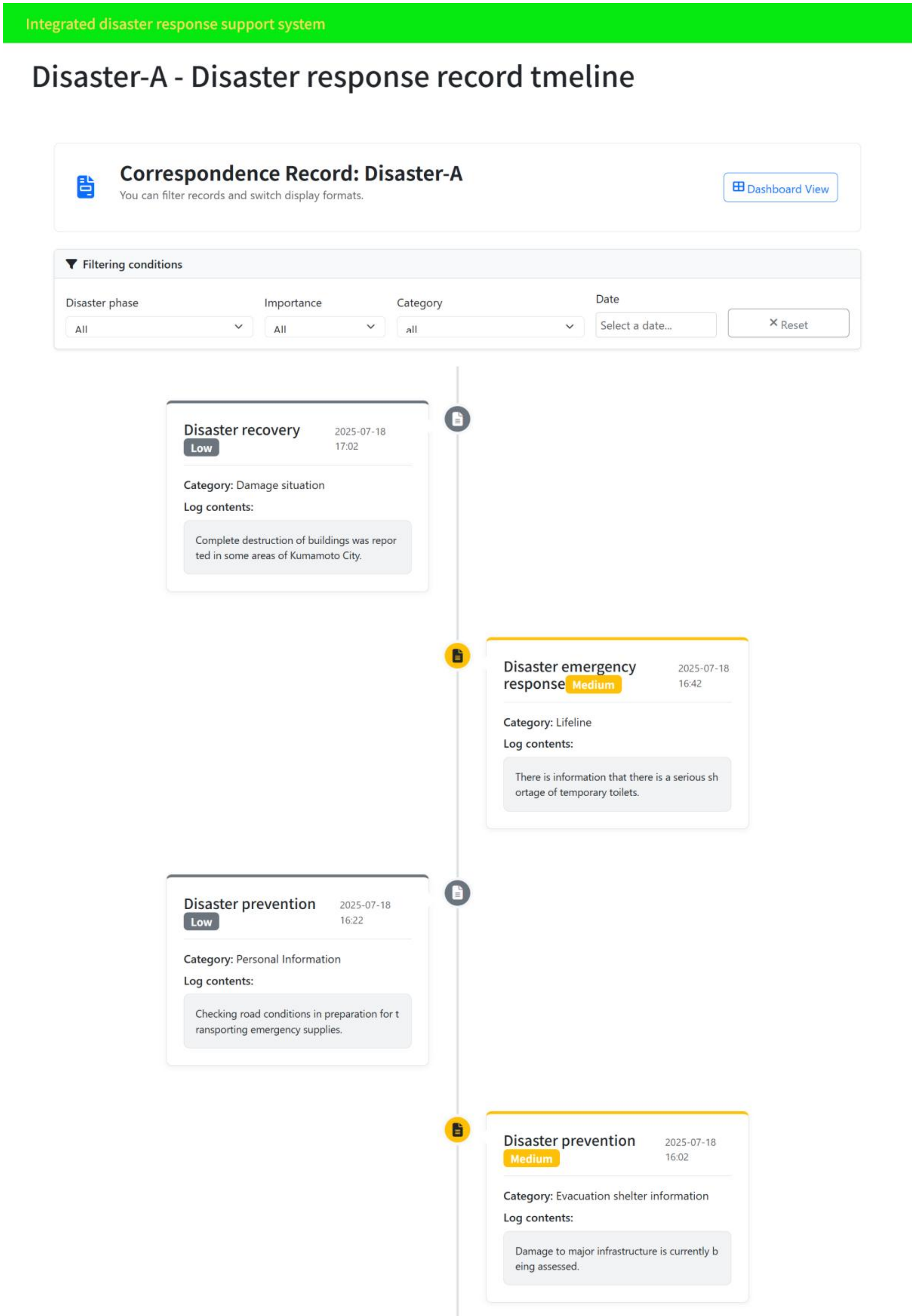


Figure 11. Chronology view screen

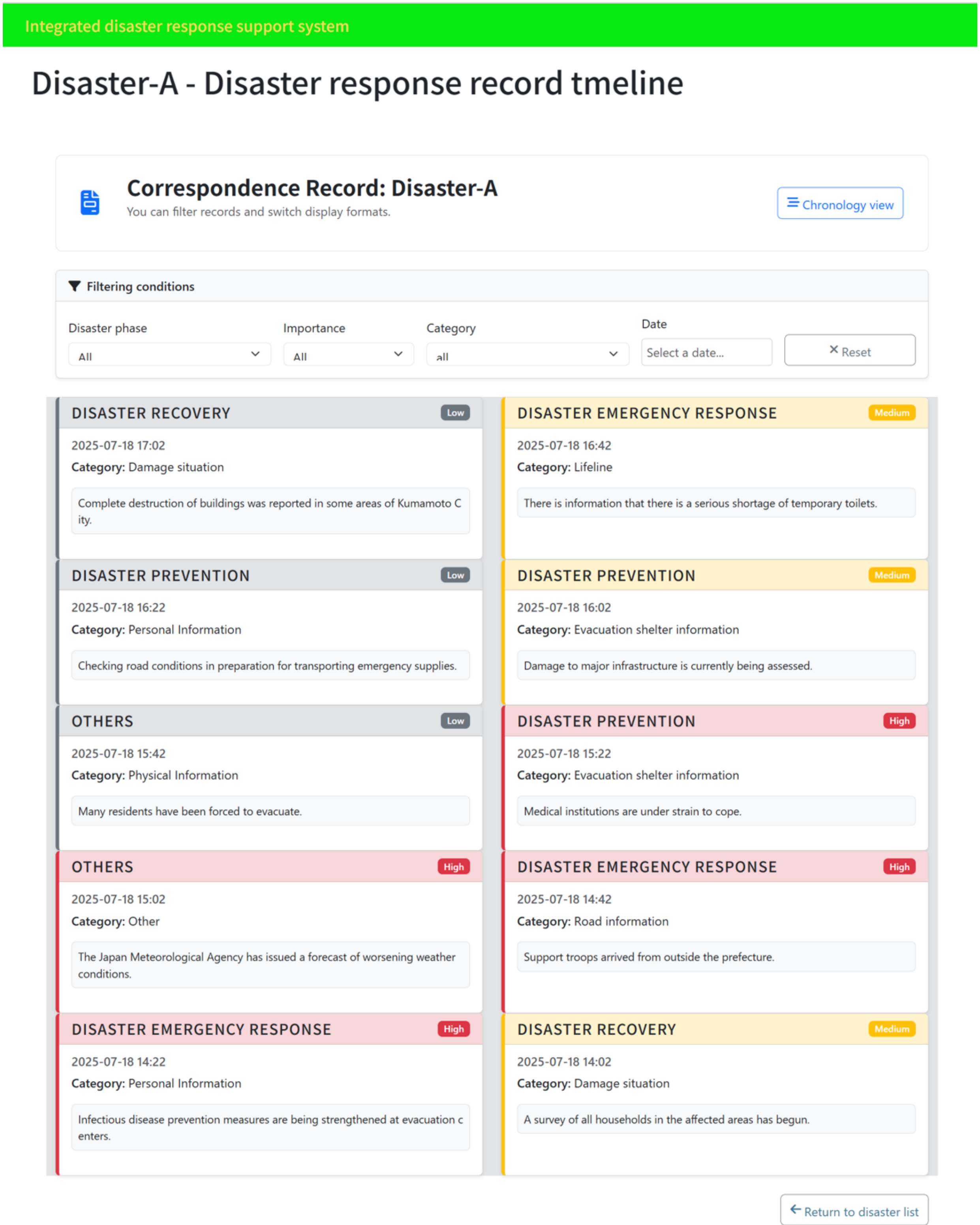


Figure 12. Dashboard view screen

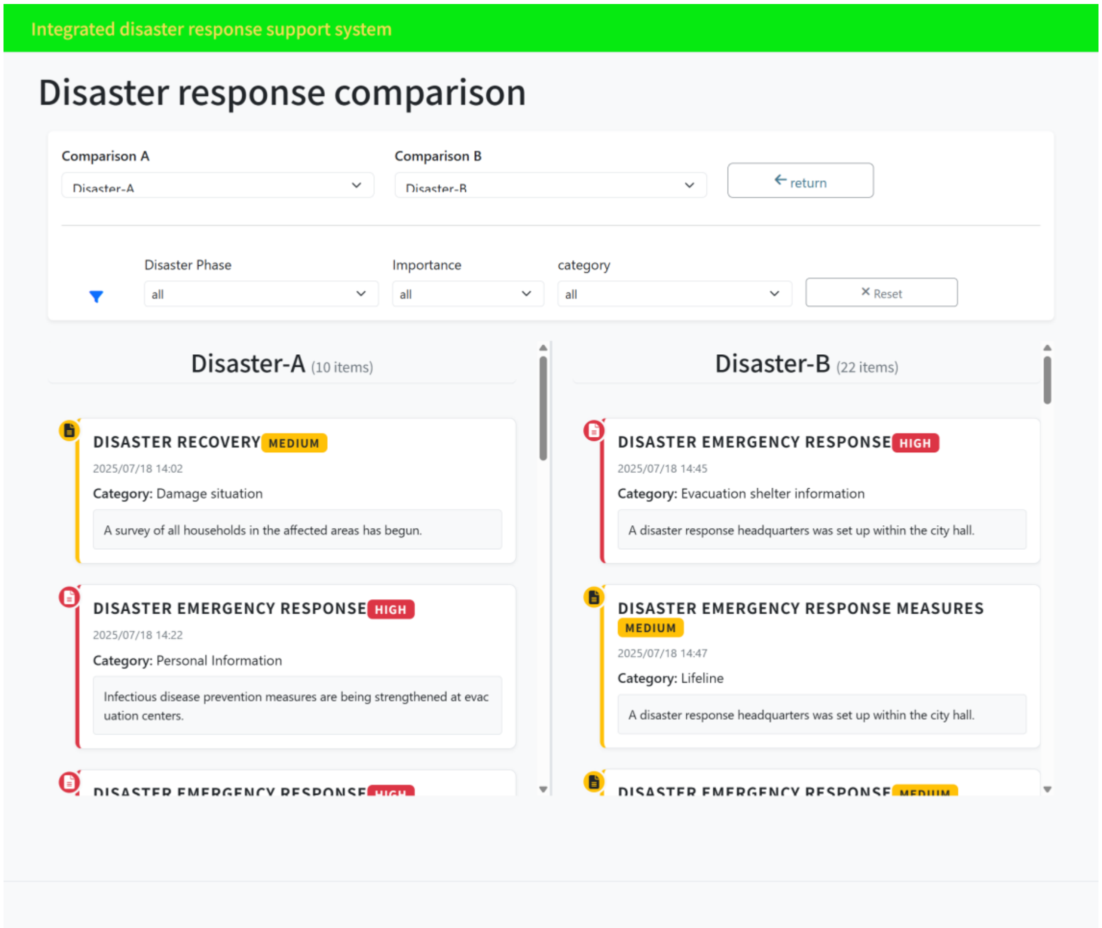
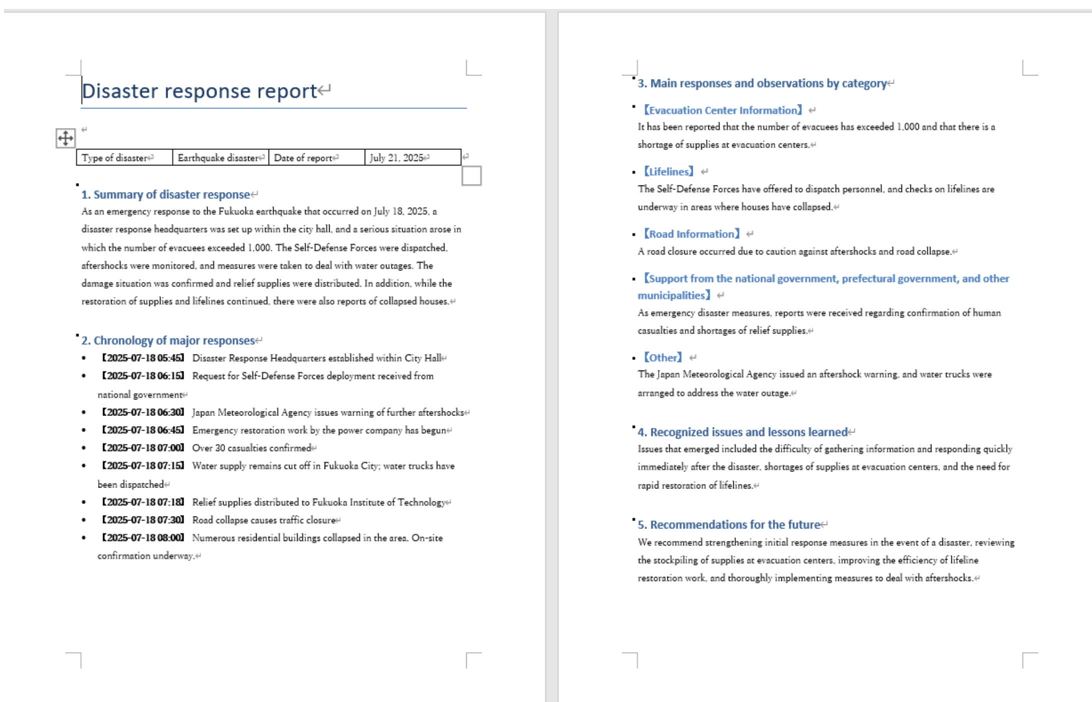


Figure 13. Disaster response task comparison analysis screen



Disaster response report

Type of disaster	Earthquake disaster	Date of report	July 21, 2025

1. Summary of disaster response

As an emergency response to the Fukuoka earthquake that occurred on July 18, 2025, a disaster response headquarters was set up within the city hall, and a serious situation arose in which the number of evacuees exceeded 1,000. The Self-Defense Forces were dispatched, aftershocks were monitored, and measures were taken to deal with water outages. The damage situation was confirmed and relief supplies were distributed. In addition, while the restoration of supplies and lifelines continued, there were also reports of collapsed houses.

2. Chronology of major responses

- 【2025-07-18 05:45】 Disaster Response Headquarters established within City Hall
- 【2025-07-18 06:15】 Request for Self-Defense Forces deployment received from national government
- 【2025-07-18 06:30】 Japan Meteorological Agency issues warning of further aftershocks
- 【2025-07-18 06:45】 Emergency restoration work by the power company has begun
- 【2025-07-18 07:00】 Over 30 casualties confirmed
- 【2025-07-18 07:15】 Water supply remains cut off in Fukuoka City; water trucks have been dispatched
- 【2025-07-18 07:18】 Relief supplies distributed to Fukuoka Institute of Technology
- 【2025-07-18 07:30】 Road collapse causes traffic closure
- 【2025-07-18 08:00】 Numerous residential buildings collapsed in the area. On-site confirmation underway.

3. Main responses and observations by category

- 【Evacuation Center Information】
It has been reported that the number of evacuees has exceeded 1,000 and that there is a shortage of supplies at evacuation centers.
- 【Lifelines】
The Self-Defense Forces have offered to dispatch personnel, and checks on lifelines are underway in areas where houses have collapsed.
- 【Road Information】
A road closure occurred due to caution against aftershocks and road collapse.
- 【Support from the national government, prefectural government, and other municipalities】
As emergency disaster measures, reports were received regarding confirmation of human casualties and shortages of relief supplies.
- 【Other】
The Japan Meteorological Agency issued an aftershock warning, and water trucks were arranged to address the water outage.

4. Recognized issues and lessons learned

Issues that emerged included the difficulty of gathering information and responding quickly immediately after the disaster, shortages of supplies at evacuation centers, and the need for rapid restoration of lifelines.

5. Recommendations for the future

We recommend strengthening initial response measures in the event of a disaster, reviewing the stockpiling of supplies at evacuation centers, improving the efficiency of lifeline restoration work, and thoroughly implementing measures to deal with aftershocks.

Figure 14. Automatically generated disaster response report

The sequence diagram shown in Figure 15 explains how the system's processes are integrated and coordinated. The overall flow involves following four phases, aligned with the decision-making processes of local government employees.

1) Task recording phase

The phase begins with a task input operation by local government employees. The submitted data undergo server-side validation and are persisted to the database with ensured data integrity. Note that the system time and associated disaster ID are assigned automatically.

2) Situation visualization phase

In response to a visualization request, the accumulated data are extracted from the database in chronological order. The server sends the data as a single HTML response, and a client-side (browser) JavaScript handles switching between the chronology and dashboard views, as well as dynamic layout adjustments.

3) Comparison and analysis phase

In this phase, which is aimed at knowledge transfer, the server individually retrieves two different user-selected disaster datasets and renders them as a comparative view. In the browser, the left and right columns can be scrolled independently, and filtering and other operations are synchronized.

4) Report generation phase

In the final stage of the process, all accumulated and analyzed records are processed by AI. Unstructured text collected from the database is converted into structured data via the OpenAI API and output as a Word document.

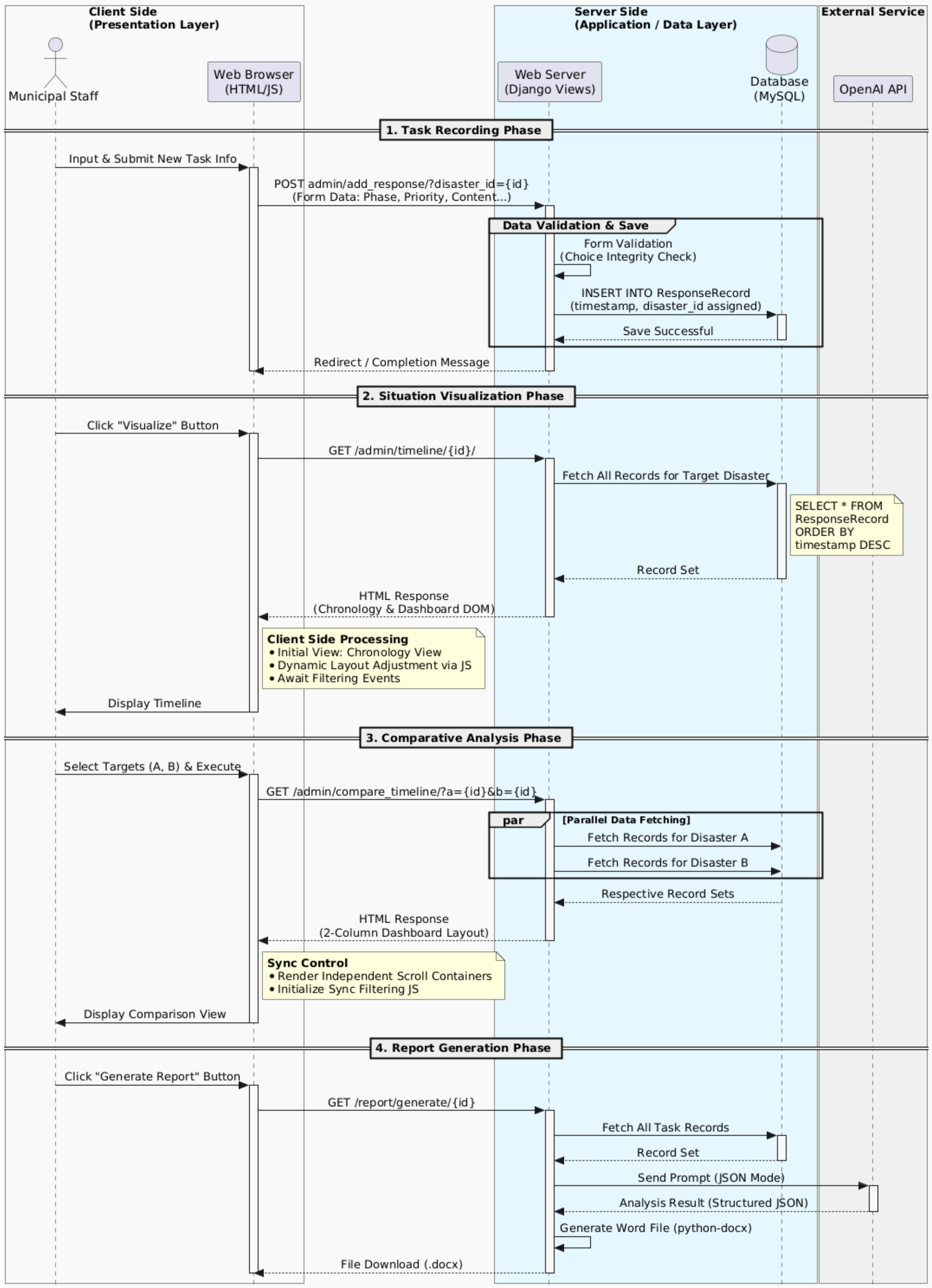


Figure 15. Disaster response task visualization system sequence diagram

5.4 RAG-based Disaster Knowledge Sharing System

Current disaster response efforts frequently fail to make full use of past response examples and disaster prevention plans. This system creates a database of local disaster prevention plans and past disaster response records created by local governments, and it extracts and acquires the required information through vector search using ChromaDB [31]. Then, the acquired information is used to generate natural language responses using an LLM, which facilitates instant search and presentation of the required information when a disaster occurs. The system uploads PDF-format disaster prevention plans and response records, and includes two types of chatbots: one dedicated to each PDF, and one that integrates all data. The dedicated chatbot presents response measures based on individual plans, and the integrated chatbot proposes comprehensive response measures across multiple datasets.

Figure 16 shows the home screen, which serves as the starting point for the chatbot function. This screen functions as a dashboard to centrally manage the uploaded PDF files. In the center of the screen, the list view displays relevant attribute information, e.g., the title, number of pages, and size (KB) of the PDF files registered in the database. Each row integrates an operational interface. Here, clicking PDF LINK navigates to the original text verification screen, and clicking DEDICATED CHAT navigates to a RAG dialogue screen focused on the context of a specific document. The action area at the top of the screen also contains a Start General Chat button. When this button is selected, the screen transitions to a comprehensive question-and-answer screen for all uploaded PDF files.

Figure 17 shows the system's PDF upload screen. This function converts and registers document assets held by local governments into a format that can be searched by the RAG system. When a file is selected and sent from the interface, the system receives the PDF file and begins the analysis process. The system is designed to handle storage in MySQL to maintain document management consistency and vector registration in ChromaDB to enable semantic search in parallel. Specifically, text, tables, and image data are extracted from the uploaded PDF, and then vector embedding is generated. Note that this goes beyond simple file storage; it automatically creates a knowledge base for AI to understand context.

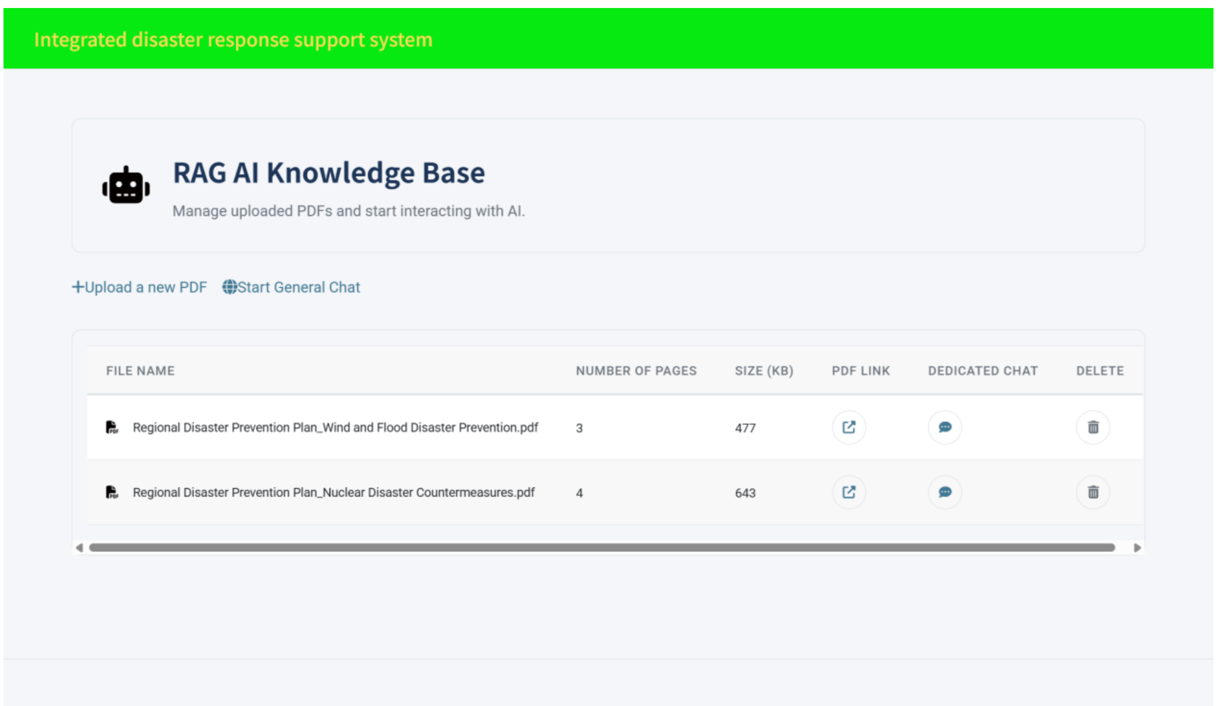


Figure 16. Home screen of RAG-based disaster knowledge sharing system

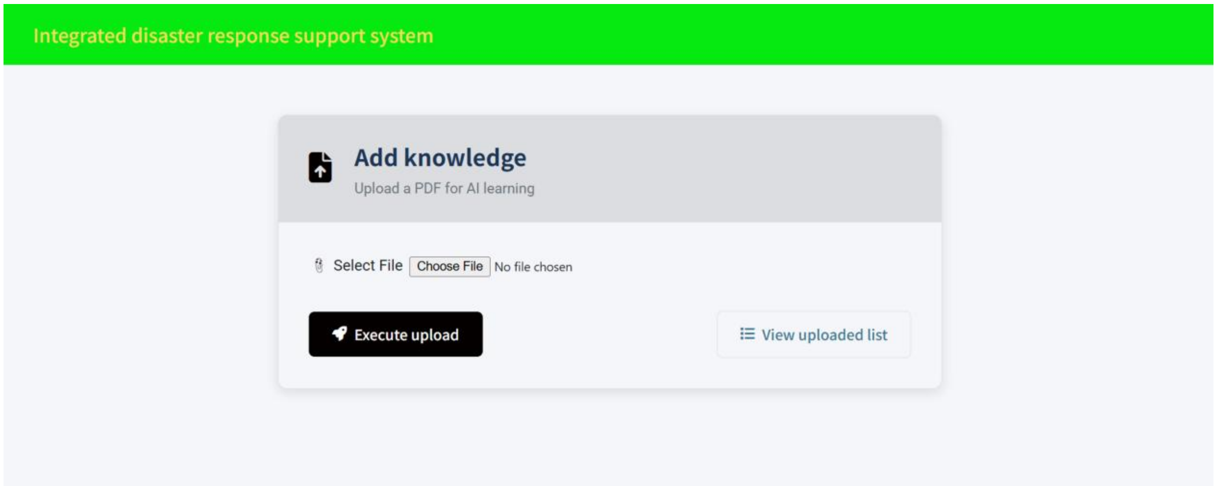


Figure 17. PDF upload screen

The chatbot screen is shown in Figure 18. Here, the user's query and the system's response are displayed on the right and left sides of the screen, respectively. When a user submits a question, the following inference process is executed in the backend.

1) Context search

The input natural language query is vectorized, and relevant document chunks are extracted from ChromaDB.

2) Prompt construction

The extracted knowledge data are combined with strict system prompts to suppress hallucination and sent to the LLM.

3) Answer generation

The answer data generated by the LLM are received and rendered in chat format.

This process enables the presentation of well-founded answers based on the uploaded disaster response plan.

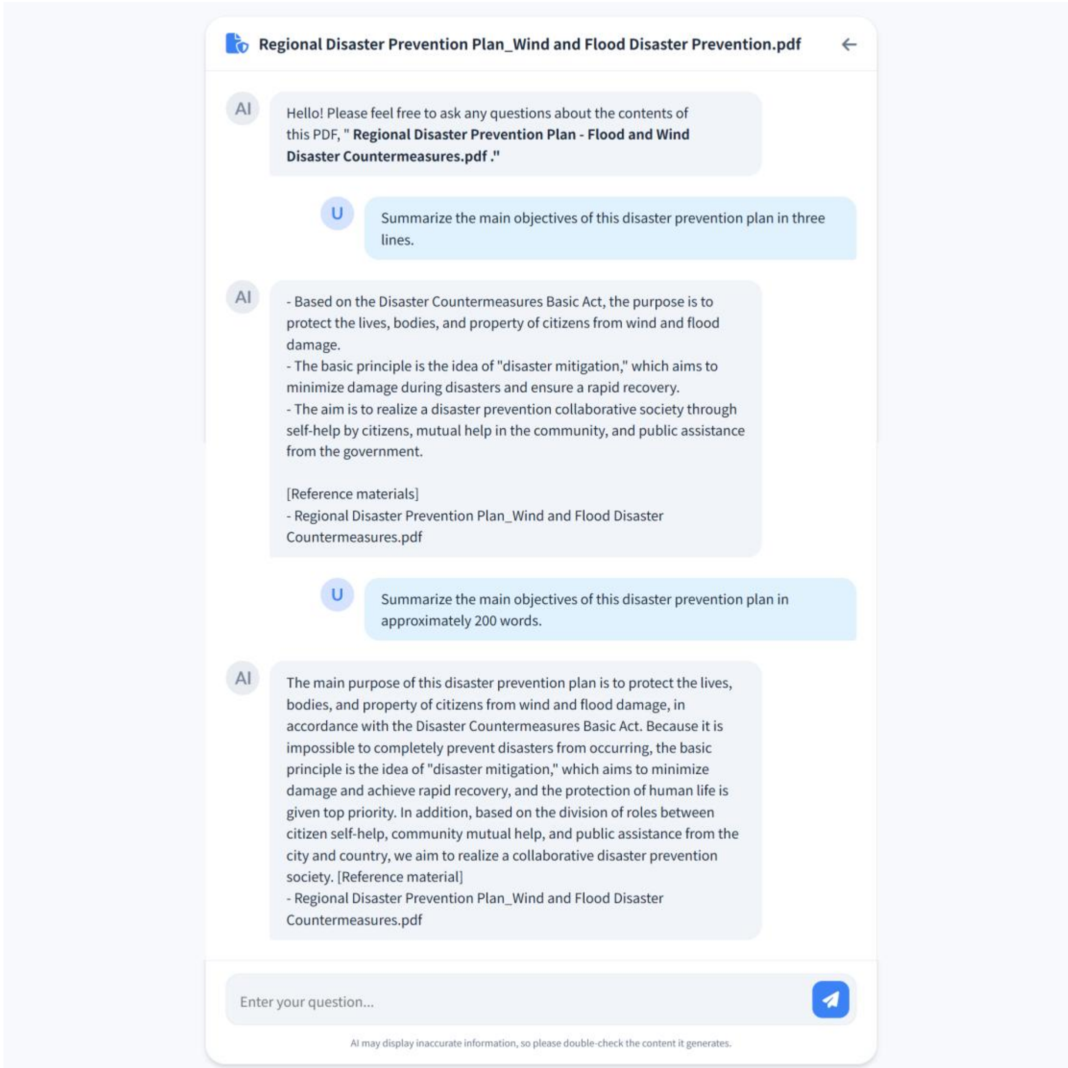


Figure 18. Chatbot screen

The operation of this system involves two main processing flows, i.e., the knowledge base construction phase, in which knowledge data are accumulated, and the information search and answer generation phase, in which answers are generated using the accumulated data. Figure 19 shows the overall sequence diagram of this system.

- Knowledge base construction process (PDF upload and vectorization)

The first flow involves an administrator registering documents, e.g., disaster response plans and response records, into the system.

1) When an administrator uploads a PDF file through the web interface (Django admin screen), the analysis module in the application server is launched, and extract, transform, and load processes are performed.

2) Text extraction and preprocessing. The PDF analysis engine pdfplumber (Python library for extracting text, images, tables, etc. from PDF files) is employed to extract text data from unstructured PDF files.

3) Vector indexing. The extracted text is divided into semantic chunks and then converted into high-dimensional vectors using an embedding model. The converted vector data is then registered in

ChromaDB with relevant metadata, e.g., the file ID. This enables searches based on keyword matches and semantic similarity.

4) Metadata management. Simultaneously, management information, e.g., file name, file size, and upload date and time, is persisted in MySQL to maintain consistency.

- Search and answer generation process (RAG inference)

The second flow is the interaction process when an end user, e.g., a local government employee, uses the chatbot.

1) Query translation and vector search. A user-submitted natural language question is first converted into a query vector using an embedding model. The system uses this vector to perform a nearest neighbor search on ChromaDB to extract text chunks that are semantically relevant to the question.

2) Prompt construction and generation. The application server integrates the extracted related knowledge and the user question according to the system prompt, and it constructs an input prompt for the LLM.

3) Answer generation and history storage. The LLM generates an answer based on the provided context and returns it to the system. The generated answer is then displayed on the UI and simultaneously recorded in MySQL as a Q&A pair.

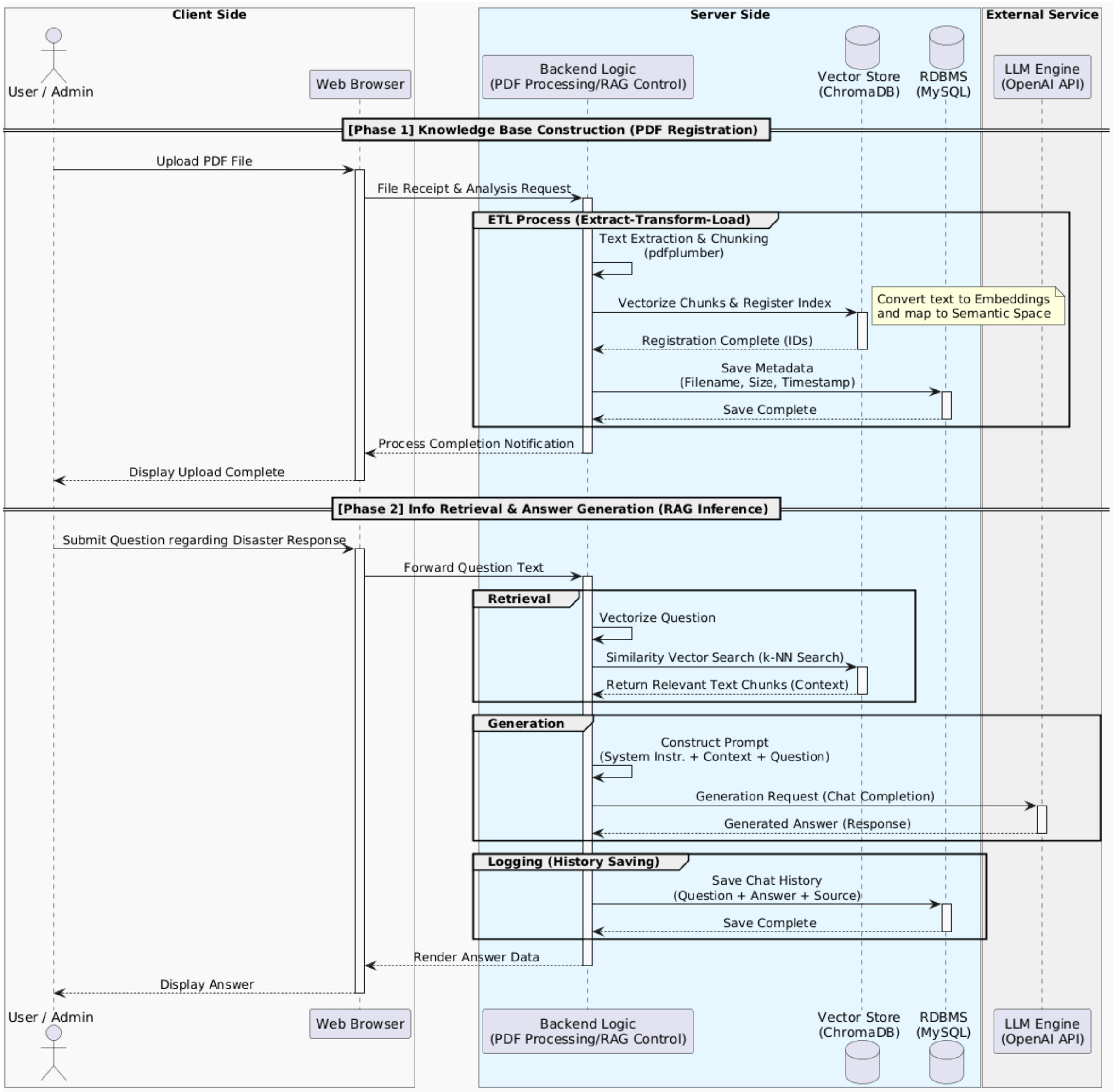


Figure 19. RAG-based disaster knowledge sharing system sequence diagram

5.5 Disaster Image Sharing System

Recently, with the spread of smartphones and social media, citizens are increasingly sharing images of disaster sites in real time during disasters. While these posted images contain useful information about the damage on the ground, the sheer volume of images poses a challenge, i.e., collecting, organizing, and utilizing the information contained in such images takes time. Thus, we propose a system that automatically summarizes, classifies, and visualizes damage situations based on citizen-submitted disaster images. Specifically, citizens submit disaster images with location information through a web form, and captions describing the damage are generated from the images. Then, an LLM is employed to generate image titles from the captions and automatically classify the damage categories based on the damage content. The images are then plotted on a map. This system is expected to structure the large number of citizen-submitted images in real time, thereby enabling an intuitive overall understanding of the disaster site. This provides an effective framework for citizen participation in disaster information sharing and supports rapid decision-making processes by government agencies and relief organizations.

In the following, we describe the implementation details, from users posting disaster images to image analysis and metadata extraction within the system.

1) Configuration of the posting interface

Figure 20 shows the interface used to post disaster images. The left and right sides of Figure 20 show the initial state and the transition screen after uploading is complete, respectively. Here, the users enter the following information through the web form.

1. Disaster image (required): a photo file of the disaster situation
2. Submitter name (optional): submitter's identifier (if not entered, the user will be treated as anonymous)
3. Location information (optional): a text area for manual input in case automatic acquisition is difficult

2) Extraction of location information from image metadata

On the server side, location information is extracted from the binary data immediately before the image is saved. Specifically, the Exif header (photographic information and metadata automatically embedded in photographs captured using a smartphone) of the image file is analyzed. If the image contains GPS information, this is recorded in "degrees, minutes, and seconds" format. This is then converted to decimal "degrees" format as follows.

$$Decimal = Degress + \frac{Minutes}{60} + \frac{Seconds}{3600} \quad (1)$$

3) Image caption generation using BLIP-2

After physical storage of the image is complete, the system immediately begins verbalizing the image content. This system employs BLIP-2 as the visual language model. Here, the caption generation function converts the input disaster image into RGB format and prompts the user with "Describe the image in detail." Then, the system generates text describing the context of the image (collapsed buildings, flooding, etc.). The generated text is stored in the image caption model and serves as input for subsequent processing by an LLM.

4) Damage classification and tag extraction using LLM

The generated captions are structured semantically using an LLM. The processing flow is summarized as follows.

1. Prompt engineering

The LLM is given captions as input and is instructed to determine the disaster classification and extract hashtags. The output format is strictly specified to prevent parsing errors.

2. Automatic disaster classification

The system classifies the disaster into one of five categories (i.e., collapse, flood, landslide, fire, and other) and records the results in a table.

3. Hashtag generation

Important objects in the image (e.g., cars, telephone poles, or rubble) are extracted as nouns and linked to the image via a hashtag model and many-to-many relationships.

This process creates a system where, when a user uploads an image file, location information is identified, the situation is verbalized, the type of damage is classified, and search tags are added automatically in real time.

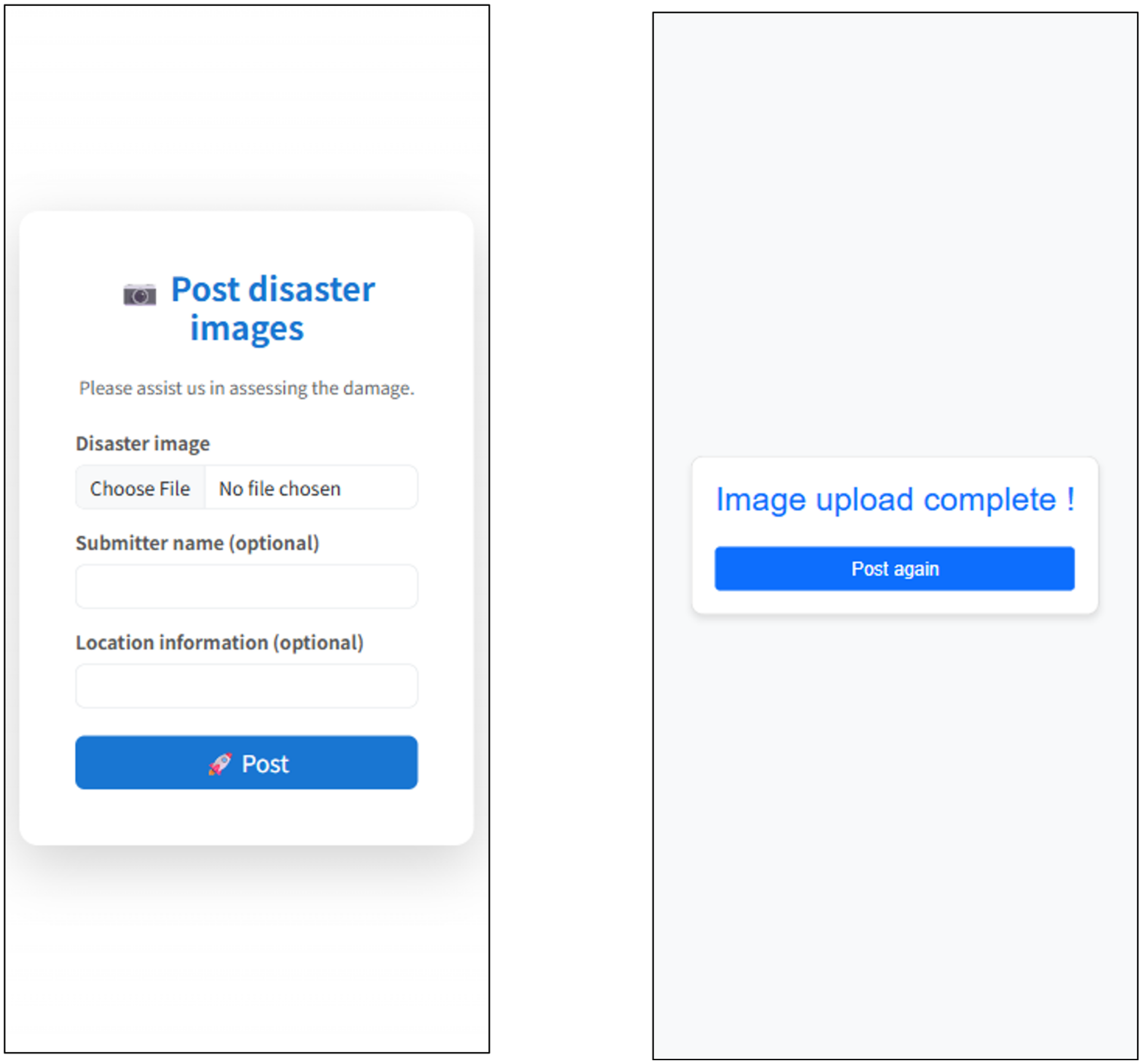


Figure 20. Disaster image posting screen

In the following, we describe the implementation of an interface that allows administrators to grasp the collected and structured disaster information in both a bird's-eye view and in detail.

1) Screen configuration and data linkage mechanism

Figure 21 shows the initial state of the administrator screen. As can be seen, this screen has a two-column layout with a map drawing area on the left and a sidebar on the right. In this system, the pins on the map and the image cards in the sidebar are closely linked.

2) Visual filtering

Figure 22 shows the color-coding rules for pins according to the disaster type. Within the JavaScript logic, predefined icon assets are assigned dynamically based on the categories (collapse, flood, landslide, etc.) classified by the LLM. This allows administrators to instantly grasp the visual trends of the damage by simply viewing the map.

3) Real-time search using multiple criteria

Each pin is associated with a pop-up window that displays the corresponding image, category, caption, and hashtag when clicked. Here, a filtering control panel is implemented at the top of the sidebar, and two types of filters are available, i.e., Category selection (checkboxes) and Hashtag search (text input). Note that filtering is completed using client-side JavaScript array manipulation without communicating with the server, which allows the map pins and image list to be redrawn without delay in response to user input, thereby enabling rapid information filtering.

With this implementation, this system can quickly extract necessary information from vast amounts of disaster image data and present it in a manner that facilitates effective and efficient decision-making processes.

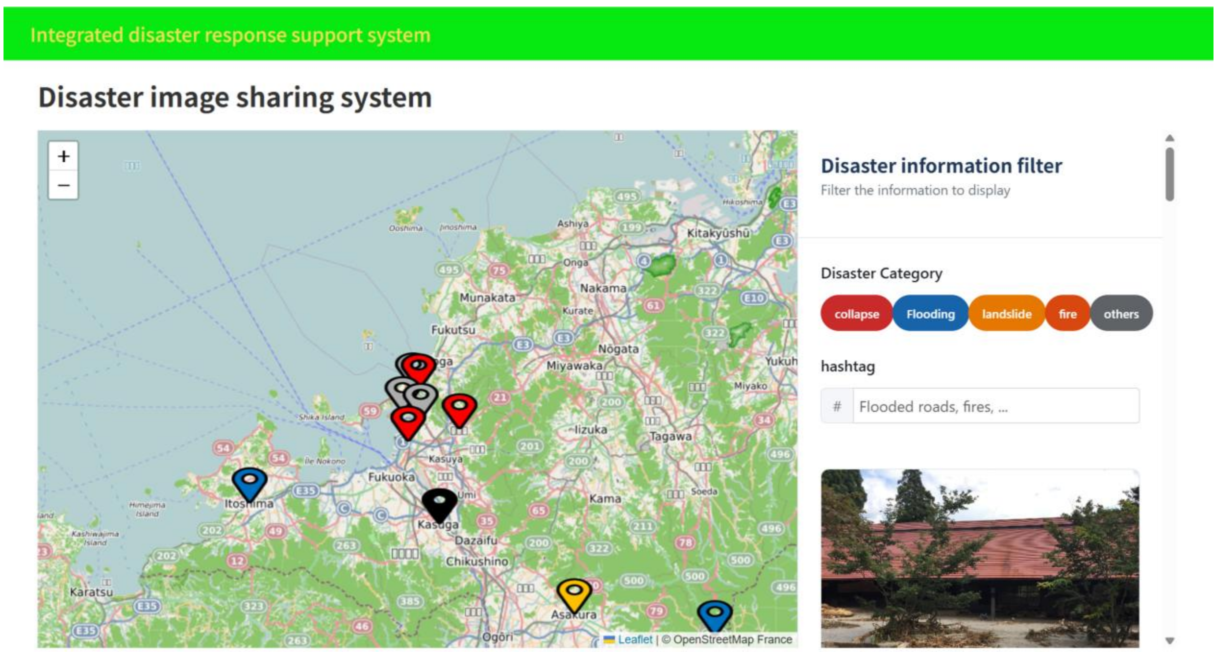


Figure 21. Initial state of disaster image sharing system's administrator interface

Figure 22. Color coding of pins by disaster category

Figure 23 shows a sequence diagram of the disaster image sharing system, which illustrates the entire information processing flow, from posting disaster images to displaying the images on a map. First, victims post images taken at the disaster site and their location information to a web server using a device, e.g., a smartphone. Then, the web server sends the received images to a caption generation agent, requesting the generation of natural language captions. The generated captions are then sent to the OpenAI API, which extracts hashtags related to the type of disaster (e.g., fire or flood) from the text. This assigns classification information to the images. The web server stores a series of information, including images, captions, classification results, hashtags, and location information, in a database for subsequent visualization. Next, when an administrator checks the disaster situation, the application server retrieves the relevant information from the database and displays pins on a map (via the OpenStreetMap API) that correspond to the location and classification of each post. Administrators can efficiently identify necessary disaster information using the filtering functions based on the classification results and assigned hashtags. As a result, the system integrates the entire process from post -> analysis -> classification -> storage -> visualization, which enables rapid information sharing.

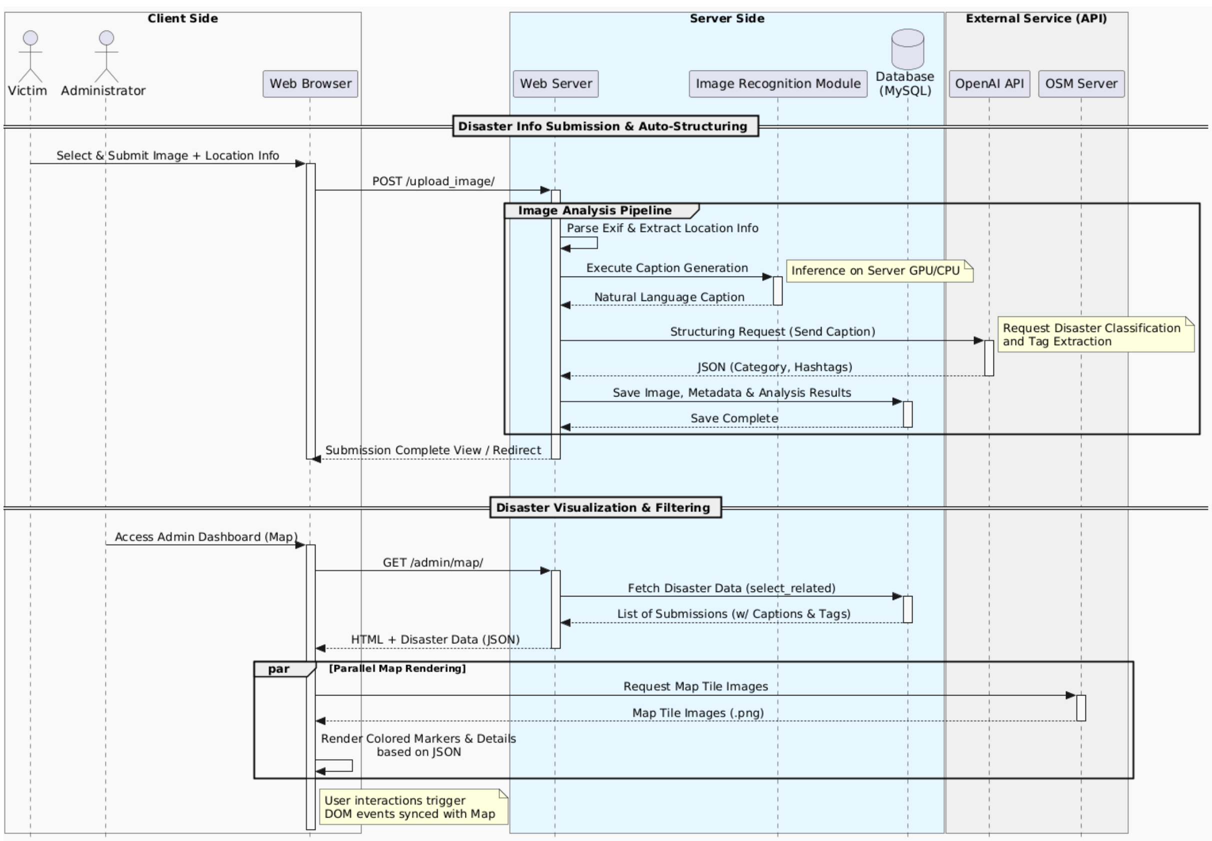


Figure 23. Disaster image sharing system sequence diagram

5.6 Victim/Volunteer Automatic Matching System

When a large-scale disaster occurs, disaster volunteer centers are opened. These centers collect information about the needs of many individual victims and disaster volunteer participation. However, the current volunteer matching system collects information on paper, which is then coordinated manually, which makes it difficult to respond quickly and effectively. Thus, the proposed system enables automatic matching between victims and volunteers. Based on the victims' requests and volunteer requirements, matching is performed automatically using an algorithm, taking into account the necessary qualifications and experience, and completing the matching quickly and efficiently. This enables the victims' needs to be satisfied quickly and is expected to facilitate smooth operation of the necessary relief activities.

Figure 24 shows the home screen, needs registration screen, and matching result screen of the disaster victim needs registration system. This system comprises two sections, i.e., the Register/edit needs and Confirm matching results sections. In the needs registration function, disaster victims select their address, phone number, recruitment date and time, number of people recruited, recruitment area, recruitment field, required qualifications/licenses for volunteers, desired qualifications/licenses for volunteers, and other/notes. After registration is complete, the system queues the data to match the registration data with volunteer data waiting in the background. If a match is found, the matching results screen displays information about the matched volunteer.

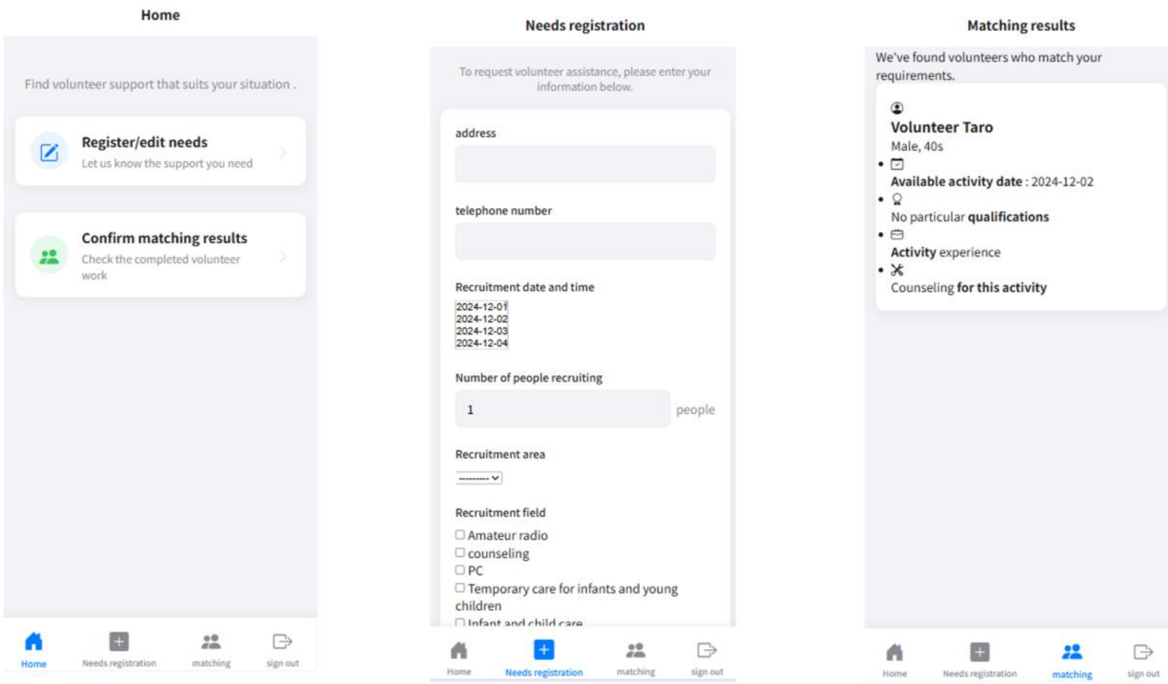


Figure 24. Home screen, registration screen, and matching result screen of the disaster victim needs registration system

Figure 25 shows the home screen, volunteer registration screen, activity report screen, and matching result screen of the volunteer registration system. This function involves three sections, i.e., the Register/edit volunteer, Confirm matching results, and Activity report sections. On the volunteer registration screen, the volunteers select their address, phone number, whether they have volunteer insurance, available participation dates and times, activity area, activity field, qualifications/licenses held, activity experience, activity details, and other/notes. After registration is complete, the system queues the data for matching calculations with disaster victim data waiting in the background. After completing a volunteer activity, the activity report screen allows users to enter the activity name, activity area, activity duration, and activity details to complete the report. If a match is found, the information about the matched disaster victim is displayed on the matching result screen.

The system's administrator home screen (dashboard) is shown in Figure 26. Here, the screen design utilizes a card-type grid layout to enable administrators to make quick decisions. In addition, the cards that lead to each function feature highly visible icons and concise descriptions, allowing for intuitive operation. The main management items displayed on the screen serve as a hub to centrally manage all information within the system and are divided into the following four categories.

1) Resource management

- Volunteer participant list. This list is used to manage the skills and availability of all registered volunteers.
- Disaster victim request list. This list is used to manage the specific needs (location and type) of disaster victims requiring assistance.

2) Matching parameter management

- Date and time, field of recruitment, and qualifications list. This list is employed to manage and update the master data that serve as the matching criteria for the matching algorithm.

3) Process execution

- Matching processing. This triggers the algorithm to calculate the optimal match.
- Matching results. This checks the pairing results based on the calculated score.

4) Activity records

- Activity reports. Activity records are accumulated after matching is completed.

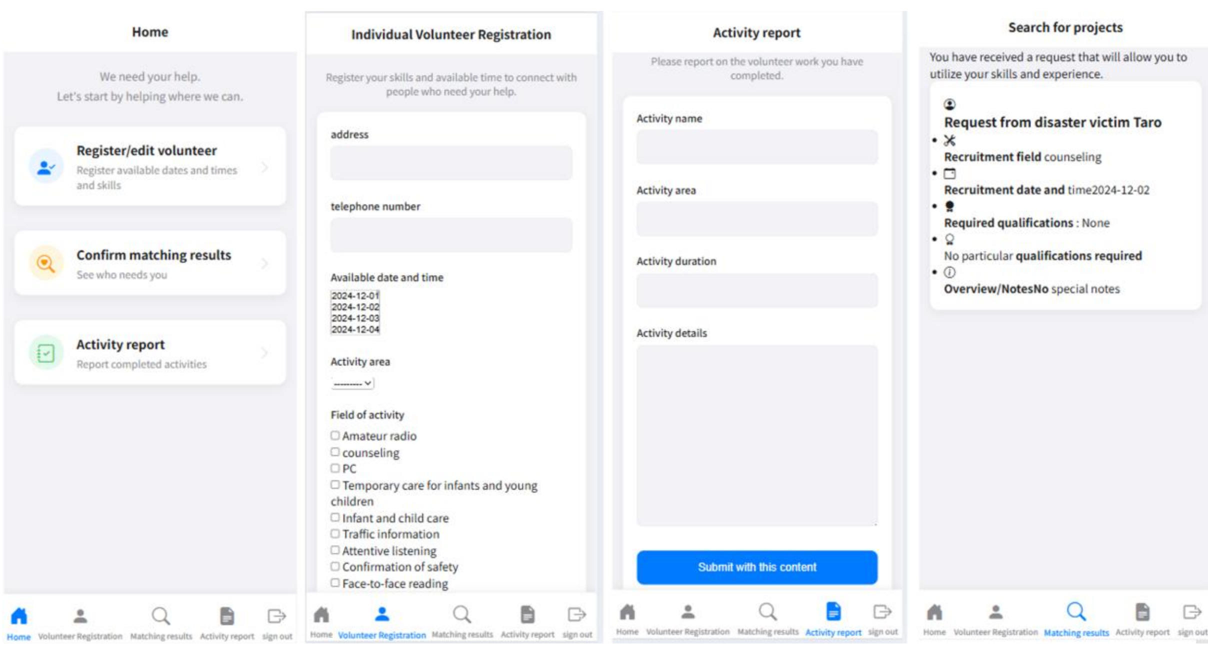


Figure 25. Home screen, volunteer registration screen, activity report screen, and matching result screen of the volunteer registration system

Note that this dashboard is designed to serve as the starting point for the entire workflow of relief activities (check registration status -> execute matching -> approve results -> report activities). Through this screen, administrators can monitor the status of each datum and quickly execute matching processes as required, thereby ensuring smooth provisioning of information to both the victims and volunteers.

To ensure fairness, the matching process queues unmatched disaster victim needs in order of registration date and time (in a first in, first out (FIFO)) manner, and an algorithm is employed to select the most suitable volunteer for each need. The overall processing flow consists of three stages, i.e., candidate filtering, scoring suitability, and confirming the match and processing the transaction.

1) Candidate filtering (matching essential conditions)

This step involves filtering all volunteers based on their essential conditions for the needs of the disaster victims. Here, the following two conditions are applied as AND conditions.

- Date and time match. The volunteer's availability (date and time) must overlap at least one day with the victim's recruitment date and time.
- Required qualifications. If the victim has specified required qualifications or licenses, the volunteer must possess those qualifications. If the required qualifications are set to "none," the filtering is based solely on the date and time condition.

2) Scoring suitability (quantifying suitability)

This step involves calculating the matching score S for the filtered candidates based on the following evaluation function to determine their priority. Here, the score S is defined as the sum of the weight ω_k for each evaluation item k and the suitability flag $x_k \in \{0,1\}$.

$$S = \sum_{k=1}^{n} \omega_k \cdot x_k \qquad (2)$$

The specific evaluation items and points are shown in Table 1. This scoring system allows personnel with past experience in similar activities E_f and skills and qualifications Q_d to be highly evaluated, thereby improving the quality of support.

Table 1. Matching score evaluation items and point allocation

Evaluation items (variables)	Detailed matching criteria	Point allocation ($\boldsymbol{\omega_k}$)
Experience details match (E_f)	The disaster victim's requested field matches the volunteer's activity experience	50 points
Qualifications match (Q_d)	The volunteer possesses the disaster victim's preferred qualifications	40 points
Field of activity match (F)	The disaster victim's requested field matches the volunteer's activity field	40 points
Region match (R)	The disaster victim's requested area matches the volunteer's activity area	30 points
Volunteer experience (E)	Whether the volunteer has previous volunteer experience	10 points

3) Confirming the match and processing the transaction

Based on the calculated scores, the candidates are sorted in descending order. Volunteers are selected from the top to the bottom until the number of disaster victims reaches the recruitment number, and then pairings are performed.

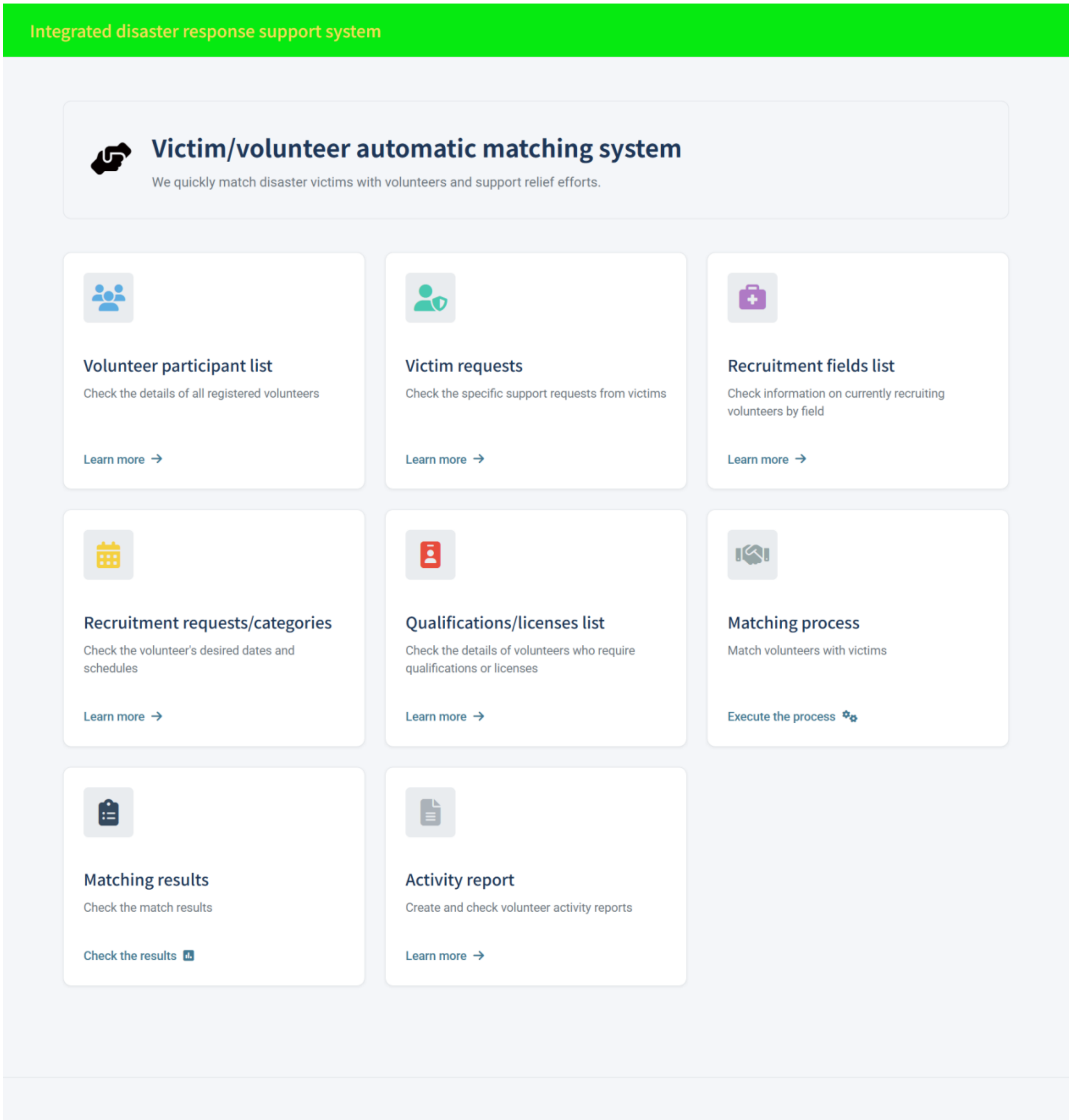


Figure 26. Administrator home screen of victim/volunteer automatic matching system

Figure 27 shows a sequence diagram of the victim/volunteer automatic matching system. The entire process comprises the information registration phase, which is performed asynchronously by each user, and the matching execution phase, which is performed collectively by the administrator. Here, both victims and volunteers enter information using a dedicated form corresponding to their respective status. The data sent from the client are stored in the corresponding tables and an intermediate table in the database. In addition, the matching status is recorded as the initial value (false). Then, the algorithm is executed in response to a request from the administrator. On the server side, the victim data are read in order of registration date and time (FIFO), and the scoring calculations are performed with matching volunteers. To maintain data consistency, pairings and status updates based on the calculation results are committed in a single transaction.

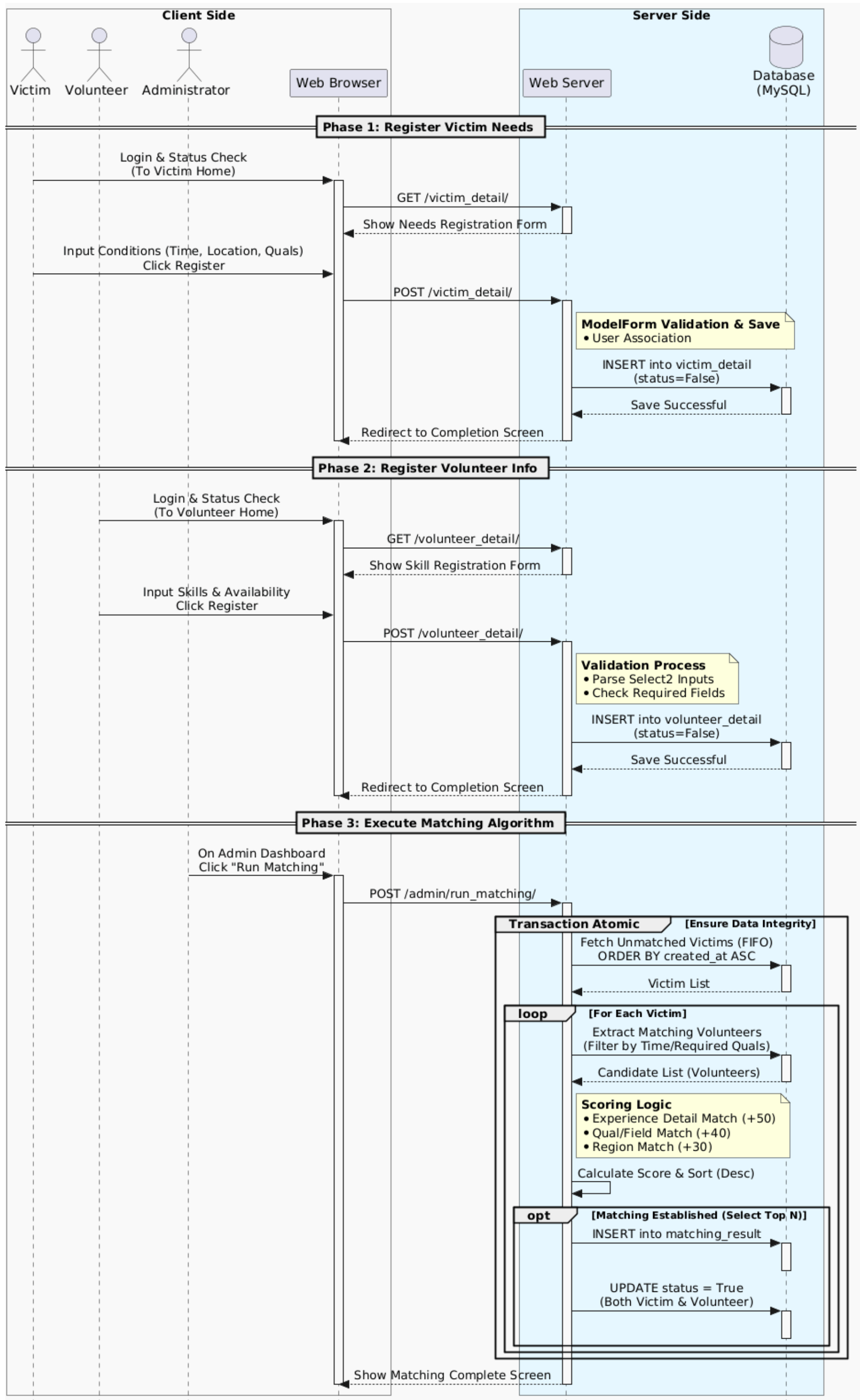


Figure 27. Victim/volunteer automatic matching system sequence diagram

6 Evaluation of Integrated Disaster Response Support System

6.1 Effectiveness Evaluation by Local Government Employees (Semistructured Interviews)

We performed a qualitative evaluation of the proposed system with local government employees who are experts in disaster response to verify its practical effectiveness at disaster sites. Here, we asked local government employees from three municipalities in Fukuoka Prefecture to participate in the evaluation, and we conducted individual semistructured interviews to obtain their opinions on the system. Each interview lasted approximately 60 min.

The responses received from nine participants are summarized as follows. Note that the evaluation results are summarized in two ways, i.e., quantitative scores based on a five-point scale and qualitative opinions based on the semistructured interviews.

- Quantitative evaluation results (five-point scale)

Table 2 shows the response distribution and the percentage of positive responses (i.e., the total of “strongly agree” and “agree”) for each question, and Figure 28 shows a stacked bar graph of the response trends.

1) Evaluation of the system as a whole and its integration

For question 8, which asked about the overall usefulness of the system, and question 9, which asked about the convenience of integrating multiple functions, all nine respondents (100%) answered either “strongly agree” or “agree.” For question 9, the number of responses that answered “strongly agree” was four (approximately 44%), representing a higher percentage than that obtained for other questionnaire items.

2) Evaluation of individual functions

All respondents provided positive ratings of the shelter management (question 1), relief supply management (question 2), disaster image sharing (question 6), and task visualization (question 7) functions.

3) Evaluation of automation technology (AI/RAG)

The effectiveness of the volunteer matching algorithm (question 4) was rated 100% positively. In terms of the effectiveness of the automatic volunteer matching function itself (question 3) and the improvement of work efficiency through RAG (question 5), eight respondents (89%) gave a positive evaluation, and one person answered “neither agree nor disagree.”

4) Evaluation of UI/UX and familiarity

Eight respondents (89%) assigned positive evaluations to the ease of familiarization through pretraining and manuals (question 11). Regarding ease of use in emergencies (question 10), two respondents answered “strongly agree,” five answered “agree” (for a total of seven (78%)), and two answered “neither agree nor disagree.”

Furthermore, no negative responses below “disagree” were observed for any of the questionnaire items.

Table 2. Questionnaire items and percentage of positive responses ($n = 9$)

Questions	Questions (summary)	Related features	Number of positive answers	Percentage of positive responses (%)
Question 1	Is it effective to track the trend in the number of people in evacuation shelters?	Shelter management	9	100%
Question 2	Is it effective to display estimated amounts of relief supplies?	Relief supplies management	9	100%
Question 3	Is automatic volunteer matching effective?	Volunteer	8	89%
Question 4	Is a matching algorithm effective?	Volunteer	9	100%
Question 5	Is streamlining through RAG effective for improving operational efficiency?	RAG operational efficiency	8	89%
Question 6	Is citizen participation in information gathering effective at disaster sites?	Disaster image sharing	9	100%
Question 7	Is visualizing task chronology and priorities effective?	Disaster task visualization	9	100%
Question 8	Is the system as a whole effective during a disaster?	Overall evaluation	9	100%
Question 9	Is it effective to handle multiple functions in a single system?	Integration evaluation	9	100%
Question 10	Is it easy to use even under limited time and high stress in an emergency?	UI/UX evaluation	7	78%
Question 11	Would it be possible to quickly master it with prior training and a manual?	Proficiency evaluation	8	89%

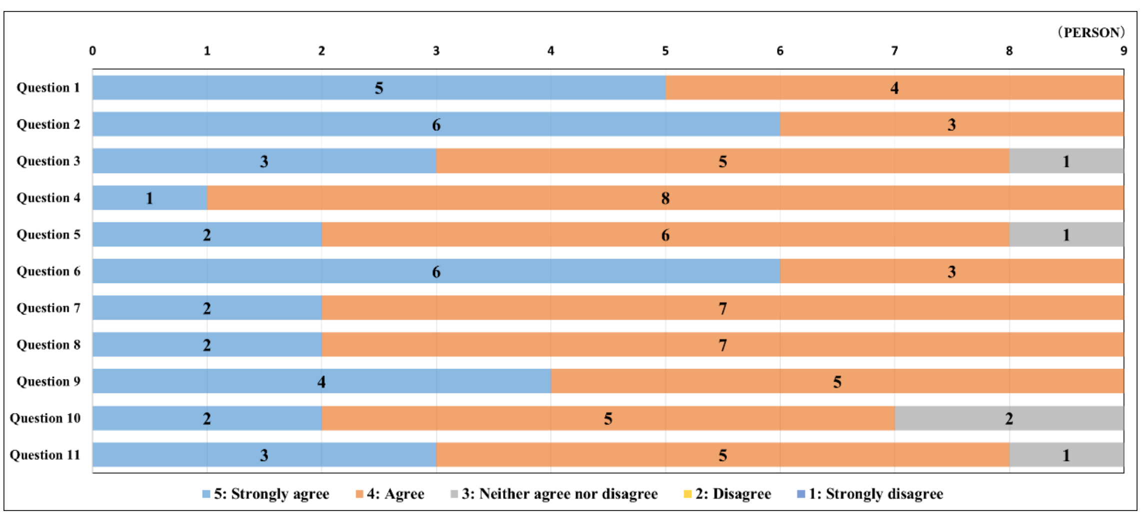


Figure 28. Results of five-point evaluation ($n = 9$)

- Qualitative evaluation results (analysis of free comments)

Analyzing the respondents' comments and free comments obtained through the semistructured interviews revealed specific findings regarding the practical value and operational challenges of the proposed system. The opinions obtained from the respondents were organized into the following four categories based on the evaluation perspective.

1) Usefulness of the integrated platform and current issues

All respondents expressed positive opinions regarding the system's centralized information management. However, one issue with current disaster response operations was noted: "The prefectural reporting system, internal communications, and public relations information are fragmented, resulting in duplicate information entry." In addition, from a decision-making support perspective, the respondents discussed the on-site situation, where "it is extremely difficult to make judgments when there is a discrepancy between weather forecasts and actual events (such as a sudden rise in river water levels)." The proposed system's visualization function, which supports real-time situational awareness, was expected to "contribute to rapid decision-making at headquarters." In particular, there was a consensus that, from an evacuation shelter management perspective, "the ability to grasp trends in the number of people and their attributes (such as the presence or absence of special needs) in real time, which currently relies on telephone confirmation, is extremely useful for resource allocation and support planning."

2) Expectations and concerns regarding AI/RAG functions and automation

Regarding the use of cutting-edge technology, specific expectations and operational concerns were raised for each function. For example, many of the respondents acknowledged the usefulness of the RAG function, which provides instant answers to questions, stating that "during a disaster response, we don't have the time to refer to thick manuals or disaster prevention plans." In particular, the RAG function was highly valued as a tool for suggesting "what to do now" for inexperienced local government employees. However, some respondents stated that "entering prompts (instructions) is itself a hurdle in an emergency." In addition, they cited maintaining up-to-date information and linking it to local information (e.g., the plans of each local government) as essential requirements, stating that "if the data on which the answers are based (regional disaster prevention plans and laws and regulations) is not always up-to-date, there is a risk of incorrect judgment." In terms of using images posted by residents and AI analysis, the respondents unanimously agreed that it "is effective in understanding on-site conditions (flooding, landslides, etc.) when local government employees cannot patrol the site." However, concerns were raised about the potential for misinformation, similar to that observed on social media platforms, as a major operational challenge. In response, they expressed hope for the system's "dedicated form with location information" to ensure reliability and for the "AI-based information filtering and automatic classification" Regarding the automatic report generation functionality, they advised that it would be good to visualize how weather conditions affect responses when managing tasks. The automatic matching approach was deemed useful; however, the respondents also stated that "there are factors (physical strength, compatibility, and the subtle needs of the field) that cannot be matched based on qualifications and skills alone." They also indicated that careful design is required for full automation, e.g., ensuring fairness to prevent volunteers from becoming concentrated at specific evacuation shelters.

3) Usability and possibility of knowledge transfer

Regarding the usability of the proposed system, the responses were generally favorable, with some commenting that it "allows for intuitive understanding of the situation." However, many requests for improvements were made regarding emergency use. From a knowledge transfer perspective, many municipalities share the structural issue of "difficulty in accumulating know-how due to staff being transferred every few years." In addition, the system's disaster response visualization function (i.e., task log recording and forecast/actual management) was highly praised, with participants saying that it "is important not only for postevent review and report creation, but also as an educational tool for passing on past disaster response history to the next generation."

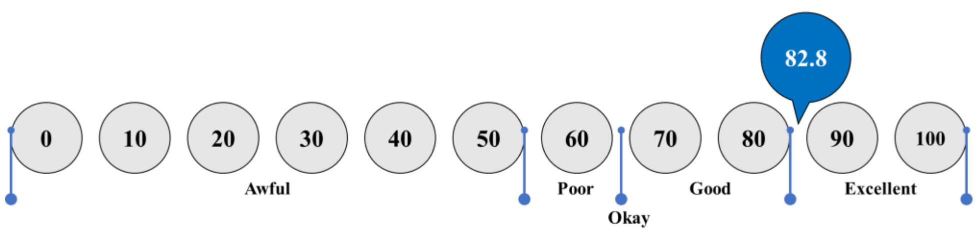
4) Practical issues for social implementation

In implementing the system, nontechnical challenges specific to government administration emerged. The most significant obstacle was identified as "integration with existing systems." Given that the Disaster Management Headquarters and Social Welfare Councils currently operate under different chains of command and systems, the need for cross-sectional integration (e.g., API integration) between the system and dedicated systems reporting to the prefecture was emphasized. In addition, based on the common understanding that "systems that are not used during normal times cannot be used in emergencies," the respondents indicated that the system should be employed in daily operations (e.g., road damage reports) and that training opportunities should be provided.

6.2 Usability Evaluation (System Usability Scale)

To quantitatively verify the intuitive operation of the system's UI/UX, we performed a usability evaluation using the system usability scale (SUS) with 50 participants.

The aggregated SUS scores are shown in Figures 29 and 30. The average SUS score for the entire system was 82.8 points. In general SUS evaluation standards, an average score of 68 is considered the standard, and a score of 80.3 or higher is classified as excellent. The score obtained by the proposed system exceeded this standard, thereby indicating high usability. In addition, the individual score distribution shown in Figure 30 revealed a minimum and maximum of 57.5 and 97.5 points, respectively. Notably, 46 of the 50 participants (92%) rated the system above the standard average of 68 points. These results demonstrate that the proposed system provides highly satisfying operability for many users.



SUS score	Grade	Adjective Rating
> 80.3	A	Excellent
68-80.3	B	Good
68	C	Okay
51-68	D	Poor
< 51	E	Awful

Figure 29. SUS results

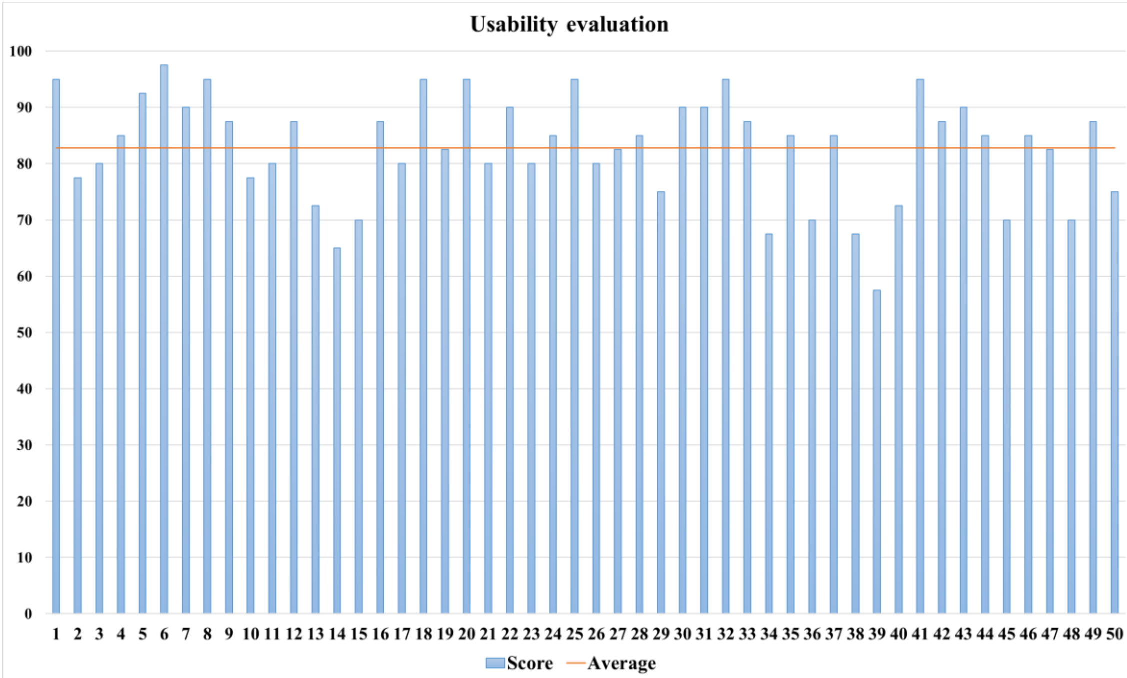


Figure 30. SUS score distribution (*n* = 50)

6.3 Efficiency Evaluation

We also compared the conventional method (i.e., keyword search using PDF viewing software) with the RAG-based chatbot function implemented in the proposed system to clarify the differences in the time and accuracy required to identify information and to quantitatively examine the impact on the time efficiency of information search tasks. To ensure the density and complexity of information in actual disaster responses, we used the Verification Report on the 2015 Joso City Kinugawa River Flood Response [32] published by the local government as the evaluation reference. Here, the participants were 60 undergraduate and graduate students. The participants were randomly divided into two groups of 30. Group A completed the experimental tasks using the proposed system's chatbot function, and Group B used the conventional method, i.e., search using PDF viewing software and Ctrl+F.

In this evaluation, the following three tasks were set, differing in terms of the level of abstraction of the information and the depth of the information provided.

- Task 1 (Number Identification). Identify the number of "total destruction" residential damage cases as of June 3, 2016.
- Task 2 (Multiple Data Aggregation). Calculate the total number of people rescued by the "police" (including helicopters and ground troops) as a result of activities from September 10–19, 2015.
- Task 3 (Time Identification). Identify the time on September 11 when the Joso City Hall main building lost power and telephone service due to flooding.

Regarding the implementation procedure and measurement indicators, a stopwatch was used in each task to measure the time from the start of the search process to the time at which the correct answer was obtained. Regardless of the search method, the time limit per question was set to 300 s in consideration of practical limitations, and any time exceeded was treated as a "timeout." In addition to measuring the time, the accuracy of the answers was also recorded.

Figure 31 compares the correct answer rates for Tasks 1 to 3. As can be seen, Group A maintained extremely high correct answer rates regardless of the task difficulty, i.e., 100% for Tasks 1 and 3, and 97% for Task 2. Group B achieved 100% correct answer rates for Task 1, representing a simple numerical search; however, Group B achieved only 90% for Task 2, which involved searching for a wide range of information, and 73% for Task 3, which involved identifying time-series information. Thus, when using the conventional method, there was a tendency for the correct answer rates to decrease as the task complexity increased.

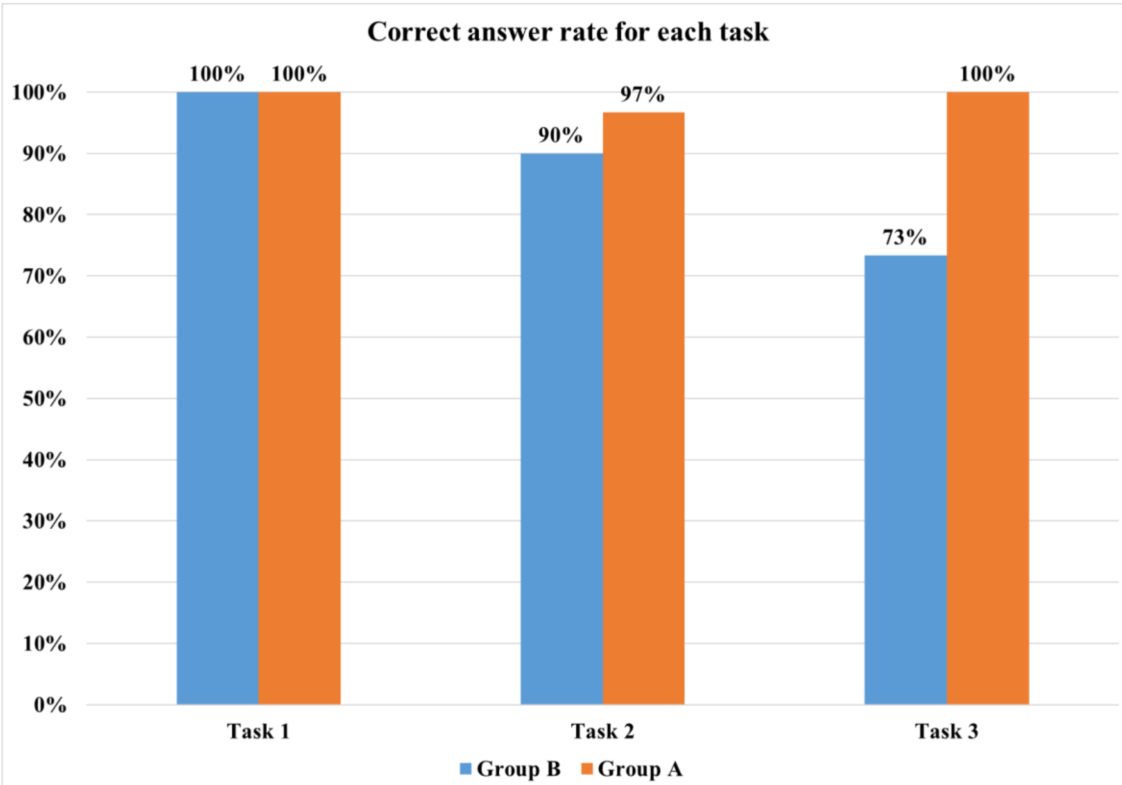


Figure 31. Correct answer rate for each task

Figure 32 shows the distribution of task completion times for the participants who obtained the correct answer in each task, and Table 3 summarizes the descriptive statistics for each group (i.e., the minimum, quartiles, median, mean, and maximum).

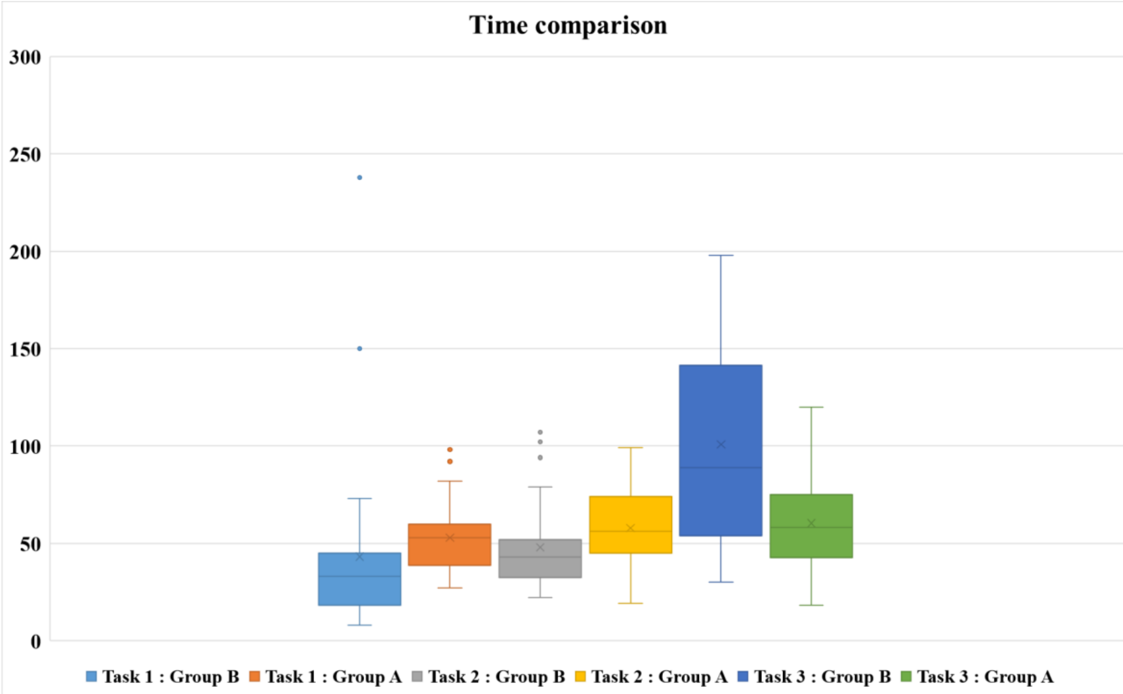


Figure 32. Comparison of task completion time

Table 3. Descriptive statistics for task completion times

Item	Minimum	1st quartile	Median	Average	3rd quartile	Maximum
Task 1 : Group B	8 s	18.25 s	33 s	43.13 s	45 s	238 s
Task 1 : Group A	27 s	38.75 s	53 s	52.87 s	59.75 s	98 s
Task 2 : Group B	22 s	32.5 s	43 s	47.93 s	52 s	107 s
Task 2 : Group A	19 s	45 s	56 s	57.79 s	74 s	99 s
Task 3 : Group B	30 s	54 s	89 s	100.77 s	141.5 s	198 s
Task 3 : Group A	18 s	43.5 s	58 s	60.3 s	74.5 s	120 s

- Task 1 (simple search) results

Group B exhibited a minimum and median of 8 s and 33 s, respectively, demonstrating high reactivity for information that could be identified using a simple keyword search. However, the maximum value was 238 s, indicating large delays when selecting search keywords and a tendency for large variations (variance) depending on individual search skills. In contrast, Group A’s median was 53 s, approximately 20 s longer than the conventional method, due to the time required for prompt input and response generation. However, the interquartile range (i.e., the width between the first and third quartiles) was relatively narrow, and the maximum value was within 98 s, which confirms that all participants completed the task within a consistent time frame.

- Task 2 (combining multiple pieces of information) results

For Task 2, which expanded the range of the information search, Group B’s median was 43 s, which was fast; however, the average was 47.93 s, representing a slight increase from Task 1. For Group A, the median and average were 56 s and 57.79 s, respectively. Thus, there was no significant change in the response time for Task 2 compared with Task 1 (median 53 s), which indicates that the participants were able to extract information consistently.

- Task 3 (context understanding and time identification) results

The most significant difference was observed in Task 3. For Group B, the median time increased sharply to 89 s, and the mean increased to 100.77 s for the more difficult search task requiring context understanding. The third quartile reached 141.5 s, which indicates that many participants struggled to identify the answer, significantly increasing their search times. In contrast, for Group A, the median and mean were 58 s and 60.3 s, respectively, which are approximately the same as the times recorded for Tasks 1 and 2. The third quartile also remained at 74.5 s, demonstrating that efficiency is not reduced as the complexity of the information increases.

These results quantitatively demonstrate that although the proposed system may be inferior for experts using conventional methods in terms of simple search speed (instantaneous power), it provides stable efficiency over a constant, short time span (approximately 60 s) regardless of the task difficulty or the skill of the searcher.

6.4 Discussion

In the following, we integrate the results of the three evaluations (i.e., effectiveness evaluation by local government employees, usability evaluation using SUS, and efficiency evaluation using task time measurement) and discuss the degree to which the three research objectives (Section 3) were achieved. We also compare and discuss our findings with related research and summarize the technical and practical significance of this study.

- Achievement of research objectives

1) First objective: integration of functions and construction of a common platform

Regarding the first research objective, i.e., removing information fragmentation and building a comprehensive support platform, 100% of the respondents positively evaluated the usefulness of the overall system (question 8) and the convenience of the integration (question 9) in the local government employee evaluation (Section 6.1). In addition, in the interviews (qualitative evaluation), "the difficulty of judging discrepancies between weather forecasts and actual conditions" was cited as an issue. The integrated dashboard implemented in the proposed system addresses this issue by enabling real-time situation sharing, suggesting that it may satisfy the practical needs of the field. Thus, we believe that the first research objective has been achieved.

2) Second objective: improved work efficiency and reduced workload

The efficiency evaluation results (Section 6.3) quantitatively support the achievement of the second research objective, i.e., automating and streamlining work processes with AI and RAG technology. In the task time measurement, Group B (using the conventional method) outperformed Group A (using the proposed system) in some cases in Task 1 (simple keyword searches). However, in more complex tasks requiring contextual understanding (Task 3), Group B's accuracy rate was reduced to 73%, and the variance in response times increased. In contrast, Group A maintained a high accuracy rate regardless of the task difficulty, and the variance in the response times was extremely small. These results suggest that the proposed system can serve as a support tool to address the on-site concern raised by local government officials in the interviews (Section 6.1), i.e., "we don't have the time to refer to thick manuals or disaster prevention plans." The proposed system's ability to consistently provide accurate information without relying on individual skill suggests that it could be used as a support tool.

3) Third objective: intuitive UI/UX and eliminating knowledge dependence

The achievement of the third objective, i.e., eliminating knowledge dependency and creating a user-friendly UI, was demonstrated by the extremely high average score of 82.8 points in the SUS evaluation (Section 6.2). Furthermore, the qualitative evaluation (Section 6.1) revealed a structural issue in local governments, i.e., staff in charge are replaced every few years, which makes it difficult to pass on knowledge. The proposed system's high usability means that even inexperienced staff can achieve information retrieval accuracy that is comparable to that of experienced staff. In addition, the evaluation of the task visualization using the disaster response visualization function (Section 6.1) combined with the ease of use demonstrated in the SUS evaluation is expected to function as a foundation for sustained system operation and the accumulation and transfer of organizational experience.

- Comparison with related studies and discussion

As discussed in Section 2, research on disaster response support systems is diverse. However, the current study builds on the results of previous studies and makes unique contributions in three key areas, i.e., functional integration, practical application of AI technology, and elimination of personalization. In the following, we discuss the position of this study through a comparison with related studies.

1) From optimizing individual functions to an integrated platform

Many previous studies, e.g., the volunteer matching projects by Sekiguchi et al. [4] and Kodama et al. [5], or the evacuation shelter management systems by Nakatani et al. [8] and Shimoda [11], have attempted to optimize specific disaster response phases or functions. These studies have made valuable contributions to solving issues in their respective fields; however, these tasks are interrelated in real-world disaster sites. The current study is unique in that it attempts to integrate these individual functions on a common platform and eliminate information fragmentation. The positive evaluation of the convenience of integration by local government officials (Section 6.1) demonstrates the importance of data integration and visualization, as advocated by Hanashima et al. [22] and Li et al. [23], by applying it to more practical workflows. Furthermore, this study is innovative in that develops a platform that comprehensively supports disaster response efforts rather than simply aggregating data.

2) Quantitative demonstration of the practical usefulness of generative AI and RAG technology

In terms of applying generative AI to disaster response, Pujiono et al. [12] and Xia et al. [15] reported the usefulness of systems that utilize RAG. The current study supports the findings of these previous studies and further clarifies the quantitative value of the proposed system by comparing it with human search capabilities through the task time measurement (Section 6.3). For example, in a task with high information complexity (Task 3), this study demonstrated that while the accuracy rate of the conventional methods declined, the proposed system maintained a high accuracy rate and stable response times. These findings support the potential of dialogue systems, as discussed by Boné et al. [13] and Urbanelli et al. [16], in terms of improving operational efficiency.

3) Approach to knowledge transfer

Technology has become increasingly sophisticated, as seen in the image analysis technologies proposed by Sogi et al. [17] and Udo et al. [18], as well as the decision-making dashboard developed by Tillekaratne et al. [21]. However, as revealed in our interviews with the local government officials, a challenge in this field is the difficulty associated with knowledge transfer due to frequent transfers. While implementing advanced image analysis and visualization technologies, this study has made them accessible to anyone through high usability (UI/UX), as demonstrated by the "excellent" rating obtained in the SUS evaluation (Section 6.2). The implementation of a system that can be operated by inexperienced officials and the proposal of a foundational system for organizational knowledge transfer distinguishes the current study from previous research that focused on technical proposals, thereby representing a unique practical and societal contribution.

- Technical contributions and practical significance

The greatest significance of this study lies in the ability to combine two elements, i.e., technological advances (AI and automation) and user-friendliness (UI/UX), in an integrated platform. Generally, advanced

AI systems tend to be complex to operate; however, this study has developed an interface that is intuitive for general users, as demonstrated by the results of the SUS evaluation. Thus, as discussed in Section 6.3, even users without specialized knowledge can complete advanced information search tasks with the assistance of AI. This contributes to improving practical resilience in disaster prevention settings in local governments, which face chronic labor shortages and challenges related to knowledge transfer. This means that a certain level of disaster response will be possible regardless of who is in charge.

7 Conclusion and Future Work

In this study, we developed an integrated disaster response support system to streamline disaster response operations and support the integrated management of information in local governments. Recent disaster response issues include a shortage of staff with specialized knowledge and difficulties transferring disaster response knowledge. In response to these issues, this study developed a system with three objectives, i.e., integrating functions, supporting operations using the latest technology, and eliminating the dependency on personal knowledge.

First, the results of an effectiveness evaluation targeting local government employees suggested that the integrated platform provided by the proposed system may be an effective means of eliminating information fragmentation in the field and supporting decision-making processes.

Second, a usability evaluation using the SUS achieved an “excellent” rating, which demonstrates that the proposed system can be operated intuitively even by inexperienced employees.

Third, an efficiency evaluation based on task time measurement quantitatively demonstrated that the RAG-based chatbot functionality implemented in the proposed system consistently achieved a high accuracy rate and efficiency, even for difficult information search tasks and regardless of individual skill.

Based on these findings, we conclude that the proposed system combines technological sophistication with ease of use. Furthermore, the proposed system is highly useful as a practical foundation for sustainably improving local government disaster response capabilities.

The following three points can be raised as future developments.

1) Use in nondisaster times and resident adoption

The most significant challenge in implementing the proposed system for broad practical application is that a system that is not used in normal times cannot be used in emergencies. Thus, to establish the system, it is essential to shift from a tool designed specifically for disaster situations to a “phase-free” design that can be utilized in everyday operations. Specifically, the disaster information visualization system (with its resident posting function) could be used to report road collapses, damaged convex mirrors, and illegal dumping. In addition, creating a cycle in which residents routinely report to the government using the system and officials respond and perform management duties using the dashboard will improve operational proficiency and increase awareness of the system, thereby enabling smooth information gathering during disaster scenarios.

2) API-based integration with existing systems

Currently, local governments utilize disaster information systems operated by prefectures; however, the independence of these systems frequently results in “double information entry” issues. To maximize the value of the proposed system, it is necessary to construct an ecosystem that automatically synchronizes data with these existing systems using APIs. Thus, our next development goal is to extend the proposed system to realize seamless integration with all information sources by implementing various features, e.g., automatic linking of damage information to prefectural systems and automatic counting and reporting of evacuees.

3) Cloud-based customizability and deployment model

In this study, we developed five subsystems to provide comprehensive support. However, the challenges and priorities faced by local governments vary depending on their regional characteristics. Thus, rather than uniformly deploying all functions as a package, it would be desirable to transition to a cloud model employing a microservices architecture that allows each local government to select and deploy only the functions (subsystems) they need. This would allow local governments to configure (customize) their systems in a flexible manner according to their budget constraints and local challenges, while optimizing deployment costs. In addition, modular updates are expected to make it easier to adapt to new technologies, which we believe will contribute to greater adoption among local governments and improve disaster response capabilities.

4) Balancing information security and system efficiency

Because this system handles personal and confidential information of evacuees, robust security measures are essential for practical implementation. As Huth et al. [33] point out, not only external attacks but also unintentional data leaks and privilege abuse from within the organization can pose serious threats. Addressing the risk of cyberattacks exploiting disaster-induced chaos is particularly critical. In addition, Kim et al. [34] discuss the importance of balancing real-time transmission of highly confidential data with reduced device power consumption and processing load in the proposed system. Our proposed system also requires future design efforts to balance security and system efficiency (power efficiency), such as adopting lightweight authentication protocols and optimizing communication frequency. Meanwhile, Kiyomoto et al. [35] point out the difficulty of maintaining a balance between availability and security in disaster-response IT systems during emergencies. Therefore, it is necessary to build a framework that balances safety and convenience by using AI for automated monitoring and detection of unauthorized access, and by strengthening access control based on user attributes.

Acknowledgement

This paper is an extended version of the work [36] originally presented at the 9th International Conference on Mobile Internet Security, Sapporo, Japan, December 16-18, 2025.

References

[1] Bündnis Entwicklung Hilft & Institute for International Law of Peace and Armed Conflict (IFHV). WorldRiskReport 2025: WorldRiskMap. https://weltrisikobericht.de/worldriskreport/, [Online; Accessed in December 2025].

[2] Iizuka, T. (2020). Institutional Building for Functional Disaster Management in Local Government - Decision Making, Organaization Design, Training Program -. Law and Politics Review, 6, 73-96. (in Japanese). https://doi.org/10.20691/kanhouseiken.6.0_73

[3] Nishimura, W. (2024). Research on Disaster Prevention and Crisis Management Systems of the Local Governments in Japan -Analysis of Survey Results for the Local Governments-. The Review of Economics & Political Science, 92(3-4), 153-183. (in Japanese). https://meiji.repo.nii.ac.jp/records/2000469

[4] Sekiguchi, H., Takai, M., Owada, Y., Oguchi, M. (2021). Disaster Volunteer Challenges and Application Design. In Proc. of the Multimedia, Distributed, Cooperative, and Mobile Symposium 2021 (pp. 495-500), Online, June 2021. (in Japanese)

[5] Kodama, S., Uetake, T. (2021). Proposal of volunteer activity participation support system for beginners. In Proc. of the 83rd National Convention of IPSJ (pp. 119-120), Online, March 2021. (in Japanese)

[6] Hijiri, R., Tsukada, K. (2023). Support system for communication among management, volunteers, and disaster victims in disaster volunteer activities. In Proc. of the 85th National Convention of IPSJ (pp. 735-736), Tokyo, Japan, March 2023. (in Japanese)

[7] Huang, A., Sun, Y. (2020). An intelligent and data-driven mobile platform for youth volunteer management using machine learning and predictive analytics. In Proc. of the 10th International Conference on Advances in Computing and Information Technology (pp. 1-15), London, United Kingdom, November 2020. https://doi.org/10.5121/csit.2020.101515

[8] Nakatani, Y., Tsukada, K. (2021). Dispersed evacuation support system for pre-flood disaster. In Proc. of the

83rd National Convention of IPSJ (pp. 665-666), Online, March 2021. (in Japanese)
[9] Nakada, H., Murotani, T., Nakamura, M. (2021). Development of Shelter Navigation with Considering Three Cs in the Corona Age. IEICE Technical Report, ICM2020-51,LOIS2020-39(2021-01), 100-107. (in Japanese)
[10] Akasaka, K., Namba, T., Isshiki, M., Abe, K. (2020). Proposal on Victims Information Management System. International Journal of Informatics Society (IJIS), 12(1), 17-27.
[11] Shimoda, E. (2023). Construction of a support system for people requiring special care at evacuation shelters using Geographical Information System (GIS). Japanese Journal of Disaster Medicine, 28(1), 12-18. https://doi.org/10.51028/jjdisatmed.28.1_12
[12] Pujiono, I., Agtyaputra, I., M., Ruldeviyani, Y. (2024). IMPLEMENTING RETRIEVAL-AUGMENTED GENERATION AND VECTOR DATABASES FOR CHATBOTS IN PUBLIC SERVICES AGENCIES CONTEXT. Journal of Computer Science and Technology, 10(1), 216-223. https://doi.org/10.33480/jitk.v10i1.5572
[13] Boné, J., Ferreira, J. C., Ribeiro, R., Cadete, G. (2020). DisBot: A Portuguese Disaster Support Dynamic Knowledge Chatbot. Applied Sciences, 10(24), 9082. https://doi.org/10.3390/app10249082
[14] Tsunoda, K., Okazaki, M., Koyanagi, R., Sakai, M., Kakuta, S., Minoura, D. (2018). Development of A Report Assistance System using Chatbot - An Application Case of Disaster Drill -. IPSJ SIG Technical Report, 2018-GN-103(13), 1-8. (in Japanese)
[15] Xia, Y., Huang, Y., Qiu, Q., Zhang, X., Miao, L., Chen, Y. (2024). A Question and Answering Service of Typhoon Disasters Based on the T5 Large Language Model. ISPRS International Journal of Geo-Information, 13(5), 165. https://doi.org/10.3390/ijgi13050165
[16] Urbanelli, A., Frisiello, A., Bruno, L. Rossi, C. (2024). The ERMES chatbot: A conversational communication tool for improved emergency management and disaster risk reduction. International Journal of Disaster Risk Reduction, 112, 104792. https://doi.org/10.1016/j.ijdrr.2024.104792
[17] Sogi, N., Shibata, T., Terao, M., Senzaki, K., Tani, M., Rodrigues, R. (2024). Disaster Damage Visualization by VLM-Based Interactive Image Retrieval and Cross-View Image Geo-Localization. In Proc. of the IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium (pp. 1746-1749), Athens, Greece, July 2024. https://doi.org/10.1109/IGARSS53475.2024.10640928
[18] Udo, H., Koshinaka, T. (2024). Reading is Believing: Revisiting Language Bottleneck Models for Image Classification. In Proc. of the 2024 IEEE International Conference on Image Processing (pp. 943-949), Abu Dhabi, United Arab Emirates, October 2024. https://doi.org/10.1109/ICIP51287.2024.10648091
[19] Yasuda, K., Aritsugi, M., Takeuchi, Y., Shibayama, A., Mendonca, I. (2025). Disaster image tagging using generative AI for digital archives. In Proc. of the 24th ACM Joint Conference on Digital Libraries (pp. 1-11), Hong Kong, China, December 2025. https://doi.org/10.1145/3677389.3702516
[20] Klerings, A., Tang, S., Chen, Z. (2019) Structuralizing Disaster-scene Data through Auto-captioning. In Proc. of the 2nd ACM SIGSPATIAL International Workshop on Advances on Resilient and Intelligent Cities (pp. 29-32), IL, USA, November 2019. https://doi.org/10.1145/3356395.3365671
[21] Tillekaratne, H. I., Wickramagamage, P., Werellagama, I., Rathnayake, U., Siriwardana, C., Bandara, A., Madduma-Bandara, C. M., Bandara, T. W. M. T. W., Abeynayaka, A. (2023). Situation report (SITREP) visualization for effective management of disaster incidents in Sri Lanka. Journal of Infrastructure Policy and Development, 7(3), 1-21. https://doi.org/10.24294/jipd.v7i3.2206
[22] Hanashima, M., Sano, H., Usuda, Y. (2024). Designing and Developing Dynamic Decision Support Information for Disaster Response. In Proc. of the International Association of Hydrological Sciences (189-195), April 2024. https://doi.org/10.5194/piahs-386-189-2024
[23] Li, W., Wang, S., Chen, X., Tian, Y., Gu, Z., Lopez-Carr, A., Schroeder, A., Currier, K., Schildhauer, M., Zhu, R. (2023). GeoGraphVis: A Knowledge Graph and Geovisualization Empowered Cyberinfrastructure to Support Disaster Response and Humanitarian Aid. ISPRS International Journal of Geo-Information, 12(3), 112. https://doi.org/10.3390/ijgi12030112
[24] Django Software Foundation. The web framework for perfectionists with deadlines | Django. https://www.djangoproject.com/, [Online; Accessed in December 2025].
[25] OpenAI. OpenAI Platform. https://platform.openai.com/docs/overview, [Online; Accessed in December 2025].
[26] Li, J., Li, D., Savarese, S., Hoi, S. (2023). BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Proc. of the 40th International Conference on Machine Learning (pp.19730-19742), Hawaii, USA, July, 2023.
[27] OpenStreetMap Wiki. About OpenStreetMap. https://wiki.openstreetmap.org/wiki/About_OpenStreetMap, [Online; Accessed in December 2025].

[28] OpenStreetMap contributors. Leaflet - a JavaScript library for interactive maps. https://leafletjs.com/, [Online; Accessed in December 2025].

[29] Bootstrap. Bootstrap ・ The most popular HTML, CSS, and JS library in the world. https://getbootstrap.com/, [Online; Accessed in December 2025].

[30] Cabinet Office of Japan. About Material Support: Disaster Prevention Information Page - Cabinet Office. https://www.bousai.go.jp/taisaku/hisaisyagyousei/push.html, [Online; Accessed in December 2025].

[31] Chroma. Introduction - Chroma Docs. https://docs.trychroma.com/docs/overview/introduction, [Online; Accessed in December 2025].

[32] Joso City. Verification Report on the 2015 Joso City Kinugawa River Flood Response. https://www.city.joso.lg.jp/data/doc/1674176111_doc_6_0.pdf, [Online; Accessed in December 2025].

[33] Huth, C.L., Chadwick, D.W., Claycomb, W.R. You, I. (2013). Guest editorial: A brief overview of data leakage and insider threats. Information Systems Frontiers, 15, 1-4. https://doi.org/10.1007/s10796-013-9419-8

[34] Kim, B., Kim, Y., Lee, I., You, I. (2007). Design and Implementation of a Ubiquitous ECG Monitoring System Using SIP and the Zigbee Network. In Proc. of the Future Generation Communication and Networking (pp. 599-604), Jeju, Korea (South), December 2007. https://doi.org/10.1109/FGCN.2007.97

[35] Kiyomoto, S., Fukushima, K., Miyake, Y. (2012). Security-and-Privacy-Related Issues on IT Systems During Disasters. In Proc. of the 7th International Conference on Availability, Reliability, and Security (.pp. 445-459), Prague, Czech Republic, August 2012. https://doi.org/10.1007/978-3-642-32498-7_33

[36] Seri, Y., Ishida, T. (2025). Development and Evaluation of a Disaster Response Task Visualization System Aimed at Supporting Decision-Making by Local Government Employees. In Proc. of the 9th International Conference on Mobile Internet Security (pp.1-12), Sapporo, Japan, December, 2025.