Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 10, No. 3, 2024

Design and Implementation of Decision Support System for Airport Emergency Disposal

Yichi Zhang, Fanliang Bu *

College of Information Network Security, People's Public Security University of China, Beijing, China
*** Corresponding author:** Fanliang Bu (Email: bufanliang@sina.com)

Abstract: To address the lack of integrated system applications in decision-support tasks and enhance the emergency response capabilities of airport authorities, this study analyzes the requirements and functionalities based on the Unified Modeling Language (UML) system analysis approach, tailored to the practical needs of emergency management. The system's functional modules and database were designed, and a decision-support engine combining "knowledge reasoning + text generation" was implemented. The resulting decision-support system for airport emergency incident management holds significant value for improving the development of airport emergency response frameworks and advancing informatization efforts.

Keywords: Emergency Management; Knowledge Reasoning; Text Generation; UML System Analysis.

1. Introduction

Emergency management is a critical component of the national security framework, responsible for risk warning and response, safeguarding national security, and serving as a key aspect of national governance [1]. Emergency plans, which encompass policies, procedures, and regulations for emergency response, provide an overarching depiction of emergency management [2]. They are essential for risk prevention, crisis mitigation, and serve as a fundamental pillar of the emergency management system [3]. Effective emergency plans can minimize the escalation of incidents and reduce potential losses caused by accidents [4].

As a key research topic within emergency plans, decision support for incident management has gained significant attention in recent years. Leveraging computer technology to provide decision support for emergency response has become a prevailing trend. With the continued expansion of airport operations, the complexity and diversity of emergencies have also increased. Efficient and reliable emergency handling has become a primary objective in emergency management and a focal concern for airport authorities. In the context of airport emergency scenarios, rapid, accurate, and reasonable decision-making is fundamental for managing incidents. Employing decision-support tools can enhance the capacity to address emergencies, minimize their impact, and ensure the orderly operation of airports.

To integrate decision-support tools systematically, this study adopts the Unified Modeling Language (UML) [5] approach for system analysis. It focuses on system analysis, design, and implementation to develop a decision-support system for airport emergency incident management. By offering an intuitive and user-friendly interface, the system significantly reduces the expertise required for decision-making and improves response efficiency.

2. System Analysis

2.1. Overall System Architecture Analysis

When an airport emergency management authority expresses the need for the system, specific user and administrator accounts are issued, and the system software is installed. The airport emergency incident management decision support system is primarily divided into two subsystems: the frontend user subsystem and the backend management subsystem. Both subsystems are accessible to the deploying organization. The frontend user subsystem is exclusively available to users with valid accounts, providing a range of services, including comprehensive decision-support workflows. In contrast, the backend management subsystem is accessible only to administrators and offers a complete set of backend management functionalities for the deploying organization. Each subsystem has distinct functional modules. The frontend user subsystem includes account self-management, case demonstration, decision support, and performance evaluation. The backend management subsystem includes case management, user management, and decision-support engine management. The overall system architecture is shown in Figure 1.

2.2. System Requirements Analysis

2.2.1. Overview of System Requirements

Based on the overall system architecture and the practical needs of emergency incident management, the system requirements are summarized. It is important to note that, under this system design framework, both "administrators" and "users" can be staff members from the system's deploying organization. Each subsystem has its own account, and staff can access different subsystems by using different accounts. The main requirements for the system are as follows:

(1) Frontend User Subsystem Requirements:

1)View information and search cases in the case library;

2)Self-management of user information (including login, registration, and information modification);

3)Emergency incident management decision support (including filling out event information forms, using knowledge reasoning and text output models, and viewing historical decisions);

4)Decision effect evaluation (including filling out evaluation forms and automatically calculating evaluation scores);

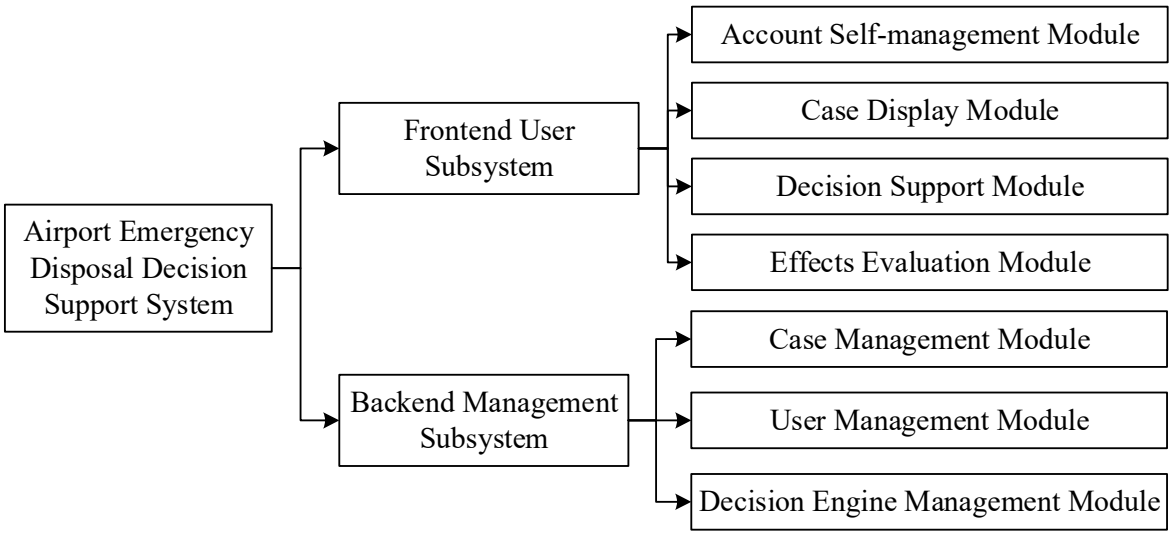

Fig 1. System Overall Architecture Diagram

(2) Backend Management Subsystem Requirements:

1)Administrator management of user information (including viewing, modifying, and deleting);

2)Administrator management of the case library (including viewing, editing, deleting cases, and entering new cases);

3)Administrator management of the decision support engine (including daily maintenance, optimization of reasoning weights, modification of reasoning rules, and updating text generation models).

2.2.2. System Use Case Diagram

Based on the above system requirements overview, the system participants are identified as the User (frontend account holder) and the Administrator (administrator account holder). As shown in Figure 2.

Based on the system requirements analysis, the use case diagram for the two participants is provided next. First, the user use case diagram. According to the frontend user subsystem's requirements analysis, the activities related to the user mainly include the following: account management, case viewing, decision support, and effect evaluation. Account management includes user registration, login, and self-management of user information. Case viewing includes case retrieval and display. Decision support includes knowledge reasoning and text generation. Based on these activities, the user use case diagram is created. The user use case diagram is shown in Figure 3.

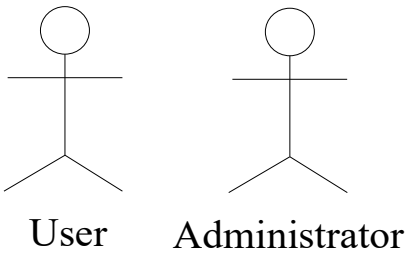

Fig 2. System Participants

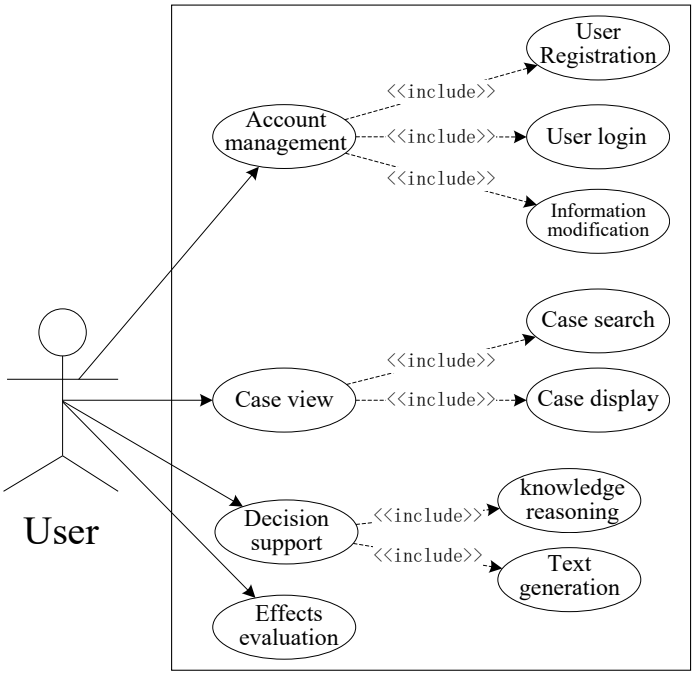

Fig 3. User Use Case Diagram

Next is the administrator use case diagram. According to the backend management subsystem's requirements analysis, the activities related to the administrator mainly include the following: case management, user management, and decision engine management. Case management includes case entry, modification, and deletion. User management includes modification and deletion of user information. Decision engine management includes optimization of reasoning weights, modification of reasoning rules, and updating of text generation models. Based on these activities, the administrator use case diagram is created. The administrator use case diagram is shown in Figure 4.

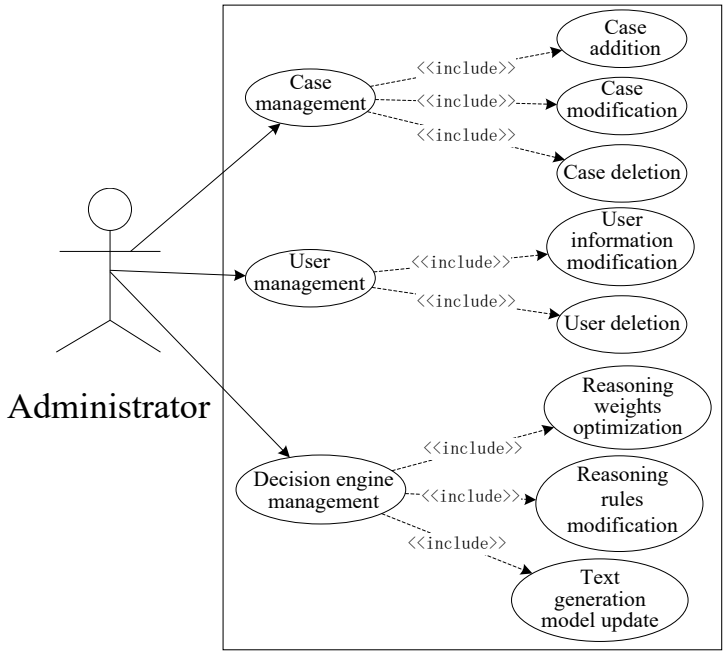

Fig 4. Administrator Use Case Diagram

By combining the user use case diagram and the administrator use case diagram, the overall system use case diagram is obtained, as shown in Figure 5.

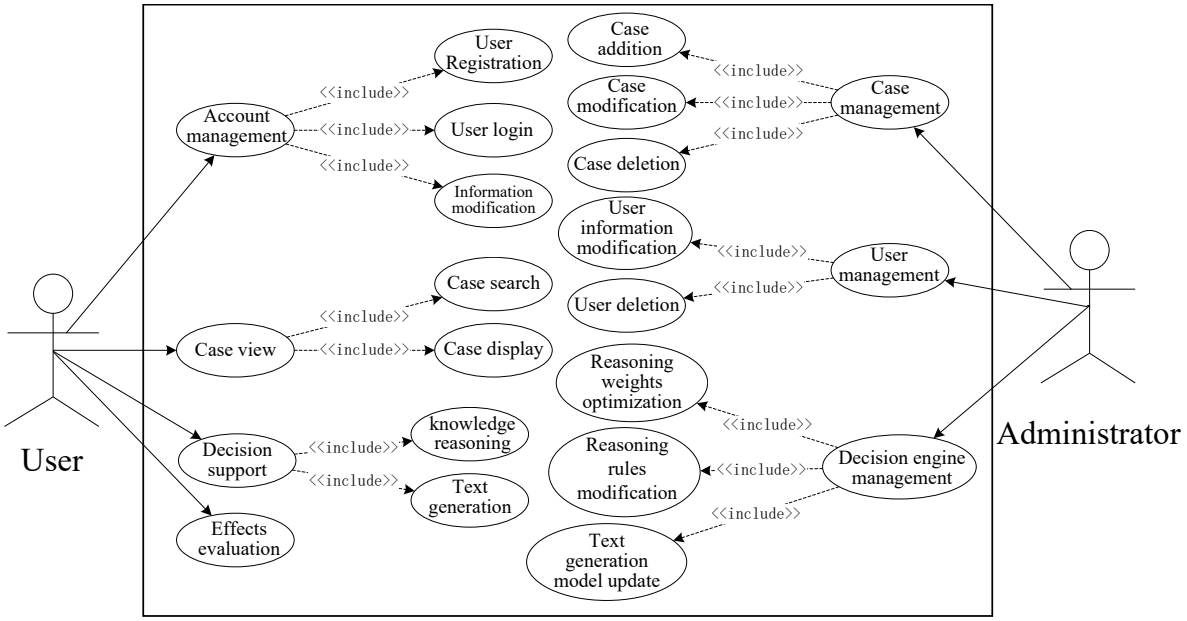

Fig 5. System Use Case Diagram

2.3. System Functionality Analysis

2.3.1. Frontend User Subsystem Functionality Analysis

The frontend user subsystem is designed for use by airport emergency management staff. These staff members directly face emergency incidents and have a need for decision support. Based on the frontend subsystem's requirements analysis, the features that users need to use are as follows:

a) Self-management of account information: This includes user registration, login, and modification of user information. For registration, users must fill out as detailed a form as possible, and the registration information will be crucial for user login and management. The system's login functionality must access the user information database, and successful

login occurs when the username and password match. As users' information may change during system use, the system should allow users to easily modify their own information, making it more convenient compared to administrator-backed modifications.

b) Case query and display: The case library is the basis for the system's decision support. Users should not be in a "black-box" state when facing these cases, as this could create a sense of distrust. Furthermore, after each decision support instance, the process should be stored in the case library as a new case. Therefore, the system should have case query and display functionalities to allow users to access relevant cases.

c) Decision support: Decision support is the core functionality of the system. The system should allow users to input relevant information about an emergency incident and receive a corresponding emergency response plan. Specifically, the system should analyze the event attributes input by the user, perform knowledge reasoning based on the case library, and use the results of the reasoning to generate a response plan text.

d) Decision support effect evaluation: Feedback after an incident management process is an important basis for improving decision support performance. The system should allow users to evaluate the effectiveness of the decision-making process by filling out feedback forms. The evaluation scores should be automatically calculated from the feedback and the response plan, and then stored in the historical decision library to be used as a reference for optimizing the decision support engine in the future.

2.3.2. Backend Management Subsystem Functionality Analysis

The backend management subsystem serves as the platform for managing and improving the decision support process from the perspective of administrators and system operators. The main management functions of the backend management subsystem are as follows:

a) User Management: Since the system is designed for user use, managing user information is crucial. The system should allow administrators to perform functions such as user cancellation, viewing all user information, and modifying certain user details.

b) Case Management: Cases are the main basis for decision support, and a well-maintained case library ensures the accuracy of decision-making. The entry of new cases, modification of case information, and deletion of outdated cases will positively impact the effectiveness of decision support. Therefore, the backend management subsystem should include functions for case entry, modification, and deletion.

c) Decision Support Engine Management: The decision support engine consists of two parts: the knowledge reasoning model and the text generation model. Management of the decision support engine should therefore focus on these two areas. For the knowledge reasoning model, improvements can be made by optimizing reasoning weights and modifying reasoning rules. Additionally, updating the text generation model will enhance the performance of the decision support engine. Therefore, the backend management subsystem should enable functions such as optimizing reasoning weights, modifying reasoning rules, and updating the text generation model.

2.4. Business Process Analysis

Due to space limitations, the business process analysis section focuses on the full-process decision support. Full-process decision support is not only the fundamental purpose of designing this system, but also the core business of the system. After an emergency incident occurs, its response process is initiated, and the system is quickly put into use. The user collects information about the emergency event and fills out the event attribute form. Once submitted, the system enters the knowledge reasoning phase. Knowledge reasoning generates a structured dataset containing the event information and corresponding response procedures. Next, the text generation model creates an emergency response plan text based on this structured data to assist decision-making. The user follows this text to handle the emergency event, which causes changes in the event's status. At this point, the system checks whether the event has ended. If not, the above process is repeated; if the event is concluded, the emergency response process ends, followed by an evaluation of the decision support effectiveness. Once the evaluation is complete, the full-process decision support business is completed. Based on the above description, the business process diagram for full-process decision support is shown in Figure 6.

3. System Design

Based on the system analysis section, design the system's functional modules and draw the system's functional module sequence diagram, including the account self-management module, case display module, decision support module, and effect evaluation module for the frontend user subsystem, as well as the case management module, user management module, and decision support engine management module for the backend management subsystem.

3.1. Frontend User Subsystem Functional Module Design

3.1.1. Account Self-Management Module

The functions of the Account Self-Management Module include user registration, login, and information modification. The user information modification function is similar to the user management module in the backend management subsystem, as detailed in section 3.2.2, so it will not be repeated here. The sequence diagrams for user registration and user login are shown in Figures 7 and 8, respectively.

3.1.2. Case Display Module

The main functions of the Case Display Module include case retrieval and viewing. After the user enters a keyword in the case search box, the case search service will access the case library and display the search results on the case search interface for the user to view. The sequence diagram for case retrieval and viewing is shown in Figure 9.

3.1.3. Decision Support Module

The Decision Support Module is the core functional module of the system. Users access this module through the decision support interface. After inputting the relevant attributes of the emergency event, the decision support service analyzes this attribute information, accesses the case library, performs knowledge reasoning, and generates a response plan text. Finally, the result of the decision support is returned and displayed on the decision support interface. The sequence diagram for the decision support module is shown in Figure 10.

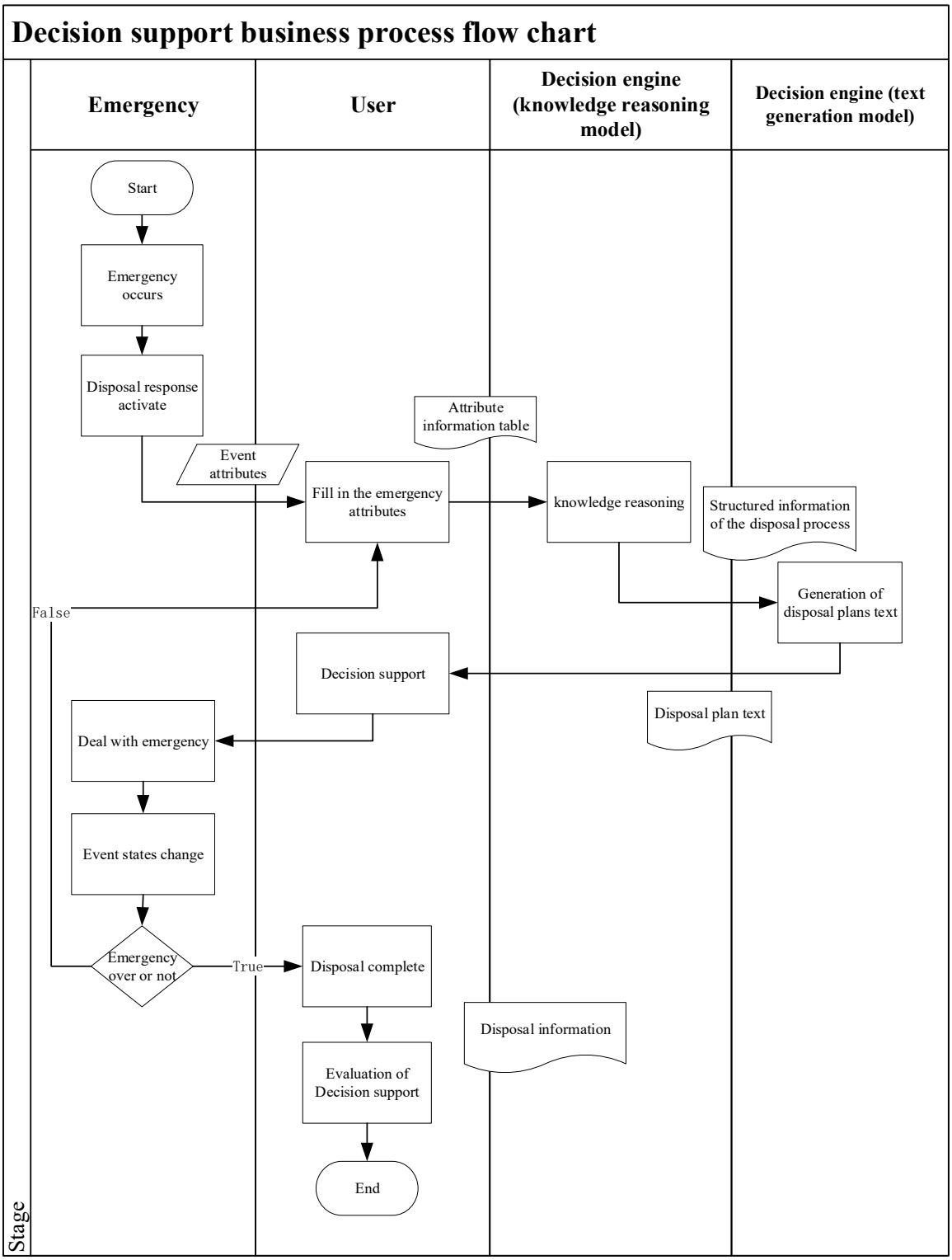

Fig 6. Full-Process Decision Support Business Process Diagram

3.1.4. Effect Evaluation Module

The main functions of the Effect Evaluation Module are twofold: filling out the effect evaluation form and calculating the evaluation score. After the emergency event is handled, the user fills out the effect evaluation form. The effect evaluation service then automatically calculates the evaluation score based on the form information and stores the result in the historical decision database, which serves as an important reference for future updates to the decision support engine. The sequence diagram for effect evaluation is shown in Figure 11.

3.2. Backend Management Subsystem Functional Module Design

3.2.1. Case Management Module

Cases are fundamental to the system's decision support. The quality of the case library largely determines the effectiveness of the decision support. The Case Management Module allows the administrator to perform a series of operations on the case library through the case management service, including the entry of new cases, modification of case information, and deletion of outdated or incorrect cases. The

sequence diagram for the Case Management Module is shown in Figure 12.

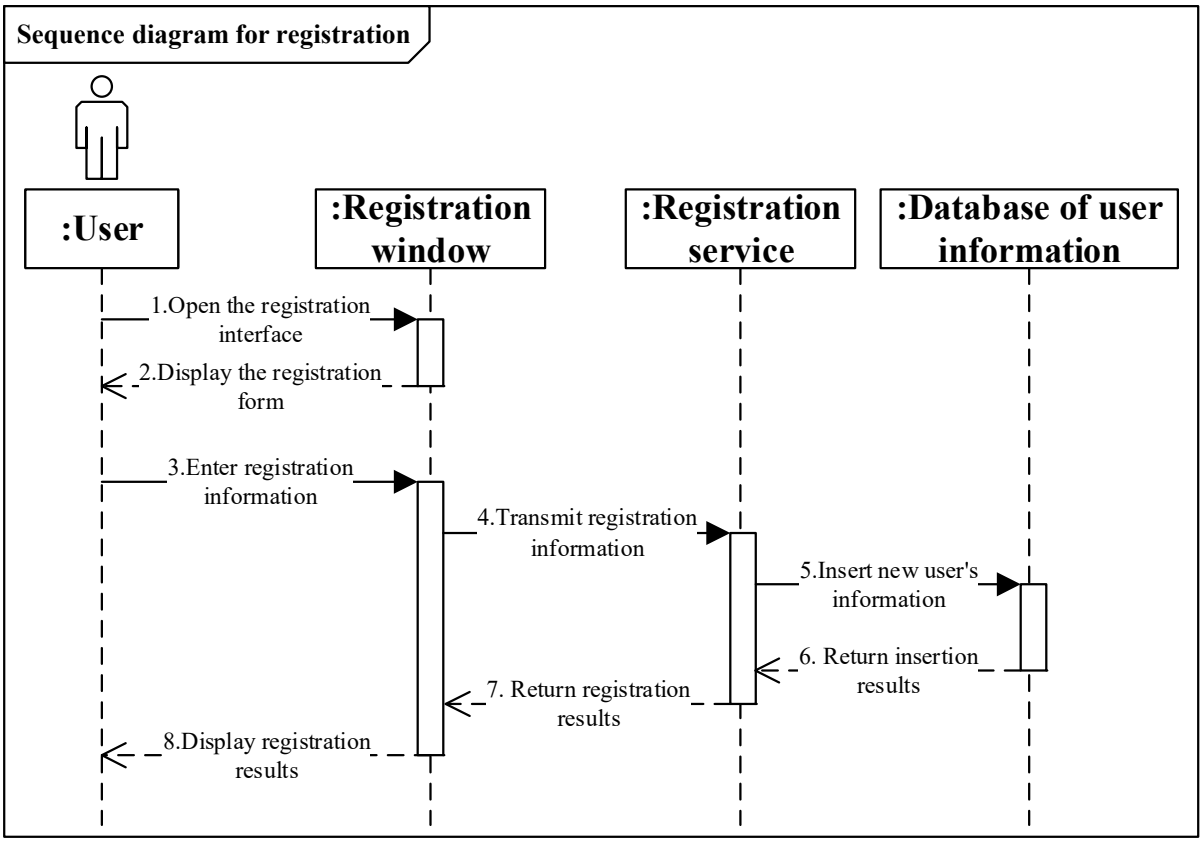

Fig 7. User Registration Sequence Diagram

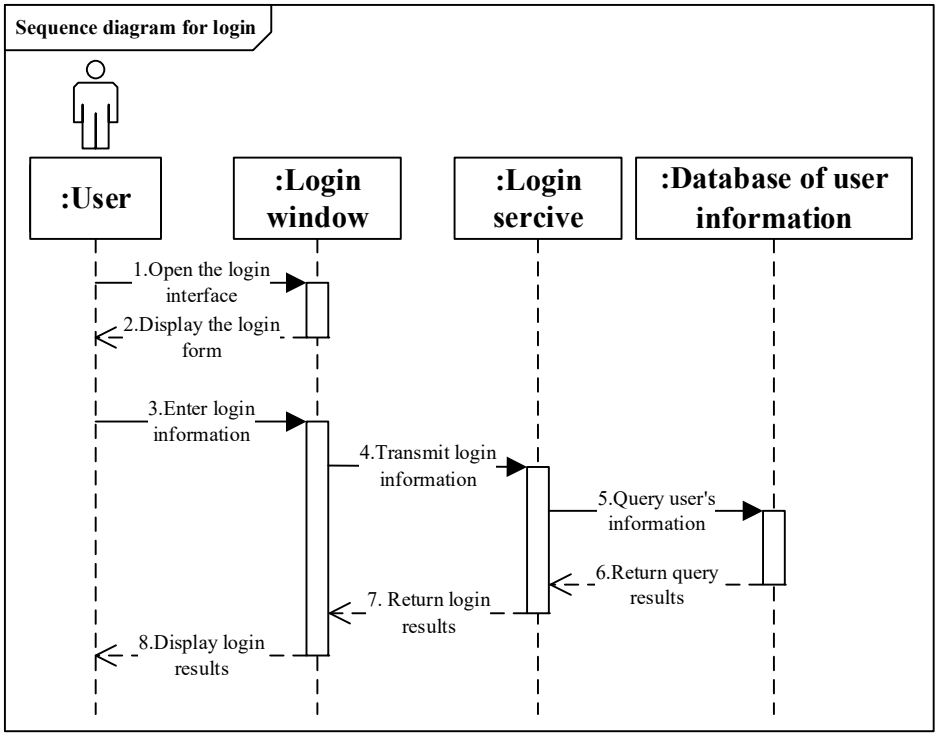

Fig 8. User Login Sequence Diagram

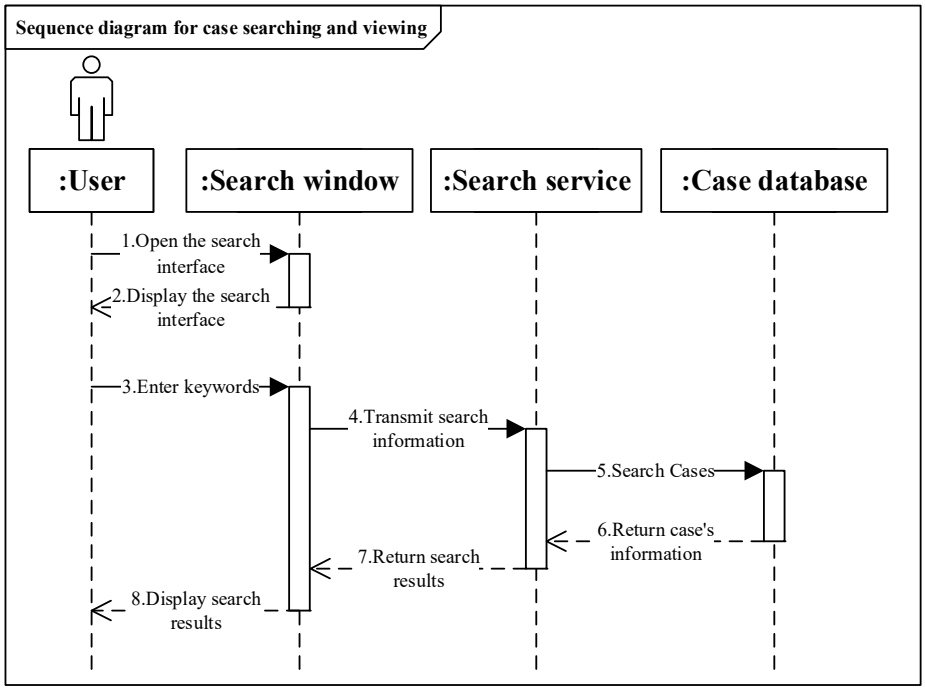

Fig 9. Case Retrieval and Viewing Sequence Diagram

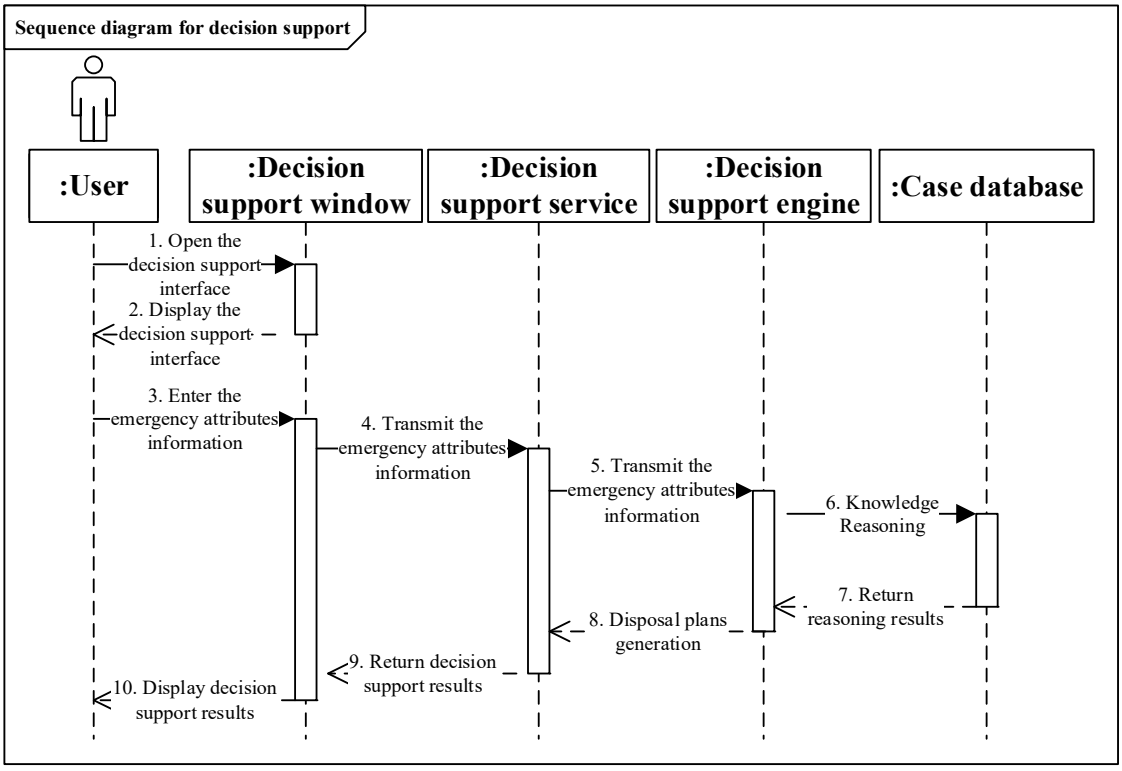

Fig 10. Decision Support Sequence Diagram

3.2.2. User Management Module

The User Management Module in the backend management subsystem allows for viewing, modifying, and deleting user information. This module is an important tool for the system's operating unit to manage user accounts in the frontend subsystem. The sequence diagram for the User Management Module is shown in Figure 13.

3.2.3. Decision Engine Management Module

The decision support engine is the core of the system, capable of processing the attribute information of emergency events and intelligently generating response decisions. To ensure its accuracy and adaptability, the system needs to design and implement management functions for the decision support engine, allowing administrators to perform decision engine management tasks through the decision engine management service, including optimizing reasoning weights, modifying reasoning rules, and updating text generation models. The sequence diagram for this functional module is shown in Figure 14.

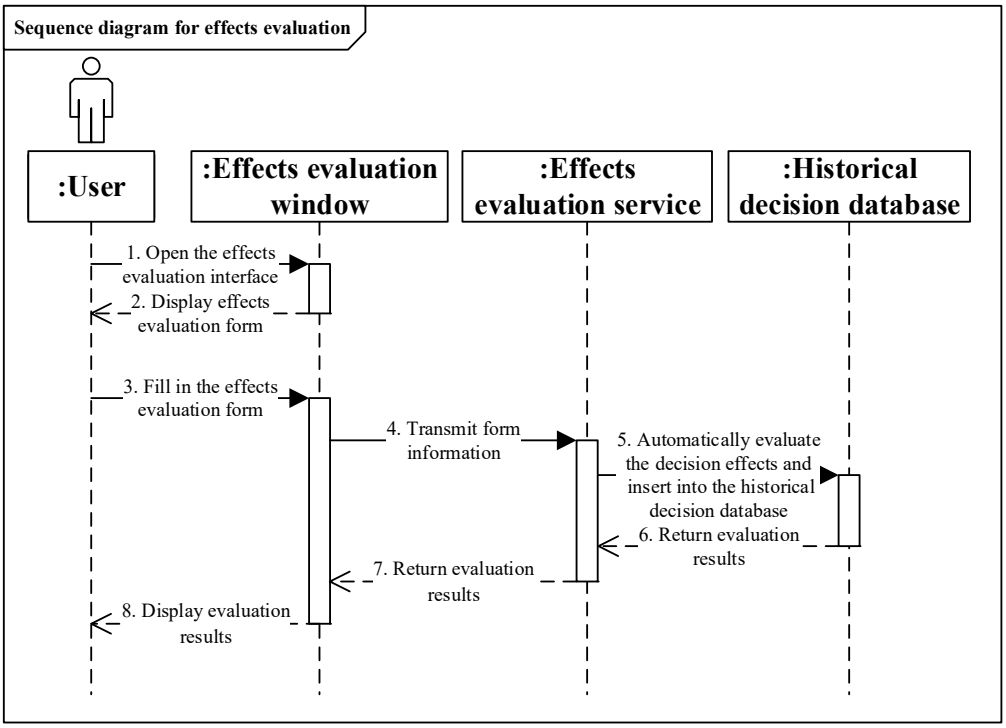

Fig 11. Effect Evaluation Sequence Diagram

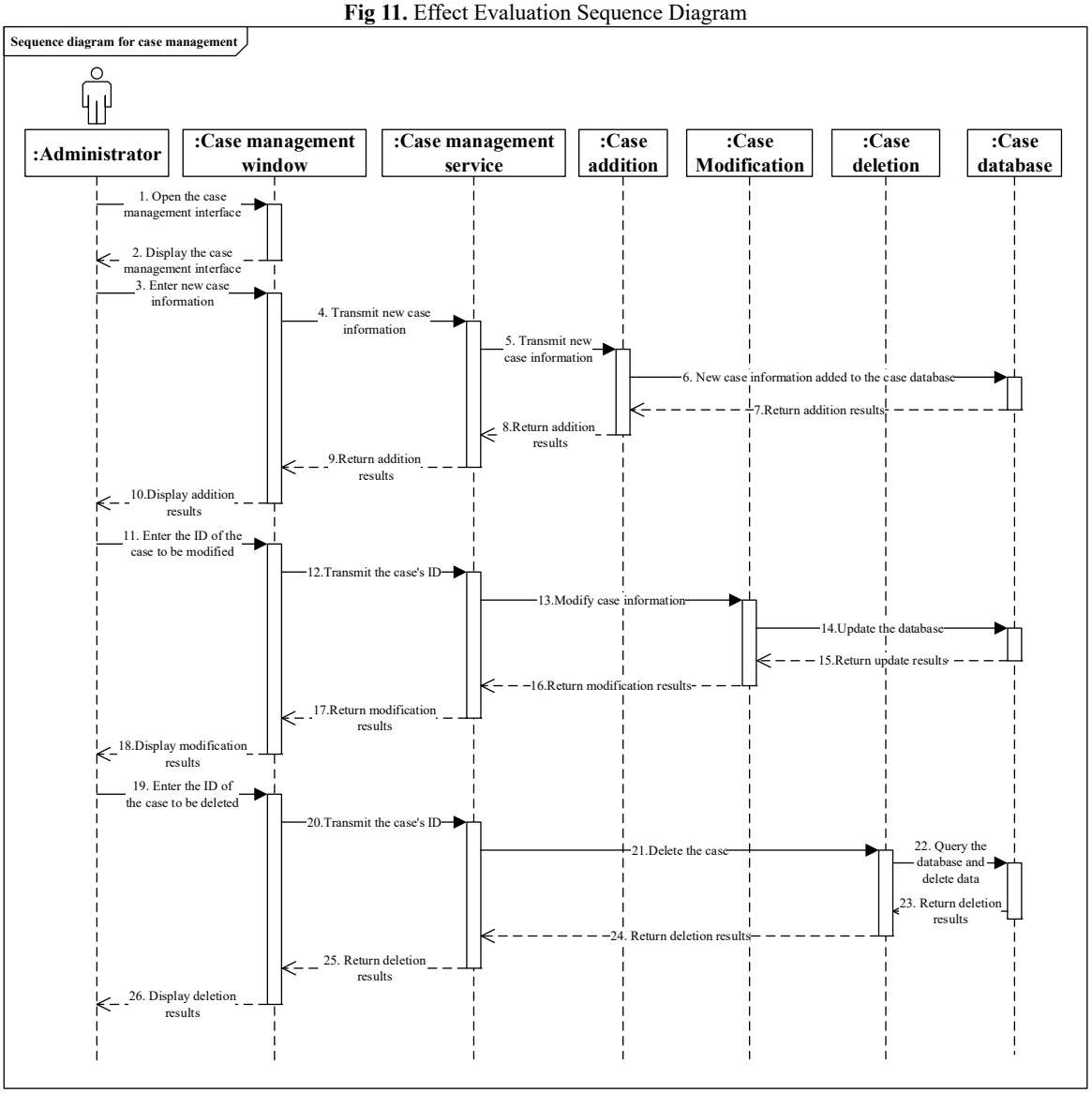

Fig 12. Case Management Sequence Diagram

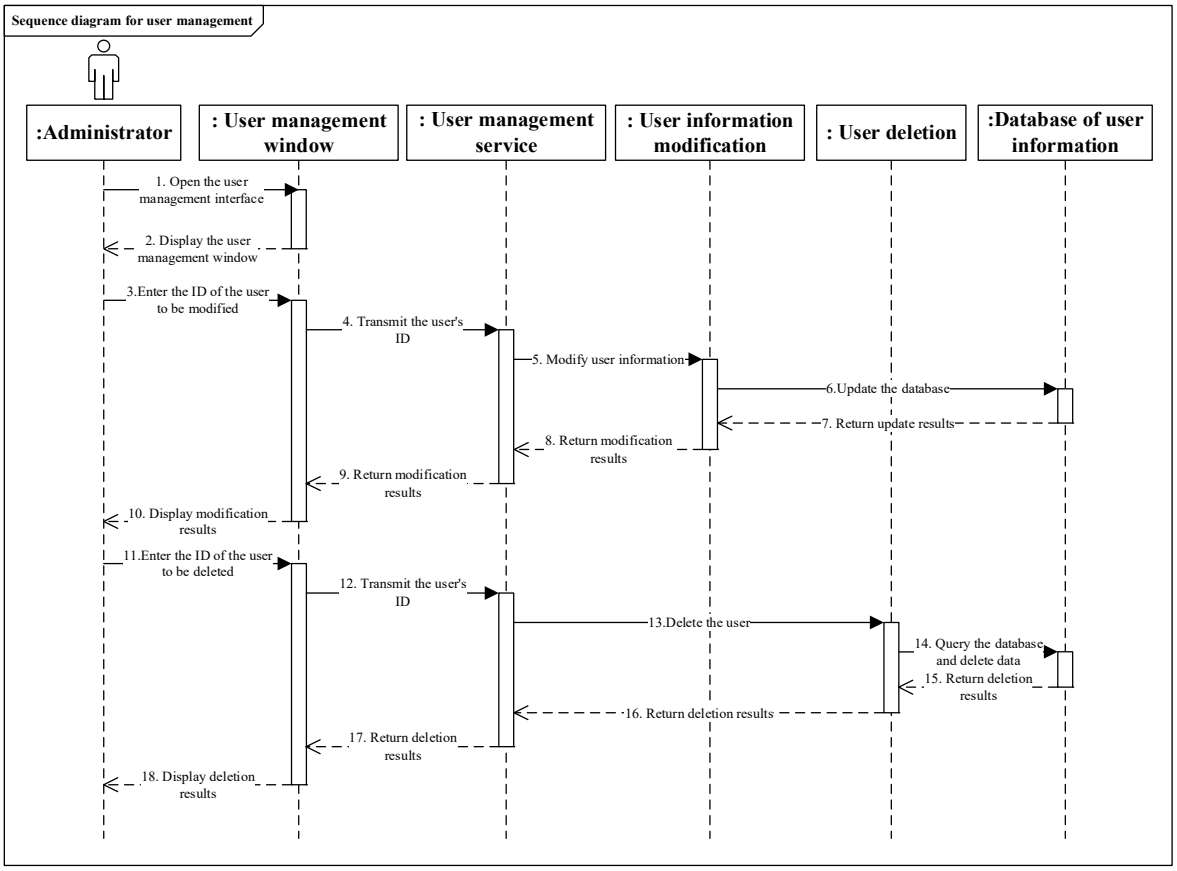

Fig 13. User Management Sequence Diagram

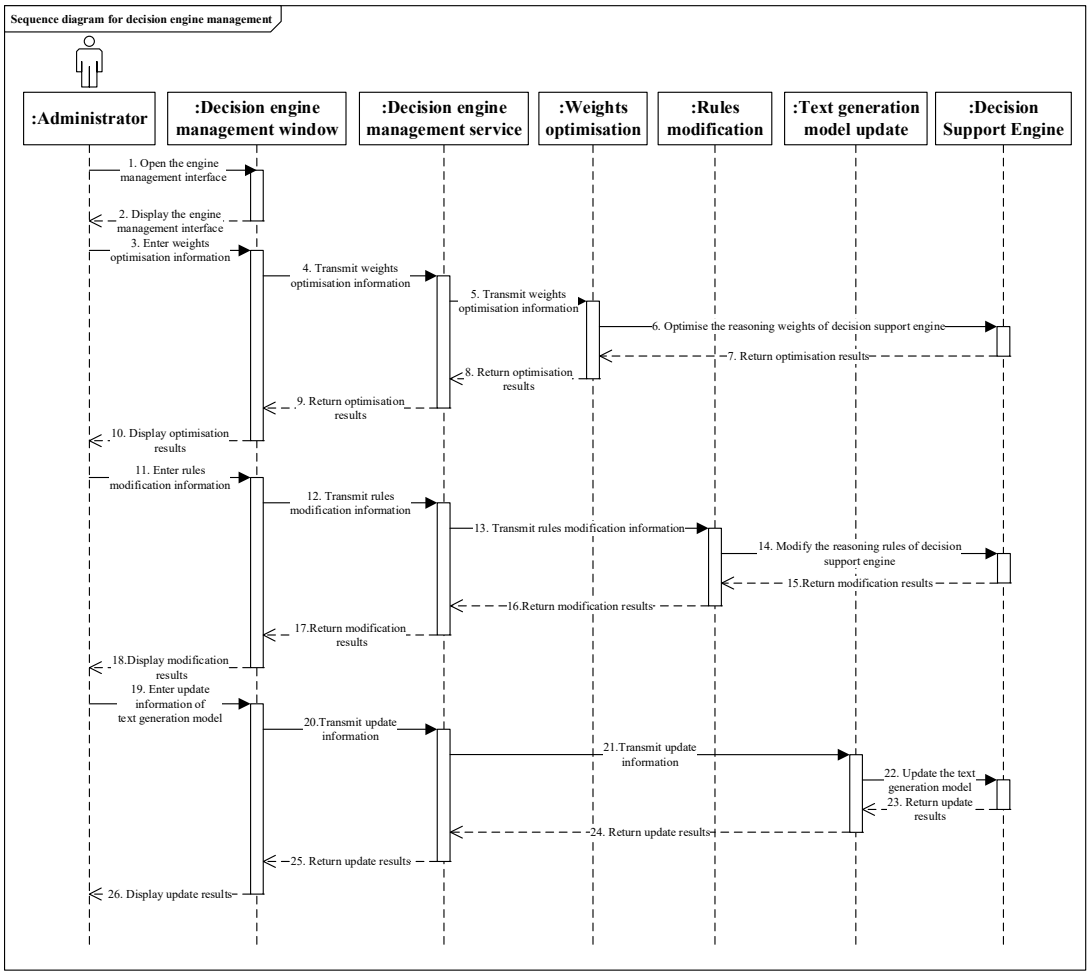

Fig 14. Decision Engine Management Sequence Diagram

3.3. Database Design

3.3.1. E-R Diagram

The database entities for the Airport Emergency Incident Response Decision Support System are divided into four entities: User Information, Administrator Information, Case Information, and Historical Decision Information. The definition of each entity is as follows:

User: {User ID, Username, Password, Position, Phone}

Administrator: {Administrator ID, Username, Password}

Case: {Case ID, Scenario Name, Terminal Name, Airport Type, Terminal Type, Event Occurrence Area, Event Phase, Event Category, Event Damage Level, Initial Response A01, Emergency Response A02, Police Response A03, Evacuation Measures A04, Firefighting Response A05, Medical Assistance A06, Subsequent Plans A07}

Historical Decision: {Decision ID, Historical Decision Step Sub-ID, Decision Details, Evaluation Score}

The system's E-R diagram is shown in Figure 15.

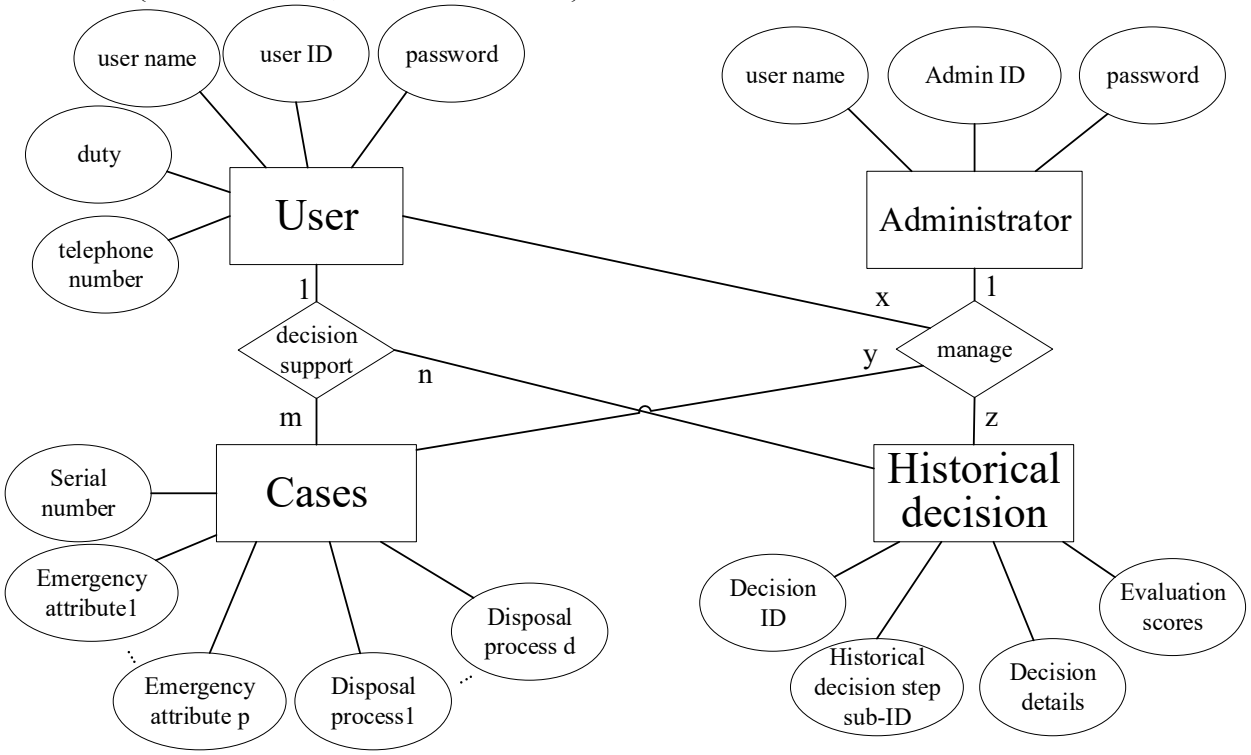

Fig 15. System E-R diagram

3.3.2. Physical Model

The physical model of the system database is shown in Table 1 to Table 4.

Table 1. User Database Physical Model

Item Name	Data Type	Length	Alias	Nullable	Primary/Foreign Key
User ID	varchar(30)	30	User ID	No	Primary Key
Username	varchar(30)	30	Username	No	
Password	varchar(30)	30	Password	No	
Position	varchar(30)	30	Position	No	
Phone	varchar(30)	30	Phone	No	

Table 2. Administrator Database Physical Model

Item Name	Data Type	Length	Alias	Nullable	Primary/Foreign Key
Admin ID	varchar(30)	30	Admin ID	No	Primary Key
Username	varchar(30)	30	Username	No	
Password	varchar(30)	30	Password	No	

4. System Implementation

4.1. Development Environment

Development Tools: Microsoft Visual Studio Code

Database Management System: Sql Server 2022

Browser: Goole Chrome

Web Server: IIS 11.0

Server Operating System: Windows 11 Professional

4.2. Main Functional Technical Implementation of the System

The core function of this system is the auxiliary decision-making for emergency incident handling. This function is achieved through the "knowledge reasoning + text generation" decision support engine. The decision engine

processes the emergency event attribute information using knowledge reasoning to obtain structured data of the emergency response process. Then, it uses text generation technology to generate response plan text based on the structured data of the response process. The decision support engine is shown in Figure 16.

Table 3. Case Database Physical Model

Item Name	Data Type	Length	Alias	Nullable	Primary/Foreign Key
Case ID	varchar(30)	30	Case ID	No	Primary Key
Scenario Name	varchar(30)	30	Scenario Name	No	
Terminal Name	varchar(30)	30	Terminal Name	No	
Airport Type	varchar(30)	30	Airport Type	No	
Terminal Type	varchar(30)	30	Terminal Type	No	
Event Occurrence Area	varchar(30)	30	Event Occurrence Area	No	
Event Phase	varchar(30)	30	Event Phase	No	
Event Category	varchar(30)	30	Event Category	No	
Event Damage Level	varchar(30)	30	Event Damage Level	No	
Initial Response A01	varchar(30)	30	Initial Response A01	No	
Emergency Response A02	varchar(30)	30	Emergency Response A02	No	
Police Response A03	varchar(30)	30	Police Response A03	No	
Evacuation Measures A04	varchar(30)	30	Evacuation Measures A04	No	
Firefighting Response A05	varchar(30)	30	Firefighting Response A05	No	
Medical Assistance A06	varchar(30)	30	Medical Assistance A06	No	
Subsequent Plans A07	varchar(30)	30	Subsequent Plans A07	No	

Table 4. Historical Decision Database Physical Model

Item Name	Data Type	Length	Alias	Nullable	Primary/Foreign Key
Decision ID	varchar(30)	30	Decision ID	No	Foreign Key
Step Sub-ID	varchar(30)	30	Step Sub-ID	No	Primary Key
Decision Details	varchar(3000)	3000	Decision Details	No	
Evaluation Score	varchar(30)	30	Evaluation Score	No	

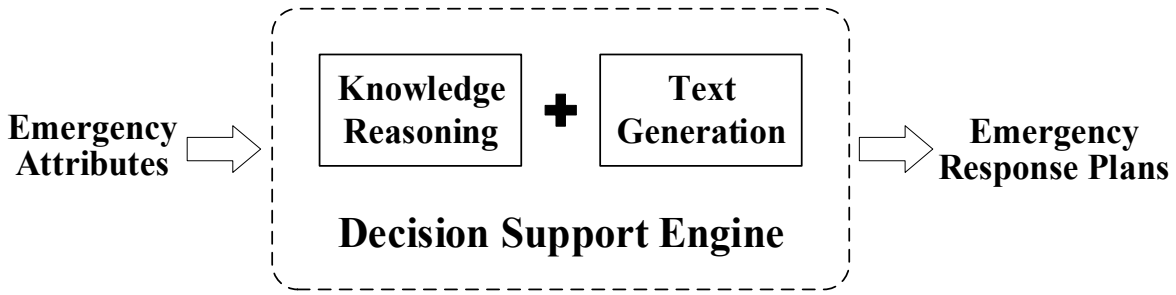

Fig 16. Decision Support Engine

4.2.1. Knowledge Reasoning

Knowledge reasoning uses a hybrid reasoning method[6]that combines case-based reasoning and rule-based reasoning. The case-based reasoning retrieves the case database, calculates the similarity, and finds the case most relevant to the current emergency event. Then, rule-based reasoning is applied to modify the retrieval results and fill in the gaps in the case, based on predefined rules. In this method, the similarity calculation of cases uses the most widely applied nearest neighbor strategy[7], and Bert[8] is used to calculate the text similarity of event attributes. The rule-based reasoning, based on case reasoning, corrects the retrieval results using a feature-weighted algorithm for attribute balancing and adjusts attribute parameters through a neural network-based weight optimization strategy. The knowledge reasoning method is shown in Figure 17.

4.2.2. Text Generation

Text generation is achieved through an improved seq2seq model in three stages, focusing on table-to-text generation. Specifically, in the first stage, bidirectional long short-term memory [9] (BiLSTM) networks and conditional random fields [10] (CRF) are used to perform sequence labeling on the record units in the table, identifying the key record units that make up the text. In the second stage, a structure-aware long short-term memory [11] (LSTM) unit is used to simultaneously encode both the table structure and record content to obtain vector representations that encapsulate the structural information. In the third stage, a pointer network [12] is first used for sequential planning, followed by decoding and prediction according to the planned content using a copy mechanism [13, 14], ultimately generating the descriptive text of the table. The three-stage text generation model is shown in Figure 18.

4.3. System Main Interface Display

The system has numerous interfaces, and displaying them all would take up a large amount of space. Here, we select the most representative interfaces from both subsystems for display, including the home page and login interface of both

subsystems, as well as the auxiliary decision-making interface of the frontend subsystem and the case entry interface of the backend subsystem.

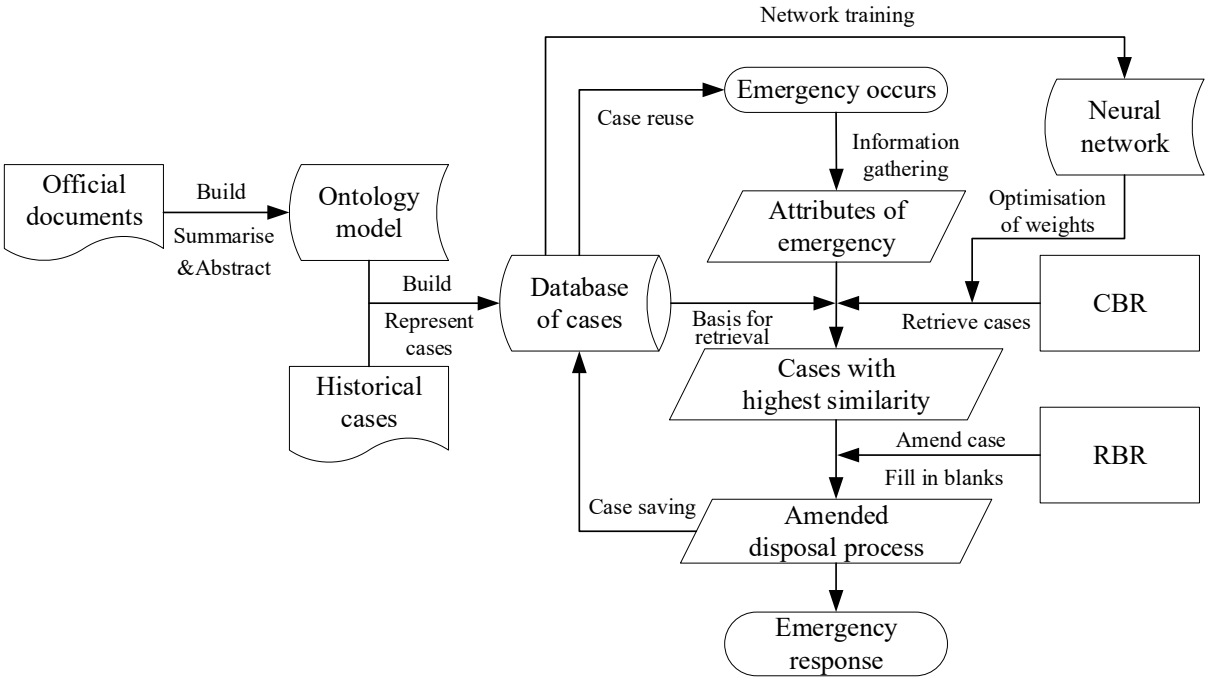

Fig 17. Diagram of the Hybrid Reasoning Method

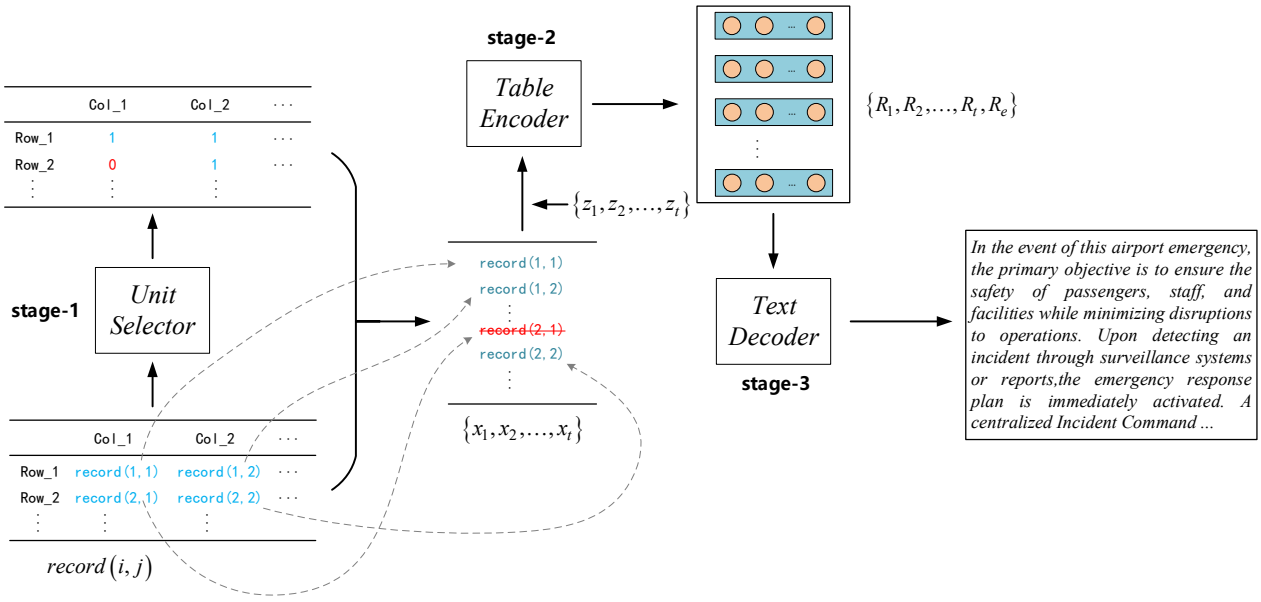

Fig 18. Three-stage Text Generation Model

4.3.1. Frontend Subsystem

When users access the front-end subsystem, they are first presented with the login interface. The login interface is simple and elegant, featuring various elements related to airports and emergency management. By entering the correct user ID and password, users can officially access the system. The login interface of the front-end subsystem is shown in Figure 19.

After logging in, the user is directed to the system homepage, which provides an easy-to-access and user-friendly interface. The homepage offers links to various functionalities, with a navigation bar on the left side. Users can access each functional module of the system through the homepage and the left navigation bar. The system homepage is shown in Figure 20.

The decision support interface is an important window for users to access the decision support function. To facilitate user interaction, the system divides the decision support function into four steps. The first step is to provide a form for users to collect information about the emergency event. Once the user completes the form with relevant event details, they can proceed to the next step. The information collection form is shown in Figure 21.

The second step enters the knowledge reasoning phase, where the user needs to choose the knowledge reasoning method to be used for the decision support. The available options are case-based reasoning, rule-based reasoning, and a hybrid reasoning combining both. After selecting the knowledge reasoning method, the user can click "Next" to start the knowledge reasoning process and proceed to the next

phase. This interface is shown in Figure 22.

Fig 19. Frontend Subsystem Login Interface

Fig 20. Frontend Subsystem Homepage

Fig 21. Decision Support - Event Information Collection Form

Fig 22. Decision Support - Knowledge Reasoning Method Selection

The next step is text generation, where the disposal plan text is generated based on the structured data obtained from knowledge reasoning. The user can click "Next" on this interface to begin the text generation. The text generation interface is shown in Figure 23.

Fig 23. Decision Support - Running Text Generation Model

Once the text generation is completed, the auxiliary decision-making process will conclude. The system will then display the decision results on the page for the user to review. The decision result viewing interface is shown in Figures 24 and 25.

4.3.2. Backend Subsystem

Similar to the frontend subsystem, the backend management subsystem also provides an aesthetically pleasing and simple login interface. This interface displays the relevant functions of the backend management subsystem and the units involved in emergency response. After logging in with an administrator account on this interface, administrators can access the backend management subsystem. The login interface is shown in Figure 26.

Similarly, the backend subsystem displays welcome information, system usage details, and provides an easy-to-use side navigation bar on the homepage, which will not be elaborated further here. The homepage of the backend subsystem is shown in Figure 27.

As one of the core functions of the backend subsystem, the case entry interface needs to implement functions such as displaying case library information, querying cases, and entering new case information. The case display function shows the data in the case library in a tabular format, with pagination. The page number to jump to can be selected in the lower-right corner. The case entry interface is shown in Figure 28.

In the case entry interface, clicking the "Add New Case" button leads to the new case information filling interface. After carefully completing the new case information form, clicking "Confirm Submission" will complete the entry of the new case. The new case information filling interface is shown

in Figure 29.

Fig 24. Decision Support - Decision Result Viewing

Fig 25. Decision Support - Decision Result Viewing

Fig 26. Backend Subsystem Login Interface

Fig 27. Backend subsystem homepage

Fig 28. Case Addition Interface

Fig 29. New Case Information Form

5. Summary

This paper, based on the practical needs of emergency management work by airport authorities, designs and develops an airport emergency response decision support system using the UML system analysis approach. The system is divided into a frontend user subsystem and a backend management subsystem. Relying on the "knowledge reasoning + text generation" decision support engine, the system implements the full-process decision support in the frontend subsystem, self-management of user information, and functionalities such as user management, case management, and decision engine management in the backend subsystem. The system features an attractive, user-friendly interface that can improve the informatization level of emergency management work and enhance the response efficiency for handling emergency events. The next step will focus on optimizing the system by improving the decision engine and enhancing system interactivity, further exploring actual user needs and feedback, and refining the system.

References

[1] Yuan Zhenlong. How to modernize emergency management systems and capabilities. Labour Protection, 2023(04):28-30.

[2] Deng Jie, Qi Mingliang, Chi Hong, et al. Modularity of emergency plan response procedures. Operations Research and Management Scie-nce, 2015,24(05):132-143.

[3] Huo Cuifang, Tang Zichao, Xue Chen. College emergency plan: generation logic, problem thinking and optimization path, Journal of China Emergency Management Science, 2023(02): 59-71.

[4] Li Hongchen, Deng Yunfeng, Liu Yanjun. Formal description of emergency plans, Journal of Safety Science and Technology, 2006 (04):29-34.

[5] Hu Zhenzhang, Zhang Zhikun. The application of UML model in information system requirements analysis. Modern Computer, 2024,30(12):62-66.

[6] Cai Shengsheng, Bu Fanliang. An assistant decision-making algorithm for public security emergencies based on case-based reasoning and rule- based reasoning, Computer and Modernization, 2019(09):7-11.

[7] Zhang Yi, Bu Fanliang. Design and implement-ation of emergency decision support system ba-sed on CBR and RBR, Software Guide, 2019,18(02):55-59.

[8] Devlin J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[9] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[10] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data: ICML, 2001. Williamstown, MA, 2001-01-01.

[11] Liu T, Wang K, Sha L, et al. Table-to-Text Generation by Structure-Aware Seq2seq Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 2018,32(1).

[12] Vinyals O, Fortunato M, Jaitly N. Pointer networks. Advances in neural information processing systems, 2015,28.

[13] Gu J, Lu Z, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning. arXiv preprint arXiv:1603. 06393, 2016.

[14] Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words. arXiv preprint arXiv:1603.08148, 2016.