Integrated Child Welfare System for Secure Reporting and Case Management in Malaysia

By

Justin Cheah Wei Hao

TP064130

APU3F2502SE

A report submitted in partial fulfillment of the requirements for the degree of

B.Sc. (Hons) in Software Engineering

at Asia Pacific University of Technology and Innovation.

Supervised by Mr. Daniel Mago Vistro

2nd Marker: Mr. Abubakar Santuraki

2025

Declaration of Thesis Confidentiality

DECLARATION OF THESIS CONFIDENTIALITY

Author's full name: JUSTIN CHEAH WEI HAO

IC No./Passport No.: 021113080389

Thesis/Project title: INTERGRATED CHILD WELFARE SYSTEM FOR SECURE REPORTING AND CASE MANAGEMENT IN MALAYSIA

I declare that this thesis is classified as:

- [x] CONFIDENTIAL
- [x] RESTRICTED
- [ ] OPEN ACCESS

I acknowledged that Asia Pacific University of Technology & Innovation (APU) reserves the right as follows:

1. The thesis is the property of Asia Pacific University of Technology & Innovation (APU).
2. The Library of Asia Pacific University of Technology & Innovation (APU) has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Author's Signature:

Date: 6 July 2025

Supervisor's Name: Daniel Mago Vistro

Date: 7 July 2025

Signature:

Figure 1: Declaration of The

Library Form

First Name: Justin

Middle Name (only if applicable) :

Last Name: Cheah

Title of the Final Year Project / Dissertation / Thesis : Integrated Child Welfare System for Secure Reporting and Case Management in Malaysia

Abstract :
To facilitate secure reporting and organised case management of child abuse cases in Malaysia, this Investigation Report examines the initial research, design considerations, and stakeholder analysis for the creation of a centralised digital system. The inquiry tackles important problems such limited public awareness, delayed interagency coordination, lack of anonymity in present reporting methods, and fear of reprisals. Relevant stakeholders including social workers, healthcare professionals, law enforcement and the public, participated in a structured survey to gain a better understanding of this issue and the needs of the system. The results highlight the necessity of secure anonymous reporting features, role-based access and case tracking. A literature review and system benchmarking against similar system informed the design considerations of the proposed solution. The findings imply that user confidence, reporting effectiveness, and interagency collaboration can all be considerably enhanced by a centralised digital reporting system with robust privacy features and collaborative procedures. With an emphasis on safe, inclusive, and transparent reporting procedures, this study aids in the creation of a child protection platform that is in line with SDG Goal 16: Peace, Justice, and Strong Institutions.

A few keywords associated with the work : Communication Technology, Risk Management, Threat Reporting, Web-Based Application

General Subject: Computer Software

Date of Submission: 17 September 2025

Figure 2: Library Form

Acknowledgement

First, the developer wishes to express gratitude to the Asia Pacific University of Technology and Innovation (APU) for providing the academic foundation, practical knowledge, supportive learning environment essential for completion of this project. The exposure to these relevant fields such as software engineering, web development, advanced programming language will be use significantly to the project implementation later. Besides that, the developer is especially grateful to Mr Daniel Mago Vistro, whose patience, guidance and valuable feedback have been fundamental throughout this journey. Furthermore, the developer would like to extend sincere thanks to Mr. Abubakar Santuraki for his time and efforts in evaluating the project as a second marker, which added valuable perspective to the overall work. The supervisor's support not only helped developer to refine on the idea of the system but also contributed greatly to personal and academic growth. Furthermore, the developer would like to extend gratitude also to the participants who participated in the survey. Their input has provided critical insights that shaped the direction of this investigation. Lastly, the developer would like to sincerely thank friends and family for their ongoing support during this process. Their expertise and drive have been essential in keeping the developer focused on specific, particularly difficult, parts of the investigation.

Abstract

To facilitate secure reporting and organised case management of child abuse cases in Malaysia, this Investigation Report examines the initial research, design considerations, and stakeholder analysis for the creation of a centralised digital system. The inquiry tackles important problems such limited public awareness, delayed interagency coordination, lack of anonymity in present reporting methods, and fear of reprisals. Relevant stakeholders including social workers, healthcare professionals, law enforcement and the public, participated in a structured survey to gain a better understanding of this issue and the needs of the system. The results highlight the necessity of secure anonymous reporting features, role-based access and case tracking. A literature review and system benchmarking against similar system informed the design considerations of the proposed solution. The findings imply that user confidence, reporting effectiveness, and interagency collaboration can all be considerably enhanced by a centralised digital reporting system with robust privacy features and collaborative procedures. With an emphasis on safe, inclusive, and transparent reporting procedures, this study aids in the creation of a child protection platform that is in line with SDG Goal 16: Peace, Justice, and Strong Institutions.

Keywords: Child abuse reporting, digital child welfare system, anonymous reporting, secure case management, interagency coordination, privacy-preserving platforms

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

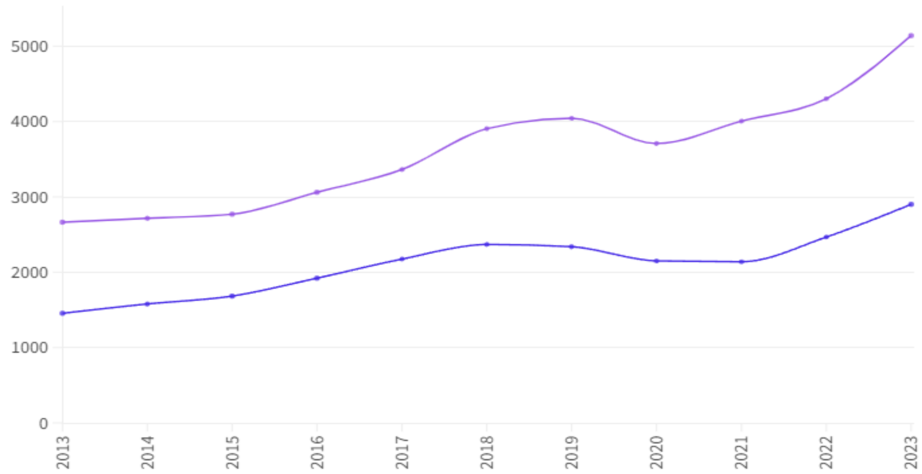


Figure 3: Number of child abuse cases reported to the Social Welfare Department (PFordten, 2024)

Child abuse and neglect are serious social problems that are receiving more and more attention everywhere, including in Malaysia. According to statistics, between 2000 and 2010, the Department of Social Welfare received complaints of between 1,000 and 3,257 cases of child abuse and neglect, demonstrating the concerning incidence of child maltreatment in Malaysia (Halim et al., 2023). More recently, in 2023, the number of reported cases surged to 8,401 which marks a 19% increase from the previous year and nearly double the figures recorded a decade earlier. Intervention measures are still under debate despite the establishment of multidisciplinary Suspected Child Abuse and Neglect (SCAN) teams in healthcare facilities across the country (Mydin et al., 2021). The COVID-19 epidemic has worsened these issues which leads to the decreased number of child protection referrals. As a result of this, there may be many cases that have gone unnoticed or the individuals who were not given the attention they sought (Garstang et al., 2020). Intervening child abuse and neglect in Malaysia goes beyond just the individual aspects of socioeconomic challenge, cultural norms, and the inefficiencies in the reporting and intervention systems (Yob et al., 2022; Ghani et al., 2019).

The unavailability of a reliable and secure online platform that facilitates individual reporting, case monitoring and interagency cooperation is also some main challenges in child welfare invention. Malaysia's reporting systems are still strongly reliant on paper form filling that causes delays, mix-ups, and writing data entry mistakes. Victims and witnesses might not come forward, besides concerns about being anonymous, security threats, and getting delayed responses. Elarousy and Abed (2019), in their research, contend that centralized databases are very significant in the child welfare sector, adding that apart from efficient data administration, better collaboration among institutions engaged in child protection programs is possible as well. Chadwick-Charuma et al. (2022) contend that ineffective channels for reporting deter victims from reporting abuse, which exacerbates the problem.

Additionally, partnerships between governmental agencies are the key to child protection efforts. As per the insights of Okato et al. (2020), the ability of social workers, law enforcement agencies, medical professionals, and child welfare organisations to effectively listen to each other clearly identifies and properly solves child abuse issues. But the lack of a digital hub that is one full-stop and is beyond any possible error removes the streamlining of access to and so increases the misunderstanding, delay, and disorganized offer, which possibly raises the already existing risk for children. The necessity for an all-encompassing digital solution that equally supports

1.2 Problem Background

1.2.1 Problem Statement 1: Lack of a Centralized Digital Case Management System

Because Malaysia does not have a single digital infrastructure for the management of child abuse cases, reports are scattered, records are redundant, and intervention is delayed. If a mechanism for the exchange of information is not available, reports are now being handled independently by different units such as social workers, police enforcers, medical experts and child welfare organizations. About the issues of inconsistent documentation, lost case files, and no follow-up activities that are not monitored, the timely intervention and protection of children at risk become very difficult. The number of reported cases is surprisingly high according to Halim et al. (2023), who found that between 2000 and 2010, there were between 1000 and 3257 recorded cases of child abuse in Malaysia. This is just a small part of the picture as the cases due to the inadequate inter-agency collaboration and underreporting are substantially more than the ones reported. Also, noncompliance in the outpatient follow-up of the child abuse cases also due to the missing tie-up report on the patient acutes healthcare agency, which leads to the fact that the patients are not getting the required follow-up treatment (Halim et al., 2023). In addition, the most known moment of this issue is that many children don't get the right care due to not timely interventions.

Also, socioeconomic conditions are an important part of the equation in cases of child abuse. Yob et al. (2022) assert that child maltreatment in Malaysia is particularly associated with economic conditions like unemployment and poverty. Child protection agencies as per their lack of centralized mechanism on socioeconomic factors monitoring and evaluation are not regularly doing their focus interventions, hence it stays the abuse cycle which is not checked. Also, due to the lack of a proper reporting system the trainings for early childhood educators in detecting and reporting child abuse are less efficient (Jaafar et al., 2024). Despite being trained to identify abuse instances, educators and medical professionals frequently struggle to navigate the intricate and disjointed reporting processes, which can result in lost case files, unmonitored follow-ups, and poor response times. The likelihood of prompt intervention and protection for impacted children is greatly diminished by the current child welfare framework's disarray.

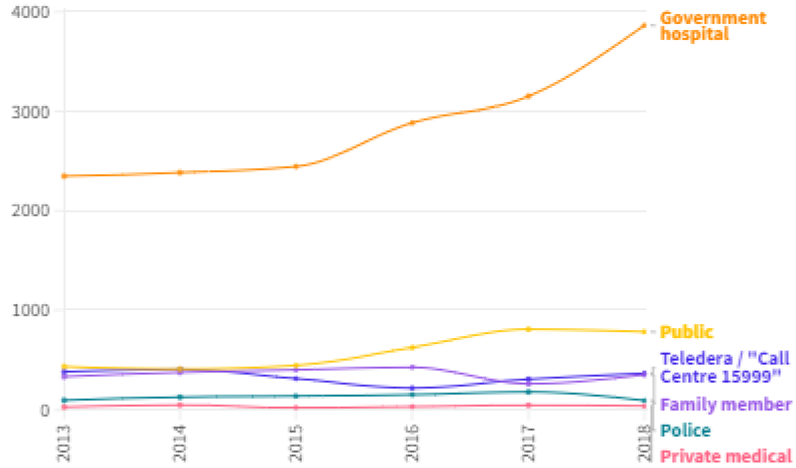


Figure 4: The number of child abuse cases reported to the Social Welfare Department and PDRM (PFordten, 2024)

Moreover, government hospitals continue to be the major place where cases of child abuse are referred, highlighting the vital role that doctors play in spotting and reporting abuse. Government hospitals referred the most child abuse cases, as shown in the above figure, with over 3,500 cases reported in 2018, a pattern that has been continuously rising. This increasing dependence on hospitals to identify misuse emphasises how crucial efficient interagency coordination is. Nevertheless, despite their vital function, the absence of a centralised system causes delays in case forwarding to appropriate authorities, including child welfare organisations and law enforcement, impeding coordinated case tracking and prompt response. In addition to hospitals, other organisations that handle reports include police stations, private clinics, and the national child abuse hotline, Talian Kasih 15999. However, because each of these organisations keeps its own records, reporting is dispersed, and case follow-ups are irregular. Due to ineffective information exchange and disjointed intervention efforts, instances may slip through administrative gaps in the absence of a centralised digital system, allowing abuse to continue unchecked.

1.2.2 Problem Statement 2: Fear of Retaliation and Lack of Anonymity in Reporting

The lack of anonymity and fear of reprisals are major obstacles to reporting child abuse in Malaysia, discouraging witnesses and victims from coming forward. Many people are reluctant to disclose abuse because they fear exposure, possible harm, or social stigma, especially children, family members, and even professionals.

Many Malaysian teachers were reluctant to report suspected child abuse, according to a study by Choo et al. (2013). The biggest concern, the lack of family and community support, was mentioned by 62 percent of them. Both the reasons given by Ja'Afar et al. (2024) and the case discussed of an early childhood education provider who does not report a suspected case because of a fear of the social stigma that goes with it and losing the relationship with the families prove that these persons are reluctant. Underreporting has also been fuelled by the statement of some providers who have expressed that they would rather not get involved in a situation that might have a negative influence on their personal security and their professional relationships.

Besides the business field, socioeconomic factors are a common and considerable reason not to report maltreatment. Generally, poverty and lack of jobs are seen to be the biggest culprits in the issue. According to Yob et al. (2022), poverty and unemployment exacerbate the issue because families that are undergoing financial struggles may fear the consequences of social rejection or economic loss if they open about the abuse. In many cases, the victims and their relatives refrain from speaking out due to their concern that it would further result in their isolation, monetary problems, or the involvement of authorities. These outcomes demonstrate the first place that tackles the implementation of a clandestine and protected reporting network which means victims and others concerned could report episodes without the risk of any adverse sequels. In addition, the insufficient statistics that feature the frontline players are ascribed to a lack of information and training. Based on the findings by Ja'Afar et al. (2024) more than 80% of the nursery staff involved in childcare were not involved at all in the training program focused on recognizing and reporting child abuse which made them feel insecure and doubtful when considering cases of possible abuse. Professionals are deterred from coming forward by this ignorance and lack of confidence, which emphasizes the necessity of safe, anonymous reporting systems to guarantee that child abuse instances are not missed.

In the absence of a systematic, safe, and anonymous reporting system, people in important professions will continue to be reluctant, enabling child abuse to continue unchecked.

Addressing this issue would require a Digital Child Welfare System that protects reporters' legal rights, maintains anonymity, and raises public awareness. Most crimes are filed by direct police reports, hotlines, or in-person complaints, all of which demand personal information from witnesses or victims. Victims face physical and psychological hazards because of the absence of safe and anonymous reporting channels, which deters them from seeking help or protection.

1.2.3 Problem Statement 3: Inefficient Interagency Coordination and Delayed Intervention

Effective child protection requires close coordination across several authorities, including child welfare organisations, social workers, law enforcement, and medical specialists. Unfortunately, there is no efficient interagency coordination mechanism in Malaysia's present child abuse response system, which results in misunderstandings, hold-ups, and poor case management. Law enforcement and child welfare specialists are essential to investigations and intervention, even if hospitals and medical personnel are the main point of referral for child abuse cases in Malaysia. These agencies still operate independently, nevertheless, and keep disparate, disorganised case files. Critical interventions are ultimately delayed because of missing case files, overlapping investigations, and duplicating efforts brought on by inadequately integrated documentation systems.

As per the research on the multi-disciplinary approaches to child abuse in other countries like the Children's Advocacy Centres (CAC) model, structured agency collaboration is the most effective way of promoting outcomes through the coordination of priorities and procedures (Herbert & Bromfield, 2019). But Malaysia has no structured mechanism for police, medical staff, social workers, and child welfare authorities to share data in real time, which contributes to the intervention being delayed and the potential of children being put at greater risk. The improvement of coordination would be attained by realization of centralized digital infrastructure that provides secure information sharing across the agencies and real-time case tracking. In addition to improving the safety and wellness of children who are at danger of abuse, this would lessen the existing inefficiencies.

1.2.4 Problem Statement 4: Lack of Public Awareness and Accessibility to Reporting Systems

Not being able to comprehend fully how to report child abuse constitutes one of the serious hindrances to the success of child protection measures because it finally causes the issue of underreporting. Numerous Malaysians, amongst them the victims, bystanders, teachers, and healthcare providers, are not informed well enough about the services, pages on the internet, and legal procedures of reporting abuse. The lack of awareness of various physical or psychological signs that children are being abused or the uncertainty of who and where to report the cases respectively, make the people ignorant of the situations and as a result, most possibilities to help these children go unnoticed.

Socioeconomic factors also inhibit individuals from disclosing their problems. It is possible that families who are living in poverty and those that are financially unstable will be afraid to report abuse due to the possibility of them not getting the government aid they need or getting a perception of being bad in the societal eyes. Commonly, international research shows that the inadequate child protective systems and the lack of public knowledge about reporting procedures are two main causes of the child abuse situation being raised although the real reports decreased in most places, especially in the case of crises or natural disasters (Seddighi et al., 2021). This indicates that the main cause of underreporting, especially in the vulnerable areas, is mostly the lack of awareness and systemic inefficiencies.

To successfully remove these barriers, a centralized digital reporting system, expert training programs, and comprehensive awareness campaigns are necessary. The implementation of these measures would ensure communication of proper procedures such as reporting and protective mechanisms to as many relevant people as possible including victims, witnesses, teachers, medical personnel, and child welfare workers. Along with the efforts to tackle the child abuse cases like overlook and neglect, it is a must to open the eyes of the public and spread the resources better to be able to report the offenses easily.

1.3 Project Aim

To design a Digital Child Welfare System that empowers users to report any incidents of child abuse through a secure and confidential reporting method, perform case tracking, and work inter-agency through a natural collaborative process to ultimately achieve a better intervention and case management of child abuse in Malaysia.

1.4 Objectives

- To investigate the primary factors that affects the effectiveness of digital reporting and case management systems in child welfare.
- To create a digital platform that is intuitive and accessible for both reporting and managing child welfare cases.
- To incorporate real-time connectivity, and organized case monitoring elements in the reporting system.

1.5 Scope

The proposed system is a fully digital reporting, monitoring, and collaboration platform that is meant to provide Malaysia's child protection agencies. The platform is the one that has been designed to ensure the discretion of all participants. Therefore, respondents and victims of the abuse can submit incidents safely, either anonymously or as registered users. With restricted access to case data, social workers, law enforcement, and government representatives will be able to examine and revise case statuses as necessary. Moreover, the system will have restricted permissions that should only be held by social workers, law enforcement officers, and authority representatives who will have the ability to view and amend the status of the cases.

Any device with an internet connection will be able to access the system since it will be designed as a web-based application. Basic case management functions like report submission, manual case status updates, and the creation of basic case records will be included. The platform will also have a secure communication system, which might be a discussion forum or chat system, which would make it easier for authorised personnel to coordinate actions and share updates. Basic data storage and access control features will also be included in the system to guarantee the security and privacy of private data.

As for the constraint, the system won't have a mobile application in its initial phase due to resource constraints. Furthermore, the platform will rely on manual reporting and expert assistance rather than real-time AI-based abuse detection or automated decision-making for investigations. To preserve the security and privacy of all users particularly victims and whistleblowers the system will adhere to data protection laws.

1.6 Potential Benefits

1.6.1 Tangible benefits

Table 1: Tangible Benefits

Benefit	Description
Centralized Case Management	Some Malaysian child protection organization work independently with fragmented paper records, which frequently results in duplicate and misplaced case files. It is anticipated that the suggested centralised approach will improve cross-agency visibility by reducing administrative redundancies and reporting duplication by 30 to 40%.
Secure and Anonymous Reporting	Due to the lack of anonymous reporting channels, many abuse cases remain unreported for fear of reprisals. Based on pattern seen in similar systems, it is projected to increase reporting rates by 15 to 25 % especially among frontline workers and vulnerable groups.
Efficient Documentation and Tracking	Response times are sometimes delayed by incomplete or misplaces case files which can cause delays in response. With digital tracking and structured case logs, the proposed system can reduce these issues by up to 50% which can improve response times.

Basic Reporting and Data Analysis	Currently, reports and statistics are compiled manually, consuming significant human resources.
Improved Interagency Collaboration	Allows child welfare agencies, social workers, and law enforcement to coordinate and track case progress effectively. It is expected that the system's dashboard and export capabilities will reduce report generation time by 70%, enabling quicker operational and policy decision-making.

1.6.2 Intangible benefits

Table 2: Intangible Benefits

Benefit	Description
Increased Public Trust in Child Protection Services	Encourages more people to report abuse cases by ensuring confidentiality and security.
Empowerment of Victims and Whistleblowers	Gives victims and witnesses a safe way to report abuse without fear of retaliation.
Raising Awareness on Child Abuse	Educates the public, caregivers, and authorities on the importance of timely intervention.
Encouraging Proactive Intervention	Enables agencies to respond faster, reducing long-term harm to children.
Ethical and Legal Compliance	Ensures the system follows child protection laws, maintaining credibility and safety.

1.6.3 Target Users

1. General Public (Victims/Witnesses)
2. Social Workers and NGOs
3. Law Enforcements (Police/Authorities)
4. Government Child Welfare Officials
5. Healthcare Professionals (Doctors / Nurses)

1.6.4 Functionalities/Deliverables

1. The system will allow General Public (Victims/Witnesses):

- To allow users to submit child abuse reports either anonymously or as registered users.
- To enable users to track the progress of their submitted reports (registered users only).
- To provide a channel for users to communicate follow-up queries through a secure contact form (registered users only).

2. The system will allow Social Workers and NGOs:

- To provide access for managing reported child abuse cases assigned to their organization.
- To enable users to update case status and document intervention progress.
- To facilitate communication with other agencies involved in the case via secure role-based messaging.

3. The system will allow Law Enforcement Officers:

- To allow officers to access detailed case reports assigned for investigation.
- To enable updates on investigation status and attach relevant findings.
- To support the upload of legal documentation such as evidence or investigation reports.

4. The system will allow Government Child Welfare Officials:

- To provide a dashboard interface for viewing analytics on case volume and status.
- To allow exporting of case in standardized formats.
- To facilitate communication with other agencies involved in the case via secure role-based messaging.
- To facilitate communication with other agencies involved in the case via secure role-based messaging.

5. The system will allow Healthcare Professionals (Doctors, Nurse):

- To allow submission of medical observation or evidence.
- To allow exporting of case in standardized formats.

6. The system will allow System Administrator:

- To manage user account and system roles.
- To monitor system performance.
- To configure access permission for all user groups.

1.7 Overview of the IR

Each of the four major chapters in this Investigative Report (IR) Introduction, Literature Review, Methodology, and Conclusion which helps readers fully understand the Integrated Child Welfare System for Secure Reporting and Case Management. The first chapter, Introduction, describes the challenges faced in Malaysian child welfare case management, with particular attention to the growing number of child abuse cases, the shortcomings of current reporting mechanisms, and the lack of safe channels of communication for victims, social workers, and law enforcement. A joint digital system for interagency coordination, secure reporting, and effective case tracking has been prioritized in the centralized digital system that gets introduced in this chapter along with the project's goals, objectives, and scope. Moreover, the system's possible advantages have been also examined with the main emphasis on its potential to enhance case management, to ensure the safety of victims, and to promote further child protection programs.

The second chapter, that is, Literature Review discusses the current studies, case studies, and equivalent systems that deal with child welfare reporting and case management. It is a study on the current method used in treating child abuse cases as well as the digital platforms used to improve security, anonymity, and interagency collaboration. It assesses alternative systems like Primero and Casebook. Through the obsolescence of older models and the emergence of new ones which are more competitive than others, internal resistance to change reduces and inasmuch it guarantees long-term survival and growth of the company. This study emphasizes the necessity for a child welfare system that is not only efficient but also safe and user-friendly and is specifically adapted to the Malaysian environment.

The development strategy, system architecture, and security concerns necessary for the proposed system's implementation are outlined in Chapter 3, Methodology. It describes the database architecture, chosen technology stack, and essential system features, such as real-time case progress tracking, secure reporting tools, and access control for various user roles. The data gathering techniques required to verify system requirements are also covered in this chapter; these techniques may involve surveys, expert consultations, or secondary research. To meet the demands of important stakeholders including social workers, law enforcement, and child protection groups, the methodology guarantees a methodical, evidence-based approach.

The last chapter, Conclusion and Future Enhancements, summarizes the main conclusions of the study and evaluates the viability and expected effects of the suggested system. It discusses difficulties encountered during development, such as problems with technology limitations and legal compliance, and it makes suggestions for possible improvements. To close current gaps and provide a safer, more efficient platform for reporting and managing child abuse cases in Malaysia, this chapter emphasizes the significance of a digital child welfare system. The system's development is based on this article, which offers a methodical framework for comprehending the technological, social, and ethical factors pertinent to the establishment of a safe, centralized online platform for child safety.

The conclusion and future enhancements, the last chapter are summaries of the essential aspects of the study and evaluation of the possibility, and the expected impact of the proposed system offered. It delineates the problems encountered during development, like the technological limitations and compliance with the law, and it proposes improvements. The single solution to the existing lack and the perfect route to a more user-friendly and safe management of cases on child abuse in Malaysia is the development of a child welfare system in digital format, as emphasized in this chapter.

1.8 Project Plan

TASK ID	TASK NAME	DURATION	START DATE	END DATE	STATUS
Ethic Form	Ethic Form	3 Days	16-Apr-25	18-Apr-25	Done
WBS-C1	**Chapter 1 - Introduction**				
WBS-C1-1	1.1 Introduction	3 Days	18-Feb-25	20-Feb-25	Done
WBS-C1-2	1.2 Problem Background	5 Days	20-Feb-25	24-Feb-25	Done
WBS-C1-3	1.3 Project Aim	1 Days	25-Feb-25	25-Feb-25	Done
WBS-C1-4	1.4 Objectives	1 Days	25-Feb-25	25-Feb-25	Done
WBS-C1-5	1.5 Scope	4 Days	25-Feb-25	28-Feb-25	Done
WBS-C1-6	1.6 Potential Benefits	4 Days	28-Feb-25	3-Mar-25	Done
WBS-C1-7	1.7 Overview of IR	1 Days	11-Apr-25	11-Apr-25	Done
WBS-C1-8	1.8 Project Plan	193 Days	18-Feb-25	29-Aug-25	Done
WBS-C2	**Chapter 2 - Literature Review**				
WBS-C2-1	2.1 Domain Research	17 Days	5-Mar-25	21-Mar-25	Done
WBS-C2-2	2.2 Similar Systems	15 Days	21-Mar-25	4-Apr-25	Done
WBS-C2-3	2.3 Technical Research	9 Days	3-Apr-25	11-Apr-25	Done
WBS-C3	Chapter 3 - Methodology				
WBS-C3-1	3.1 System Development Methodology	4 Days	11-Apr-25	14-Apr-25	Done
WBS-C3-2	3.2 Data Gathering Design	10 Days	14-Apr-25	23-Apr-25	Done
WBS-C3-3	3.3 Analysis	5 Days	24-Apr-25	28-Apr-25	Done
WBS-C4	**Chapter 4 - Design and Implementation**				
WBS-C4-1	4.1 Introduction	4 Days	7-Jul-25	10-Jul-25	Done
WBS-C4-2	4.2 Design	4 Days	11-Jul-25	14-Jul-25	Done
WBS-C4-3	4.3 Database Design	2 Days	15-Jul-25	16-Jul-25	Done
WBS-C4-4	4.4 Interface Design	4 Days	17-Jul-25	20-Jul-25	Done
WBS-C4-5	4.5 Test Case Preparation	4 Days	21-Jul-25	24-Jul-25	Done
WBS-C4-6	4.6 Execution	6 Days	25-Jul-25	30-Jul-25	Done
WBS-C4-7	4.7 Summary	1 Days	31-Jul-25	31-Jul-25	Done
WBS-C5	**Chapter 5 - Result and Discussion**				
WBS-C5-1	5.1 Unit Testing	5 Days	1-Aug-25	5-Aug-25	Done
WBS-C5-2	5.2 User Acceptance Testing	15 Days	6-Aug-25	20-Aug-25	Done
WBS-C5-3	5.3 Summary	1 Days	21-Aug-25	21-Aug-25	Done
WBS-C6	**Chapter 6 - Conclusion**				
WBS-C6-1	6.1 Critical Evaluation	1 Days	22-Aug-25	22-Aug-25	Done
WBS-C6-2	6.2 Limitation	1 Days	23-Aug-25	23-Aug-25	Done
WBS-C6-3	6.3 Future Enhancement	1 Days	24-Aug-25	24-Aug-25	Done

Figure 5: Project Plan

Chapter 2: Literature Review

2.0 Introduction

A revolutionary chance to strengthen interagency cooperation, improve case management, and guarantee prompt interventions for children in danger is presented by the growing dependence on digital platforms for child welfare reporting. However, resolving important issues with data privacy, system design, reporting mechanisms, and operational efficiency is necessary for the successful deployment of such systems. Aspects of digital child welfare systems that are examined in this chapter include interagency coordination models, mandatory reporting frameworks, technological solutions for secure reporting, and the social and legal contexts that influence system adoption.

In a study, through their comparative legal analyses on obligatory reporting processes of domestic violence cases, have brought out the import of systematic reporting frameworks, explicit legal tasks, and integrated systems design to enhance reporting results and prevent the vulnerable population (Randawar et al., 2024). These elements of the child's welfare reporting systems are the lessons that feature telling points while legal transparency and operational effectiveness are key factors to the acclimatization of such systems. Taking advantage of a thorough exploration of pertinent literature and real cases, this section explores the difficulties and potential ways to optimize child protection reporting systems. Focus is given to enhancing user trust, improving system transparency, strengthening operational coordination, and ensuring alignment with national and international child protection standards.

2.1 Domain Research

2.1.1 Effectiveness Factors in Digital Child Welfare Reporting Systems

A digital child welfare reporting system is an electronic system created to secure, fast, and inclusive procedures for reporting and managing cases of child abuse. Notwithstanding, its actual usefulness relies on a mixture of technology, society, institution, and the individual human factor. These elements affect system integration and further decide the effectiveness of reporting, coordination, and intervention outcomes. This paragraph is a scrutinization of four basic interrelated factors that performance and digital child welfare platform trust depend directly on.

2.1.1.1 Accessibility, Usability and Emotional Barries

Both the technical capabilities of a digital child welfare reporting platform and the level of assistance it gives to the users who are in a dire situation will be the things it depends on to be successful. The said users can be different like witnesses, victims, carers, social workers, teachers, and medical personnel. A platform that is not user-friendly, especially for individuals facing emotional problems and lacking IT skills, may be a challenge and a source of frustration for them. At this point, the platform should not only act properly but also give out emotional clarity, simplicity, and reliability. To make this system better, the challenge lies in ensuring both the accessibility and usability across all levels of society.

Interestingly, a study has revealed that in areas that have scanty digital literacy, the existence of sites that are not mobile supported or have very complex interfaces inhibits the users from even starting the report (Yu et al., 2024). The emotional or cognitive burden which is now present has increased the level of this usability problem. There are challenging situations like making a report on child abuse that little interface errors is the reason which users condemn the process. Moreover, the poor design of the system can also result in unexpected problems where the users decide to drop out of the process due to stress or confusion. A paper focused on cognitive walkthroughs in the context of medical information systems claims that user fatigue and dissatisfaction can stem from a misalignment between the design of the system and user experience (Bagheri et al., 2023). A system that does not consider users' needs and perceptions is poorly designed and may create administrative challenges and, ultimately, the disapproval of the system. In the same way, the mandatory user-centred design in preventing the technological-off sponsoring is substantiated in the research conducted on assistive technology which says that the users uninvolved in the design period are more likely to reject the technologies because of the issues with their usability (Pinjatela, 2024).

The collective effect of these three studies is to highlight the fact that the design flaws not only irritate the users but also can forbiddingly hinder some of the crucial interventions by making obstacles at the emotionally significant places. The expression of the situation is that although there may be numerous potential reporters, the platform that does not set user needs as the priority won’t have them finish their contributions, thus leaving considerable incidents of abuse uncovered and basically uninvestigated within child welfare settings. This contributes to a silent underreporting crisis, where systemic design failures become indirect barriers to justice and protection. It is not simply a matter of improving aesthetics or layout; it is about removing invisible roadblocks that prevent timely intervention.

The emotional distress involved complicates the situation even further. A report in the year 2024 reviewed the fact that even if the users are conscious of the relevance of the reporting, the emotional burden of that choice most often is what makes them unable to take it (Lee et al., 2024). The act of reporting the suspected child abuse is not a simple act at all. It can bring about feelings of anxiety, fear of revenge, or concern over having hurts the family concerned. Several caregivers and inhabitants of the community are more preoccupied with the idea of making mistakes or being judged than their wish to do something about the situation. Individuals who may be on occasion of getting the reporting process underway perhaps doubt themselves, fearing the possibility of legal entanglement or possible social backlash in case the case become more complicated than expected. This kind of statistic is covered in public data like the figure below, which gives a list of the most often cited reasons why people do not complain to the authorities about the suspected child abuse cases (Lee et al., 2024). Uncertainty on whether the conditions are appropriate for calling it abuse, concern about personal confidentiality, and fear of the legal consequences are the most frequently named reasons. These worries are rooted in fear, confusion, and a lack of system support rather in selfishness nor indifference. This reflects how poorly thought-out digital tools can accidentally worsen emotional distress. This implies that, in addition to functionality, technological design must consider emotional comfort, especially when working with vulnerable or fearful people.

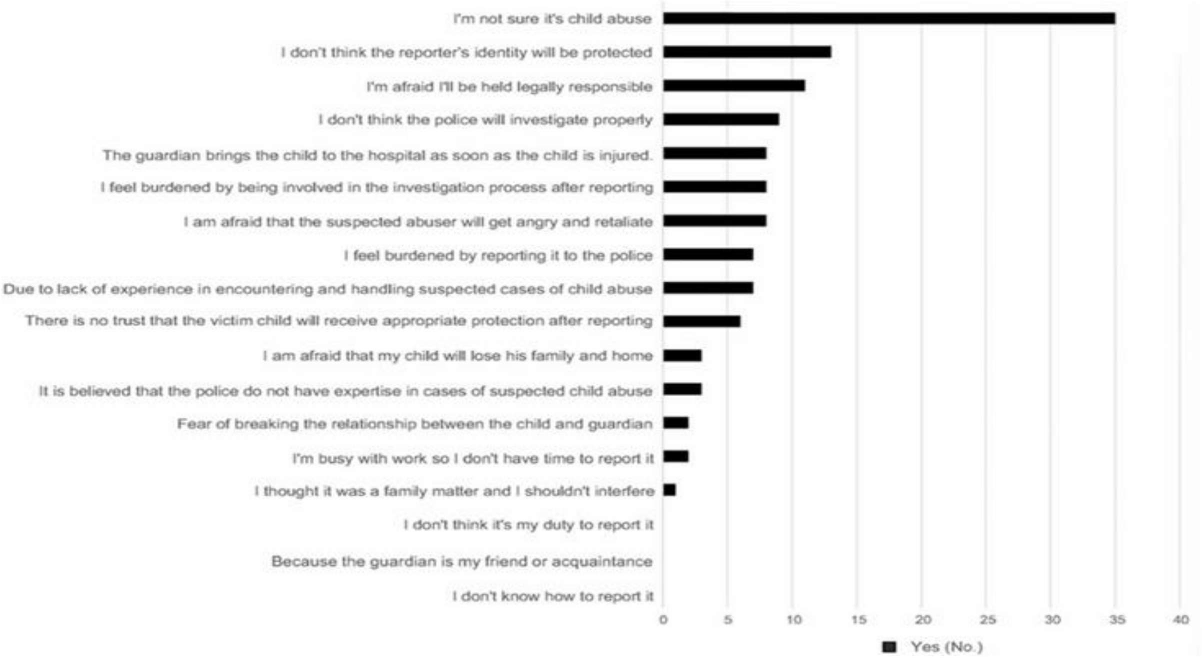


Figure 6: Top three reasons were uncertainty about whether the situation constituted abuse, concerns over confidentiality, and fear of legal responsibility (Lee et al., 2024)

Digital reporting platforms should be premeditated with an emphasis on human behaviour and emotion to mitigate these barriers. Research indicates that frontend systems, which have their own guides in steps, provide few instructions, and give real-time feedback, assist users in being more confident and safer when they make a report (Nwaimo et al., 2024). To battle emotional fear and the decrease of usability in this case, the inclusion of elements that offer comfort and the natural progression for the user through every step are a must. It is very important to use visual cues, plain language and contextual help options which all are keys to having a good sense of direction and safety during the reporting experience. These features are therefore not to be taken as optional improvements but as basic elements of a child protection platform that functions well.

Next, a study has focussed the importance of culturally appropriate language and the inclusion of multiple language options in increasing engagement with digital systems (Kwon et al., 2022). User comfort and trust are significantly influenced by cultural relevance in addition to usability and emotion. When users realizes that the platform acknowledges their native language or background, users will be more likely to trust the process, whether this involves submitting a report or reviewing and escalating an existing case. This applies not only to interface language but also to tone, layout, and usability. In a multicultural society like Malaysia, it is critical to ensure all user groups feel represented and respected. To further increase interaction with digital systems, the study also highlighted the significance of using terms that is culturally relevant and offering a variety of language alternatives. Whether the job is to file a report or evaluate and escalate an existing case, users are more likely to trust the process and finish their duties when they see that the platform understands their original language or cultural background.

Importantly, emotional hesitation does not apply to public only by also by professionals. Some studies focus specifically on parents or caregivers that are in low-resource settings who usually turn to informal community networks for support, the insights extend to other users as well (Doelman et al., 2024; Doh et al., 2021; Howard et al., 2025). Victims, witnesses and even trained professionals can experience similar overwhelm or uncertainty when formal system feel intimidating due to concerns which can lead to delays or disengagement from the system entirely. Moreover, the absence of emotional and technical guidance during the reporting and case management process can also lead to system abandonment

Additionally, study showed that digital systems must create a feeling of emotional security before users can interact with them (Kwon et al., 2022). This involves having a calm visual design, offering role specific support, ensuring transparency about what happens after a report is submitted or reviewed, and providing clear assurances that both personal identity and case data will remain protected which will make the users feel recognised. If users feel recognised, understood, and supported in their duties, they are more likely to act. These platforms that can enhance coordination, boost involvement, and finally help to better protect children by means of inclusive design, support of various user roles, and trust building. Usability, emotional clarity, and cultural design are all connected components. Regardless of how technically sound digital platforms are, without them, users may view them as intimidating, unfriendly, or dangerous. Through inclusive design, support for many user roles, and trust-building, these platforms may improve coordination, increase engagement, and ultimately aid in improved child protection.

2.1.1.2 Community Awareness and Trust Building

Lack of awareness or lack of trust will cause even the best-designed systems to work poorly. Promoting broad use of digital reporting systems requires raising public knowledge of them and boosting trust in their operation. Community programs that educate parents on their legal rights and how to use digital tools have shown strong potential to increase both the volume and quality of reporting (Tahir & Cobigo, 2025). When families feel informed and empowered, they are more likely to engage with these platforms and view them as allies rather than threats. Moreover, the long-term trust that is created in organizations with these values of transparency, responsiveness, and inclusive design gets strengthened particularly in settings where people often do not disclose the information due to the fear of exposure or social judgement. Nevertheless, just being aware of the issue is not enough. Besides, users must be assured of their safety, acceptance, and safeguard during the whole operation of the system. Thus, the integration of the system is not just a technical issue, it is also a social one, which is emotional security and assurance in the results. Theres an examination of child protection systems in Philippines wherein the issues identified were disorganized record-keeping and insufficient communication among the stakeholders that made it difficult to make timely interventions and caused the compromise of coordinated care that can be discussed (Bastian et al., 2021; Roche & Flynn, 2020). The wrongdoings that interfere with intervention are also the ones that cause the lacking trust in the users in charge of the reporting systems. Furthermore, when families or professionals run into a situation where their cases are misplaced or wrongly dealt with, they can get disheartened from future activity in the reporting system.

Next, findings suggest that the transition from the traditional paper system to the digital platform is known to be more effective, and transparent in handling child protection cases is becoming more accepted. To build trust with families and communities, the use of digital technology can help to decrease human error, increase the quality of documentation and provide more quicker follow up (Paki et al., 2023). There is growing evidence that digital case management solutions can contribute to the restoration of public trust by providing organised case tracking and increasing system transparency, even if empirical research on long-term digital adoption is still in its infancy. In this sense, making an investment in dependable digital infrastructure is not just a technological advancement but also a first step in promoting more accountability and involvement in child welfare procedures. In this way, investing in reliable digital infrastructure is a first step towards encouraging more responsibility and participation in child welfare processes, in addition to being a technological breakthrough. In the end, trust is a result of reliability as much as messaging. Public and professional confidence is strengthened by a responsive and open system, which turns reporting into a shared duty rather than a burden to be feared.

2.1.2 Secure and Privacy-Preserving Frameworks in Child Welfare Platforms

In modern digital platforms that handle sensitive user data, privacy and security is crucial for preserving public trust and promoting active engagement (Berger & Torres, 2024). Users expect that their personal information will be shield against unauthorized access and misuse whether in healthcare system, financial services or social welfare applications. This expectation is even more critical in systems related to child welfare where reports often contain highly sensitive and emotional content. To ensure user feel safe, privacy-preserving and secure design principles must be incorporated into any of the platform (Wharton, n.d.). This section explores key considerations in building trust through confidentiality protection, secure data handling, and transparent privacy guarantees, focusing on their relevance to digital child welfare reporting platforms.

2.1.2.1 Anonymous Reporting and Confidentiality

The fear of punishment, social rejection, or legal issues often causes people to remain sileng when they are suspecting child abuse cases. The fact that reporting might lead to negative consequences or create misunderstandings between the people involved, is the reason why these feelings are very strong in communities knitted tightly or in family-oriented societies. But the facility of anonymous reporting is the way to remove these obstacles by giving individuals the chance to provide information anonymously.

Backing this up, studies have shown that the anonymous reporting option is the best way to incentivize people to come forward about suspected child abuses, especially in the cases where retaliation fear and stigma are a strong barrier (Afrouz & Vassos, 2024; Towhid et al., 2023). These options allow the users to have the emotional protection they needed to act on their worries particularly in the situations where they might fear the community's negative responses or being wrongly accused as instigators.

In addition, a study has shown not only the psychological barrier related to cyber-dating abuse victims but also more general psychological barrier corresponding to the victims who are dissuaded from seeking help by fear of judgment and revenge (Afrouz & Vassos, 2024). The findings are true in these settings where similar emotional and social misgiving may stop people form reporting the suspected case. Towhid et al. (2023) also embrace a feature of anonymous reporting in a study on the protection of minors form online sexual exploitation to promote the reporting of unsafe practices, especially in those environments where trust in authorities might be an obstacle to the disclosure. The individuals who discussed the issue referred to the possible benefits of anonymity in the case of reporting behaving safer, which is especially true in the areas where there is a mistrust of unauthorized representatives or fear of system abuse that is likely to hinder their disclosure. But in such contexts, it is the proper channels of communication which usually lack that would have prompted the individuals to contribute information on their own accord.

Additionally, study conducted a systematic review of technology-based child abuse prevention programs that focuses on the role of technology-facilitated programs in encouraging more participation form the parents and the members of community (Altunpul & Kocturk, 2024).. Results of the survey shows that the fact the easiness of access and the safety of digital platform can gave rise to more awareness, more responsiveness and the overall progress in child protection measures.

The argument for programming digital systems to minimize emotional and procedural barriers is further validated by the above observations. A critical step forward is the integration of anonymous reporting channels into digital child welfare platforms. This approach can significantly increase user participation, reduce reporting hesitations, and foster a stronger sense of safety and trust within communities. It ensures that individuals whether they are witnesses, victims, or concerned professionals feel empowered to act without fear of judgment or retaliation.

2.1.2.2 Basic Data Encryption and Access Control

These days, it is a necessity for most internet platform, especially those which are responsible for handling sensitive data and very personal information, to have secure transmission and storage system. To keep trust of users and avoid instruction, it is better for internet platform to follow the route of strong data protection policies that adapt to constantly changing privacy legislation. Data encryption, by converting the information into a code, is essentially a defence that even the attacker with access to the site is still unable to read the data in there.

An outcome from a recent survey given by Channel Impact (2024), underlined the gravity of encryption flaws by mentioning that data breaches about 33% were partly due to the lack of encryption while and additional 25% were due to poor encryption protocols like using weak keys. The results of this research where substantiated by the work that was mentions that the use of inefficient or poorly managed encryption mechanism accounts for a considerable portion of the security breaches that are noted in crucial sectors such as healthcare, social welfare and finance (Alsharif et al., 2022). So as one of the key findings, the researchers showed that the digital ecosystems must urgently create and enforce strong encryption standards as a major element of their cybersecurity plan.

The case of child welfare reporting system illustrates the significance of encryption in the context of securing sensitive data. Abuse reports are often the ones which give identifiable information about the victims, offenders and witnesses and hence, they become the ones being compromised if the proper protection dint applied. But these vulnerabilities, through the introduction of the encryption measures such as HTTPS, encrypted database and secure APIs, it can be lowered significantly (Yosef, 2025). As a result, the implementation of this measure not only secures the report’s content but also the metadata may designate the reporter’s identity.

Controlling access to sensitive information is being made possible with the help of access control mechanisms alongside encryption. Using role-based access control (RBAC) framework which is a security approach that assigns access rights according to the organizational roles, it becomes possible to minimize the threat of internal breaches (strongdm, 2025). Research conducted by Butt et al. (2024) demonstrates that the system that combine structured access controls with trust mechanisms were the ones to get more user trust and better operational security results. In the same manner, besides the encrypted database and strict access controls, the new system can be dynamic for the creation of a reliable ecosystem to the public and thus encouraging more involvement in protection of a child's. However, even the most secure system should have to go through the process of convincing users about its reliability. Aside from the technical measures, which are only good if users understand them, trust is built not just by the architecture but primarily by declared privacy guarantees.

2.1.2.3 Trust through Privacy Guarantees

While technological tools such as encryption and access control are necessary perse for preventing loss of sensitive data, user's perceptions of privacy are no less important in influencing their desire to use a digital reporting platform. In software applications dealing with sensitive data such as child abuse, where the concepts of fear of exposure, backlash, or distortion of facts prevails, the clientele frequently considers credibility based on privacy disclosure and messaging, not only the computer-based means of security used covertly.

The reason behind this is the users fail to possess the technical knowledge required for the direct evaluation of system security and vulnerabilities. Thus, their only trust left is on visible indicators of privacy assurances such as consent prompts, privacy labels and institutional endorsements to make a platform as secure for use. This belief is explained by a study, who pointed out that digital service present along with parents are accustomed to using privacy-specific visual signs, to estimates the level of safety a particular platform provides. If technical or legal jargon is complicated or hard to grasp, cues such as in-app permission, security badges, and straightforward privacy summaries become more essential. The safety of a system often depends on the presence of obvious cues, which, when absent, could lead to it being displayed as insecure, thus triggering the user's reluctance or complete withdrawal. This statement evidently emphasizes the necessity of both establishing robust privacy protocols and enabling the user to understand them easily and gain access to them easily. A user's feeling of security regarding the control a system confers to them over their information, the openness with which it stipulates its data usage policies, and the commitment frequency with which it adheres to its declared privacy obligations are all factors that influence that sensation of security.

Studies have always confirmed that user engagement and trust are promoted by straightforward and clear disclosures of privacy policies. The mentioned study carried out by Yadav and his co-authors (2024) revealed that 74 percent of the users were more interested in interacting with companies that kept their privacy policies transparent, while 82 percent of users preferred brands that made their data practices visible. Most of the respondents or data suggest that being transparent is not only a legal requirement but also a tangible factor that determines customers decision on using a digital platform. Besides, this desire for openness is way more important in sensitive areas such as child protection, where individuals are asked to disclose very private and often traumatic situations.

It is necessary for the users to receive the guarantee that their data is not only technically safe but also that they are able to understand and manage its use. It is the technical security that stands alone is not sufficient and this fact is proven. The users might feel reluctant or get disconnected from the services regardless of how solid the backend is owing to the lack of clear norms, comfortably accessible user agreements, and consistent privacy notifications. Openness perceived by the clients is a prerequisite for joining in the use of systems set aside for vulnerable groups, like abused children, or to those who represent them. The theory of digital ecosystems with visible and strong privacy guarantees experiencing higher user trust, satisfaction, and loyalty through the analysis results of Saura et al. (2023). These researchers underline the issue of trust, which is not a sudden gain, but a gradual one. This is accomplished mainly, by the immediate response and the detailed explanation of the data flow which also involving user empowerment. For example, for child protection contexts dynamic privacy notifications, or user control over data sharing pathways, and clear compliance indicators that are established by data rules may be applied. Implementing these strategies will help platforms make the shift from passive protection to proactive reassurance, which in turn will increase the public's willingness to report and re-engage.

Besides that, as well as enhancing the credibility of the system, compliance with such famous legal frameworks as the EU's General Data Protection Regulation (GDPR) and Malaysia's Personal Data Protection Act (PDPA) increases user trust and legal credibility. The dominant value placed on user rights concerning information, control of their data, and withdrawal of consent by these frameworks is in addition to data protection (). Users are more often to view their data as safe and with integrity when the system logical like the access to audit trail, the data management dashboard, and the downloadable privacy policies include these principles.

2.1.3 Integration of Real-Time Coordination and Structured Case Management

Besides having a user-friendly layout and strong security features, all system especially a child welfare reporting system must guarantee prompt and organised processing of instances that are reported. Structured case management refers to the planned, controlled, and coordinated of services and interventions in a systematic way that is but through a predetermined design layout to accomplish the desired outcomes for client cases (Giardino & De Jesus, 2023). As real-time coordination and structured case management functions are key elements in the child protection initiatives to operational responsiveness, the delays and non-conformance rates need to be decreased, and the results should be improved. Modern digital systems that are equipped with features such as real-time alerts, ordered tracking of cases, and linked interagency communication could facilitate the quick routing of workflows, prioritizing the urgent cases, and enabling the faster interventions. This section presents a detailed explanation of essential elements necessary for case management organization and real-time coordination through child welfare reporting platforms focusing mainly on notification systems, case tracking and multi-agency integration.

2.1.3.1 Real-Time Notification and Response Systems

Real-time notifications are a very useful digital tool which one can use to inform the concerned parties of any major happenings immediately. In forms of child welfare reporting, this feature guarantees that the case officials such as case workers, law enforcement officials and the like are immediately informed about urgent update that need to act immediately. For example, the system may send out and in-app or email notification to the assigned worder when a new child abuse report is submitted or updated, which ensure that case is acknowledge and progress without any delay.

This type of function is crucial in the system because it can assist in the reduction of the response wheel and the occurrence of procedural bottlenecks. In cases of child abuse, any possible delay in reporting or confirming might increase the chances of a child suffering further injury or traumatic experience. Research revealed that the psychological impacts of childhood abuse can continue into adulthood thus confirming the necessity for intervention at the earliest possible stage (Machpheerson et al., 2021). The research results underscore the importance of simultaneous identification of abuse and the initiation of appropriate case management at the time of receiving the reports.

In addition to the physical risk, a delay in the response can also increase mental health problems. Zaydlin and Laras (2021) discovered that the insufficiently fast psychiatric measures taken for sexually abused children caused them to suffer trauma longer and that it was more likely for them to develop psychological distress later. Thus, real-time notifications which are part of child welfare platform can act as connectivity bridge between report intake and actual intervention. Furthermore, the use of technology for alerting in other safety issues has evidence of performance success. Spoorthi et al. (2023) presented a child monitoring system that detects real-time threats that parents might face. Even though the system was intended for domestic use, is an example of how the use of timely digital notifications can improve the protective action. The basic idea behind this where awareness of the situation leads to an appropriate reaction could also be valid for the use of similar system by caseworkers and agencies for the treatment of abuse.

A study identified that in the reflection on AMBER Alert systems most child kidnappings were in safety eventually but only 26.7% of the recoveries (Griffin et al., 2023). These recoveries, however, were proportionally related to the alert. The authors refer to this as the system's design that mostly includes abductions seen as low risk, like those with family members and the absence of consistent impact in the truly high-risk cases. As per the findings, the real-time notifications can aid in enhancing the performance of the adherence. Still, the actual level of their application and the involvement in the common response procedures will absolutely decide their efficiency. In institutional child welfare systems, this emphasizes the necessity of mixing real-time alerts with organized processes, escalation protocols, and case tracking to provide systematic and timely interventions which was also discussed ealeir in real-time systems.

2.1.3.2 Interagency Coordination and Workflow Integration

According to a study, disjointed system and lack of real-time data sharing are some of the causes that disrupt coordinated agency responses (UNICEF, 2020). The problem underscores the fact that responsiveness is an important measure of the efficiency of digital reporting system. Just a few delays in the process of handing over coordinate reports or inter-agency assignment can result in missed chances for the protection of a victim or extend the damage more to the case itself. If the official users such as the police, child welfare officers, healthcare providers do not coordinate their information properly, instances may then fall into situation of being captured by bureaucratic cycles or inadvertently be repeated. This would lead to a consequence where action is delayed even when it is needed the most.

Additionally, case routing system which allocate high-risk reports and facilitate the communication, can not only reduce the waiting time for the affected families but also the emotional distress that comes along with them. As child welfare involve many different actors in various departments, enabled by synchronized data flows and collaborative workflows, efficient coordination will be important between them. The findings show that in some areas such has South Asia, it is often the disconnected systems that lead to delay in intervention, while digital case management tools, in conjunction with the coordination and response times have been a positive indicator (UNICEF, 2020).. The instruments are excellent for sharing the agency's actual updates, tracking case processing across different sections, and making sure no report is missed because of the administrative fragmentation. Hence, the use of online integrated software is seen more often as a strategic necessity in the areas suffering from lack of resources and having loads of cases.

Another important factor that determines the efficacy of a system is interagency cooperation. Since many parties are involved in child protection initiatives, including social workers, medical professionals, educators, police enforcement, and legal authorities, digital solutions must be made to allow for organised yet secure departmental information exchange. Communication that is fragmented or compartmentalised increases the likelihood of contradictory replies, missing interventions, or duplicate activities. According to Kiedinger (2024), institutional frameworks that encourage interoperability and cross-sector collaboration are necessary to overcome fragmented welfare coordination, particularly in complex and resource-constrained social care situations. In addition to increasing response consistency, efficient interagency communication makes ensuring that no susceptible case is missed during institution-to-institution handovers.

2.1.3.3 Structured Case Tracking and Workflow Automation

A digital child welfare reporting system long-term efficacy ultimately depends on how successfully it tracks, manages and monitors cases over time, even while real-time alerts guarantee that incoming complaints receive fast attention, and interagency collaboration allows for joint accountability. Operational continuity is based on structured case tracking. It guarantees that a report is actively moved through a defined series of processes as soon as it enters the system, without needless delays, omissions, or duplication. This becomes especially crucial in settings when several organisations, including social services, law enforcement, and healthcare professionals, are working together. In the absence of structured tracking, the risk of fragmented responses increases. Cases may slip through the gaps, crucial information may be overlooked, and follows-ups may be postponed especially for high-risk reports that need constant and urgent attention. Redundant case entries, miscommunication across departments, and bottlenecks in approval or referrals are common issues that often stem from ineffective workflow monitoring. To overcome these issues, digital workflow automation introduces a systematic logic to the processing, prioritized and escalation of reports.

Automation capabilities in modern systems can be designed to dynamically monitor the progress of each case, ensuring that timelines are maintained, and no critical step is left pending (Shetty, 2025). These findings highlight the necessity of systems that, beyond simply logging reports, can also escalate unresolved cases automatically to ensure timely monitoring and intervention. The system may, for instance, automatically remind assigned caseworkers when tasks are due, alert supervisor if case not are not updated within the time and such. In settings with limited resources when manual oversight is difficult or unsustainable, such mechanisms can be particularly effective. Case routing intelligence, which is one of the application forms, means that a case may go to any of the available workers based on different factors such as location, case type, staff workload or language preference. That is why it is important that the reports are assigned to the person with the right power to act. In fact, handling the report in such a way positively affects the period when misallocation of duties can take place.

The efficient delegation of tasks and the lessening of the backlog make these mechanisms the best choice. The literature has provided sufficient evidence for the capability of these mechanisms. Munnangi (2024), for example, reviewed automated case management systems in public legal services and discovered that workflow routing, automatic alerts, and unresolved case escalation are the main features that would significantly increase response times and system transparency. The report revealed a 38% decrease in case processing time and a 73% decrease in human intervention by using artificial intelligence tools for decision-making, which reflects the contribution of automation in fast-paced service situations. This kind of automation is not only helpful with delay issues but also imposes procedural uniformity.

In the same vein, another study mention that the workflow automation based on AI not only help increases the speed of task completion but also reduces the cost of operations and increases accountability in sensitive areas, including legal and welfare services. The recent advances in workflow-alignment show an extraordinary gain in the management of legal and welfare cases. Figure 4 illustrates that the components such as the Workflow Automation Engine and Integration API Endpoints had success rates of 99.5% and 98.5% respectively, along with similar high accuracy levels (Munnangi, 2024) Through these statistics, it is clear that the follow-up with the process orchestration is indeed beneficial in order to improve the system value, reduce the human-initiated delays and maintain the procedure consistency. The incorporation of mechanisms like these has been notably advantageous in areas where the measurement of time taken for responses and the adherence to responsibilities are the utmost important aspects and these are the ones being frequently talked about in digital child protection structures.

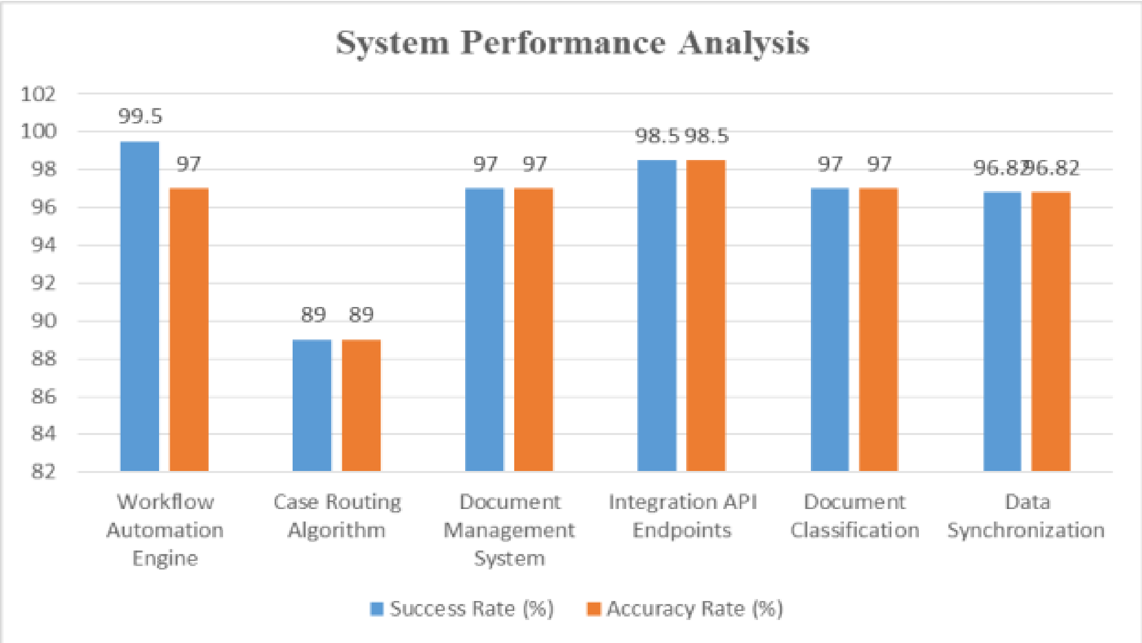


Figure 7: Performance Metrics of Core Architecture Components in Leal Case Management (Munnangi, 2024)

Tracking systematic activities is also an external advantage of providing transparency and trust. The situation of a case that one obtains through a system is advantageous to all sides engaged, including professionals dealing with this case and the reporting parties such as victims, family members, or the public. With the appropriate access control, end users can contribute their additional evidence or explanation through guided prompts, get reports on actions taken place, and also, find out how their reports are taking shape. Once the detected trends for intervention become pointing to inefficiencies, the administrators could make rapid decisions since they are supported with the necessary for visualization as well as the appropriate overview of cases that are pending, delayed, or closed.

Proper workflows contribute to agencies recognizing risk ownership and improving their auditability trusted by many as the long-term accountability of child welfare systems is only achievable through maintaining such practices, as founded by Wu et al. (2022). Their findings support the point that, the multi-agency settings of continuous planning streamline the process of making roles and expectations explicit. This leads to a collective, transparent, and proactive approach to protecting children, resulting in fewer obstacles caused by jurisdictional uncertainty or compartmentalized action. Research collocated by Ball et al. (2024) and Driscoll et al. (2022) affirms the usefulness of structured case tracking in improving information flow, duplication reduction, and interagency responsiveness augmenting. Their investigation displays that the adoption of digital workflows during the COVID-19 pandemic increased the speed of case completion, cut down on the duplicate of work, and established an improved coordination of communication. They argue that a structured digital roadmap decreases the service fragmentation, and it provides the opportunity for organizations to work together by means of sharing the visualization of the case progress.

With the integration of automatic processes and systematic case monitoring over time, a digital child welfare platform able to transform from a passive archive to an active, intelligent system that enables quick, well-coordinated, and reliable interventions. It not only offers trustworthy insights to the managers and policymakers but also brings the frontline operations into line with the standard procedures. This systematic methodology guarantees that reporting is not just a submission procedure but the start of an outcome-oriented, monitored, and responsive protection pathway as the complexity and volume of instances increase.

2.2 Similar Systems

2.2.1 Primero/The Child Protection Information Management System Plus (CPIMS+)

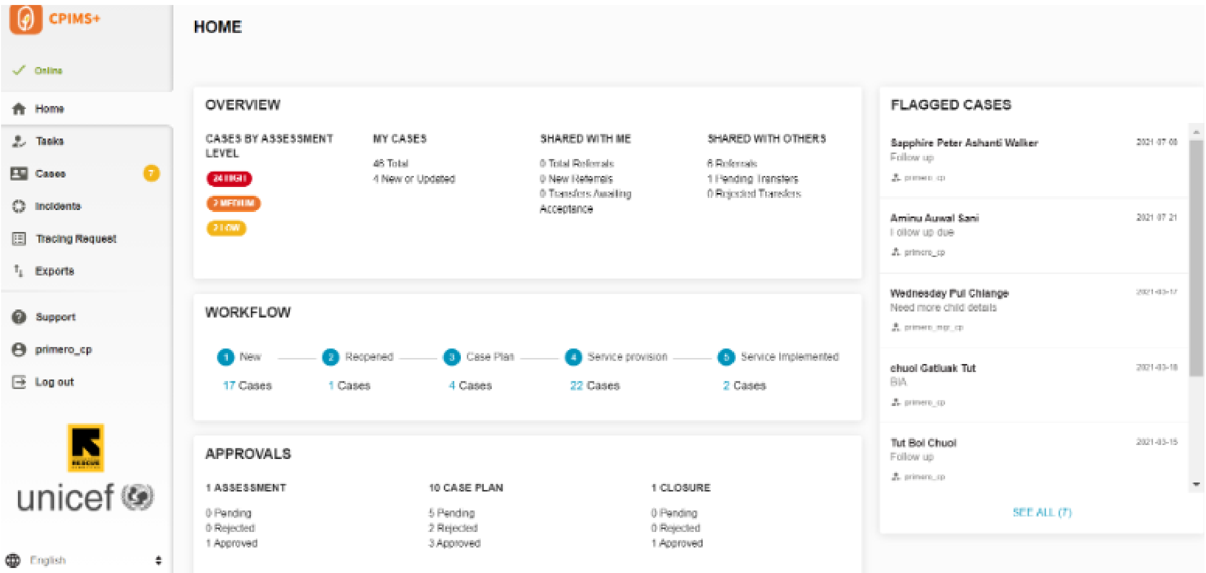


Figure 8: Dashboard of CPIMS+

CPIMS+ is a specialized digital case management solution developed under the Primero platform by UNICEF, in collaboration with a coalition of global child protection stakeholders.

It helps provides a highly ordered, safe, and flexible digital environment that greatly enhances the coordination, responsibility, and efficacy of child protection services as a committed instrument for handling delicate child protection situations. National governments, NGOs, and humanitarian organizations as well as long-term social service projects employ it in both crisis and development environments (CPIMS+, 2024).

Mainly, the CPIMS+ application allows the first-line child protection staff to record, manage and track electronically the Cases all along the protection process. The technological tool provides adaptable digital processes that are in line with the practices of different countries or agencies. The platform provides standardizing and traceability in the treatment of all individual cases from first intake, registration, and evaluation to case planning, service delivery, follow-up, and ultimate closure (CPIMS+, 2024). This platform not only promotes consistency but also supports the application of best practices and the compliance of legal or organizational norms.

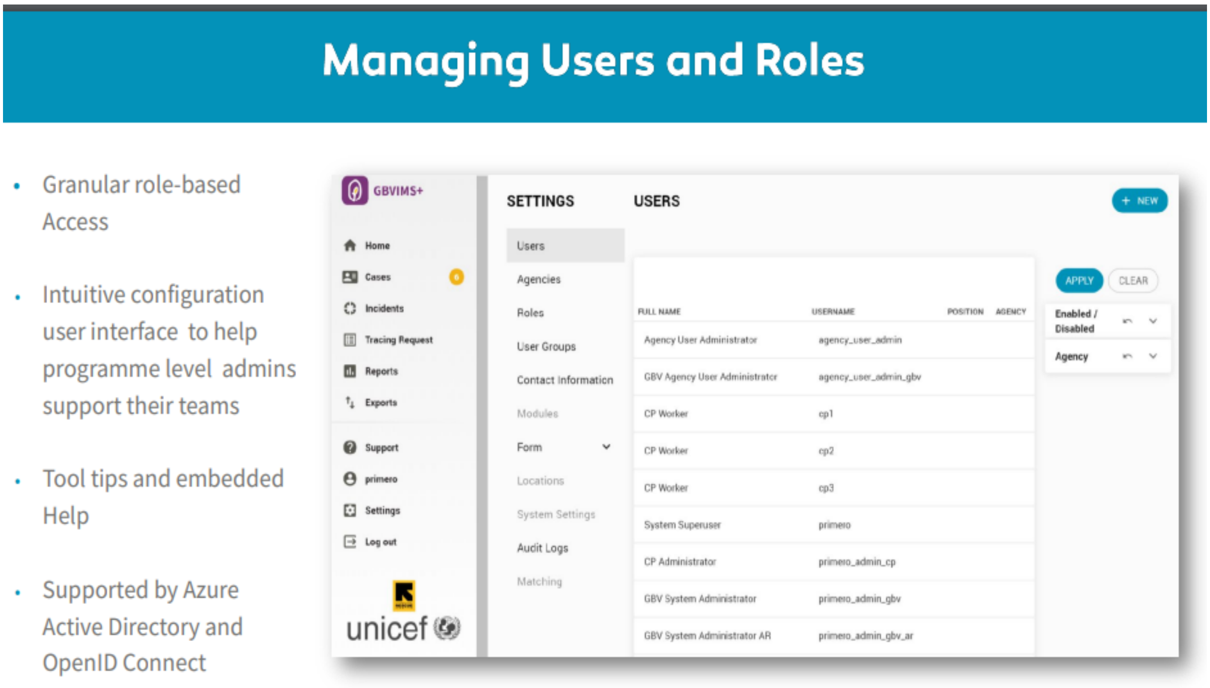


Figure 9: User and Role Management Interface in CPIMS+

CPIMS+ is highlighted for its robust security infrastructure that ensures the confidentiality and privacy of personal kid information. It is built on the Role-Based Access Control (RBAC) model, which controls the viewing and editing access depending on the user roles such as caseworker, supervisor, administrator. Thus, it is guaranteed that only the authorized staff members can view private information, and the risk of illegal data disclosure is reduced. CPIMS+ provides a digital environment that is secure and in compliance with global data safety standards like GDPR. This is achieved through end-to-end encryption, secure login authentication, and audit logging features (Primero, 2024).

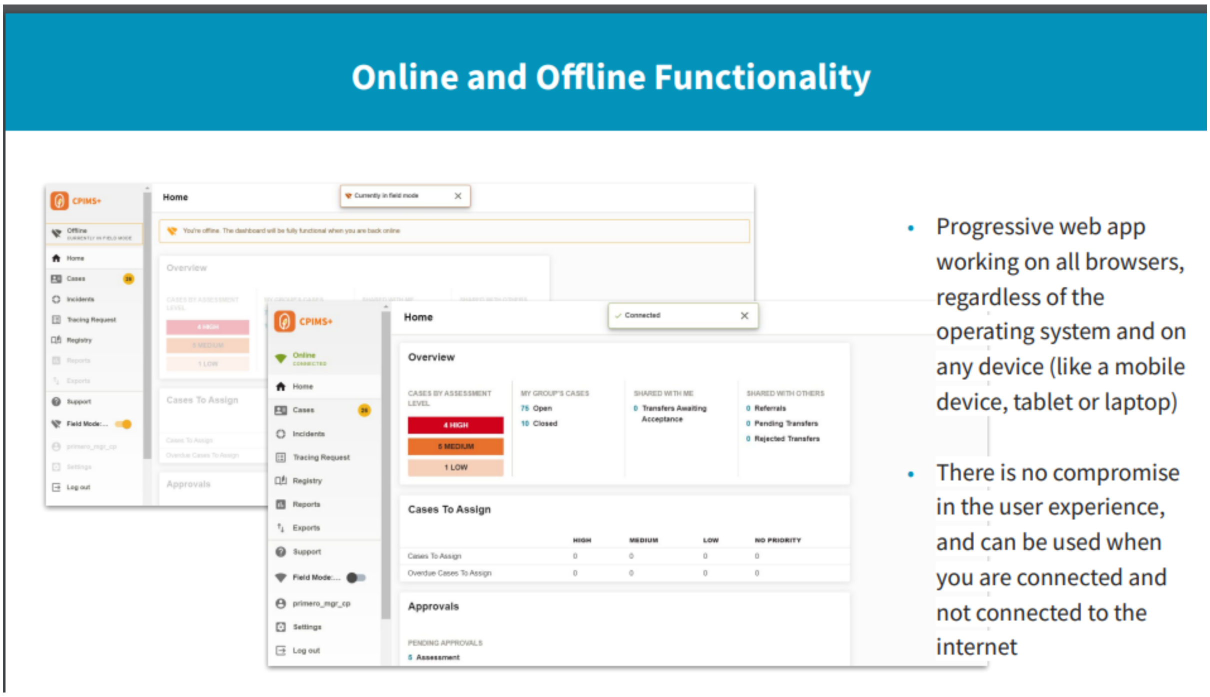


Figure 10: Online and Offline Functionality

This system also comes with the offline application which is the reason why it is primarily fit for demanding field environments (Primero, 2024). Being a PWA (Progressive Web Application) that works without the internet, CPIMS+ assists caseworkers in their duties. Data offline written would be safely stored on the device and would be automatically synchronized with the main database as soon as the internet connection is made available again. This ensures that the same method is used in invulnerable and difficult conditions where restricted technical infrastructure is found. In that way, the data collecting and handling such places are consistent.

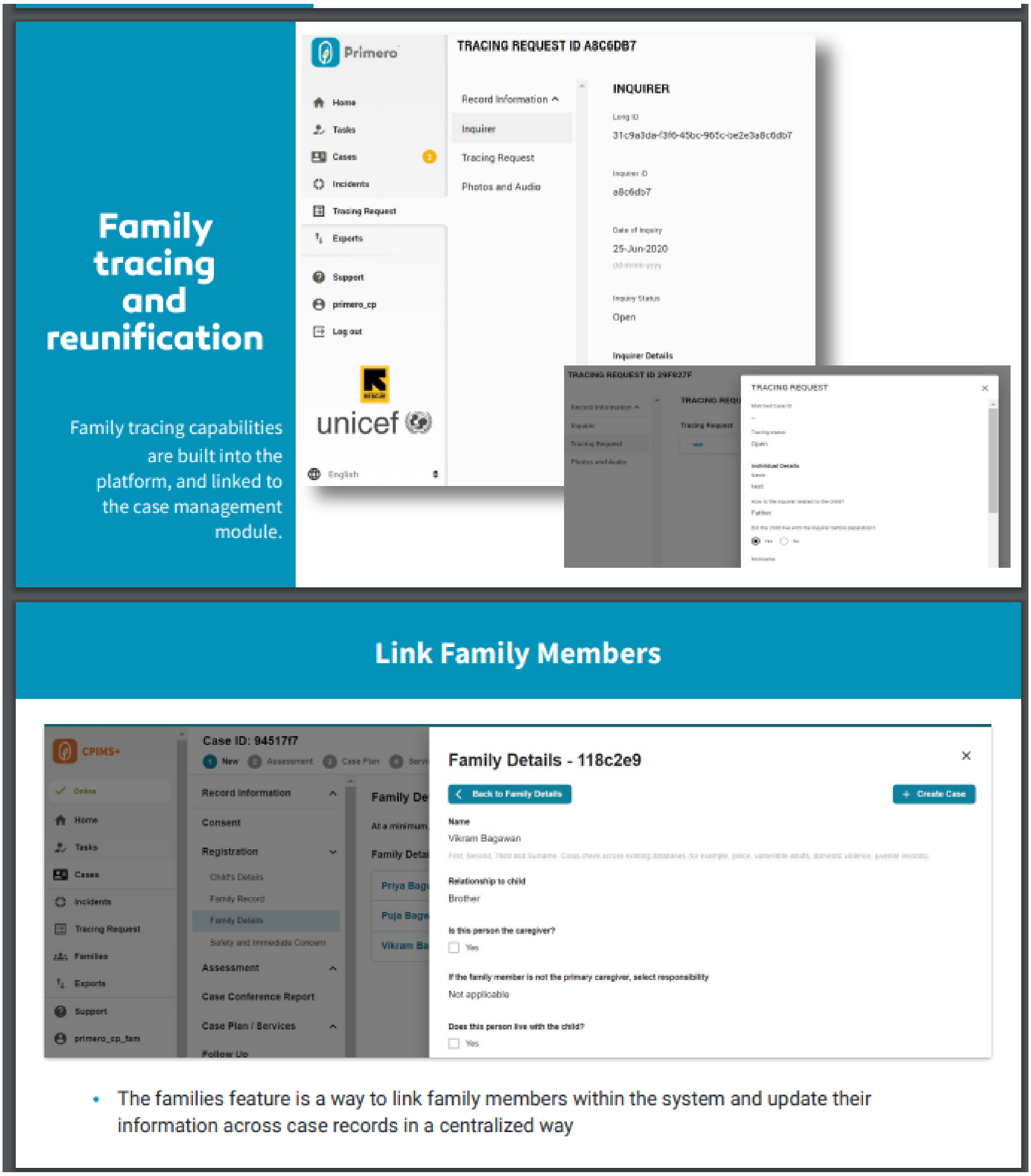


Figure 11: Family Tracing and Reunification Features AND ALSO Link Family Members Feature

This platform consists of a Family Tracing and Reunification module primarily aimed at disasters such as the conflicts arms, natural disasters, and significant displacements (Primero, 2024). This module proposes an innovative solution that uses matching algorithms whereby kid data and tracking requests from caregivers are compared to reconstructs the network. The algorithm, by calculation of the chances, helps the caseworkers in finding possible solutions, hence the reunification process is accomplished quickly and efficiently while not sacrificing security and accuracy.

CPIMS+ intertwines interoperability in its building process to elevate the cross-organizational collaboration. The product is fast to connect to different systems in the digital universe via shared APIs, that includes government databases, health systems, as well as NGO case management applications. This not only helps to continue care through the transfer of cases outside the organizational lines but also manages safe referrals between agencies and limits duplicates in data. Namely, CPIMS+ works with OSCaR, which is an NGO-based system in Cambodia to allow automatic referral of data between the two platforms (Primero, 2024). The system in Sudan constitutes of ten states and more than twenty partner organizations that together tackle close to 35,000 cases of vulnerable children (UNICEF, 2022). Built-in capabilities for data reporting, analysis, and visualization also abound in CPIMS+ These tools let users track key performance indicators, create dashboards for operational control, generate case-level and system-level reports, and track trends in child abuse or neglect. This real-time information helps decision-makers to effectively distribute resources, assess the success of interventions, and promote evidence-based policy development (Primero, 2024).

Strengths

- **Role-Based Access Control:** Enforces strict data privacy based on user roles (caseworkers, supervisor, admin).
- **Offline Functionality:** Progressive Web App (PWA) lets users record data in low-connectivity settings and auto-sync later.
- **Family Tracing Module:** Uses matching algorithms to reunite families in crisis scenarios.
- **Interoperability:** Integrates with other Platforms through APIs (OSCaR, health systems).
- **Built-in Reporting Tools:** Includes case tracking dashboards and data visualization for policy analysis.

Weaknesses

- **Complexity of Deployment:** May require technical expertise and significant setup for local adaption
- **Global Orientation:** Not tailored specifically to Malaysia's reporting ecosystem.

2.2.2 CaseBook

Figure 12: Casebook (CaseBook, 2024)

Developed by the Annie E. Casey Foundation, Casebook is a cloud-based case management platform that supports a variety of social service initiatives, including public safety, victim assistance, and child welfare (Casebook, 2024). Agencies may maintain case records, improve overall case management across departments and stakeholders, and expedite communication using its customisable features.

Casebook's cloud-first, web-based architecture guarantees accessible across platforms including desktop computers, tablets, and smartphones, hence defining some of its fundamental features. Even when they are remote, this lets field officers, law enforcement, and social workers securely access and update case data in real time (CaseBook, 2024). Hence, this helps to increase operational flexibility and removes the necessity for local server infrastructure. Besides that, Casebook uses Role-Based Access Control (RBAC), in which case data view or modification system rights are assigned depending on user roles like caseworker, supervisor, or administrator (Casebook, 2024). Important for preserving confidentiality in child protection operations, this function not only safeguards sensitive information about abuse of children but also guarantees responsibility and reduces the possibility of illegal access.

Its flexible workflows and forms which enable agencies create their own intake systems, evaluation forms, and follow-up templates is also one of its noteworthy characteristics (Casebook, 2024). This customisability guarantees the system conforms with local rules and organisational norms and allows fast adaption to policy changes or procedural improvements without requiring software rebuilding. This system also offers improved case notes capability where users can electronically document thorough stories, observations, or case-based choices taken (Casebook, 2024). Crucially for legal responsibility and interagency collaboration, this substitutes manual documentation, lessens administrative burden, and guarantees a safe, traceable record of case history.

Furthermore, Casebook includes real-time reporting tools and integrated dashboards that let agencies monitor performance metrics, service delivery data, and case load trends (Casebook, 2024). These awareness helps to promote data-driven decision-making, assist in the identification of service shortages, and let managers more wisely distribute resources according on real case patterns. Caseworkers may link clients with appropriate services such shelters, healthcare, legal assistance, or counselling using the built-in service directory and referral tracking module. It guarantees that support services are really followed through and not only documented, therefore increasing the general well-being of impacted children and families by tracking the delivery and status of recommendations.

Every system activity including data modifications, access attempts, or role changes is recorded in Casebook's thorough audit logs, therefore ensuring transparency and compliance. This guarantees traceability of all system actions, therefore supporting internal audits, legal documentation, and regulatory compliance. Finally, Casebook is built for flawless open API interface with other systems (Casebook, 2024). This encourages continuity of treatment and effective multi-agency cooperation by letting agencies trade data with government databases, healthcare systems, or NGO tools without repeating entries.

Strength

- **Cloud-Based Accessibility:** Works on desktop, tablet, and mobile for remote or field access.
- **Customizable Workflows:** Agencies can design their own intake, evaluation, and reporting templates.
- **Integrated Dashboards:** Real-time case updates, performance metrics and visualization tools.

- **Audit Trails:** Every action is logged for compliance and accountability.
- **Service Referral Tracking:** Ensures follow-up on services like healthcare, legal aid and shelters

Weakness

- **Broad Scope:** Designed for general social services
- **Connectivity Reliance:** Need consistent internet access to only be fully functional

2.2.3 Enhanced Child Abuse Information Management System (ECAIMS)

The Enhanced Child Abuse Information Management System (ECAIMS) is a web-based software system developed in Nigeria to provide a digital framework for reporting and documenting incidents of child abuse. It was introduced to solve the challenges of inefficient and disorganized manual processes in child protection, where paper-based documentation often resulted in lost case files, delays in investigations, and a lack of coordinated follow-up (Odikwa et al., 2022). Designed with the Nigerian Police Force, Human Rights Commission, and members of the public in mind, the system facilitates a more effective response to child abuse reports through digital channels.

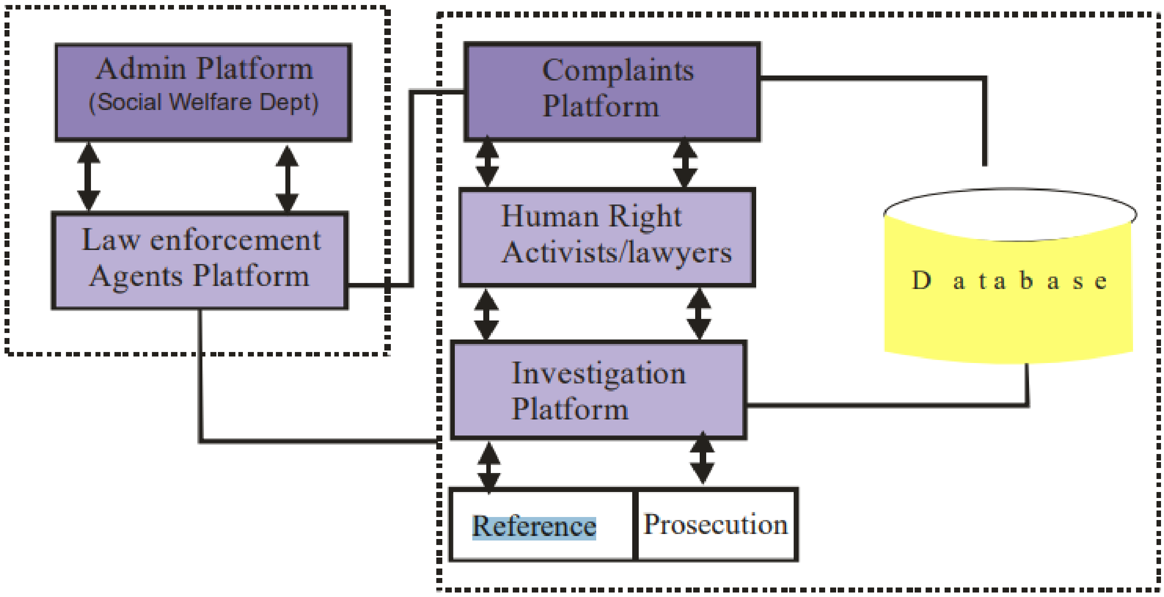


Figure 13: Architecture of Child Abuse Information System (Odikwa et al., 2022)

The system is structured into 7 major modules which is the Admin Platform, Complaint Platform, Law Enforcement Interface, Human Rights Platform, Investigation Platform, Reference Platform and Prosecution Platform. The Complaint Platform enables victims or observers to report incidents online, where they are stored securely in a centralized database. This database stores essential information such as the name of the alleged abuser, type of abuse, time, location, and any supporting details provided by the reporter (Odikwa et al., 2022). The design aims to prevent information loss and eliminate duplication common in paper-based systems.

The Complaints Platform is designed to give the opportunity to the reporters to file online abuse reports without registering. The reports contain the type of abuse, number of times it occurred, place, and name of the accused, and are stored in a centralized MySQL database. The Admin Platform oversees the user account management and also updates the system data. After submission, these complaints become visible to the police officers who can check if the claims are true and enter their investigation results by means of the Law Enforcement Platform.

Similarly, the Human Rights Platform allows lawyers and activists to view these cases and assist with follow-ups, legal interventions, or referrals (Odikwa et al., 2022) This multistakeholder access increases transparency and facilitates quicker action toward justice. Through the investigation platform, case updates and summaries of concluded cases are publicly posted, promoting transparency and community awareness In addition to managing current cases, the refence platform serves as a legal archive which helps the practitioners learn from past case outcomes and penalties (Odikwa et al., 2022). Finally, the prosecution platform lists offenders and relevant legal actions taken which the forms a complete legal trail.

One of the system most distinguishing features is its commitment to public transparency. Unlike old system that keep case data internal, ECAIMS provides visibility of resolved cases and actions to educate the public and deter abuse However, ECAIMS does have limitations. The system lacks real-time status tracking for reporters, meaning users cannot monitor the progress of their submitted cases. It also does not offer built-in communication tools for interagency collaboration between law enforcement, legal advocates, and social workers. Features such as mobile accessibility, offline functionality, and automated notifications are also absent, reducing the system's adaptability in remote or low-infrastructure settings (Odikwa et al., 2022). While access to the system is login-restricted by user roles, no granular role-based permission settings or user audit logs are documents. Additionally, there is no analytical dashboard or reporting module to support data-driven decision-making or policy review.

ECAIMS is, thus, a significant move towards opening up child welfare systems to increased transparency especially in low-resource settings. However, it is low interactivity, lack of collaborative support, and no advanced security or monitoring features make a way for further improvement.

Strengths

- **Multi-Agency Access**: Interface for police, human rights groups and public promoting transparency.
- **Centralized Complaint Platform**: Allow public to report cases securely online.
- **Legal Reference Archive**: Public access to concluded cases increases awareness and deterrence.

Weakness

- **No Real-Time Case Tracking:** Users can't follow up on case progress after submission.
- **No built-in messaging tools:** No direct communication between users.

2.2.4 Comparison of Similar System

Table 3: Comparison of Similar System (Casebook, 2024; CPIMS+, 2024; Odikwa et al., 2022).

Feature/ Criteria	CPIMS+	Casebook	ECAIMS	Integrated Child Welfare System (SinDA)
Anonymous Reporting	Not supported, public users must register.	Not available, identity required for all access.	Anonymous reporting supported, no registration required.	Supports anonymous and registered reporting.
Role-Based Access Control (RBAC)	Stric access based on user roles.	Highly customizable permission settings.	Basic modular access with limited control.	Basic role-based access.
Real-Time Case Tracking	Dashboard supports real-time updates.	Timeline tracking and real-time dashboard.	No live tracking after report submission.	Live tracking after report submission.
Secure Interagency Messaging.	Supported via system integration (no direct chat).	Tagging and note-based internal collaboration.	Not available in system design.	In system chat feature between involved users.
Audit Trails and Logs.	Tracks changes and user activity logs.	Comprehensive audit logs per user session.	No logging or version control features.	Basic logging for key user actions such as logins and

				case updates with timestamps.
Localization to Malaysian Context	Global focus, not tailored for Malaysia.	Designed for Western institutional use.	Built for African government context.	Fully localized to Malaysian legal and agency structure.

The above table shows the comparison between the similar system including the proposed system SinDA against six core features which are relevant to digital child protection solutions. Anonymous Reporting is one of the functions to encourage public participation in abuse reporting without fear of exposure. Only ECAIMS and Sinda support this feature while CPIMS+ and Casebook restrict submission to only registered users which may discourage broader public engagement. Besides that, Role-Based Access Control (RBAC), CPIMS+ and Casebook offer structures user access, where Casebook supports customizable permission. ECAIMS employs login-based access but lacks detail's role control while as SinDA will offers basic administrative vs general user distinctions.

As in terms of Real-Time Case Tracing, CPIMS+ and Casebook have provided bot dashboards and timelines for case monitoring which enables authorized user to monitor them cases progress. In contrast, ECAIMS does not offer any form of live status updates to reporters after a complaint is submitted. SinDA, however is designed to allow registered users who are directly involved in the case to track the progress of their submitted reports in real time, thereby improving user trust and system responsiveness. Secure Interagency Messaging is not offered by any of the existing systems in a real-time chat format. Casebook uses tagging and case notes, CPIMS+ integrates with external system but lacks direct communication tools, and then ECAIMS lacks such functionality entirely. SinDa introduces in-system chat to bridge this gap which then improve coordination. Finally for localization, all the similar system were designed for cases outside Malaysia whereby SinDA is fully tailored for local needs. The unique position of SinDa stands out in terms of fulfilling local legal requirements, public accessibility, and interagency coordination gaps that are observed in current international system.

2.2.5 Conclusion

The examination of CPIMS+, Casebook, and ECAIMS reveals notable structural and functional variations in the global implementation of digital child welfare systems. Strong internal controls, organised dashboards, and comprehensive user permissions catered to organisational users are elements that CPIMS+ and Casebook provide, however they do not permit anonymity or public access. On the other hand, by allowing unregistered users to submit reports and view resolved case outcomes, ECAIMS encourages transparency that is visible to the public. As for the professional users, it is not provided live collaboration, status tracking, or audit trail features. The systems offer these options but still none of them are completely balanced to meet the interests of the parties involved regarding technological, legal, and social issues. Nonetheless, all of them showcase the strengths that correspond to their regional goals.

These systems consistently show a gap in terms of real-time interagency communication and a lack of specific, role-based audit logs. On top of that, the platforms analysed never incorporated Malaysia's legal, cultural, and procedural environment into their design. This was evident in the local context where the platforms could be only used in conjunction with government officers, law enforcement, medical professionals, and NGOs participation. Besides, public reporters tend to the severe shortage of available case updates, particularly in the systems where the anonymous report is accepted but no feedback loop is included.

The results are valid justifications for the configuration of the Integrated Child Welfare System (SinDA) presented, which combines anonymous public reporting, real-time progress tracking for authorized users, interagency messaging, and basic audit trails which all these features are included in a localized Malaysian framework. The main goal of SinDA is to provide a proper mix of accessibility, transparency, and secure collaboration by making it a practical, context-aware improvement over the existing global systems.

2.3 Technical Research

2.3.1 System Requirement Analysis

To build and deploy the proposed system locally, the developer's system uses the following hardware and software resources:

Physical Hardware Requirements

Components	Specification
Processor	Intel Core i3 or higher
Memory (RAM)	Minimum 4GB
Storage	At least 10GB free disk space

Software Requirements

Components	Specification
Operating System	Windows 11
Web Browser	Brave/ Edge / Google Chrome
IDE	Visual Studio Code
Web Server	Apache via XAMPP

2.3.2 Programming Language Chosen

Figure 14: Laravel

The main programming language for developing the Integrated proposed system is PHP, implemented through the Laravel 12.x framework. PHP is a general-purpose, open-source scripting language especially designed for web development and has been a foundational technology for countless large-scale and secure web applications. Laravel, built on PHP, is a modern web application framework that simplifies complex backend operations such as authentication, routing, session handling, and database management.

In addition to PHP and Laravel, the proposed system also incorporates basic web development language which is HTML, CSS and JavaScript to manage the front-end components ese technologies are used to structure the web pages, style the user interface, and add interactivity and dynamic behaviour to key modules such as form validation, dashboard features, and user interactions. While PHP and Laravel power the backend logic and server-side functionality, these client-side languages ensure a responsive, accessible, and user-friendly interface for all system users.

Justification on Using HTML, CSS and JavaScript for Front End

Figure 15: HTML5 and CSS3

The core technologies for creating adaptable and accessible web interfaces, HTML5, CSS3, and JavaScript, are used in the development of the frontend of the planned Integrated Child Welfare System (SinDA). HTML or Hypertext Markup Language is the common language used to structure web content on the internet It defined the meaning and structure of web content and enables the developers to create meaningful page layouts through elements like headers, forms, navigation bar and more (w3schools, n.d.). The latest version HTML5, enhances clarity and accessibility by introducing new structural tags which can improve screen reader compatibility and assist ser with visual or cognitive impairments. For the implementation of SinDA, HTML5 will be used to layout system content clearly and ensure that critical section like the abuse reporting forms and user dashboards are presented in a n organized manner especially for users with limited technical literacy.

CSS or Cascading Style Sheets is a stylesheet language that used to describe the presentation of a document written in HTML (mdn web docs, n.d.). CSS3, the current standard, has introduced a sophisticated layout tool that allows the content to adjust dynamically according to the screen size. This guarantees that the system is scalable and works properly on any platform, which is a must-have for this proposed system. Besides that, the strong contrast colour schemes and the easily readable typography are the main instruments of assistance for panic users, and they are the great addition to the usability of the interface. These design considerations align with established web accessibility best practices and contribute to a more inclusive and user-friendly experience.

A high-level programming language called JavaScript makes it possible for web applications to include dynamic, interactive features. Real-time input validation, content changes without page reloads, and customised interactions are all made possible by this browser-based application (mdn web docs, n.d.). For example, without requiring server connectivity, JavaScript can dynamically modify the interface to display pertinent form fields or dashboard data when a user picks a role. This lessens the cognitive load during crucial reporting activities by improving responsiveness and usability. Future improvements like lightweight charting components for real-time dashboard analytics or AJAX-based updates are also made possible by JavaScript (w3schools, n.d.). A smooth experience catered to the emotional and functional requirements of different user types is produced by its interaction with HTML and CSS. Hence, is essential to maintaining the frontend's responsiveness, user-friendliness, and ability to provide a productive and encouraging user experience in a variety of situations.

The choice to employ these reliable, established technologies guarantees wide compatibility, effective operation, and simplicity of maintenance. Although frameworks like Vue.js and React provide strong component-based structures, they add dependencies and complexity that are superfluous for SinDA's current purpose. A lightweight, scalable foundation that meets project objectives while preserving accessibility, clarity, and interaction across user roles and devices is provided by the usage of HTML, CSS, and JavaScript.

Justification on Using PHP and Laravel Framework for Back End

Over three-quarters of all online applications are powered by PHP (Hypertext Preprocessor), making it one of the most widely used server-side scripting languages (W3Techs, 2025). It is ideal for dynamic, database-driven systems like proposed system, which requires secure form submission, user registration, role-based dashboards, and case data handling. PHP integrates smoothly with HTML, CSS, and JavaScript, and supports relational databases like MySQL, making it suitable for full-stack development. Being platform-independent, it runs on major operating systems such as Windows, Linux, and macOS (InfoStride, 2022). PHP's built-in modules, reusable libraries, and rich documentation help reduce development time and complexity. Additionally, it offers faster load times in low-bandwidth environments, a critical factor for emergency case reporting in underserved areas (GeeksforGeeks, 2022). These features make PHP a strategic and efficient choice for the proposed system.

Then, the justification for using Laravel is that it is an innovative free PHP framework that applies the Model-View-Controller (MVC) design pattern, which encourages a distinct division between the data, presentation, and application logic levels (Ibrahim, 2024). Advanced features like task batching improvements, native support for JSON: API resources, quicker route caching, and enhanced security settings are all included in the recently released Laravel 12, which is compatible with PHP 8.2+ (Laravel, n.d.).

Furthermore, Laravel simplifies and automates many routine tasks, including email verification, password reset, and user role management. Its built-in features like Eloquent ORM, middleware, route groups, and Blade templating engine help to maintain a clean, structured, and secure codebase (Laravel, n.d.). Laravel 12 is especially well-suited to create a system like SinDA, which handles extremely sensitive child abuse case data and requires robust security and data integrity. It has built-in support for CSRF prevention, authentication scaffolding, and encrypted communications (Laravel, n.d.). Additionally, the proposed system may be expanded into a contemporary, scalable, and API-ready platform using Laravel's vast ecosystem resources, including Laravel Breeze, Jetstream, Horizon, and Sanctum.

Advantages and Disadvantages

The advantage of using PHP with Laravel is the rapid development speed due to Laravel’s scaffolding, templating, and prebuilt solutions for everyday tasks like login, registration, and form validation (Barvaliya, 2024). Then, Laravel also provides powerful database tools, including migration systems and rollback functionality, which ensure that the database schema is version-controlled and easily manageable (TiDB Team, 2024). These features are particularly beneficial in systems like the proposed system, where changes are made in data structure.

As for the disadvantages, PHP is loosely typed, meaning the variables do not require explicit declaration of type (GeeksforGeeks, 2024). This can lead to logic errors or inconsistent data handling if the input is not validated correctly. While PHP 8.2 improves support for stricter types, developers must follow good practices. As for Laravel, it can introduce performance overhead if excessive packages, middleware, or services are used unnecessarily (Rafael, 2023). For instance, using multiple third-party analytics or logging packages such as Sentry, Telescope, and Debugbar simultaneously can add unnecessary background processing, which increases server load and slows down application performance during peak usage. Its abstraction and automation may also limit control over the core behaviour of the application unless developers fully understand the underlying architecture. Furthermore, not all Laravel packages are standardized, leading to inconsistent performance and behaviour across different modules.

Comparison with other programming languages

Table 4: Comparison of other programming language

Criteria	Laravel (PHP)	Spring (Java)	Django (Python)
Architecture Pattern	Follows Model-View-Controller (MVC), separating	Also follows MVC. supports robust layering and	Uses MTV (Model-Template-View) pattern which is

	logic, UI, and data for organized development (Hanif, 2023).	dependency injection via Spring Boot.	Django’s variation of MVC that separates business logic, data handling, and presentation cleanly (Yong, 2021).
Development Speed	Fast due to built-in tools and clean routing simplify rapid prototyping (Barvaliya, 2024).	Slower due to its verbose Java syntax and complex configuration but offers powerful flexibility (Yadav R. , 2022).	High due to its simplicity and its scaffolding which make its ideal for rapid development (Barvaliya, 2024).
Security Features	Good due to build in CSRF protection, hashed passwords, encrypted storage and authentication scaffolding (Laravel, n.d.).	Enterprise-grade security. It supports OAuth2, JWT, CSRF, and HTTPS by default; preferred in regulated sectors (Alex et al., n.d.).	Good. Offers security middleware for CSRF, SQLi, XSS, plus strong password hashing and session control (Khan, 2025).
Performance	Moderate. Laravel caching improves response time in low-resources servers (Ali, 2024).	High due to Java multithreading, JVM optimization, and low-level memory management (Sharmal, 2023).	Moderate. Python’s interpreted nature can be slower under high concurrency without optimization.
Community	Strong due to active PHP/Laravel forums, GitHub repositories and Stack Overflow presence.	Excellent due global enterprise community with tons of Spring Boot integrations, plugins,	Strong. Have open-source community with thousands of reusable packages and Python libraries.

		and corporate support.	

Based on the table of comparison between languages above, the developer will choose Laravel which is PHP as it emerges as the most suitable framework for developing the proposed system. First, Laravel offers a clear division between logic, user interface, and data flow by adhering to the Model-View-Controller (MVC) design. A modular system like the system which might include case tracking, user roles, and secure form submissions, would benefit greatly from its design, which makes maintainability and code organization easier.

With its integrated routing, form validation, and authentication scaffolding capabilities, Laravel provides a notable benefit in terms of development time. In contrast to Spring's intricate configuration requirements, these characteristics allow for quicker iteration cycles and rapid prototyping, enabling the developer to fulfil delivery schedules effectively. Besides that, Laravel has strong built-in security features, including encryption, hashed passwords, middleware-based access control, and CSRF tokens. These are essential for safeguarding private information on the proposed system. Although Django and Spring Boot offer robust security as well, Laravel's simplicity of use makes it more appropriate for this proposed system with a limited timeframe.

With built-in caching and query optimization, Laravel shows moderate but effective performance improvements, greatly enhancing response time in low-resource contexts. Laravel's performance profile provides the ideal combination between functionality and efficiency, especially as the proposed system may be used in places with inadequate infrastructure. Finally, Laravel also offers robust long-term support through its vibrant international community, extensive documentation, GitHub resources, and third-party packages. This lowers the possibility of bottlenecks during the project lifetime and guarantees that the developer can discover solutions fast.

2.3.4 Interactive Development Environment (IDE)

As for the interactive development environment, the developer will be using Visual Studio Code to build and manage the proposed system. Developed by Microsoft, VS Code is a fast, free, open-source, and cross-platform code editor that balances the simplicity of a text editor with the advanced features of an IDE which makes it ideal for building Laravel-based PHP applications in a modular and scalable way.

One of the strengths of this IDE is its performance efficiency and low system resource consumption, making it suitable for mid-range developer machine (Uspenski, 2023). Despite its small footprint which is under 200MB, it supports hundreds of programming languages, and its extension marketplace enables easy integration with frameworks like Laravel, database clients, Git, and SSH tools. Besides that, VS Code also offers multifunctionality, with built-in tools for editing, debugging, version control (Git), and terminal access within a single interface.

The vibrant and close-knit developer community focused on the VS Code extension marketplace is another noteworthy benefit. In addition to contributing top-notch tools like PHP Intelephense and Laravel Blade Snippets, developers also evaluate, update, and support them, creating a more reliable and trustworthy environment. Productivity is increased and the chance of technical interruptions is reduced. These advantages, together with VS Code's cost-free, open-source status, cross-platform compatibility, and regular feature upgrades, make it the most effective, versatile, and developer-friendly IDE for creating safe, modular, and practical systems like the proposed system.

2.3.5 Database Management System chosen

MySQL was selected as the Database Management System (DBMS) for the proposed system development. Due to its dependability, scalability, and interoperability with web applications, particularly those developed using the Laravel framework, MySQL is an open-source relational database system (Erickson, 2024). Through Eloquent ORM, Laravel's integrated object-relational mapper, which makes database operations like querying, inserting, and updating records easier with user-friendly PHP syntax, MySQL and Laravel interact effortlessly

(Laravel, n.d.). When working with complicated datasets like user roles, child abuse case reports, case status logs, and audit trails, this facilitates clearer, more maintainable code.

Advantages and Disadvantages

MySQL is a popular choice because of its organised schema design and support for ACID compliance (Atomicity, Consistency, Isolation, Durability), which guarantees the reliable and safe storage of sensitive data, including reports filed by the public or law enforcement (MySQL Database, n.d.). To enforce data integrity across related tables, MySQL further provides foreign key restrictions. Furthermore, the proposed system can benefit from MySQL's strong speed optimisation for read-heavy activities when doing its purpose such as viewing reports, analysing data, or tracking cases. It is appropriate for applications that could grow to accommodate higher case volumes over time because to its compliance with indexing, query caching, and data partitioning.

The database schema may be easily set up, versioned, and managed throughout the project lifetime thanks to MySQL's support for tools like phpMyAdmin, MySQL Workbench, and Laravel's own migration mechanism (Laravel, n.d.). MySQL's security and access control capabilities aid in the efficient enforcement of limited data access and logging procedures, as the proposed system entails multi-user access with varying permission levels.

However, MySQL also presents some notable limitations. One of it is that it lacks advanced built-in development and debugging tools, making tasks like performance tuning and query optimization more challenging compared to enterprise-grade databases (GeeksforGeeks, 2022). Additionally, MySQL is not fully SQL-compliant, with limited support for certain features such as full outer joins and check constraints. These missing features can create challenges when designing highly complex relational queries or enforcing strict data validation rules. For example, if the system needs to show all users and their cases even those without any records, doing this without full outer join support would require more complicated SQL workarounds. This increases development complexity and places greater responsibility on the backend logic for maintaining data consistency.

Comparison with other DBMS

Table 5: Comparison with other DBMS

Criteria	MySQL	PostgreSQL	MongoDB
Performance	Optimized for fast read operations and lightweight workloads. Great for web applications.	High performance in complex, concurrent, and transactional workloads.	Very fast for unstructured data. Ideal for real-time and big data apps.
Data Integrity	Strong ACID compliance with InnoDB engine. Supports foreign keys and transactions.	Excellent ACID compliance. Better enforcement of constraints and complex relationships.	Weaker consistency. Does not support joins or relational constraints natively.
Security	Good built-in features. Depends on proper configuration.	Offers advanced security. Row-level security, encryption, and user-defined roles.	Supports encryption and access roles but less granular than relational DBMSs.
Ease of use	Easy to set up, widely supported, beginner-friendly tools like phpMyAdmin simplify management.	More complex to configure. Better suited for experienced developers or larger teams.	Simple to install and scale. Ideal for JSON-based data but requires careful schema planning.
Cost	Free and open source; supported by almost all shared and cloud hosts	Free and open source; some advanced features may require more system resources	Free and open source; requires more powerful hosting for high-volume data workloads.

2.3.6 Operating System Chosen

For the development and testing of development system, the operating system chosen is Windows 11. Launched in 2021, Windows 11 is Microsoft's latest operating system, known for its modernized interface, improved system performance, and development-friendly tools (Bishop, 2021).

One of Windows 11's main benefits is its redesigned user interface, which includes features like Snap Layouts, a centred taskbar, and improved workspace organisation (Raymond, 2024). These functions improve efficiency and multitasking, especially for developers working on their systems. Furthermore, Windows 11 has better memory and resource management, which boosts speed and makes it more responsive and effective for both general and development use (Raymond, 2024).

Windows 11 does, however, have a few significant shortcomings. Its rigorous hardware requirements, which include the need for TPM 2.0 and compatibility with current CPUs, are its most significant drawback and may prevent installation on older computers. For developers or organisations utilising somewhat antiquated hardware, this might provide accessibility issues (Raymond, 2024). In addition, the OS enables less customisation than Linux-based systems and has had problems with compatibility with legacy software. Despite these concerns, Windows 11 remains a practical and developer-friendly choice for academic and local development. Its widespread adoption, support for modern development tools, and improved system responsiveness make it suitable for efficiently developing, testing, and managing the proposed system.

Chapter 3: Methodology

3.1 System Development Methodology

3.1.1 Introduction

The success of any system heavily relies on adopting a suitable software development methodology (Khakhkhar, 2025). A well-chosen method provides a structured framework that guides the planning, development, testing, and deployment stages, ensuring the project remains manageable and efficient (Krasteva, n.d.). For this project, the Kanban technique was selected as the development methodology due to its simplicity, adaptability, and suitability for solo developers. Given that the proposed system consists of multiple modules such as administrative dashboards, user role management, and more, Kanban's visual workflow and emphasis on continuous task delivery offer significant advantages. This section outlines the rationale behind Kanban and details its application throughout the system development lifecycle.

3.1.2 Methodology choice and justification

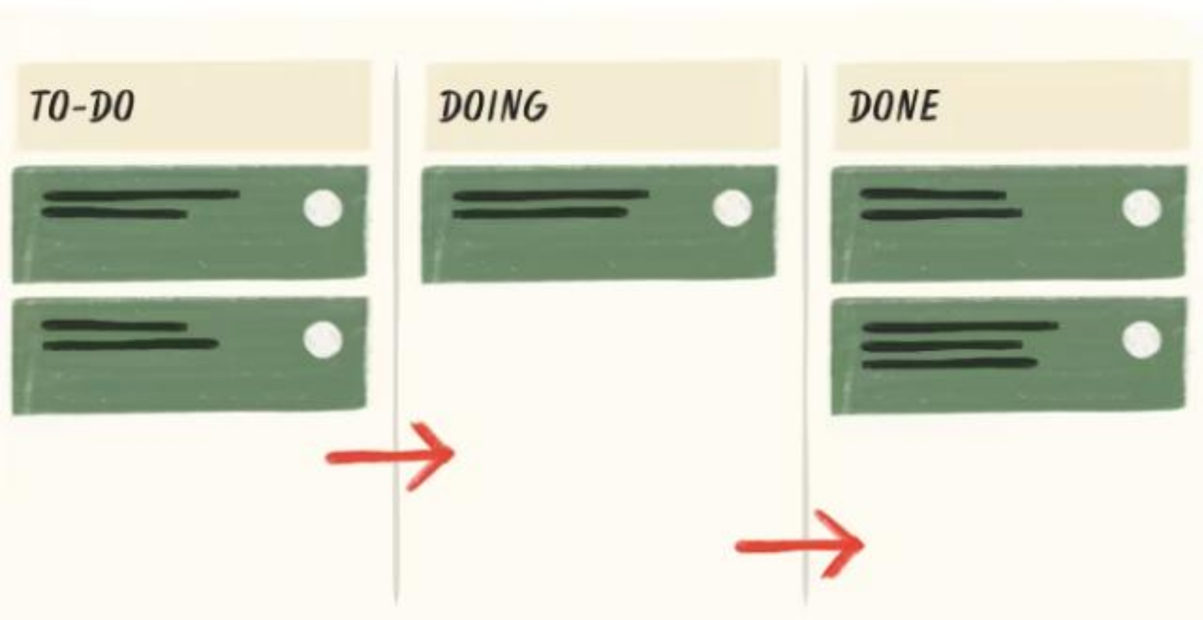


Figure 16: Kanban Workflow Visualization Showing Task Progress from To-Do to Done (Adegbuyi, n.d.)

The Kanban methodology was chosen above alternative approaches, such as Scrum or Waterfall, as the development methodology for this project. Developed initially as a component of the Toyota Production System in the late 1940s by Japanese engineer Taiichi Ohno, Kanban is a lightweight framework categorised under Agile software development (Martins, 2025; K&C, n.d.). It was later modified for software engineering to increase task management flexibility, visualise progress, and streamline workflows.

Table 6: Comparison Between Kanban and Scrum (planview, n.d.)

Criteria	Kanban	Scrum
Structure Type	Continues flow	Iterative Sprints
Team Dependency	No predefined roles needed	Team-based where it requires Scrum Master, Product Owner, and Development Team
Task Management	Visual boards with flexible task movement	Sprint backlog planned at the start of each sprint
Adaptability	High as tasks can be added or reprioritized anytime.	Low as changes are discouraged mid-sprint.
Progress Tracking	Based on visual task status	Based on sprint completion and burndown charts
Responsiveness	Very responsive to new tasks, or bug fixes	Changes are limited once sprint has started.

Kanban methodology is suitable for this project as it helps the development process for the proposed system become more adaptive, with a strong focus on visual task tracking through columns such as “Backlog”, “To Do”, “In Progress”, “Review”, and “Done” (Martins, 2025). Tasks can be added, re-prioritized, or updated in real-time without rigid sprint cycles. For instance, if an unexpected bug is discovered in the case reporting module during testing, a new task card can be immediately created and slotted into the “To Do” column without disrupting the progress of other modules already in development. This makes Kanban highly suitable for solo developers, where flexibility and personal workload control are essential (K&C, n.d.).

Furthermore, Kanban is also beneficial for this project since it is flexible enough that it allows developers to apply changes based on testing or other outcomes instantly (Hakoune, 2023). This helps guarantee that the system stays in line with user expectations throughout its lifespan and lessens the need for rework in subsequent phases. Moreover, Kanban is also a visual work management system that represents work progression within progress (S, 2024). Hence, its visual structure enhances transparency and accountability by clearly displaying the task statuses. This makes it significantly easier for the developer to get a comprehensive project overview, monitor progress, and refer to completed or ongoing work when needed.

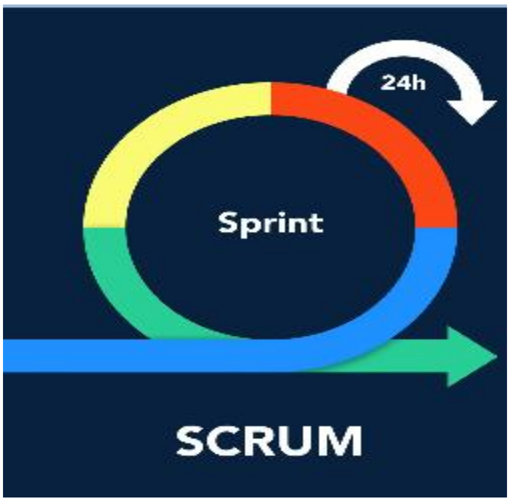


Figure 17: Scrum

Scrum, on the other hand, is also an Agile framework that organises development around sprints, which are time-boxed iterations (Digitai.ai, n.d.). Each sprint begins with a planning session where the team commits to a set of tasks from the sprint backlog. In addition, it makes extensive use of established roles, such as the Development Team, who carry out the tasks, the Product Owner, who establishes priorities, and the Scrum Master, who assists the process. For team collaboration and continual development, Scrum also incorporates regular meetings like daily stand-ups, sprint planning, sprint reviews, and retrospectives (Digitai.ai, n.d.).

Such approaches add needless complexity and overhead in a solo development setting, even if they work quite well for collaborative teams. It becomes ineffective and impractical for a single developer working on a system to oversee several Scrum roles and have formal meetings. Additionally, it is challenging to integrate input or fix unforeseen defects mid-cycle due to the strict structure of Scrum sprints, which is frequently required in projects where needs may change over time (Kamal, 2024). As a result, even if Scrum is a well-known Agile framework, it is not as suitable for the development of the suggested system, where adaptability, flexibility, and autonomy are crucial. On the other hand, Kanban's visual and lightweight methodology facilitates task monitoring, improves solo productivity, and guarantees a more seamless project development process.

3.1.3 Activities and Process in the chosen methodology

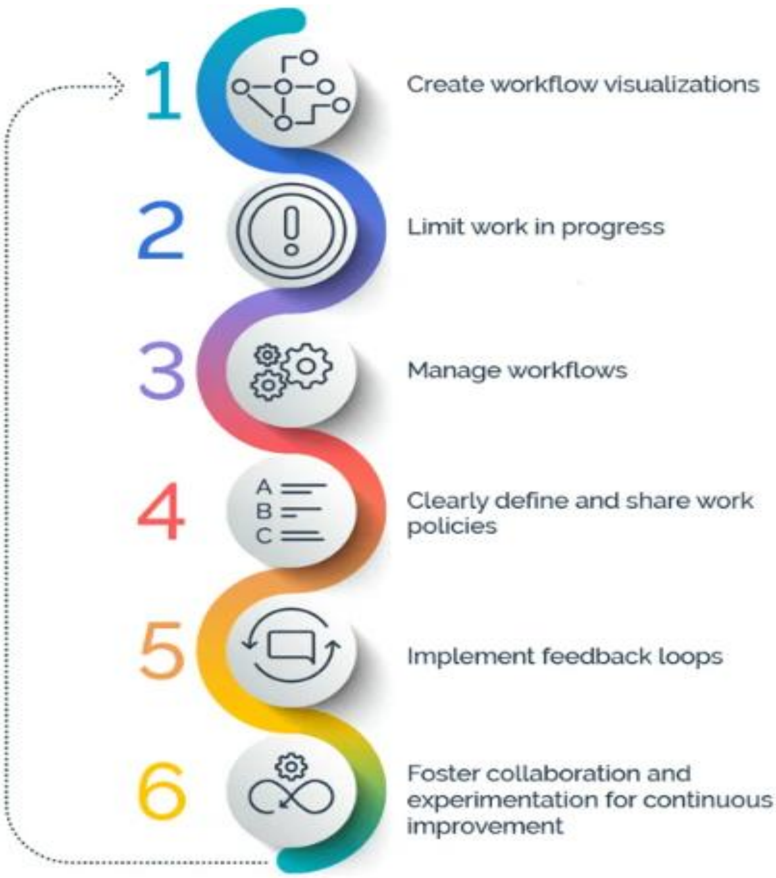


Figure 18: Key Principles of Kanban Methodology for Workflow Optimization

The development activities in this project were structured using the Kanban methodology. Unlike traditional models with rigid phases, Kanban promotes a visual and flexible flow of tasks through a continuous delivery process. To ensure an effective application, this project followed six core steps of Kanban implementation derived from Lean principles.

Step 1: Create workflow visualizations

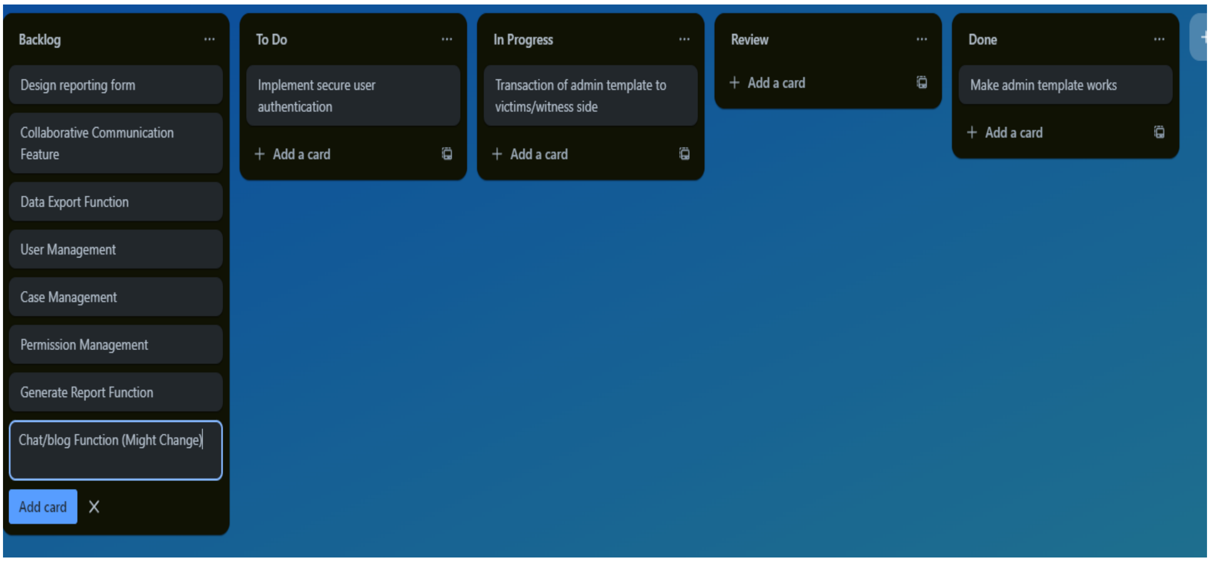


Figure 19: Kanban Board Representation of System Development Tasks

For this phase, it usually involves mapping and visually representing the entire production process either thought physical or digital Kanban boards, maps or panels (Oliveira, 2023). For instance, in this project, a digital Kanban board was implemented using Trello, a visual project management tool that supports Agile task tracking. The board was structured with five columns which is Backlog, To Do, In Progress, Review, and Done, representing the full life cycle of a development task from ideation to completion.

Each task card represented a specific feature or functionality required for the proposed system such as “Implement secure user authentication,” “Design reporting form,” or others. These tasks were initially placed in the Backlog, then moved to To Do when prioritized for development. Once the developer began working on a task, it was transitioned into In Progress, followed by Review after implementation and testing, and finally into Done upon approval or successful completion. This visualization provided a real-time overview of development progress, ensured transparency, and helped identify any stagnation in task flow.

Step 2: Limit Work in Progress

This phase emphasizes limiting the number of tasks being actively worked on at any given time to maintain focus, reduce context-switching, and increase productivity. According to Kanban best practices, implementing WIP limits helps avoid multitasking and encourages developers to complete current tasks before taking on new ones (Martins, 2025).

In this project, the principle was applied through a Trello board, where a practical WIP limit of two active cards was enforced within the “In Progress” column. As shown in the figure above, at the time of capture, only one task was present in “In Progress”: “Transaction of admin template to victims/witness side.” Meanwhile, “Implement secure user authentication” was prepared in the “To Do” column but deliberately left unstated to comply with the WIP limit. This approach ensured that no more than two development tasks were being handled simultaneously, preventing overload and improving task throughput. A previously completed task, “Make admin template works,” had already moved to the “Done” column, indicating successful focus and follow-through. Without WIP limits, it would have been easy to begin multiple features from the “Backlog” simultaneously, which could have led to scattered attention and incomplete implementations.

Step 3: Manage workflow

Managing workflow involves continuously monitoring how tasks move across the Kanban board to identify and address delays, bottlenecks, or inefficiencies (). For an instance for this project, when the task “Transaction of admin template to victims/witness side” remained longer than expected in the “In Progress” column, the developer will then need to pause new task initiation and focused on resolving the template layout bugs and compatibility issues. Hence by paying attention to the follow interruptions, the developer will be able to maintain a smooth progression of tasks and avoid stagnation.

Step 4: Make process policies explicit

This stage involves defining policies or rules that guide how tasks move through the workflow (Martins, 2025). For instance, in this project, a task could only be moved to the “Review” column after the feature was fully implemented and manually tested. Additionally, no task was allowed to move to “Done” unless all related bugs were resolved, and proper documentation was written. Hence, by doing so it helped the developer maintain a consistent standard for task completion and ensured that only verified, production-ready features progressed through the workflow. These clearly defined policies reduced uncertainty, improved development discipline, and minimized the risk of overlooking critical quality assurance steps.

Step 5: Implement Feedback Loop

This step emphasizes incorporating regular feedback into the development cycle to support continuous improvement. In this project, feedback was primarily gathered during scheduled meetings with the supervisor or from the stakeholder of the proposed system through the chosen data gathering method, where new ideas or features can be reviewed, and evaluated. For instance, after presenting the chat feature to the supervisor, the super supervisor suggested adding a blog like feature instead of chat. This feedback will then be converted into a new into a new Trello task and placed back into the "To Do" column for refinement (Avatour, n.d.). Hence doing feedback loops helps to identify issues early, refine requirements and adjust priorities dynamically and ensure the proposed system is more aligned with the requirements of the project.

Step 6: Continuous improvement

The final step in Kanban implementation involves continuously evolving the development process based on observations, challenges, and lessons learned. This principle encourages developers to refine workflows, tools, and practices incrementally to improve efficiency and output over time (Oliveira, 2023; Martins, 2025). In this project, several workflow adjustments were made as development progressed. For example, a "Bug Fixes" column was added to the Trello board after identifying that post-review issues were being mixed with new tasks, which caused confusion and disrupted prioritization. Additionally, task templates were created for recurring modules such as form validation and role permission checks, reducing the need to rewrite standard procedures. These small but strategic changes allowed the workflow to adapt to the actual project needs and challenges, thereby increasing development efficiency and quality. Embracing continuous improvement ensured the project remained agile, responsive, and aligned with evolving goals throughout the development of proposed system.

3.2 Data Gathering Design

3.2.1 Data Collection Method

Survey has been chosen as the main technique of data gathering for this project to guarantee that the features and design of the SinDA system are aligned with the real requirements of the parties that involved in child protection. The use of surveys allows the project to gather quantifiable data that can be analysed systematically, providing concrete insights into the challenges faced in current reporting and case management processes (Bhandari, 2021). Additionally, the method allows for standardization of questions, ensuring consistency in responses across different respondent categories (Bhandari, 2021). Not only that, but surveys are also widely recognized as one of the most reliable tools for conducting research, especially when attempting to make data-driven decisions based on the needs, attitudes, and feedback of specific target groups. Surveys enable researchers to gain direct feedback, uncover patterns in stakeholder behaviour, and validate assumptions which are especially important when building systems intended for sensitive issues like child abuse reporting (SurveyMonkey, n.d.). For this reason, a structured questionnaire was developed to explore existing gaps in the child abuse reporting process, assess the usability of digital systems, and identify the expectations of potential users. The survey also aims to determine how features such as anonymity, inter-agency collaboration, and case progress tracking are perceived by different user groups.

3.2.2 Comparison between Data Collection Methods

Table 7: Comparison between Data Collection Methods

Methods	Strength	Weakness
Survey (DeFranzo, n.d.)	Efficient for collecting data from many respondents without cost due to its scalability. Besides that, participants can answer the form anytime they wanted.	Survey with lots of close-ended question may restrict respondents from fully exploring their opinions, which can limit the depts and richness of collected data.
Interview (Indeed, 2025)	Allow in-depth exploration of individual experience and provide chances to clarify	Can be time-consuming to conduct and analyse, especially when dealing with multiple participants or

	responses through follow-up questions.	complex topics like child welfare.
Document Analysis (Indeed, 2025)	Allow researcher to gather background information and understand existing policies or regulations without direct involvement.	Documents like Statutory forms or case records form NGOs may contain limited or outdated information which can result in gaps when addressing current research question or stakeholder needs.
Observation (Prasanna, 2024)	Enables researchers to examine how frontline employees and system users behave in this scenario's genuine working environment. It will assist in identifying problems in the present that are not disclosed by the survey.	Since reporting child abuse involves delicate and private exchanges that are unethical to witness firsthand, observation is less feasible in this study. Furthermore, if users are aware that they are being watched, they can behave differently, which could lessen the validity of the results.
Focus Groups (StudySmarter, n.d.)	Encourage open discussion among participated which make it possible to explore shared concerns, generate ideas and understand collective views on digital reporting features.	Participants may hesitate to speak openly in group settings and discussions may be dominated by stronger personalities, potentially skewing the results

Following an evaluation of several data collection based on table above, surveys were determined to be the best technique among the rest for this project. When used in a delicate area like reporting child abuse, each approaches have drawbacks even though it has unique advantage such as depts offered by interviews or the contextual insights from observation.

Privacy consideration may make observation inappropriate and focus group participants may be unwilling to share personal information in public. However, surveys offer a useful and trustworthy means of gathering opinions from a variety of stakeholders and it allows participants to answer them anytime they want.

3.2.3 Survey Structure

Table 8: Survey Structure

Section	Question Type	Number of Questions	Purpose
Section A - Background Information	Multiple Choice (MCQ)	3	To gather demographic information such as gender, age group, and respondent role for segmentation and trend analysis.
Section B - Current Child Welfare Reporting Challenges	Likert Scale (1–5) and Multiple Choice	9	To assess the effectiveness, accessibility, coordination, and limitations of existing child abuse reporting systems from the stakeholder's perspective.
Section C - Perception of the Proposed System	Likert Scale (1–5) and Multiple Choice	9	To evaluate acceptance and expectations related to the proposed digital platform features such as anonymity, tracking, inter-agency coordination, and perceived implementation barriers.

The survey was created to be straightforward, organised, and simple enough for different child protection stakeholder groups to understand. To address various research objectives that are important to the proposed system, the 21 questions are arranged into three main sections. Each section's goal is to gather specific information from law enforcement, social workers, medical professionals, and the public which are all parties involved in child welfare. The Background Information segment is structured to gather the basic demographic data such as role, age, and gender of respondents. Accordingly, the information will be utilized in their classification for later comparison and trend analysis with other stakeholder groups. The second section aims to find the problems and the efficiency of the current child abuse reporting processes in Malaysia. It covers issues like interagency collaboration, reporting delays, anonymity in reporting, and the level of trust in the system that they can have. The systemic flaws and pain points that can make the reporting or case resolution difficult will be easily found with these insights. The final section asses the expectations for respondents about possible digital features. This covers views on role-based access, digital case tracking, interagency messaging, anonymity, centralised reporting platforms, and safe data handling. Anticipated implementation obstacles including privacy issues and technological preparedness are also covered in this section. The survey's overall results will direct the enhancement of the SinDA system's functionalities to better meet stakeholder expectations and real-world requirements.

3.2.4 Questions Objective

Table 9: Questions Objective

Question	Objective
Section A – Background Information	
1. What is your gender?	To categorize responses by gender for analysing reporting behaviour and perception trends across different demographics.
2. Which age group do you belong to?	To segment respondents by age to identify patterns in awareness, accessibility, and digital readiness.
3. Which category best describes your current role?	To distinguish responses based on stakeholder type (e.g., public, NGO, police) and analyse each group’s needs and perceptions.
Section B – Current Child Welfare Reporting Challenges	
1. The current process for reporting child abuse is efficient.	To evaluate public and professional perceptions of the speed and clarity of current reporting workflows.

2. Victims or witnesses can make anonymous reports easily.	To assess the availability and effectiveness of anonymity features in existing systems.
3. There is good coordination between agencies (e.g., police, welfare, NGOs).	To identify whether current systems support smooth inter-agency collaboration in handling child abuse cases.
4. Communication breakdowns or delays often affect case outcomes.	To determine the extent to which delayed communication impacts resolution speed and child safety.
5. Have you ever experienced or been informed of issues with case reports being lost, duplicated or ignored?	To assess the reliability and perceived integrity of existing reporting systems in handling and processing submitted reports.
6. Victims or their families feel discouraged from reporting due to fear or lack of trust.	To explore emotional and social barriers that prevent individuals from making child abuse reports.
7. A lack of centralized digital reporting makes inter-agency work harder	To determine whether the absence of a unified reporting system contributes to inefficiencies in collaboration between stakeholders.
8. A lack of public awareness about existing reporting system affects the effectiveness of child abuse reporting	To measure how public knowledge gaps affect usage, accessibility, and effectiveness of current reporting mechanisms.
9. Frontliners (e.g., social workers, police, NGOs) currently use different tools for recording and updating case progress.	To identify system fragmentation among agencies and assess the need for a standardized, centralized platform.
Section C - Perception of the Proposed System	
1. A centralized digital platform could improve the efficiency of child abuse reporting.	To assess stakeholder agreement on the value of a unified system in reducing delays and improving overall reporting flow.
2. The option to report cases anonymously is essential for protecting victims and witnesses.	To evaluate how critical anonymity is in encouraging report submission and ensuring emotional safety.

3. A digital case tracking feature would help ensure better follow-up and accountability.	To determine the perceived importance of real-time case progress tracking in ensuring transparency and sustained intervention.
4. Real-time communication tools (e.g., chat, comments) between agencies would improve coordination.	To understand whether stakeholders believe integrated communication tools could strengthen multi-agency cooperation.
5. Role-based access in the system would help safeguard confidential case data.	To assess stakeholder support for controlled access features that protect sensitive information based on user roles.
6. Using digital tools would reduce paperwork and manual documentation.	To evaluate whether stakeholders recognize the operational benefits of digitalization in reducing administrative workload.
7. The adoption of digital methods for managing child-related reports is a practical improvement.	To measure general acceptance of transitioning from manual processes to digital workflows in child case management.
8. Digital platforms are capable of securely handling sensitive information related to child welfare.	To determine confidence in the ability of digital systems to maintain data confidentiality and security.
9. What is the biggest challenge to implementing a centralized digital platform for reporting child abuse?	To identify key concerns among stakeholders regarding the implementation of digital systems, such as privacy, infrastructure, or resistance to change.

3.3 Analysis

The data collected from a structured questionnaire that was given to 31 responders including law enforcement, social workers, non-governmental organisations, medical professionals, and members of the public is analysed in the part that follows. Finding issues with the current child abuse reporting procedure and assessing the applicability and acceptability of suggested improvements for the SinDA (Integrated Child Welfare System) platform are the objectives. Three areas serve as the framework for the analysis which are stakeholder impressions of the planned SinDA system features, the demographic profile of respondents, and the constraints in the current reporting ecosystem. This method assists in determining the demands of stakeholders, important operational gaps, and feature priorities for the next stage of development.

3.3.1 Demographic Information

This section's demographic analysis is intended to gain a detailed insight into the variety of backgrounds of the individuals who took part in the survey. This section, through the collection of data on stakeholder role, age group and gender, aims to reveal the possible impact of these attributes on perceptions, priorities, and expectations toward child abuse reporting systems in Malaysia (SurveyMonkey, n.d.). The comprehension of these traits is paramount to creating a digital space where users from different operational backgrounds, responsibilities, and degrees of interaction with the existing child protection process are accommodated.

Thorough knowledge about the demographics of the respondents will help in drawing inferences of a deeper level from the survey results. As an illustration, general users may give more importance to accessibility and anonymity but professionals like police officers or healthcare workers might be more concerned about procedural integrity, secure data management, and role-based access. Technology comfort levels and the stance toward digital systems may also differ among the age groups, with the younger respondents being more likely to prefer the real-time features and modern interfaces.

Apart from giving an overview of the current stakeholder landscape, the demographic insight is also an approach to the designing, customizing, and communicating of the proposed system features. Beyond that, it helps to identify critical target user segments for the outreach, thus the system would not just cover the practical needs but also it would reflect the values and the expectations of end-users. The outlined trends also play a key role in the decision-making process concerned with technological development. Their influence is also evident in user-centred design decisions that are made in the system’s development.

Question 1

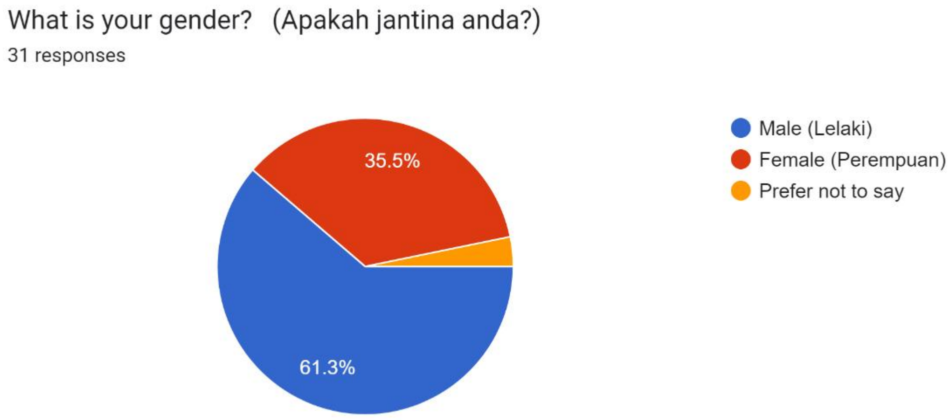


Figure 20: Gender Distribution of Survey Respondents

Based on above graph, 61.3% were males while 35.5% were females, 3.2% preferred not to disclose his/her gender. With a slight majority of males, this result still shows a balance gender distribution. Given the cross-functional nature of child protection responsibilities, the inclusion of both male and female viewpoints offers a comprehensive understanding of user expectations and concerns. Users' perceptions of urgency, sensitivity, and trust in the reporting process may be influenced by their gender-based experiences. For example, for female respondents particularly those employed in social services or healthcare may prioritise emotional safety and secrecy when submitting their reports. Along with this, male participants particularly the ones who work in law enforcement could be most likely to focus on the functioning of the system or the accuracy of the procedures. The requirement for a universal platform that accommodates the choices of all user groups without compromising the respect and seldom privacy of those not willing to disclose their gender is the implication of the insight into these differences.

<u>Question 2</u>

Figure 21: Age Group Distribution of Survey Respondents

The 31 responders' age distribution based on above graph shows a wide range of generational backgrounds. The largest group, which is 41.9% consists of individuals aged below 25, followed by those aged 25 to 34 which is 29 %, 35 to 44 which is 22.6%, and a smaller segment aged 45 and above which is 6.5%. Strong participation from younger demographics is shown in this spread which suggest that digital native users are more engaged or aware. Besides to being more open to reporting tools that focuses on speed, interactivity, and user-friendly designs, these groups are also more likely to adopt mobile or web-based solutions. Respondents that are age 35 or older, on the other hand, might be more experienced professionals such as senior officers or managers where their opinions are important in shaping secure, structured workflows and policy-aligned features within the proposed system.

Question 3

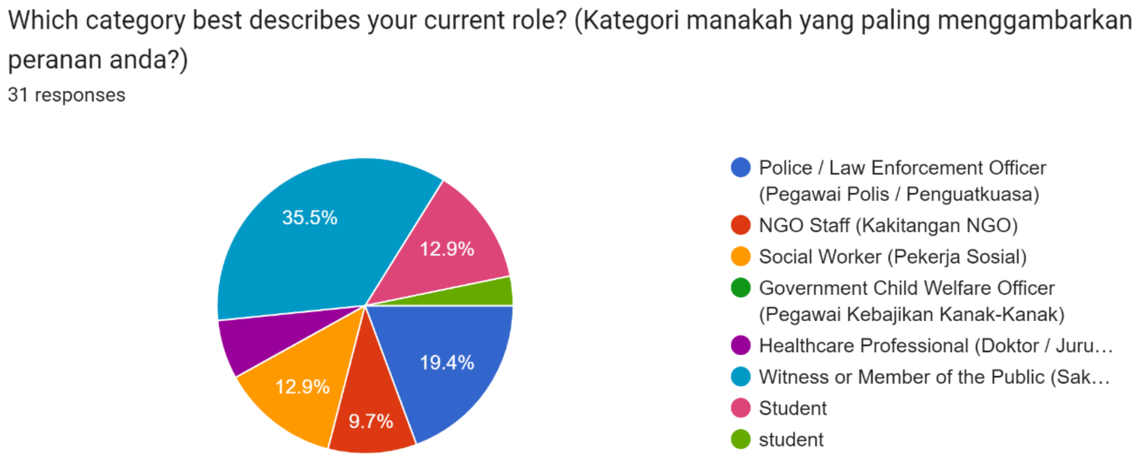


Figure 22: Stakeholder Role Distribution of Survey Respondents

The distribution of stakeholder roles illustrates a wide range of involvement in the child protection ecosystem. The largest proportion of respondents which is 35.5% identified as witnesses or members of the public, suggesting that many are positioned to observe and report abuse but may lack direct access to formal channels. This reinforces the importance of developing a platform that enables public users to report cases with ease, confidentiality, and minimal technical barriers. For this group, anonymity, clear instructions, and feedback on case status are likely to be valued.

Police and law enforcement officers represent 19.4% of the participants who plays crucial input to an investigation and enforce the law. The fact that they were included in the survey really indicates the need for tools that would support case tracking, evidence submission, and secure communication with other organizations. These individuals may benefit from process maps as well as case history completion for interrogation and reporting needs. The portion of students marked a proportion of 16.1%, reflecting the growth of young people or potential professionals who learn about this problem in related areas. Although they are not dealing with it currently, they can still present ideas for the new system. Such a system should be user-friendly, digital-accessible, and should be designed in accordance with what the future digital-first generations are expected of.

Social workers, of whom 12.9% are in the graph, are usually the frontline responders to respond to abuse cases. Very often, their work covers tasks like generating reports, working with other agencies, and overseeing the safety of children for a lengthy period. Accordingly, they need a system that allow multi-user interaction, updates on the project status, and secure data management. Likewise, non-governmental organization staff (NGOs) which are 9.7% typically take the role of the case facilitators as well as the field-level actors, and they would gain from the functions like community portals, modular reporting, and issue escalation. Healthcare professionals, even though their numbers are merely 6.5%, are the ones who are predominantly responsible for the detection of abuse indicators through medical checkups. They need very strict privacy controls, and they might prefer features such as medical document upload, flagging mechanisms, and a professional notes section to protect sensitive information.

This distribution reveals the need for SinDA to support both technical and non-technical users through a role-sensitive interface. Features must be tailored not just to access levels, but to the responsibilities and expectations of each role. The diversity of respondents underscores the platform's responsibility to be both secure and inclusive, ensuring that all stakeholders whether community-based reporters or institutional actors can engage effectively in the child protection process.

3.3.2 Current System Challenges

The quality of the reporting system is often the starting point for the effectiveness of child abuse intervention. This section sets out to investigate the impressions of the respondents regarding the child abuse reporting process that is currently in practice in Malaysia, including the confronting problems such as efficiency in reporting, delays in follow-up, lack of interagency collaboration and the perceived absence of confidentiality and trust. These questions were examined in part B of the survey, with respondents indicating their degree of agreement using a Likert scale.

The main aim of the present test is to detect critical structural or procedural deficiencies that diminish the trustworthiness of the system currently in use. This may include the inconsistent communication between the agencies, the limited public access to relevant information, and the fact that people are being judged or exposed by the time they report to the authorities. In addition, professionals may also encounter ambiguous procedures or disjointed platforms that hinder information exchange and coordinated reactions. Considering the main problems that both public users and professionals encounter, this part stresses the critical areas to be solved in the proposed SinDA platform. The knowledge gained from these areas will help to make the greatest contribution in promoting the elements of design that rebuild trust, encourage user confidence, and enhance the time taken to respond during the child protection reporting process.

<u>Question 1</u>

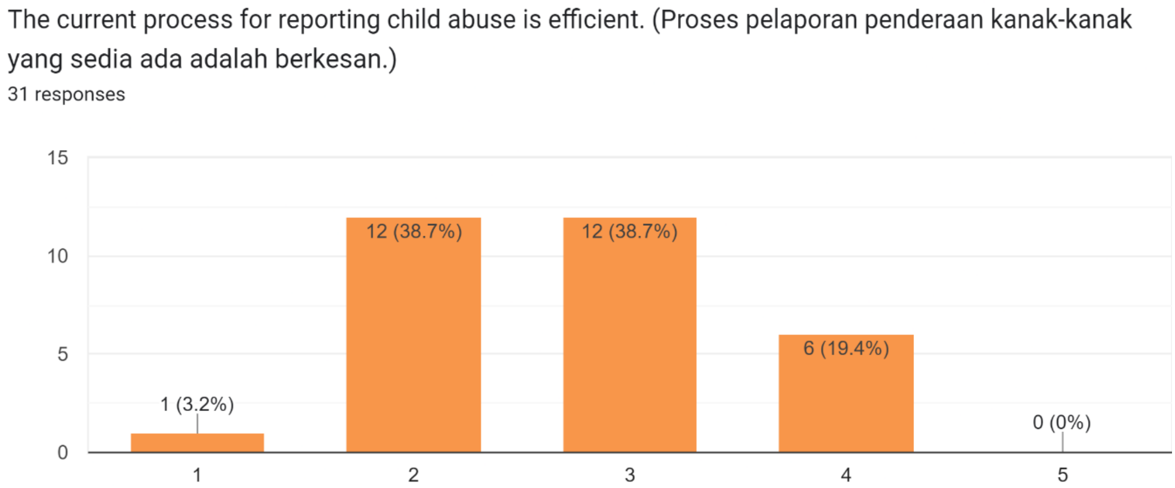


Figure 23: Respondents Perception of current Reporting Process Efficiency

Based on the result above, most respondents do not perceive the current child above reporting process as efficient. A combined of 77.4% selected rating 2 or 3 on the scale, with 38.7% choose 2 and another 38.7% choose 3. 19.4% rated it as a 4 and none selected the highest rating of 5 which is strongly agree. A very small portion however which is 3.2% strongly disagreed, selecting 1. The results of the research reveal that the current system is not trusted at all with respect to efficiency and quickness. The neutral and negative responses mentioned above indicate that the professionals may have difficulties with the unclear procedures and bureaucratic bottlenecks when dealing with cases. This might lead public users to think twice in lodging reports, even as they do not want the authorities to treat their issues with no delay or at all. The requirement for SinDA to integrate automated reporting and tracking workflows, as well as inter-agency coordination tools, is densely reinforced by this scenario. Removing the unnecessary or perceived bottlenecks from the above-mentioned system can not only improve the trust of users but also help ensure rapid responses and decrease the chances of cases being missed due to the errors in the procedures.

Question 2

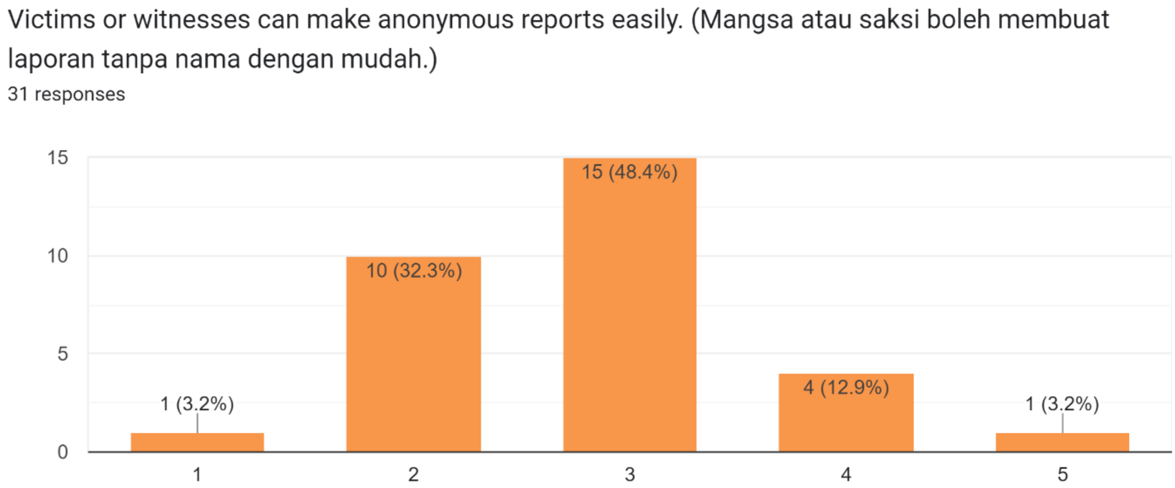


Figure 24: Respondents Perception of Anonymous Reporting Accessibility

Based on the results, it shows that a significant number of the respondents are either unsure or dissatisfied with how easily victims or witnesses can submit anonymous reports in the current system. Almost half of the respondents, making up to 48.4 percent have selected a neutral rating of 3 whereas another 32.3% disagreed by selecting a rating of 2 which indicates they believe anonymity is not affectively supported. Then, a smaller group which is 12.9%, agreed by selecting a rating of 4 while only 1 respondent strongly agreed by selecting a rating of 5 and another one strongly disagreed by selecting a rating of 1.

The results illustrate that the anonymity, which plays an essential role in motivating individuals who have suffered and observed sensitive child abuse to report these cases, is mainly seen as frail or unreliable in the present system. Due to problems such as fear of exposure and the rest that is mentioned when submitting a child abuse report, user may feel unsafe about it. This is deeply concerning because it can the discourage people from reporting which makes some abuse cases to remain hidden or unresolved. Additionally, professionals who works within the system may also know the weaknesses may lack the tools or protocols to ensure the report submitted is stored confidentially.

Secure and reliable anonymous reporting features are a necessary requirement for the SinDA platform as these findings suggest. Simply advertising the anonymity of the system is not enough; the software must actively apply guarantees of identity protection at all levels of processing. In addition, the system should provide clear directions regarding the processing of anonymous submissions, display system messages that confirm confidentiality and have secure backend processes that prevent the unintended release of confidential information. Strengthening this aspect of the system can significantly improve user trust and increase the likelihood that both victims and witnesses will report cases without fear or hesitation.

Question 3

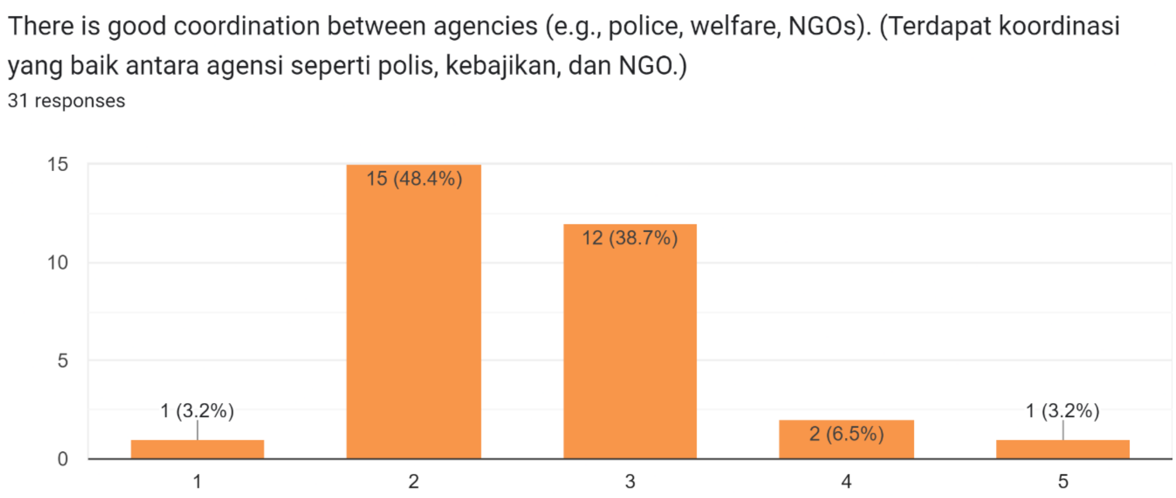


Figure 25: Respondents Perception of Interagency Coordination

The result shows that there is poor coordination between agencies such as the police, welfare departments, healthcare professionals and other user that are related when handling child abuse cases according to the responders. Nearly half of the respondents, or 48.8%, selected a rating of 2, indicating disagreement with the statement that agencies work well together. Another

38.7% provided a neutral response by selecting a rating of 3, showing uncertainty or mixed perceptions. Only a small portion agreed, with 6.5 % selecting a rating of 4, and an even smaller portion, just 3.2 percent, strongly agreed by selecting a rating of 5. Additionally, 3.2% strongly disagreed by selecting a rating of 1.

These results highlight that a large segment of both public and professional respondents perceive a lack of effective collaboration across agencies. Weak coordination can result in fragmented case handling, missed updates, duplication of work, and, most seriously, delayed or failed interventions for vulnerable children. Such gaps may arise from incompatible systems, a lack of shared communication platforms, or unclear protocols for case handovers between agencies. These insights have emphases the need to design a robust communication for interagency and coordination features. The proposed system should also have a role-based data sharing, secure messaging channels and clear workflows which allow agencies to work effectively without compromising privacy. Since users are more willing to interact with a system that have guarantees seamless and timely intervention across institutional boundaries.

<u>Question 4</u>

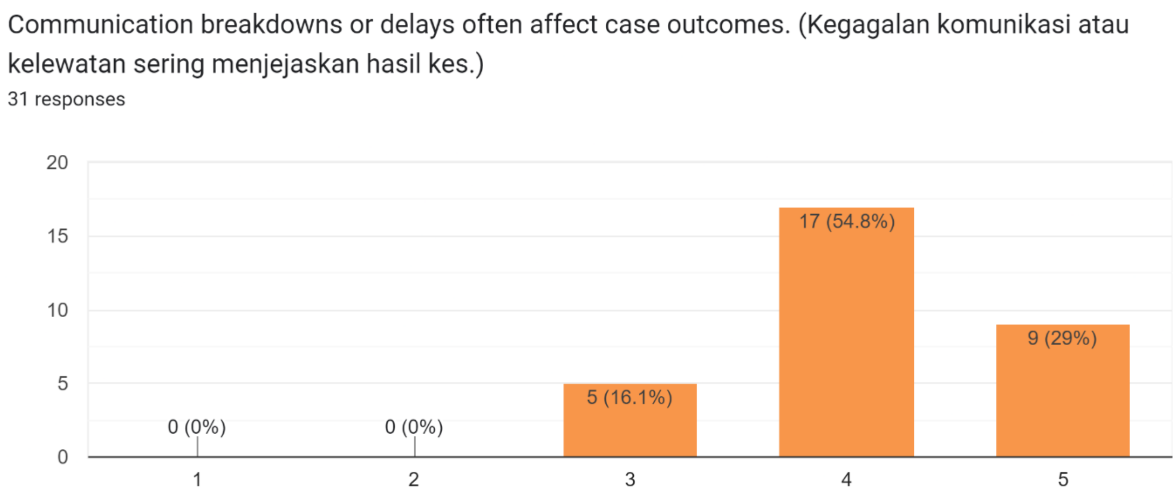


Figure 26: Respondents Perception of Communication Breakdowns Affecting Case Outcomes

The survey result shows a that there is broad agreement that poor or delayed communication often affects how child abuse turn out. More than half of the respondents, 54.5%, agreed with this statement by selecting the rating of 4 while another 29% strongly agreed by selecting a rating of 5. Notably, no ratings 1 and 2 were selected and 16.1% have selected a neutral rating of 3.

The findings point out that a considerable portion of both respondents from the public and professionals consider that there is no effective cooperation between the organizations. Weak coordination is responsible for the case being handled in parts, causing the absence of the necessary updates, doing the same work, etc., and in the most serious case, intervening vulnerable children on time or not at all. The root of the problem could be different systems that cannot function with each other, no common communication platforms, or unclear case transfer arrangements between parties.

To tackle this issue, the proposed system design should strive to incorporate straightforward communication channels that promote transparency and accountability. Some of the features would be the inclusion of visual enterprise dashboards for a real-time status, the automated status updates, and the functionality of collaborative workspaces where agencies can monitor joint case actions. Moving from individual agency workflows to a common shared platform is the first step for the proposed system to transform agencies working in the case properly and achieve the maximum outcomes by ensuring consistent and smooth information flow.

Question 5

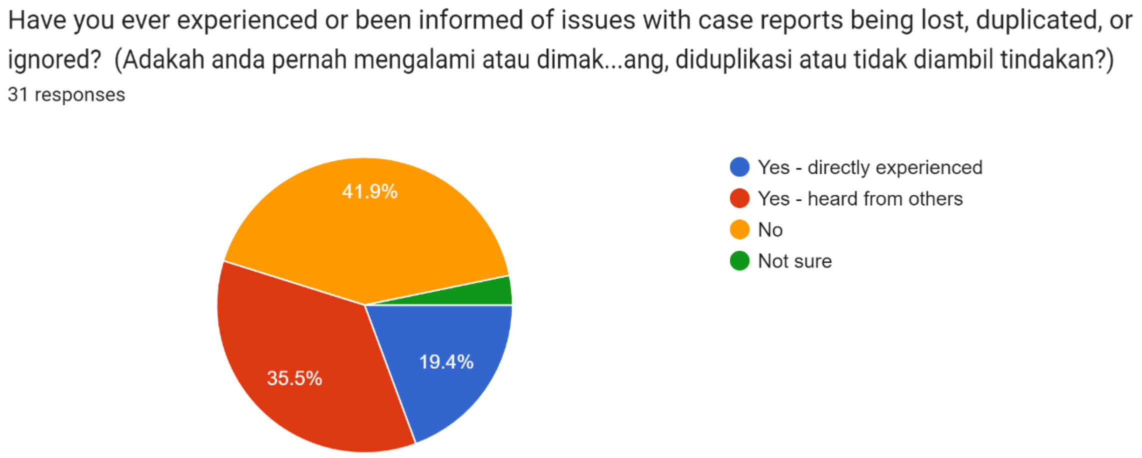


Figure 27: Respondents Experiences or Awareness of Issues with Lost, Duplicated or Ignored Case Reports

The survey results in relation to whether respondents have personally gone through or have been informed about problems with case reports being lost, duplicated, or ignored underline some major issues. Of the 31 participants, only 19.4% admitted that they have been through such cases directly, whereas another 35.5% claimed that they have heard about these cases from others. Notably, 41.9% of respondents indicated that they have not had any personal experience with such issues, and a few of them expressed their doubts on this matter. The fact that more than half of the respondents included such personal and typical experiences demonstrates that the community is aware of cases of mishandled reports, even though not all of them were affected by the issue personally.

The current handling of cases has shown some of the major deficiencies through the said findings. In the absence of an effective system, the reports may either be misplaced or inadvertently duplicated while important details may be omitted on the account of either human error or a lack of uniformly applied tracking procedures. The failure to properly and sequent keep records of the cases in the agencies does not allow them to guarantee that each report was disregarded, the action was not taken, and there was no follow-up to the resolution. In the long run, this somehow causes the public to lose trust because the victims and witnesses and even the professionals start feeling that if they report, there will be no real action taken.

To address these problems, the proposed system might focus on implementing a essential case management features like by using unique case identification number to ensure that each report is recorded as a unique entry

The software is not up to date with the latest features as it can lack automated escalations, interagency integrations, or full audit trails yet. However, it is still the first step towards a better case management through the assurance that all the reports are securely stored, accessible, and clearly documented. Improvements in these basic areas could make the system more effective at helping the agencies handle child abuse reports in a more honest and reliable manner, which in turn, is the key to winning back user trust and, consequently, better intervention results.

Question 6

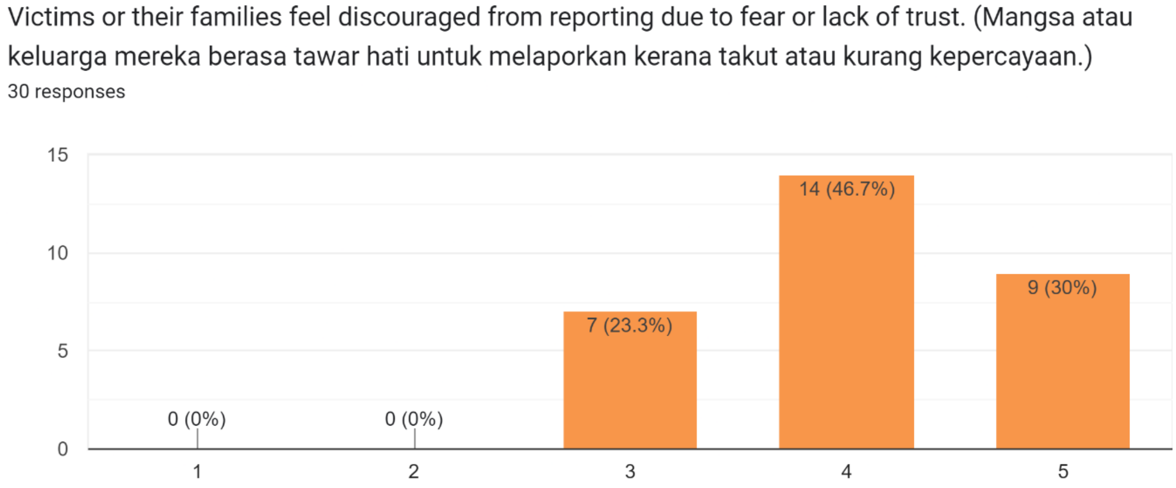


Figure 28: Respondents Agreement That Victims or Families Feel Discouraged from Reporting Due to Fear or Lack of Trust

The result shown notable regarding victims and family's hesitation to report abuse due to fear or lack of trust. Out of 31, 46.7% have agreed where they rate 4 that victims or their families feel discouraged from reporting, while an additional 30 percent strongly agreed which the respondent's rate 5. Meanwhile, 23.3% choose neutral rating and not one respondent selected disagreement ratings which is 1 or 2.

The deep consensus on this point is impressive and stresses the existence of a barrier in the reporting process, which is related to emotional and psychological obstacles. Even when technical or procedural systems run well, victims, and their relatives may have hesitations in the reporting process. They may feel that the system is unable to protect them, that they will get exposed to retaliation, or that the system will ultimately not take any meaningful action. These fears are responsible for the gradual loss of trust not only in the reporting system itself but also in the whole network of agencies concerned with child protection, like social workers, the police, and health personnel.

According to the analysis, fixing technical system problems on their own is not enough. The proposed system must also incorporate design elements that actively build user trust. For example, the system should provide clear options for anonymous or confidential reporting to reduce the fear of personal exposure. It should also present transparent status updates so that users can see that their report is being received, reviewed, and acted upon, rather than disappearing into an invisible process. Additionally, maybe with a well-written privacy notices and consent agreements embedded into the system interface can help reassure users that their data is handled securely and responsibly.

It is possible to overcome larger cultural and societal factors of trust issues. System suitability will demonstrate that this is not the only way to do it. The design itself can go a long way in eliminating the barriers of reporting by highlighting features like privacy, transparency, and user empowerment. As a result, the system would not only increase the reporting rates but also contribute to the implementation of its goal of protecting the children who are at risk of being violated more effectively.

Question 7

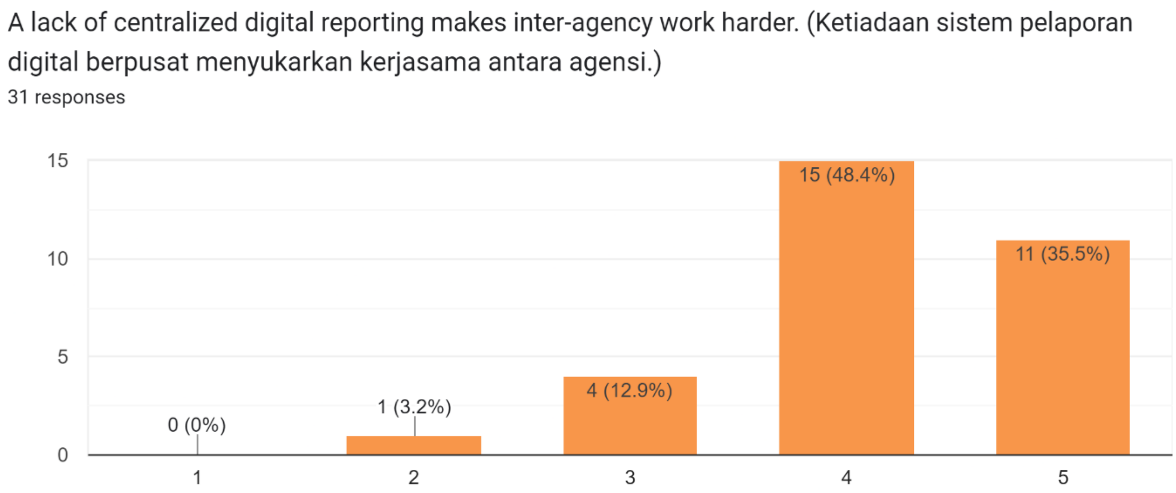


Figure 29: Respondents Agreement That Lack of Centralized Digital Reporting Makes Interagency Work Harder

As for this question, this result also shows strong agreement that the absence of a centralized digital reporting system creates challenges for interagency collaboration. Out of 31 respondents, 18.4 percent agreed with the statement while another 35.5% strongly agreed by selecting rating of 5. Only 4 respondents chose 3 which is neutral and minimal disagreement was recorded with just one respondent selecting a rating of 2 and none selecting a rating of 1.

Hence based on these findings, where most participants also agreed on, without a centralized platform, information flow between agencies becomes fragmented and inefficient. Each agency operates its own processes which results in disjointed communication, incomplete case awareness or maybe duplicate efforts as well. Due to this lack of coordination, this can significantly slow down interventions and ultimately affecting the effectiveness of child protection efforts.

Question 8

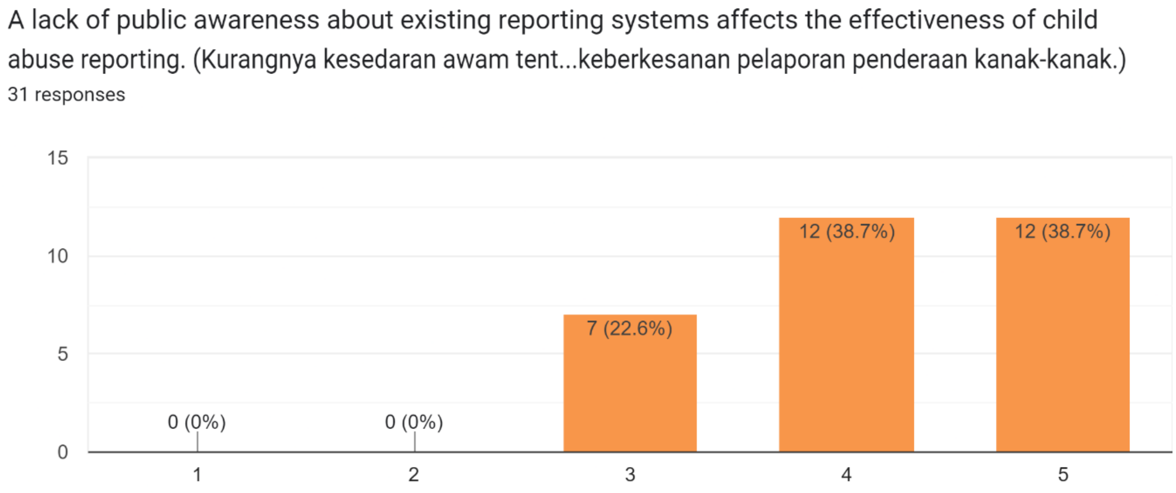


Figure 30: Respondents Agreement That Lack of Public Awareness About Existing Reporting Systems Affects the Effectiveness of Child Abuse Reporting

Next up, result clearly shows that the respondents believes that due to lack of public awareness,it affects the current system efficacy for reporting child abuse. 38.7% have given the statement rating of 5 and 4 which show strong agreement on this statement. Additionaliy, no participant disagrees with this statement since 1 and 2 are nothing. Then 22.6% gave a neutral reaction for this statement. The findings highlight that the condition of the public's knowledge on how to find and use the systems considerably influence the efficiency of the digital reporting systems, despite their being. The deficiency of outreach and educational programs can lead many community people, victims, and witnesses to be wronged without a channel to send a report or seek assistance. Sites that are not easily found will leave people without the proper tips for using them, thus resulting in the closure of many cases just because people are not adequately informed about the useful channels.

To bring about these changes, the projected system will prioritize the intuitive and informative modules ensuring that the first users can use it independently. A well-designed interface with brief guidance and learning material that states the procedure for filing a report, the needed information, and the status after submission is included in the plan. Furthermore, the visible help sections, contact points, and accessible FAQs integrated into the system will also assist the user in grasping the content.

Question 9

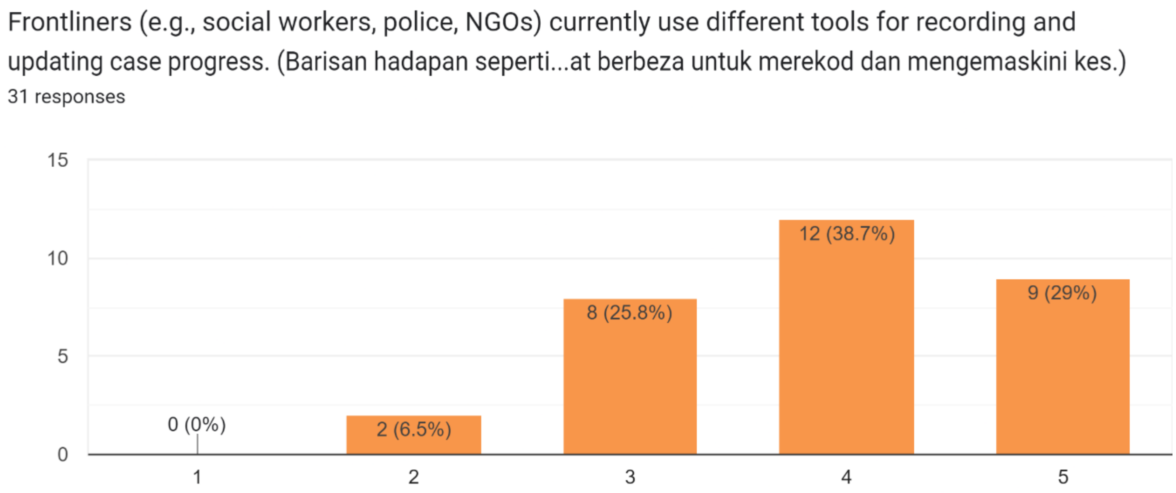


Figure 31: Respondent Agreement That Frontliners Currently Use Different Tools for Recording and Updating Case Progress

The data in figure above shows that the respondents agrees that frontline actors currently rely on separate tools for recording and updating case progress. Specifically, 12 respondents selected a rating of 4 and 9 respondents strongly agreed with a rating of 5. This high level of agreement indicates that's the stakeholders are aware of the fragmented nature of case management tools across agencies. Then, 8 respondents gave a neutral rating 3 which hints some uncertainty or varied experiences depending on their organizational context. With only 2 respondents, where they selected 2 rate and none selecting rating 1 reinforce that few see the current tools as standardized.

3.3.3 Perception and Acceptance of Proposed Features

The last part of the analysis focuses on the stakeholders' perceptions and the level of acceptance of the suggested features of the proposed system. In Section C of the survey, the data were collected regarding their views about some of the system components such as centralized reporting platform, integrated case tracking, interagency collaboration tools, and data security measures. The feedback the respondents gave in this part of the survey provided very important information about which options for the stakeholder's regard as most important, their expectations related to the use of the system in the processes of child protection, and the overall willingness to embrace the platform. This section's analysis of these answers attempts to identify feature priorities and discuss possible issues that ought to be taken into account when the system is being developed.

Question 1

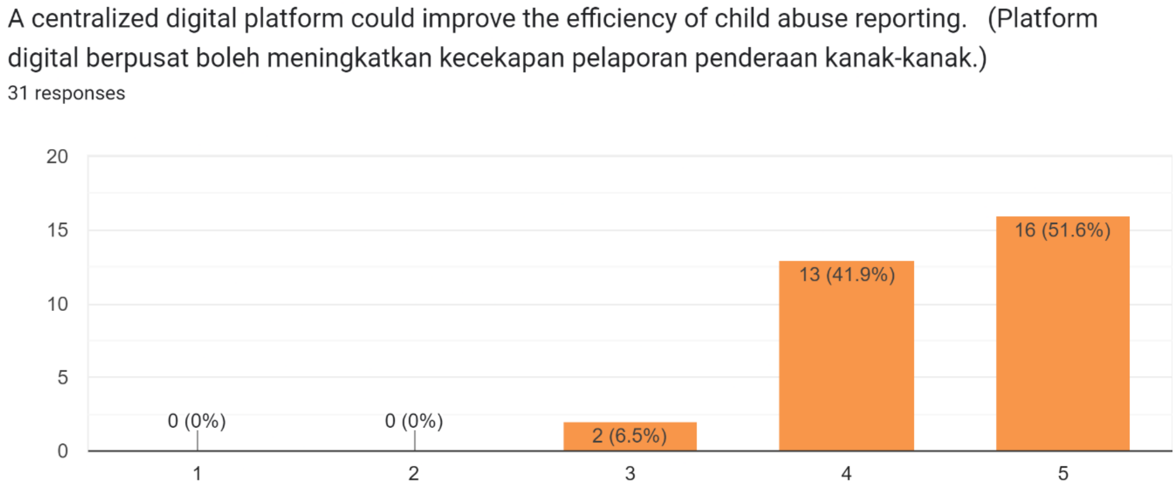


Figure 32: Respondent Agreement on Whether a Centralized Digital Platform Could Improve the Efficiency of Child Abuse Reporting

Figure above shows that stakeholders strongly believe a centralized digital platform could enhance the efficiency of child abuse reporting. 41.9% specifically selected a rating of 4 while majority of respondents gave the highest agreement rating of 5. Then only 2 respondents were neutral about this and none disagreed, indicating near-unanimous support. The strong stakeholder consensus emphasizes the critical importance of developing a unified system that enhances reporting workflows, minimizes procedural delays, and prevents duplicated efforts. Respondents clearly see that a centralized digital platform will accelerate case communication, improve tracking accuracy, and strengthen cross-agency coordination. These insights signal that SinDA's design must place high priority on robust centralized reporting features to align with user expectations and maximize operational improvements.

Question 2

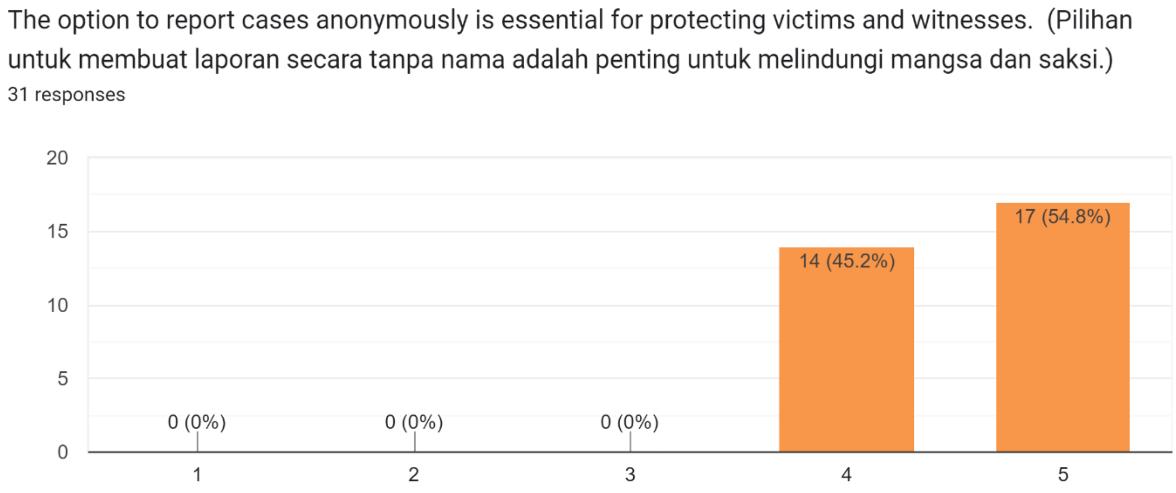


Figure 33: Respondent Agreement on the Importance of Anonymous Reporting for Protecting Victims and Witnesses

Based on the data above, responders agree that having an anonymous reporting option is important for protecting witnesses and victims in child welfare. All respondents have agreed on this statement where 14 respondents give a grade of 4 and 17 respondents give the hight possible rating. The shared position of all stakeholder groups has been highlighted as well since there's no responder voiced any disagreement for this statement. The agreement shows that some child abuse exposure reporting is very delicate and very often, it is life-threatening to the reporter. Personal safety, danger, or stigma can make it harder for people to disclose issues. Therefore, if the proposed system were to include anonymous reporting feature, the users would effectively help to overcome the fears and prejudice that impede the citizens' disclosures, thus, leading to better case detection and speeding the intervention processes. These outcomes give the impression that guaranteeing anonymity is not an option but the basic condition for creation of public confidence and increase of system's capacity.

<u>Question 3</u>

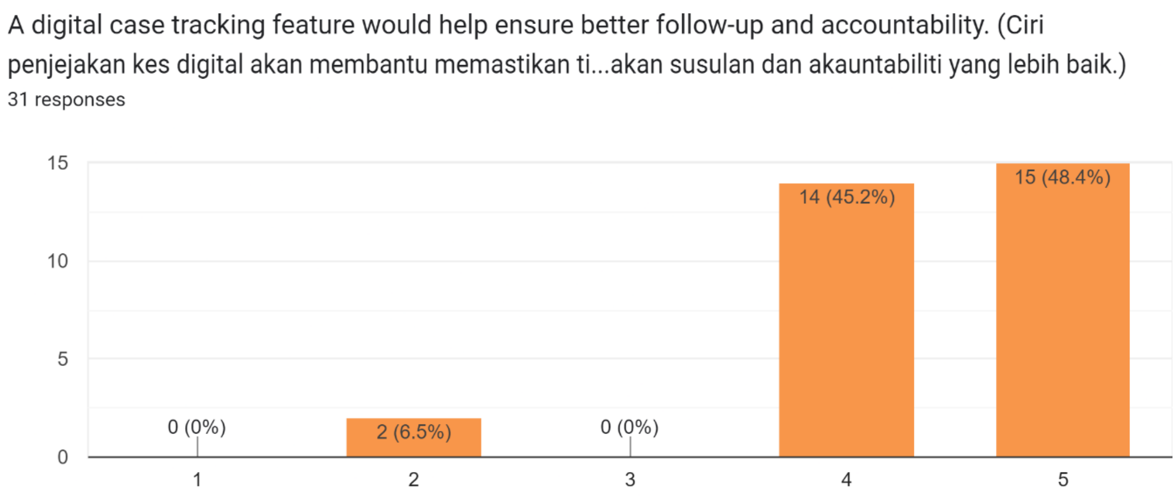


Figure 34: Respondent Agreement on the Importance of a Digital Case Tracking Feature for Ensuring Better Follow-Up and Accountability

Data above shows that nearly all respondents support the inclusion of a digital case tracking feature to enhance follow-up and accountability within the child protection system. 45.2% which is 14 respondents rated it 4 and 48.4% which is 15 respondents gave it the maximum rating 5. However, 2 respondent offered low agreement which they have rated 2 and no one selected neutral or full disagreement. Based on the data, stakeholder expectations for system transparency and the ability to monitor case progress is a must. In addition to keep them updated, a strong monitor tool will boost institutional responsibility by highlighting any gaps in assistance. To satisfy stakeholder request for traceability and foster trust in the system operational dependability, the proposed system will need to have an open case monitoring capability.

<u>Question 4</u>

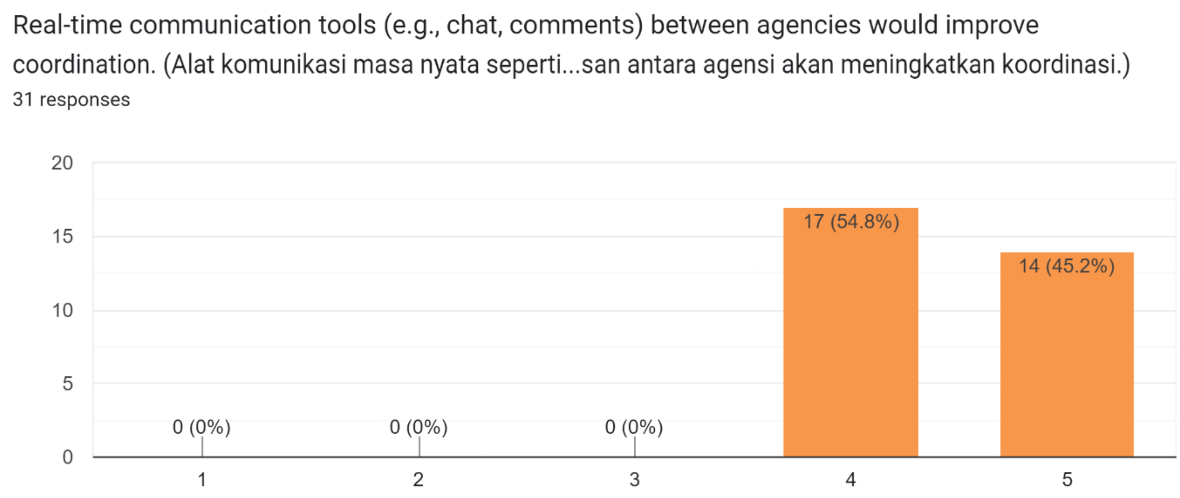


Figure 35: Respondent Agreement on the Importance of Real-Time Communication Tools for Enhancing Cross-Agency Collaboration

As can be seen from the answers, the participants firmly believe that real-time communication, including chat and comment functions, should be added to help the agencies coordinate better. Most of the respondents considered it the rating of 4, and only 45.2% rated it with the maximum mark of 5. Then, there were respondents who rated it neutral or disagreed, which points at total agreement of all participants. The need of having instantaneous and interactive channels to facilitate efficient cross-agency cooperation is widely acknowledged by stakeholders. By introducing the feature, user can find out information quicker, get a satisfactory customer service job done sooner, and thus the company can increase its productivity. These two bundles of features would be of utmost importance not only for proposed system daily process efficiency but also for the fostering inter-agency collaboration and well-coordinated, prompt, actions.

Question 5

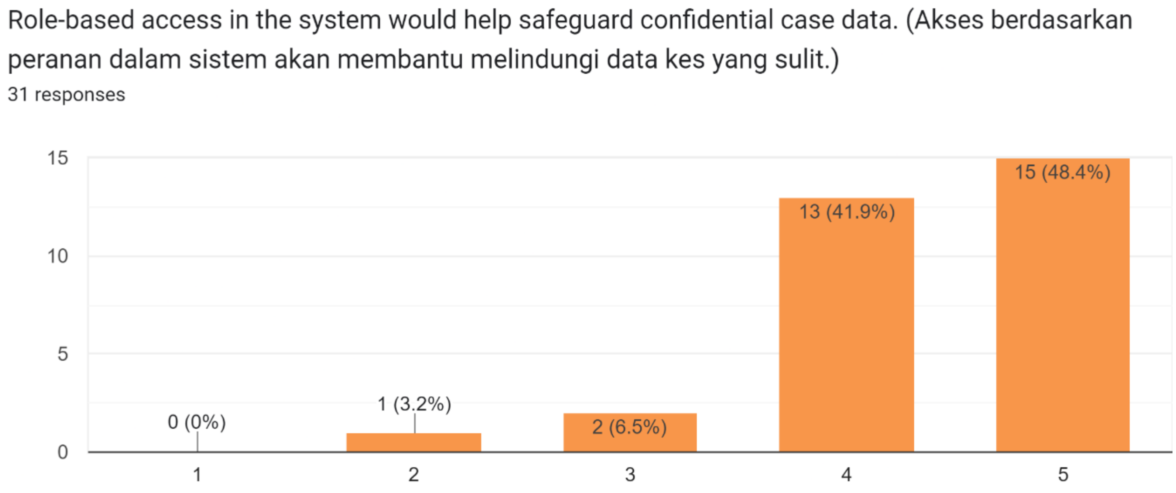


Figure 36: Respondent Agreement on the Importance of Role-Based Access Controls for Safeguarding Confidential Case Data

According to the statistics, most stakeholders are in favour of implementing role-based access restrictions to safeguard sensitive case data. In particular, only a tiny minority where 9.7% and 3 respondents indicated lesser agreement with ratings of 2 or 3, whereas 41.9 percent awarded this characteristic a rating of 4 and 48.4% which is 15 respondents provided the highest rating of 5.The discovery emphasizes that data confidentiality is a significant issue of interest to stakeholders, and therefore, it is necessary to apply the strictest access control concerning user roles for avoiding unauthorized viewing or manipulation of case details. Additionally, for SinDA, applying stringent role-based access is not only a measure of compliance with data protection laws and standards but also a tool to earn the confidence of frontline workers, victims, and the public. Focusing on the secure access software design will allow the correct person to do their work while the information stays hidden.

<u>Question 6</u>

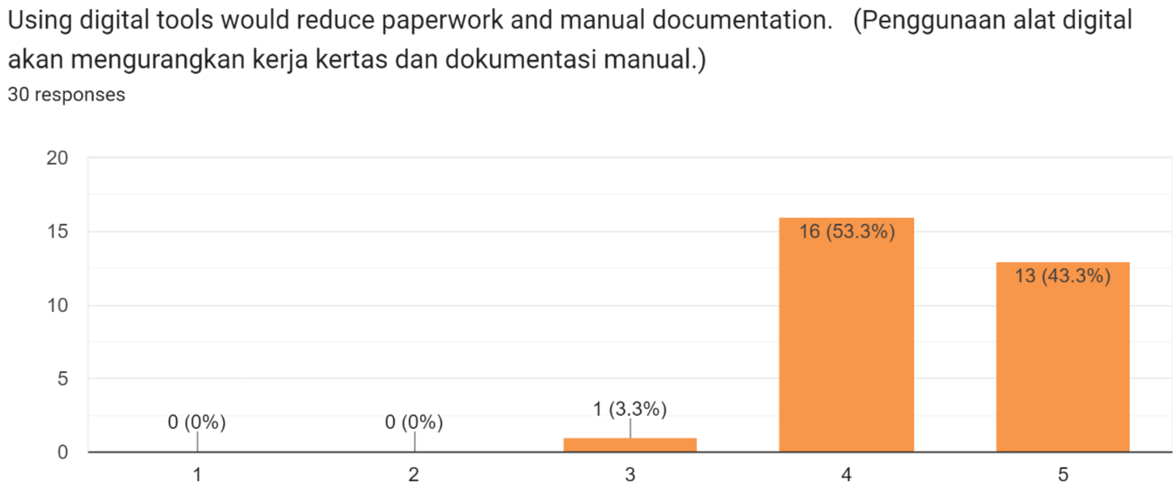


Figure 37: Respondent Agreement on the Importance of Digital Tools for Reducing Paperwork and Manual Documentation

According to the research, almost all respondents agreed that using digital tools may greatly lessen the workload associated with paperwork and physical record-keeping. Another 43.3% gave this feature the highest possible rating of 5, while more than half gave it a rating of 4. There was broad support for the system's ability to simplify administrative duties, as evidenced by the fact that just one respondent was neutral and that none disapproved. Thus, stakeholders consider digitalization not only as a comfort and an option to daily processes but also as a necessary tool to improve company operations and thereby, it is evident to them that digitalization is a must. By reducing the number of manual activities, the proposed system can decrease errors made by the human workforce, save time, and allow the frontline staff to be more involved in meaningful and influential tasks. Deploying solid digital documentation functionalities shall be the first step forward to process optimization and the effective functioning of the case management system.

<u>Question 7</u>

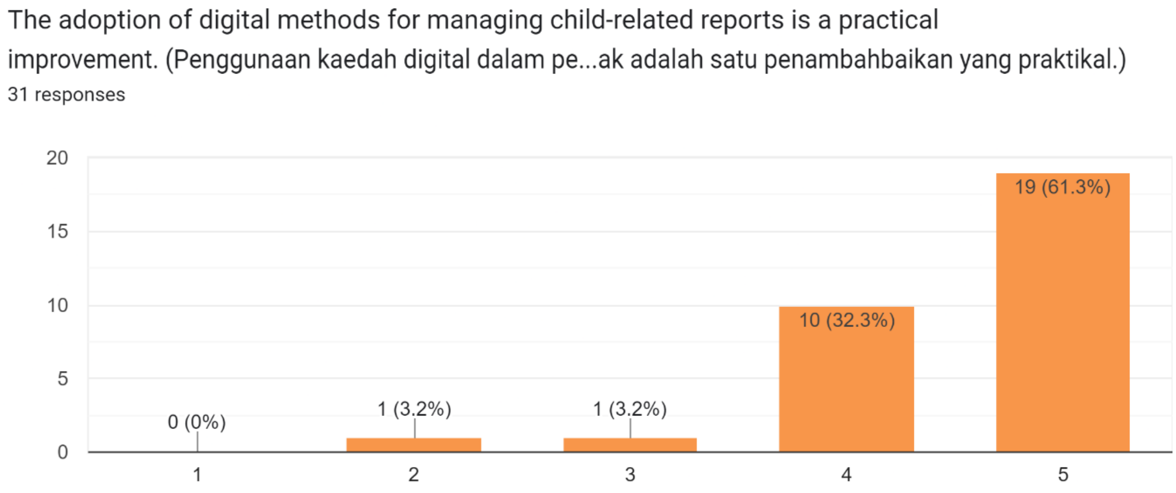


Figure 38: Respondent Agreement on the Adoption of Digital Methods as a Practical Improvement for Managing Child-Related Reports

The data presented in the figure evidently indicates that the stakeholders overwhelmingly believe using digital methods to handle child-related reports is a practical and advantageous step forward. To be exact 61.3% of the respondents gave it the highest points possible which is 5. On the other hand, 32.3% of the respondents gave it 4 points. A negligible percentage of 6.4 raised the disagreement tones which makes it clear that most of the digitalization of the current practice. The response pattern suggests that stakeholders view digital adoption as a realistic, possible enhancement with obvious benefits for day-to-day operations, rather than just as a theoretical upgrade. According to SinDA, integrating digital technologies is not only an innovation in and of itself, rather it is a critical step in addressing the tangible demands of users and bringing system operations into line with the reality of child welfare reporting and case management on the ground.

Question 8

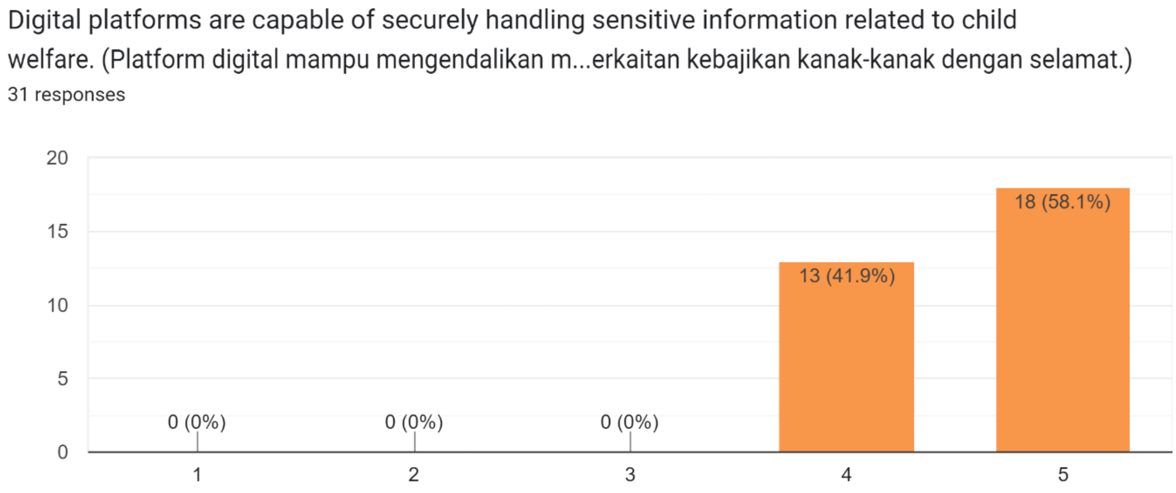


Figure 39:

The figure illustrates that all participants in the survey were certain that digital platforms are capable of safely managing confidential child welfare data. To be precise, 41.9% of the respondents rated this statement at 4, while 58.1% gave the highest rating of 5. Neutral or disagreeing responses were not given by any participant, hence showing unanimous agreement among the stakeholder group. The emphasis of this result is on the stakeholders' confidence in the capabilities of digital systems to meet the security and confidentiality requirements, which is key to handling highly sensitive and critical information. This confidence allows the proposed system to lay a strong foundation for implementing advanced security protocols, where data protection is seen not only as a matter of compliance but also as a vital support mechanism to maintain user trust and protect the affected population from harm.

Question 9

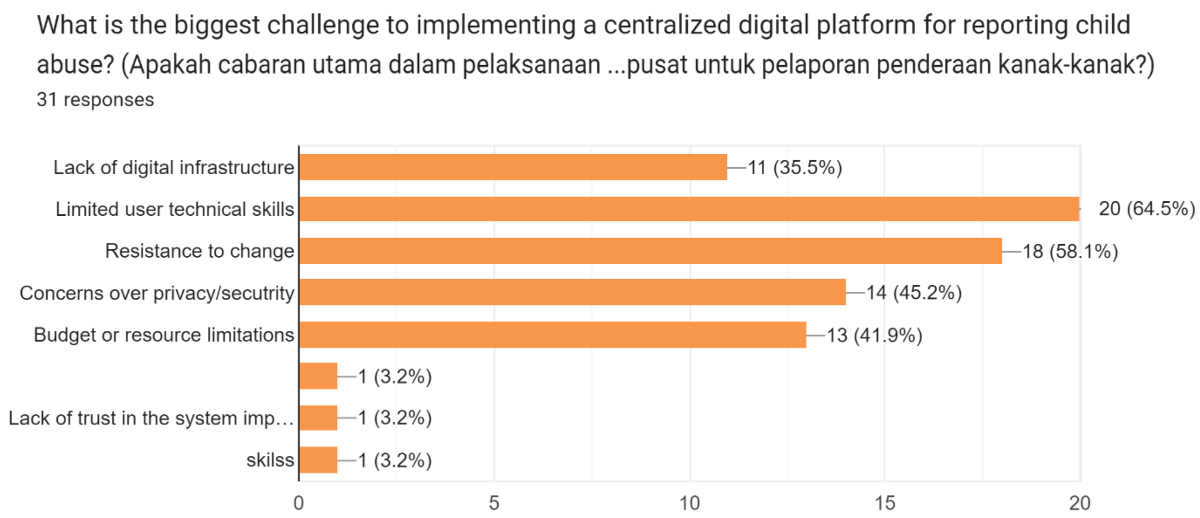


Figure 40: Respondent Views on the Biggest Challenges to Implementing a Centralized Digital Platform for Reporting Child Abuse

As depicted in the figure, the major barriers to the implementation of a centralized digital platform for reporting child abuse, according to the stakeholders, were the lack of users' technical skills and the resistance to change. Other challenges worth noting include fears of privacy and security, budget or resource constraints, and a lack of digital infrastructure. A minimal percentage of the respondents which is 3.2% each only touched on issues such as the absence of their own belief in the system followed by some other insignificant factors. The results bring to the forefront that technical updates are inevitable. There is greater hindrance in the form of human and organizational facets, especially the user's preparedness and willingness to integrate new systems. Concerning the proposed system roll-out, it entails that in addition to the creation of strong technical solutions, an equal amount of attention should be directed toward the provision of sufficient training, effective change management, and addressing user concerns regarding privacy, security, and resources. Dealing with these issues in a comprehensive way will be the key to ensuring these platforms are successful and supported in the long term.

Summary of the findings

The survey results provided valuable insights into the perceptions, expectations, and experiences of key stakeholders involved in child welfare reporting in Malaysia. Drawing on the analysis across demographic profiles, current system challenges, and the reception of proposed features, several dominant themes emerged that will directly shape the development priorities of the SinDA (Integrated Child Welfare System) platform.

Diverse Stakeholder Profiles

According to the findings of the demographic study, the gender representation was balanced as the majority 61.3 percent of the participants were male, 35.5% were female, and 3.2% were not willing to disclose their gender. Involving stakeholders comprised police officers, social workers, NGOs, healthcare providers, students, and public members, which was a secure way to ensure a wide range of voices. The younger respondents mainly concentrated on features such as mobile access and real-time updates; on the other hand, the senior professionals equally concentrated on the interest of procedural integrity and data security. This diversity points out the significant role of both a role-sensitive and user-friendly design which caters to both technical and non-technical users.

System Inefficiencies and Trust Gaps

The current reporting system was found to be badly criticized in the survey. On efficiency of the system, which is at question 1 section B, 77.4% of the respondents says that the system was badly rated. 38.7% of them rate it 2 and 38.7% of them rate it 3 mentioning the reasons like delays and bureaucratic barriers. In the case of anonymous reporting, just 12.9% which is 4 respondents showed to be satisfied where they have rated it as 4. Then, 48.4% were neutral and 32.2% of people were against it. This clearly reflects the major concern that people have about their privacy and safety. Regarding fragmented tools, 25.8% of respondents were indifferent, while 67.7% of respondents, or 21 people, agreed or strongly agreed that agencies use disjointed systems. Additionally, 83.8% agreed or strongly agreed that case outcomes are adversely affected by communication delays. Another significant obstacle was public awareness, with 77.4% of respondents agreeing or strongly agreeing that a lack of knowledge prevents prompt reporting and intervention. All of these results point to how urgently a comprehensive, open, and effective reporting platform is needed.

Stakeholder Support for Proposed Features

Support for the suggested system's characteristics was overwhelmingly positive. To increase productivity and cut down on redundant work, almost all respondents supported the establishment of a centralised digital platform. All 31 respondents emphasised the importance of anonymous reporting, emphasising its need to safeguard witnesses and victims. With over 93% (29 respondents) saying it would enhance accountability and follow-ups digital case monitoring was highly liked. Additionally, stakeholders stressed the value of role-based access restrictions to protect sensitive case data and real-time communication capabilities to promote interagency collaboration. These findings indicate that the suggested system's design priorities are rather obvious.

The overall results of the survey suggest that the stakeholders are mainly inclined to the implementation of a safe, open, and modern technological reporting system for child welfare. The observations accentuate the significance of prioritizing the SinDA platform's user-centric design, robust privacy safeguards, and fast inter-agency collaboration. It is important to note that other than the company fixing existing system flaws, the features they introduce such as digital tracking, anonymous reporting, and safe cooperation would enhance both public trust and professional ethics. These insights will direct the next phases of stakeholder participation and system development.

Chapter 4: Design and Implementation

4.1 Introduction

The proposed system concept and implementation are presented in this chapter, which translates the results of the problem background, literature study, and methodology into a workable digital solution. The chapter focusses on how the fundamental components of the system, including inter-agency communication, organised case management, secure role-based access, and anonymous reporting, were conceived and created to meet the issues found in Malaysia's child protection ecosystem.

The overall system architecture, functional modelling, and data modeling which includes entity-relationship diagrams (ERDs), activity diagrams, use case diagrams, and use case specifications are introduced in the design part. These design objects offer a clear implementation roadmap and guarantee that the suggested solution is in line with the project's goals. Furthermore, navigation flows and interface wireframes are provided to show how users such as administrators, social workers, healthcare providers, law enforcement, and public reporters will engage with the system in a safe and convenient way.

The practical development process, including the technologies, frameworks, and tools used, is then covered in the implementation section. This entails integrating DataTables for dynamic case and user administration, role-based authentication, safe data processing, and Laravel for backend functionality. To confirm that the system satisfies the required functionality and matches the planned design, screenshots of the generated modules are given. This chapter bridges the gap between the deployed solution and theoretical requirements by integrating design documentation with implementation data. It shows how the goals stated in previous chapters were achieved by turning the suggested system into a functional prototype that facilitates safe reporting, effective case monitoring, and coordinated action among child welfare partners.

4.2 Design

4.2.1 Use Case Diagram

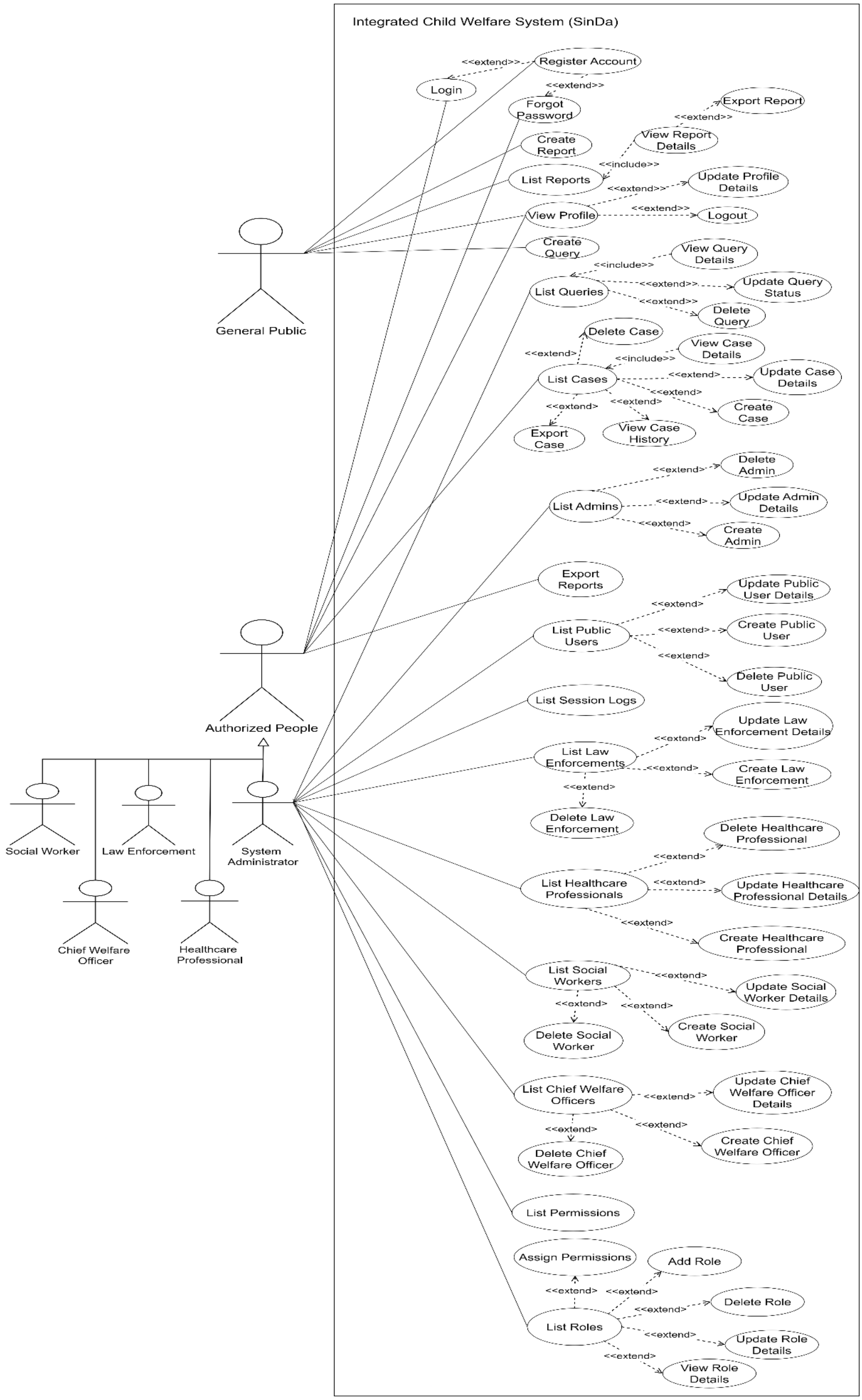


Figure 41: SinDa Use Case Diagram

4.2.2 Use Case Specification

4.2.2.1 UC1_Register Account

Use Case	**UC1_Register Account**
Description	Allow public user creates a new by providing their username, email address, and password.
Actors	Public User
Preconditions/Dependency	- User is not currently logged into the system. - Database is accessible and operational.
Postconditions	- New user account created in the database. - User is redirected to the sign-in page.
Standard Process	1. System displays Landing page. 2. Users select "Login" button. 3. System redirects user to the Sign In page. 4. User selects "Sign Up" button. 5. System redirects user to Sign Up page. 6. User fills in the registration form with username, email address, password and password confirmation. 7. User accepts Terms & Condition checkbox. 8. User clicks "Sign Up" button. 9. System receives the inputs. 10. System validates all input fields. 11. System checks if email is unique in the database. 12. System redirects user to sign-in page with success message.
Alternative Process	<u>8a. User wants to go to sign in page</u> 8a.1. User select "Sign In" button. 8a.2. System redirect user back to Sign In page. <u>8b: User wants to go back to landing page</u> 8b.1. User select "Back to Home" button. 8b.2. System redirects use back to Landing page.

Exception Flow	10a. Invalid Form Data 10a.2. Client-side validation shows error message. 10a.3. User can proceed to Step 6 to fill the form again. 11a. Email already exist 11a.1. System displays error message informing the email is already registered. 11a.2. User can proceed to Step 4 to fill the form again.

Table 10: UC1_Register Account

4.2.2.2 UC2_Login

Use Case	**UC2_Login**
Description	This use case allowed a registered user of SinDa to sign in into the system.
Actors	All Actors
Precondition/Dependency	- User has a valid account in the system - User is not currently logged in - Database is accessible and operational - User has access to the sign-in page
Postcondition	User is successfully authenticated and logged in.
Standard Process	1. System displays Landing page. 2. Users select "login" button. 3. System redirects user to the Sign In page. 4. User enters a valid email address and password. 5. User clicks "Sign In" button. 6. System receives the inputs. 7. System validates all input fields. 8. System validates the credentials against the database. 9. If valid, system authenticates the user and redirects them to their dashboard based on role.
Alternative Process	4a. User want to reset password 4a.1. User redirects to UC3_Forgot Password step 2. 4b. User wants to go back to landing page

	4b.1. User select “Back to Home” button. 4b.2. System redirects use back to Landing page. <u>4c User want to go back to Sign Up page</u> 4c.1 User click on “Sign Up” button. 4c.2. System redirects user to Sign Up page.
Exception Flow	<u>7a. Missing Required Fields</u> 7a.1 System displays validation error message. 7a.2. User can proceed to Step 4 to fill the form again. <u>8a. Invalid Credentials</u> 8a.1. System displays error message. 8a.2. User can proceed to Step 4 to fill the form again.

Table 11: UC2_Login

4.2.2.3 UC3_Forgot Password

Use Case	**UC3_Forgot Password**
Description	This use case allows a registered user who has forgotten their password to request a password reset link via email
Actors	All Users
Precondition/Dependency	- User has a valid account in the system. - User has access to the sign-in page. - Database is accessible and operational. - User in Sign In page.
Postcondition	- Password reset link is sent to user's email. - User can successfully reset their password.
Standard Process	1. User clicks on “Forgot Password?” button. 2. System opens password reset modal dialog. 3. User enters their registered email address. 4. User clicks “Send reset link” button. 5. System validates the email format. 6. System generates secure reset token for the email. 7. System store token in database with 60-minute expiry.

	8. System send password reset email. 9. System display success message. 10. User click on the link in their email and reset their password.
Alternative Process	N/A
Exception Flow	5a. Invalid Email Format 5a.1. System displays validation error message. 5a.2. User can renter their email from Step 3.

Table 12: UC3_Forgot Password

4.2.2.4 UC4_Create Report

Use Case	**UC4_Create Report**
Description	This use case allows a public user (registered or anonymous) submits a child protection report to the system by providing detailed information about a suspected case of child abuse, neglect, or exploitation.
Actors	Public User
Precondition/Dependency	- System is operational and database is accessible. - User has access to the report submission page.
Postcondition	- Report is successfully created and stored in the database for review.
Standard Process	1. System displays Landing page 2. User selects "Report Abuse" button. 3. System redirects user to the Report page. 4. User fills in the report form with victim information, incident details, and optional evidence. 5. User confirms information accuracy by checking consent checkbox. 6. User clicks "Submit Report" button. 7. System receives the inputs. 8. System validates all required fields and data formats. 9. System creates new report record in database. 10. System displays success message to user.

Alternative Process	6a. User wants to reset the form 6a.1. User clicks "Reset" button. 6a.2. System clears all form fields.
Exception Flow	8a. Missing Required Fields 8a.1. System displays validation errors. 8a.2. User can proceed to Step 4 to fill the form again or proceed to Step 6a.

Table 13: UC4_Create Reports

4.2.2.5 UC5_List Reports

Use Case	**UC5_List Reports**
Description	This use case allows a public user to view a list of their own submitted reports through the landing page interface.
Actors	Public User
Precondition/Dependency	- User is authenticated and logged into the system. - User has submitted at least one report. - Database is accessible and operational.
Postcondition	User successfully views a list of their own submitted reports with status and details.
Standard Process	1. System displays Landing page. 2. User clicks "My Reports" button. 3. System redirects user to My Reports page 4. System queries database for reports that was reported by the authenticated user. 5. System displays list of user's own reports.
Alternative Process	N/A
Exception Flow	4a. No Reports Found 4a.1. System displays "No Reports Found" message. 4a.2. System shows "Create Your First Report" button.

Table 14: List Reports

4.2.2.6 UC6_View Report Details

Use Case	**UC6_View Report Details**

Description	This use case allow user to view their own submitted report in a modal.
Actors	Public User
Precondition/Dependency	- User is authenticated and logged into the system. - User has submitted at least one report. - Database is accessible and operational. - User is in My Reports page.
Postcondition	User successfully views detailed information about a specific report.
Standard Process	1. User clicks "View Details" button on a report row. 2. System fetches the details of the chosen report from database. 3. System opens modal dialog with detailed report information.
Alternative Process	3a. User wants to export report 3a.1 Flow to UC6_Export Report.
Exception Flow	N/A

Table 15: UC6_View Reports

4.2.2.7 UC7_Export Report

Use Case	**UC7_Export Report**
Description	Allow user to export their submitted report.
Actors	Public User
Precondition/Dependency	- User is authenticated and logged into the system. - User is viewing report details in modal dialog in My Reports page.
Postcondition	User successfully downloads a PDF file containing the complete report information.
Standard Process	1. User clicks "Export Report" button. 2. System fetches complete report data from database. 3. System generates PDF document with report data. 4. System automatically downloads PDF to user's device. 5. System show successful message.

Alternative Process	N/A
Exception Flow	N/A

Table 16: UC7_Export Reports

4.2.2.8 UC8_View Profile Details

Use Case	UC8_View Profile Details
Description	Allows authenticated users to view their own profile information.
Actors	All Users
Precondition/Dependency	User must be authenticated and logged into the system.
Postcondition	- User profile information is displayed in a readable format. - Profile data is presented according to user role.
Standard Process	1. User chooses to access their profile: 1a. Public User • 1a.1. Click on their email shown in the navigation bar. • 1a.2. System redirects them to the Profile page. 1b. Authorized User • 1b.1. Clicks on their avatar. • 1b.2. System shows a dropdown. • 1b.3. User selects “Profile Details” button”. • 1b.4. System redirects them to the Profile page. 2. System fetches current user id. 3. System retrieves the user profiles information from the database based on the current user id. 4. System displays the profile details according to the user role.
Alternative Process	4a. User want to update their profile 4a.1. User flow to UC10_Update Profile. 4b. User wants to return to previous page 4b.1. User press “Cancel” button.

	5b.2. System redirects them to the associate page.
Exception Flow	N/A

Table 17: UC8_View Profile

4.2.2.9 UC9_Update Profile Details

Use Case	**UC9_Update Profile Details**
Description	Allow authenticated users to update their profile information including personal details, contact information, and role-specific fields.
Actors	All Users
Precondition/Dependency	- User is currently logged into the system. - Database is accessible and operational. - User currently in Profile page.
Postcondition	User profile information is successfully updated in the database.
Standard Process	1. User modifies desired profile fields. 2. User clicks "Save Changes" button. 3. System validates all input fields according to role-specific rules. 4. System retrieves user ID. 5. System cleans and formats data. 6. System updates profile based on the user id. 7. System displays success message.
Alternative Process	<u>2a. User wants to cancel</u> 2a.1. User press "Cancel" button. 2a.2. System redirects them to the previous page.
Exception Flow	<u>3a. Validation Errors</u> 3a.1. System displays validation error messages. 3a.2. User may retry by going to Step 1.

Table 18: UC9_Update Profile

4.2.2.10 UC10_Logout

Use Case	**UC10_Logout**

Description	Allow authenticated users to securely log out of the system, terminating their session and clearing authentication data.
Actors	All Users
Precondition/Dependency	User is currently logged into the system
Postcondition	- User session is terminated and cleared. - User is redirected to appropriate page based on their role. - Logout event is recorded in activity logs.
Standard Process	1. User initiates logout 1a. Public Users • 1a.1. Clicks "Log out" in the top navigation. 1b. Authorized User • 1b.1. User clicks avatar. • 1b.2. System show dropdown. • 1b.3. User select "Log out". 2. System verifies a valid session exists. 3. System invalidates the session on the server. 4. System redirects the user based on the appropriate page.
Alternative Process	N/A
Exception Flow	N/A

Table 19: UC10_Logout

4.2.2.11 UC11_ List Queries

Use Case	**UC11_List Queries**
Description	Allow authorized users to view and browse contact queries in a structured list format.
Actors	Authorized Users (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view queries.
Postcondition	Contact queries are displayed in a structured list format.
Standard Process	1. User selects "Contact Queries" from side bar.

	2. System redirects users to Contact Queries Management page. 3. System retrieves all queries from the database with role-based filtering. 4. System displays the queries in a table.
Alternative Process	4a. View Query Details 4a.1. User flow to UC12_View Query 4b. Delete Query 4b.1. User flow to UC14_Delete Query
Exception Flow	N/A

Table 20: UC11_List Queries

4.2.2.12 UC12_ View Query Details

Use Case	**UC12_View Query Details**
Description	Allow authorized users to view detailed information of a specific contact query.
Actors	Authorized Users (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view queries. - Contact query exists in the system. - User in contact query management page.
Postcondition	Detailed contact query information is displayed to the user in a comprehensive view format.
Standard Process	1. User clicked the 3-dots icon 2. System display action dropdown. 3. User clicks "View Details" action button for a specific query from the Contact Queries list. 4. System fetches the selected query id. 5. System retrieves the complete contact query information from database. 6. System redirect user to Contact Query Details page. 7. System displays the selected contact query information.

Alternative Process	<u>7a. Update Query Status</u> 7a.1. User flow to UC13_Update Query Status. <u>7b. User want to go to Contact Queries Management</u> 7b.1. User click the back arrow icon. 7b.2. System redirects user to Contact Queries Management page.
Exception Flow	N/A

Table 21: UC_12 View Query

4.2.2.13 UC13_ Update Query Status

Use Case	**UC13_Update Query Status**
Description	Allow authorized users to update the status of contact queries from pending to in-progress or resolved.
Actors	Authorized Users (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to edit queries. - User in Contact Query Details page.
Postcondition	Contact query status is successfully updated in the database and reflected in the system.
Standard Process	1. User select any status button. 2. System validates the status change request. 3. System fetches current query IDs. 4. System updates the query status in the database. 5. System directs user to Query Management Page. 6. System shows successful message
Alternative Process	N/A
Exception Flow	N/A

Table 22: UC13_Update Query Status

4.2.2.14 UC14_ Delete Query

Use Case	**UC14_Delete Query**
Description	Allow authorized users to delete contact queries.

Actors	Authorized Users (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to edit queries. - User in Contact Queries Management page.
Postcondition	Contact query is permanently removed from the system.
Standard Process	1. User clicks 3-dots icon. 2. System displays the action dropdown. 3. User click the "Delete" button. 4. System displays a confirmation dialog. 5. User select "Delete" button. 6. System fetches the selected query ID. 7. System removes the query from the database 8. System display success message
Alternative Process	5a. Cancel Deletion 5a.1. The user clicked on "Cancel" button. 5a.2. The system closes the confirmation modal.
Exception Flow	N/A

Table 23: UC14_Delete Query

4.2.2.15 UC15_ List Cases

Use Case	**UC15_List Cases**
Description	Allow authorized users to view and browse cases in a structured list format.
Actors	Authorized Users
Precondition/Dependency	• User is currently logged into the system. • User has appropriate permissions to view cases. • System must be operational and database accessible.
Postcondition	User successfully views a structured list of cases.
Standard Process	1. User selects "View Cases" from side bar. 2. System redirects users to Case Management page. 3. System retrieves all cases from the database with role-based filtering. 4. System displays the cases in a table.

Alternative Process	4a. Create Case 4a.1. User flow to UC16_Create Case. 4b. View Case Details 4b.1. User flow to UC17_Create Case Details. 4c. Update Case Details 4c.1. User flow to UC18_Update Case Details. 4d. Delete Case 4d.1. User flow to UC19_Delete Case. 4e. View Case History 4e.1. User flow to UC20_View Case History.
Exception Flow	N/A

Table 24: UC15_List Cases

4.2.2.16 UC16_Create Case

Use Case	**UC16_Create Case**
Description	Allow authorized users to create new cases in the system with all required case information and details.
Actors	Authorized User
Precondition/Dependency	- User must be authenticated and authorized to create case. - User must have appropriate permissions to add new case. - User in Case Management page.
Postcondition	New case is successfully created and stored in the database.
Standard Process	1. User clicks on “Add Case” button. 2. System displays a “Add New Case” modal. 3. User fills in the case form with case title, description, category, priority, and assignment details.

	4. User selects appropriate case type, department, and responsible personnel. 5. User clicks "Create Case" button. 6. System receives the inputs. 7. System validates all required fields and data formats 8. System creates new case record with unique case ID in database. 9. System displays success message to user. 10. System closes the modal.
Alternative Process	5a. User want to cancel 5a.1. User clicks “Cancel” button. 5a.2. System closes the modal without saving any data.
Exception Flow	7a. Invalid or missing Fields 7a.1. System display validation error message. 7a.2. User can proceed to Step 3 to fill the form again or proceed to Step 5a.uc17

Table 25: UC16_Create Case

4.2.2.17 UC17_ View Case Details

Use Case	**UC17_View Case Details**
Description	This use case allows authorized users to view comprehensive case details including case information, status, priority, assignments, evidence files, and case history through a dedicated case details page.
Actors	Authorized User
Precondition/Dependency	- System is operational and database is accessible. - Case exists in the system and has appropriate role permission. - User in Case Management page.
Postcondition	User successfully views complete case details with access to all authorized case information, evidence files and case messaging.
Standard Process	1. User clicks on the 3-dot icon.

	2. System displays the action dropdown. 3. User clicks "View Case Details". 4. System fetches selected cases ID. 5. System retrieves case data, assignments, and related information from database. 6. System redirects user to case details page. 7. System displays comprehensive case details page.
Alternative Process	7a. Case Messaging 7a.1. User flow to UC18_Case Messaging. 7b. Update Case Details 7b.1. User clicks green edit icon. 7b.2. User flow to UC19_Update Case Details Step 7. 7c. Export Case 7c.1. User click on "Export PDF" button. 7c.2. User flow to UC22_Export Case Step4.
Exception Flow	N/A

Table 26: UC17_View Case Details

4.2.2.18 UC18_ Case Messaging

Use Case	**UC18_Case Messaging**
Description	his use case allows authorized users to send, receive, and view secure messages within a specific case context.
Actors	Authorized User
Precondition/Dependency	- System is operational and database is accessible. - User in View Case Details page. - Case has at least one assigned personnel. - Case status is not closed.
Postcondition	User successfully sends/receives messages within the case.
Standard Process	1. User clicks on the message icon. 2. System navigates user to secure messages section. 3. System loads existing message thread for the case

	4. User composes new message in the message input field 5. User clicks “Send Message” button. 6. System validates message content. 7. System creates new message record in database 8. System refreshes message thread to display new message.
Alternative Process	N/A
Exception Flow	6a. Message exceeds 5000 characters 6a.1. System prevents submission. 6a.2. User can retry again by going Step 4.

Table 27: UC18_Case Messaging

4.2.2.19 UC19_ Update Case Details

Use Case	**UC19_Update Case Details**
Description	This use case allows authorized users to modify existing case information including case details, status, priority, assignments, and evidence files.
Actors	Authorized User
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to edit cases. - User in Case management page.
Postcondition	Case information is successfully updated and stored in the database.
Standard Process	1. User clicks on the 3-dot icon on any case. 2. System displays the action dropdown. 3. User clicks "Edit" button. 4. System fetches selected case ID. 5. System displays “Edit Case” modal. 6. System retrieves current case data from database. 7. System populates the edit form. 8. User modifies the required case information in the form fields.

	9. User clicks "Update Case" button. 10. System receives form inputs. 11. System validates fields. 12. System updates the case record in the database with new information. 13. System logs the update action with user details and timestamp. 14. System displays success message.
Alternative Process	9a. User wants to cancel the case update 9a.1. User clicks "Cancel" button in modal footer 9a.2. System closes the modal without saving any changes and returns to previous state
Exception Flow	11a. Invalid data fields 11a.1. System displays validation errors message. 11a.2. User can proceed to Step 8 to modify the form again or proceed to Step 9a.

Table 28: UC19_Update Case Details

4.2.2.20 UC20_ Delete Case

Use Case	**UC20_Delete Case**
Description	This use case allows authorized users to permanently remove cases from the system.
Actors	Authorized User
Precondition/Dependency	• System is operational and database is accessible. • User is authenticated and has appropriate permissions to delete cases. • User is in Case Management page.
Postcondition	Case is permanently removed from the system along with all associated evidence files, assignments, and related data.
Standard Process	1. User clicks the 3-dot icon on any case row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal.

	5. User clicks "Delete Case" button to confirm the deletion. 6. System receives delete request with the case ID. 7. System permanently removes the case record from the database. 8. System display success message to user.
Alternative Process	5a. User wants to cancel the case deletion 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 29: UC20_Delete Case

4.2.2.21 UC21_ View Case History

Use Case	**UC21_View Case History**
Description	This use case allows authorized users to view the complete audit trail and history of changes made to a specific case.
Actors	Authorized User
Precondition/Dependency	• System is operational and database is accessible. • User is authenticated and has appropriate role permissions. • User is in Case Management page.
Postcondition	User successfully views complete case history with detailed timeline of all changes, modifications, and actions performed on the case, including user information and timestamps.
Standard Process	1. User clicks the 3-dot icon. 2. System displays the action dropdown menu. 3. User clicks "View History" button from the dropdown menu. 4. System displays "View History" modal. 5. System retrieves case history records from the database. 6. System populates the modal with comprehensive history timeline.

Alternative Process	6a. User wants to close the modal 6a.1. User click on “Close” button. 6.a.2 System closes the modal.
Exception Flow	N/A

Table 30: UC21_View Case History

4.2.2.22 UC22_ Export Case

Use Case	**UC22_Export Case**
Description	This use case allows authorized users to export individual case information to PDF format through a direct download process
Actors	Authorized User
Precondition/Dependency	• System is operational and database is accessible. • User is authenticated and has appropriate permissions to export cases. • User is in Case Management page.
Postcondition	Case information is successfully exported to PDF format and downloaded to user's device.
Standard Process	1. User clicks the 3-dot icon. 2. System displays the action dropdown menu. 3. User clicks "Export PDF" button from the dropdown menu. 4. System retrieves case data from database 5. System generated PDF document with case data. 6. User receives PDF file in their download folder.
Alternative Process	N/A
Exception Flow	N/A

Table 31: UC22_Export Case

4.2.2.23 UC23_ List Admins

Use Case	**UC23_List Admins**
Description	This use case allows authorized users to view and browse admins in a structured list format.
Actors	Authorized User (System Administrator)

Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view admins. - System must be operational and database accessible.
Postcondition	User successfully views a structured list of admins.
Standard Process	1. User clicks on "Admins" menu item. 2. System navigates user to Admin Management page 3. System retrieves all admins from the database with role-based filtering. 4. System renders the admins in a table.
Alternative Process	<u>4a. Create Admin</u> 4a.1. User flow to UC24_Create Admin. <u>4b. Update Admin Details</u> 4b.1. User flow to UC25_Update Admin Details. <u>4c. Delete Admin</u> 4c.1. User flow to UC26_Delete Admin.
Exception Flow	N/A

Table 32: UC23_List Admins

4.2.2.24 UC24_ Create Admin

Use Case	**UC24_Create Admin**
Description	Allow authorized users to create new admin in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User must be authenticated and authorized to create admin. - User must have appropriate permissions to add new admin - User in Admin Management page.
Postcondition	New admin is successfully created and stored in the database.
Standard Process	1. User clicks on “Add” button. 2. System displays a “Add Admin” modal.

	3. User fills in the admin form including the account details, admin profile and contact information. 4. User clicks "Create" button. 5. System receives the inputs. 6. System validates all required fields. 7. System checks database for unique constraints. 8. System creates new admin record in the database. 9. System displays success message to user.
Alternative Process	<u>4a. User want to cancel</u> 4a.1. User clicks "Cancel" button. 4a.2. System closes the modal without saving any data.
Exception Flow	<u>6a. Invalid Fields</u> 6a.1. System display validation errors message. 6a.2. User can proceed to Step 3 to fill the form again. <u>8a. Duplicate email</u> 8a.1. System displays error message to user. 8a.2. user return o Step 3 or proceeds to Step 4a.

Table 33: UC24_Create Admin

4.2.2.25 UC25_ Update Admin Details

Use Case	**UC25_Update Admin Details**
Description	Allow authorized users to modify existing admin in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to modify admin details. - User in Admin Management page.
Postcondition	Admin details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a row. 2. System displays the action dropdown menu.

	3. User clicks "Edit" button. 4. System fetches selected admin ID. 5. System retrieves the selected admin data from the database. 6. System displays the "Edit Admin" modal. 7. System populates the form with selected admin details. 8. User modifies any required details in the form fields. 9. User clicks "Save Changes" button. 10. System validates all required fields. 11. System updates the admin record in the database. 12. System displays success message to the user.
Alternative Process	<u>9a. User wants to close the modal</u> 9a.1. User clicks “Close” button. 9a.2. System closes the modal without saving any data.
Exception Flow	<u>10a. Invalid data fields</u> 10a.1. System display error message. 10a.2. User can proceed to Step 8 to modify any field again.

Table 34: UC25_Update Admin Details

4.2.2.26 UC26_ Delete Admin

Use Case	**UC26_Delete Admin**
Description	This use case allows authorized users to permanently remove an admin from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete admins. - User is in Admin Management page.
Postcondition	Selected Admin is permanently removed from the system.
Standard Process	1. User clicks the 3-dot icon on any admin row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal.

	5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the admin ID 7. System permanently removes the admin record from the database. 8. System displays success message to user.
Alternative Process	3a. Cancel the case deletion 3a.1. User clicks "Cancel" button. 3a.2. System closes the modal.
Exception Flow	N/A

Table 35: UC26_Delete Admin

4.2.2.27 UC27_ Export Reports

Use Case	**UC26_Export Report**
Description	This use case allows authorized users to export comprehensive system data including cases, users, contact queries, and session logs from the admin dashboard in multiple formats.
Actors	Authorized User
Precondition/Dependency	- System is operational and database is accessible - User is authenticated and has appropriate permissions - User is in Admin Dashboard page
Postcondition	Selected data is successfully exported in the chosen format (CSV or PDF) and downloaded to user's device.
Standard Process	1. User clicks on "Export Reports" dropdown button. 2. System displays dropdown menu with export options. 3. User select specific export option. 4. System applies role-based data filtering based on user permissions. 5. System retrieves filtered data from database 6. System generates export file based on selected format. 7. System triggers automatic download of the generated file. 8. User receives export file in their downloads folder.
Alternative Process	N/A

Exception Flow	N/A

Table 36: UC27_Export Reports

4.2.2.28 UC28_ List Public Users

Use Case	**UC27_List Public Users**
Description	This use case allows authorized users to view and browse public users in a structured list format.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view public users. - User currently in Dashboard. - System must be operational and database accessible.
Postcondition	User successfully views a structured list of public users.
Standard Process	1. User clicks on "Public Users" menu item. 2. System navigates user to the Public User Management page 3. System requests all public users from the database with role-based filtering. 4. Database returns all public users records to system with role-based filtering. 5. System renders the public users in a structured table.
Alternative Process	<u>4a. Create Public User</u> 4a.1. User flow to UC29_Create Public User. <u>4b. Update Public User Details</u> 4b.1. User flow to UC30_Update Public User. <u>4c. Delete Public User</u> 4c.1. User flow to UC31_Delete Public User.
Exception Flow	N/A

Table 37: UC28_List Public Users

4.2.2.29 UC29_ Create Public User

Use Case	**UC29_Create Public User**
Description	Thise use case allow authorized users to create new public user accounts in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User must be authenticated and authorized to create public users. - User must have appropriate permissions to add new public users. - User is on the Public User Management page.
Postcondition	New public user is successfully created and stored in the database.
Standard Process	1. User clicks on "Add" button. 2. System displays an "Add Public User" modal. 3. User fills in the public user form field including name, email, password and confirm password. 4. User clicks "Create" button. 5. System receives the inputs. 6. System validates all required fields. 7. System checks database for unique constraints. 8. System creates a new public user record in the database. 9. System displays success message to user.
Alternative Process	4a. User wants to cancel 4a.1. User clicks "Cancel" button. 4a.2. System closes the modal without saving any data.
Exception Flow	6a. Invalid Data Format 6a.1. System displays error message. 6a.2. User can return to Step 3 to correct the form. 7a. Duplicate Email 7a.1. System displays error message. 7a.2. User can return to Step 3 to correct the form.

Table 38: UC29_Create Public User

4.2.2.30 UC30_ Update Public User Details

Use Case	**UC30_Update Public User Details**
Description	Allow authorized users to modify existing public user details in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to modify public user details. - User is on the Public User Management page.
Postcondition	Public user details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a public user row. 2. System displays the action dropdown menu. 3. User clicks "Edit" button. 4. System fetches selected public user ID. 5. System displays the "Edit Public User" modal. 6. System retrieves selected public user data from the database. 7. System populates the form with existing details. 8. User modifies any required details in the form fields. 9. User clicks "Update" button. 10. System validates all required fields. 11. System updates existing public user record in the database. 12. System displays success message.
Alternative Process	<u>9a. Cancel the updation</u> 9a.1. User clicks “Cancel” button. 9a.2. System closes the modal without saving any changes.
Exception Flow	<u>10a. Invalid Fields</u> 10a.1. System displays validation errors message. 10a.2. User can return to Step 8 to correct details or proceed to Step 9a.

Table 39: UC30_Update User Details

4.2.2.31 UC31_ Delete Public User

Use Case	**UC31_Delete Public User**
Description	This use case allows authorized users to permanently remove a public user from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete public users. - User is on the Public User Management page.
Postcondition	Selected public user is permanently removed from the system.
Standard Process	1. User clicks the 3-dot icon on any public user row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the public user ID. 7. System permanently removes the public user record from the database. 8. System displays success message to user.
Alternative Process	<u>3a. User wants to cancel the deletion</u> 3a.1. User clicks "Cancel" button. 3a.2. System closes the modal without deleting.
Exception Flow	N/A

Table 40: UC31_Delete Public User

4.2.2.32 UC32_ List Session Logs

Use Case	**UC32_List Session Logs**
Description	This use case allows authorized users to view a list of all recorded system session logs.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible.

	- User is authenticated and has permission to access session logs. - User in Dashboard. - Session log data exists in the system.
Postcondition	User successfully views the session log records filtered by selected criteria
Standard Process	1. User clicks on "Session Logs" button on sidebar. 2. System redirects user to Session logs page. 3. System retrieves session logs records from database. 4. System displays the session log records in a table format.
Alternative Process	<u>4a. User wants to filter session logs.</u> 4a.1. User applies filters 4a.2. User clicked on "Apply Filters". 4a.3.System sends a filtered query to the database based on the selected criteria. 4a.4. Flow moves to Step 4. <u>4b. User wants to clear filter.</u> 4b.1. User clicks "Clear Filters" button. 4b.2. System resets the filter fields to default. 4b.3. Flow moves to Step 3.
Exception Flow	N/A

Table 41: UC32_List Session Logs

4.2.2.33 UC33_ List Law Enforcements

Use Case	**UC33_List Law Enforcements**
Description	This use case allows authorized users to view and browse law enforcement officers in a structured list format.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view law enforcement officers.

	- System must be operational and database accessible.
Postcondition	User successfully views a structured list of law enforcement officers.
Standard Process	1. User clicks on "Law Enforcement" menu item. 2. System navigates user to the Law Enforcement Management page. 3. System retrieves all law enforcement officer records from the database with role-based filtering. 4. System renders the law enforcement officers in a structured table.
Alternative Process	4a. Create Law Enforcement Officer 4a.1. User flow to UC34_Create Law Enforcement Officer. 4b. Update Law Enforcement Officer Details 4b.1. User flow to UC35_Update Law Enforcement Officer Details. 4c. Delete Law Enforcement Officer 4c.1. User flow to UC36_Delete Law Enforcement Officer.
Exception Flow	N/A

Table 42: UC33_List Law Enforcements

4.2.2.34 UC34_ Create Law Enforcement

Use Case	**UC34_Create Law Enforcement**
Description	Allow authorized users to create new law enforcement officers in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User must be authenticated and authorized to create law enforcement officers. - User must have appropriate permissions to add new law enforcement officers. - User in Law Enforcement Management page.

Postcondition	New law enforcement officer is successfully created and stored in the database.
Standard Process	1. User clicks on “Add” button. 2. System displays a “Add Law Enforcement” modal. 3. User fills in the law enforcement form including the account details, law enforcement profile and contact information. 4. User clicks "Create" button. 5. System receives the inputs. 6. System validates all required fields. 7. System checks database for unique constraints. 8. System creates new law enforcement record in the database. 9. System displays success message.
Alternative Process	4a. User want to cancel 4a.1. User clicks “Cancel” button. 4a.2. System closes the modal without saving any data.
Exception Flow	6a. Invalid Field 6a.1. System display validation errors message. 6a.2. User can proceed to Step 3 to fill the form again. 7a. Duplicate Email Entered 7a.1. System displays error message to user. 7a.2. User returns to Step 3.

Table 43: UC34_Create Law Enforcement

4.2.2.35 UC35_ Update Law Enforcement Details

Use Case	**UC34_Update Law Enforcement Details**
Description	Allow authorized users to modify existing law enforcement officer details in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible.

	- User is authenticated and has appropriate permissions to modify law enforcement officer details. - User in Law Enforcement Management page.
Postcondition	Law enforcement officer details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a law enforcement officer row. 2. System displays the action dropdown menu. 3. User clicks "Edit" button. 4. Fetches the selected law enforcement officer ID. 5. System displays the "Edit Law Enforcement" modal. 6. System retrieves the selected law enforcement officer data from the database. 7. System populates the form with the officer's details. 8. User modifies any required details in the form fields. 9. User clicks "Save Changes" button. 10. System validates all required fields. 11. System updates the existing law enforcement officer record in the database. 12. System displays success message to the user.
Alternative Process	<u>9a. User wants to close the modal</u> 9a.1. User clicks "Close" button. 9a.2. System closes the modal without saving any data.
Exception Flow	<u>10a. Invalid Data</u> 10a.1. System display validation error message. 10a.2. User can proceed to Step 8.

Table 44: UC35_Update Law Enforcement Details

4.2.2.36 UC36_ Delete Law Enforcement

Use Case	**UC36_Delete Law Enforcement**
Description	This use case allows authorized users to permanently remove a law enforcement officer from the system.
Actors	Authorized User (System Administrator)

Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete admins. - User is on the Law Enforcement Management page.
Postcondition	Selected law enforcement officer is permanently removed from the system.
Standard Process	1. User clicks the 3-dot icon on a law enforcement officer row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the officer's ID. 7. System validates user permissions and officer existence in the database. 8. System permanently removes the law enforcement officer record from the database. 9. System displays success message to user
Alternative Process	<u>5a. User wants to cancel the case deletion</u> 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 45: UC36_Delete Law Enforcement

4.2.2.37 UC37_ List Social Workers

Use Case	**UC37_List Social Workers**
Description	This use case allows authorized users to view and browse social workers in a structured list format.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view social workers.

	- User in Dashboard. - System must be operational and database accessible.
Postcondition	User successfully views a structured list of social workers.
Standard Process	1. User clicks on "Social Workers" menu item. 2. System navigates user to the Social Worker Management page. 3. System retrieves all social worker records from the database with role-based filtering. 4. System renders the social workers in a structured table.
Alternative Process	<u>4a. Create Social Worker</u> 4a.1. User flow to UC38_Create Social Worker. <u>4b. Update Social Worker Details</u> 4b.1. User flow to UC39_Update Social Worker Details. <u>4c. Delete Social Worker</u> 4c.1. User flow to UC40_Delete Social Worker.
Exception Flow	N/A

Table 46: UC37_List Social Workers

4.2.2.38 UC38_ Create Social Worker

Use Case	**UC38_Create Social Worker**
Description	Allow authorized users to create new social worker accounts in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is authenticated and authorized to create social workers. - User has permission to add new social workers. - User is on the Social Worker Management page.
Postcondition	New social worker is successfully created and stored in the database.
Standard Process	1. User clicks on "Add" button. 2. System displays an "Add Social Worker" modal.

	3. User fills in the form with account details, social worker profile, and organization information. 4. User clicks "Create" button. 5. System receives the inputs. 6. System validates all required fields. 7. System checks database for unique constraints. 8. System creates new social worker record in the database. 9. System displays success message to user.
Alternative Process	<u>4a. User want to cancel</u> 4a.1. User clicks “Cancel” button. 4a.2. System closes the modal without saving any data.
Exception Flow	<u>6a. Invalid field</u> 6a.1. System display validation errors message. 6a.2. User can proceed to Step 3 to fill the form again. <u>7a. Duplicate Email Entered</u> 7a.1. System displays error message to user. 7a.2. User returns to Step 3.

Table 47: UC38_Create Social Worker

4.2.2.39 UC39_ Update Social Worker Details

Use Case	**UC39_Update Social Worker Details**
Description	Allow authorized users to modify existing social worker details in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to modify law enforcement officer details. - User in Social Worker Management page.
Postcondition	Social Worker details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a social worker row.

	2. System displays the action dropdown menu. 3. User clicks "Edit" button. 4. System fetches the social worker ID. 5. System retrieves the selected social worker data from the database. 6. System returns the selected social worker data from the database. 7. System displays the "Edit Social Worker " modal. 8. System populates the form with the social worker details. 9. User modifies any required details in the form fields. 10. User clicks "Save Changes" button. 11. System validates all required fields. 12. System updates the existing social worker record in the database. 13. System displays success message to the user.
Alternative Process	10a. User wants to cancel 10a.1. User clicks “Close” button. 10a.2. System closes the modal without saving any data.
Exception Flow	11a. Invalid Fields 11a.1. System display validation errors message. 11a.2. User can proceed to Step 3 to fill the form again.

Table 48: UC39_Update Social Worker Details

4.2.2.40 UC40_ Delete Social Worker

Use Case	**UC40_Delete Social Worker**
Description	This use case allows authorized users to permanently remove a social worker from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete social workers. - User is on the Social Worker Management page.

Postcondition	Selected social worker is permanently removed from the system.
Standard Process	1. User clicks the 3-dot icon on a social worker row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the social worker ID. 7. System permanently removes the social worker record from the database. 8. System displays success message to user
Alternative Process	<u>5a. User wants to cancel the case deletion</u> 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 49: UC40_Delete Social Worker

4.2.2.41 UC41_ List Chief Welfare Officers

Use Case	**UC41_List Chief Welfare Officers**
Description	This use case allows authorized users to view and browse chief welfare officers in a structured list format.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view chief welfare officers. - User in Dashboard. - System must be operational and database accessible.
Postcondition	User successfully views a structured list of chief welfare officers.
Standard Process	1. User clicks on "Chief Welfare Officers" menu item. 2. System navigates user to the Chief Welfare Officers Management page.

	3. System retrieves all officer's records from the database with role-based filtering. 4. System renders the officers in a structured table.
Alternative Process	<u>4a. Create Chief Welfare Officer</u> 4a.1. User flow to UC42_ Chief Welfare Officer. <u>4b. Update Chief Welfare Officer Details</u> 4b.1. User flow to UC43_Update Chief Welfare Officer Details. <u>4c. Delete Chief Welfare Officer</u> 4c.1. User flow to UC44_Delete Chief Welfare Officer.
Exception Flow	N/A

Table 50: UC41_List Chief Welfare Offices

4.2.2.42 UC42_ Create Chief Welfare Officer

Use Case	**UC42_Create Chief Welfare Officer**
Description	This user case allows authorized users to create new chief welfare officer accounts in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is authenticated and authorized to create chief welfare officers. - User has permission to add new chief welfare officers. - User is on the Chief Welfare Officer Management page.
Postcondition	New chief welfare officer is successfully created and stored in the database.
Standard Process	1. User clicks on "Add" button. 2. System displays an "Add Chief Welfare Officer "modal. 3. User fills in the form with account details, chief welfare officer profile, and contact information. 4. User clicks "Create" button. 5. System receives the inputs.

	6. System validates all required fields. 7. System checks database for unique constraints. 8. System creates new Chief Welfare Officer record in the database. 9. System displays success message to user.
Alternative Process	<u>4a. User want to cancel</u> 4a.1. User clicks “Cancel” button. 4a.2. System closes the modal without saving any data. <u>7a. Duplicate Email Entered</u> 7a.1. System displays error message to user. 7a.2. User returns to Step 3.
Exception Flow	<u>6a. Invalid fields</u> 6a.1. System display validation errors message. 6a.2. User can proceed to Step 3 to fill the form again.

Table 51: UC42_Create Chief Welfare Officer

4.2.2.43 UC43_ Update Chief Welfare Officer Details

Use Case	**UC43_Update Chief Welfare Officer Details**
Description	Allow authorized users to modify existing chief welfare office details in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to modify chief welfare officer details. - User is on the Chief Welfare Officer Management page.
Postcondition	Chief Welfare details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a Chief Welfare Officer row. 2. System displays the action dropdown menu. 3. User clicks "Edit" button. 4. System fetches the selected officer’s ID.

	5. System retrieves the selected chief welfare officer data from the database. 6. System returns the selected chief welfare officer data from the database. 7. System populates the "Edit Chief Welfare Officer " modal. 8. User modifies any required details in the form fields. 9. User clicks "Save Changes" button. 10. System validates all required fields. 11. System updates the chief welfare officer record in the database. 12. System displays success message to the user.
Alternative Process	9a. User wants to close the modal 9a.1. User clicks “Close” button. 9a.2. System closes the modal without saving any data.
Exception Flow	10a. Invalid fields 10.a.1. System display validation errors message. 10a.2. User can proceed to Step 8 to fill the form again.

Table 52: UC43_Update Chief Welfare Officer

4.2.2.44 UC44_ Delete Chief Welfare Officer

Use Case	**UC44_Delete Social Worker**
Description	This use case allows authorized users to permanently remove a chief welfare officer from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete social workers. - User is on the Chief Welfare Officer Management page.
Postcondition	Selected chief welfare officer is permanently removed from the system.
Standard Process	1. User clicks the 3-dot icon on a chief welfare officer row.

	2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the chief welfare officer ID. 7. System permanently removes the chief welfare officer record from the database. 8. System displays success message to user
Alternative Process	5a. User wants to cancel the case deletion 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 53: UC44_Delete Welfare Officer

4.2.2.45 UC45_ List Healthcare Professionals

Use Case	**UC45_List Healthcare Professionals**
Description	This use case allows authorized users to view and browse healthcare professionals in a structured list format.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view healthcare professionals. - System must be operational and database accessible.
Postcondition	User successfully views a structured list of healthcare professionals.
Standard Process	1. User clicks on “Healthcare Professionals” menu item. 2. System navigates user to the Healthcare Professional Management page. 3. System retrieves all officer’s records from the database with role-based filtering. 4. System renders the healthcare professionals in a structured table.

Alternative Process	4a. Create Healthcare Professional 4a.1. User flow to UC46_Healthcare Professional. 4b. Update Healthcare Professional 4b.1. User flow to UC47_Update Healthcare Professional Details. 4c. Delete Healthcare Professional 4c.1. User flow to UC48_Delete Healthcare Professional.
Exception Flow	N/A

Table 54: UC44_List Healthcare Professionals

4.2.2.46 UC46_ Create Healthcare Professional

Use Case	**UC46_Create Healthcare Professional**
Description	This user case allows authorized users to create new healthcare professional accounts in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is authenticated and authorized to create healthcare professional. - User has permission to add new healthcare professional. - User is on the Healthcare Professional Management page.
Postcondition	New healthcare professional is successfully created and stored in the database.
Standard Process	1. User clicks on "Add" button. 2. System displays an "Add Healthcare Professional "modal. 3. User fills in the form with account details, healthcare professional profile, and contact information. 4. User clicks "Create" button. 5. System receives the inputs. 6. System validates all required fields.

	7. System checks database for unique constraints. 8. System creates new Healthcare Professional record in the database. 9. System displays success message to user.
Alternative Process	4a. User want to cancel 4a.1. User clicks “Cancel” button. 4a.2. System closes the modal without saving any data.
Exception Flow	6a. Invalid fields 6a.1. System display validation errors for missing fields. 6a.2. User can proceed to Step 3 to fill the form. 7a. Duplicate Email Entered 7a.1. System displays error message to user. 7a.2. User returns to Step 3.

Table 55: UC46_Create Healthcare Professional

4.2.2.47 UC47_ Update Healthcare Professional Details

Use Case	**UC47_Update Healthcare Professional Details**
Description	Allow authorized users to modify existing healthcare professional details in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to modify healthcare professional details. - User is on the Healthcare Professional Management page.
Postcondition	Healthcare Professional details are successfully updated and stored in the database.
Standard Process	1. User clicks the 3-dot icon on a Healthcare Professional row. 2. System displays the action dropdown menu. 3. User clicks "Edit" button. 4. System fetches selected healthcare professionals ID.

	5. System retrieves the selected healthcare professional's data from the database. 6. System returns the selected healthcare professional's data from the database. 7. System populates the form with the healthcare professional details. 8. User modifies any required details in the form fields. 9. User clicks "Save Changes" button. 10. System validates all required fields. 11. System updates the healthcare professional record in the database. 12. System displays success message to the user.
Alternative Process	<u>9a. Close the modal</u> 9a.1. User clicks "Close" button. 9a.2. System closes the modal without saving any data.
Exception Flow	<u>10a. Invalid fields</u> 10a.1. System display validation errors for data formats. 10a.2. User can proceed to Step 8 to fill the form again.

Table 56: UC47_Update Healthcare Professional Details

4.2.2.48 UC48_ Delete Healthcare Professional

Use Case	**UC48_Delete Healthcare Professional**
Description	This use case allows authorized users to permanently remove a healthcare professional from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to delete healthcare professional. - User is on the Healthcare Professional Management page.
Postcondition	Selected healthcare professional is permanently removed from the system.

Standard Process	1. User clicks the 3-dot icon on a healthcare professional row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives the delete request with the healthcare professional ID. 7. System permanently removes the healthcare professional record from the database. 8. System displays success message to user
Alternative Process	5a. User wants to cancel the case deletion 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 57: UC48_Delete Healthcare Professional

4.2.2.49 UC49_ List Permissions

Use Case	**UC49_List Permissions**
Description	This use case allows authorized users to view available system permissions.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User in Dashboard. - User has appropriate permissions to view permissions. - System must be operational and database accessible.
Postcondition	User successfully views an available list of permission available in the system.
Standard Process	1. User clicks on “View Permission” on menu item. 2. System retrieves all permission records from the database. 3. System returns all permission records from the database.

	4. System redirects user to Manage Permission page. 5. System displays the permissions in a structured list.
Alternative Process	N/A
Exception Flow	N/A

Table 58: UC49_List Permissions

4.2.2.50 UC50_ List Roles

Use Case	**UC50_List Roles**
Description	This use case allows authorized users to view available roles in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User is currently logged into the system. - User has appropriate permissions to view roles. - User in Dashboard. - System must be operational and database accessible.
Postcondition	User successfully views an available list of roles available in the system.
Standard Process	1. User clicks on "Manage Role" on menu item. 2. System retrieves all role records from the database. 3. System returns all role records from the database. 4. System redirects user to Manage Roles page. 5. System displays the roles in a structured list.
Alternative Process	5a. Add Role 5a.1. User flow to UC51_Add Role. 5b. View Role Details 5b.1. User flow to UC52_View Role Details. 5c. Update Role Details 5c.1. User flow to UC53_Update Role Details. 5d. Delete Role 5d.1. User flow to UC54_Delete Role Details.

	5e. Assign Permissions 5zd.1. User flow to UC55_Assign Permission.
Exception Flow	N/A

Table 59: UC49_List Roles

4.2.2.51 UC51_ Add Role

Use Case	**UC51_Add Role**
Description	Allow authorized users to add new cases in the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- User must be authenticated and authorized to add role. - User in Role Management page.
Postcondition	New role is successfully created and stored in the database.
Standard Process	1. User clicks on "Add Role" button. 2. System displays a "Add Role" modal. 3. User fills in the role form with role name and role description. 4. User clicks "Create Role" button. 5. System receives the inputs. 6. System validates all required fields. 7. System checks against database for unique constraint. 8. System creates new role record with unique role ID in database. 9. System displays success message to user.
Alternative Process	4a. User want to cancel 4a.1. User clicks "Close" button. 4a.2. System closes the modal without saving any data.

Exception Flow	6a. Invalid Fields 6a.1. System display validation errors message. 6a.2. User can proceed to Step 3 to fill the form again. 7a. Duplicate Role 7a.1. System display duplication errors for role. 7a.2. User can proceed to Step 3 to fill the form again.

Table 60: UC51_ Add Role

4.2.2.52 UC52_ View Role Details

Use Case	**UC52_View Role Details**
Description	This use case allows authorized users to view role details.
Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - Role exists in the system and has appropriate role permission. - User in Role Management page.
Postcondition	User successfully views role details page.
Standard Process	1. User clicks on the 3-dot icon. 2. System displays the action dropdown. 3. Users click “View”. 4. System fetches selected role ID. 5. System retrieves selected role details from database. 6. System returns selected role details from database. 7. System redirects user to role details page with populated information.
Alternative Process	N/A
Exception Flow	N/A

Table 61: UC52_View Role Details

4.2.2.53 UC53_ Update Role Details

Use Case	**UC53_Update Role Details**
Description	This use case allows authorized users to view role details.

Actors	Authorized User (System Administrator)
Precondition/Dependency	- System is operational and database is accessible. - Role exists in the system and has appropriate role permission. - User in Role Management page.
Postcondition	User successfully views role details page.
Standard Process	1. User clicks on the 3-dot icon on any role. 2. System displays the action dropdown. 3. User clicks "Edit" button. 4. System fetches selected role ID. 5. System displays “Edit Role” modal. 6. System retrieves current role data from database. 7. System populates the edit role form. 8. User modifies the role information in the form. 9. User clicks “Update Role” button. 10. System validates all required fields. 11. System checks against database for unique constraints. 12. System updates the existing role record in the database with new information. 13. System displays success message to user and closes the modal.
Alternative Process	<u>9a. User want to close edit role form</u> 9a.1. User click on “Close” button. 9a.2. System closes the edit for without saving.
Exception Flow	<u>10a. Invalid field</u> 10a.1. System display error message. 10a.2. User can continue to Step 8. <u>11a. Duplicate Role Error</u> 11a.1. System displays duplicate role error message. 11a.2. User can continue to Step 8.

Table 62: UC53_Update Role Details

4.2.2.54 UC54_ Delete Role

Use Case	**UC54_Delete Role**
Description	This use case allows authorized users to permanently remove a role from the system.
Actors	Authorized User (System Administrator)
Precondition/Dependency	• System is operational and database is accessible. • User is authenticated and has appropriate permissions to delete Role. • That role doesn't have any users on it. • User is in Role Management page.
Postcondition	Chosen role is permanently removed from the system along with all associated evidence files, assignments, and related data.
Standard Process	1. User clicks the 3-dot icon on any role row. 2. System displays the action dropdown menu. 3. User clicks "Delete" button. 4. System displays "Confirm Delete" modal. 5. User clicks "Delete" button to confirm the deletion. 6. System receives delete request with the role ID. 7. System permanently removes the role record from the database. 8. System display success message to user.
Alternative Process	<u>5a. User wants to cancel the role deletion</u> 5a.1. User clicks "Cancel" button. 5a.2. System closes the modal.
Exception Flow	N/A

Table 63: UC54_Delete Role

4.2.2.55 UC55_ Assign Permissions

Use Case	**UC55_Assign Permissions**
Description	This use case allows a system administrator to assign or remove permissions for a specific role.
Actors	Authorized User (System Administrator)

Precondition/Dependency	- System is operational and database is accessible. - User is authenticated and has appropriate permissions to assign permission. - User is in Role Details page of a role.
Postcondition	This use case allows a system administrator to assign or remove permissions for a specific role.
Standard Process	1. User clicks on "Manage Permission" button. 2. System retrieves the current permissions assigned to the role from the database. 3. System displays the Permissions Management interface 4. User selects/deselects permissions. 5. User clicks the "Save Permissions" button to confirm changes. 6. System updates the role-permission mapping records in the database. 7. System refreshes the page to display the updated permissions. 8. Systems show a success message.
Alternative Process	<u>5a. Cancel assigning permission.</u> 5a.1. User clicks on "Cancel" button. 5a.2. System redirects user to Role Management page.
Exception Flow	N/A

Table 64: UC55_Assign_Permissions

4.2.2.56 UC56_Create Query

Use Case	**UC56_Create Query**
Description	This use case allows authorized users to permanently remove a role from the system.
Actors	Public User
Precondition/Dependency	• System is operational and database is accessible. • User has access to the contact page. • User in landing page. • User have valid email address and contact information.

Postcondition	Contact query is successfully created and stored in the database.
Standard Process	1. User navigates to contact page by clicking contact button in navigation bar. 2. System directs user to contact page. 3. System display contact form. 4. User filles in the contact form. 5. User clicks “Send Message” button. 6. System receives inputs. 7. System validates form data. 8. System creates new record. 9. System returns results from database. 10. System displays success message.
Alternative Process	N/A
Exception Flow	7a. Invalid data 7a.1. System returns results. 7a.2. System displays error message. 7a.3. User may continue again from Step 4

4.2.3 Class Diagram

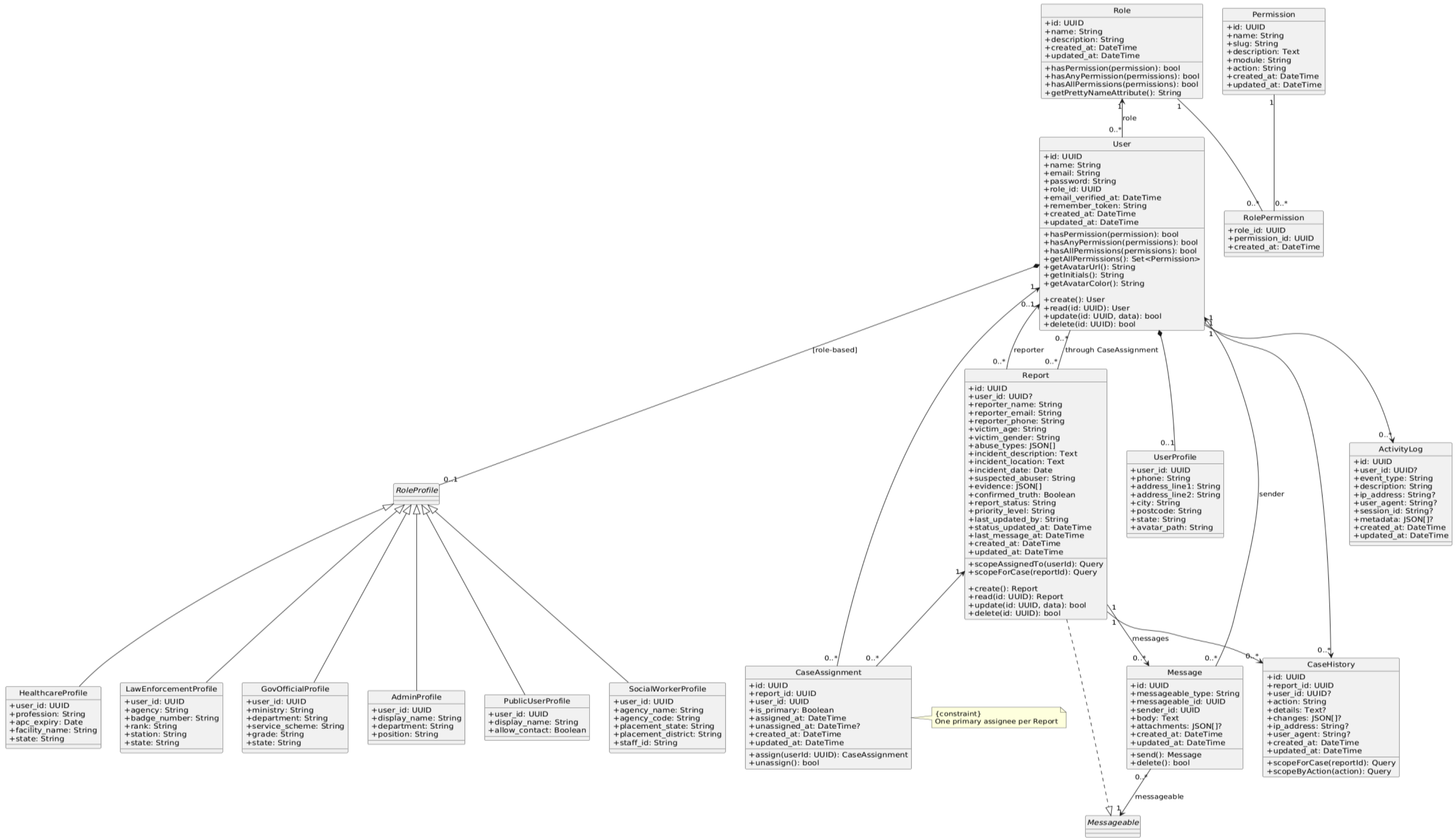


Figure 42: SinDa Class Diagram

4.2.4 Activity Diagram

4.2.4.1 UC1_Register Account

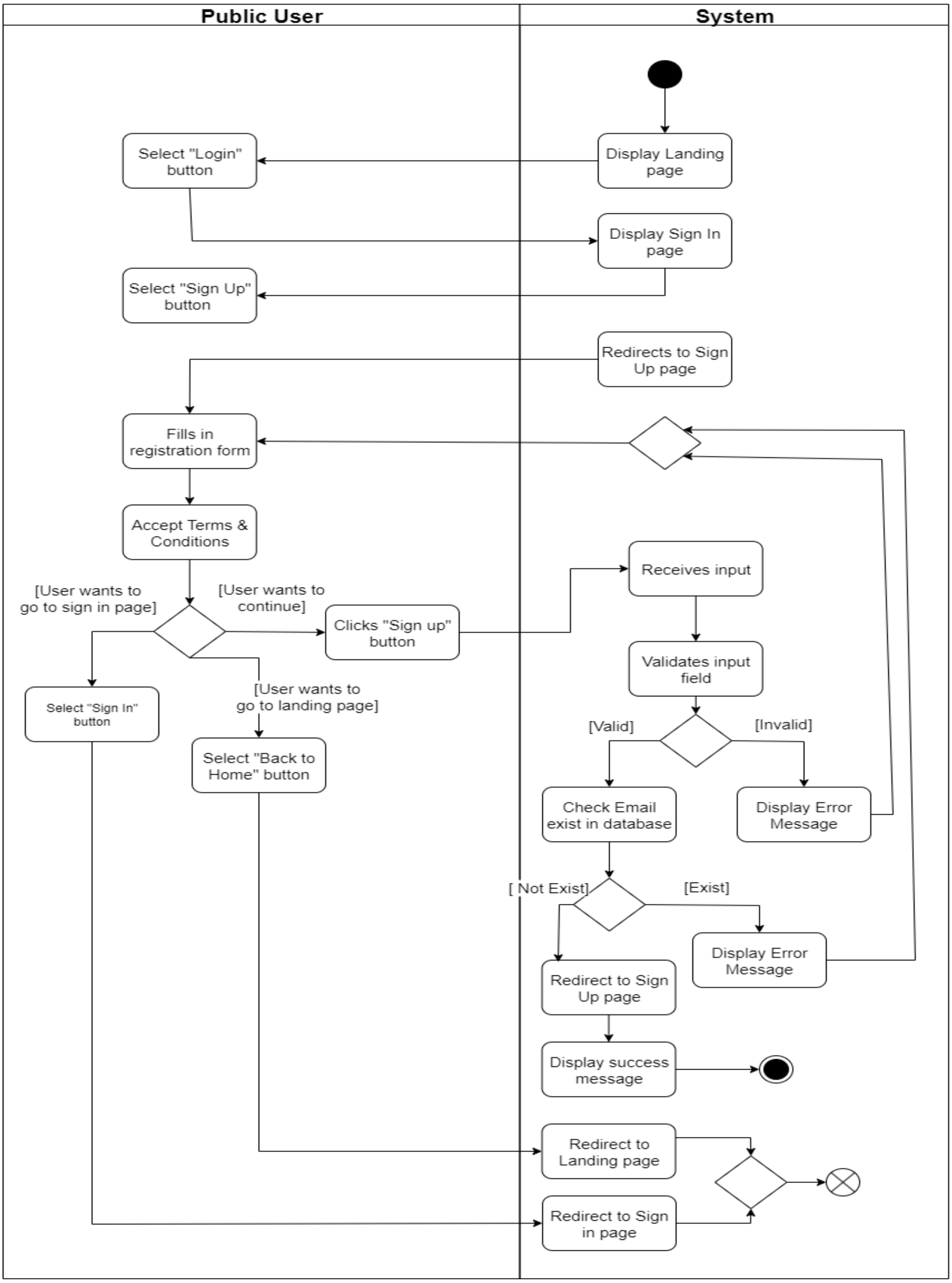


Figure 43: UC1_Register Account

4.2.4.2 UC2_Login

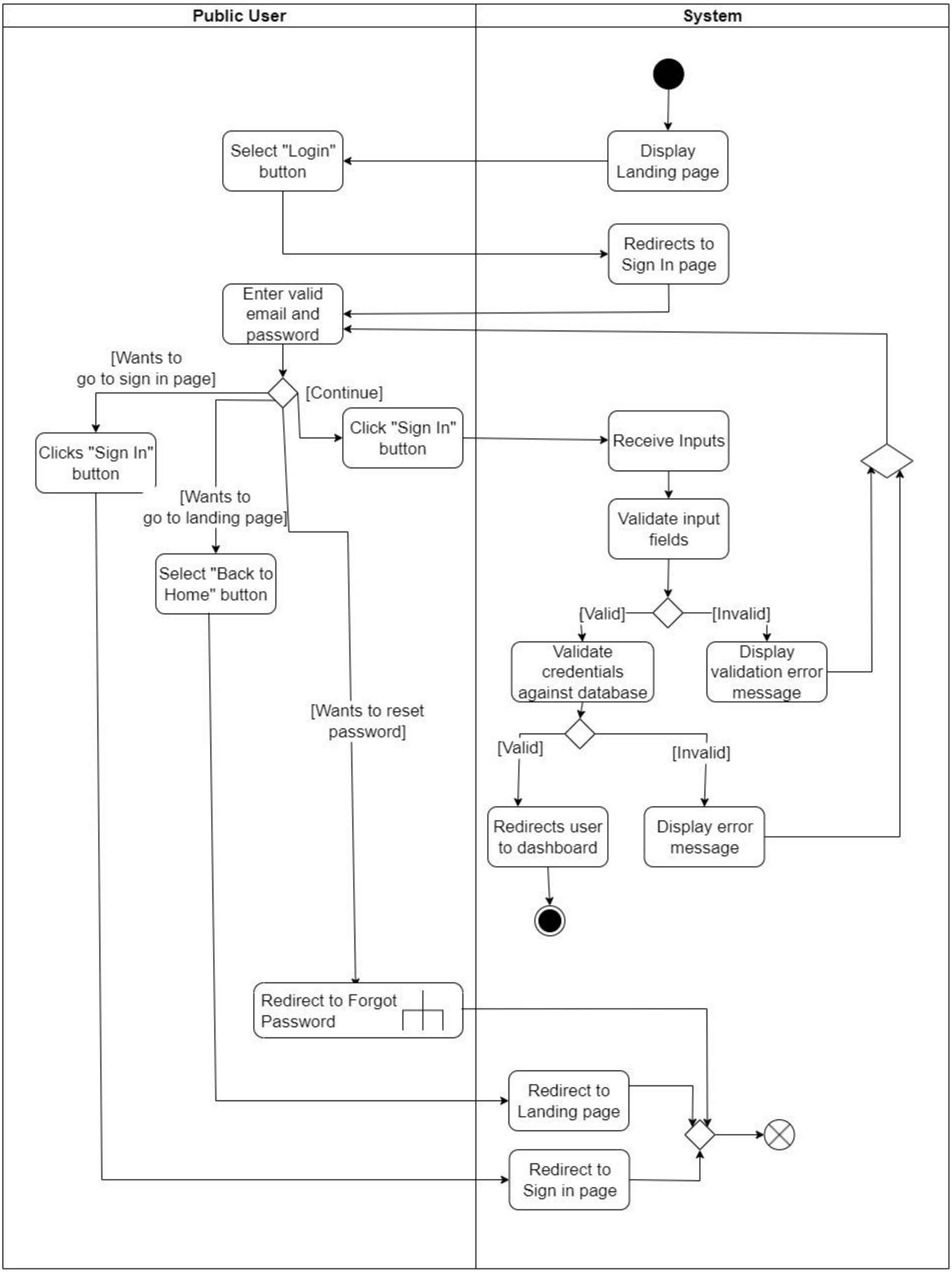


Figure 44: UC2_Login

4.2.4.3 UC3_Forgot Password

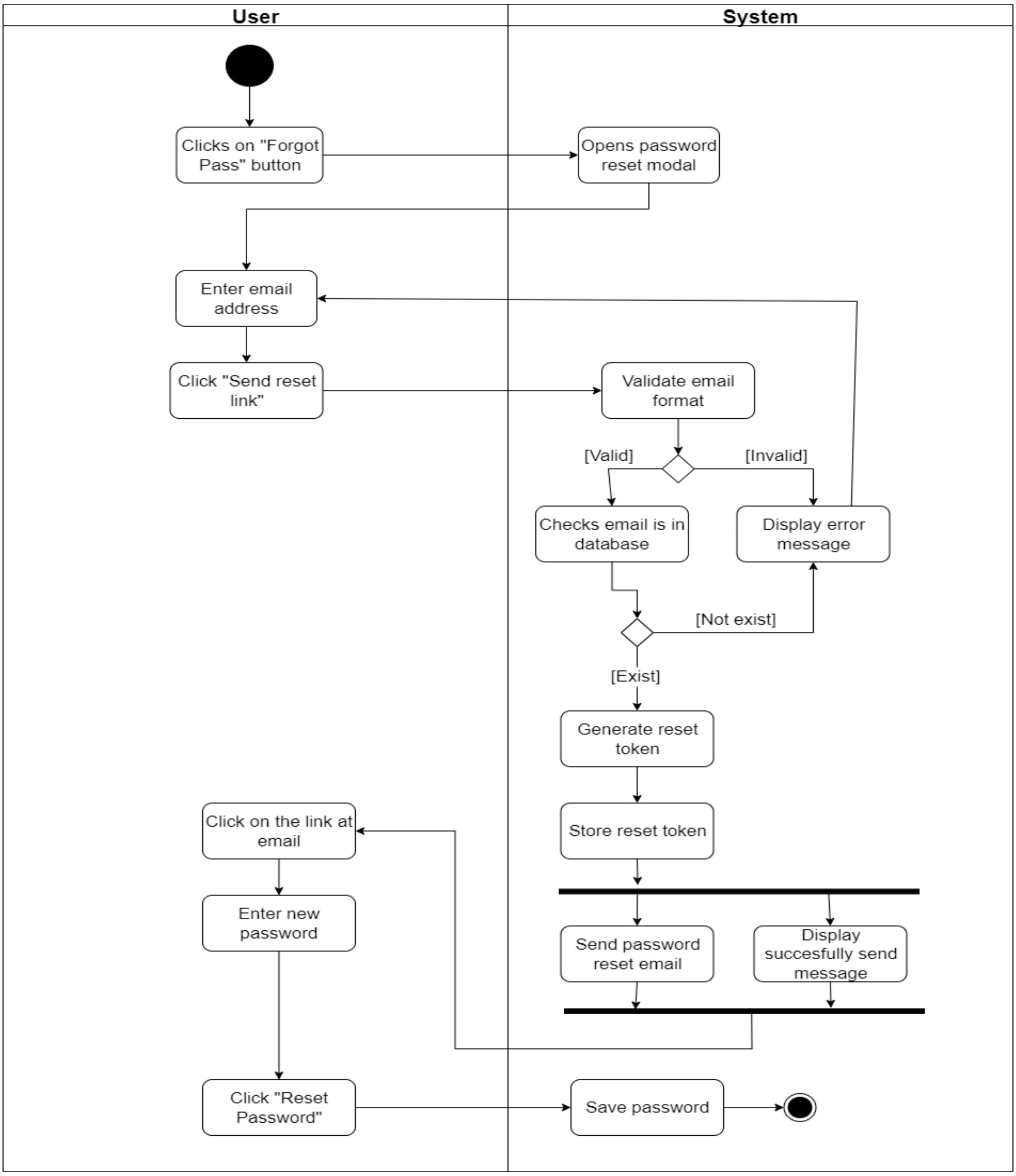


Figure 45: UC3_Forgot Password

4.2.4.4 UC4_Create Report

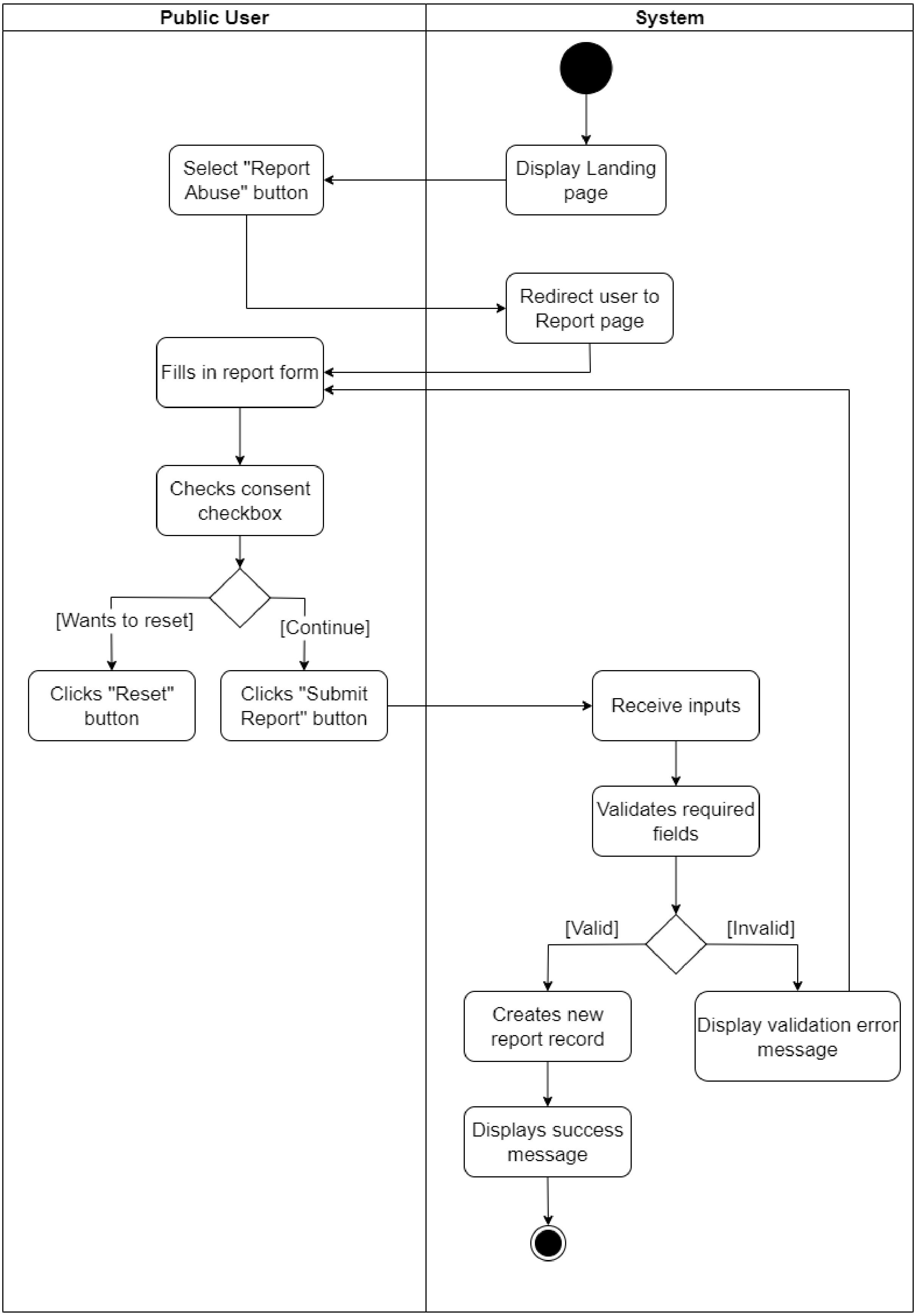


Figure 46: UC4_Create Report

4.2.4.5 UC5_List Report

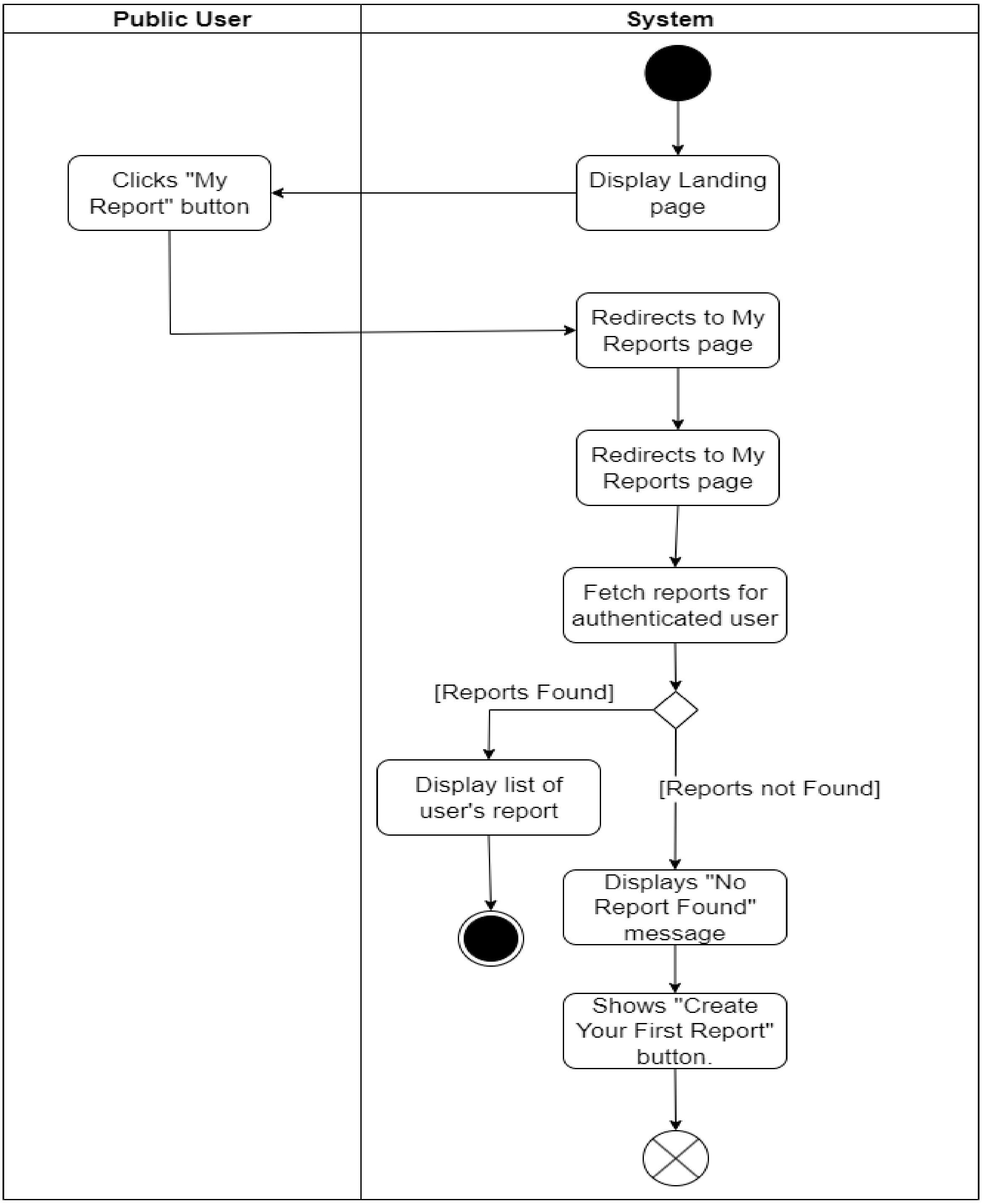


Figure 47: UC5_List Reports

4.2.4.6 UC6_View Report Details

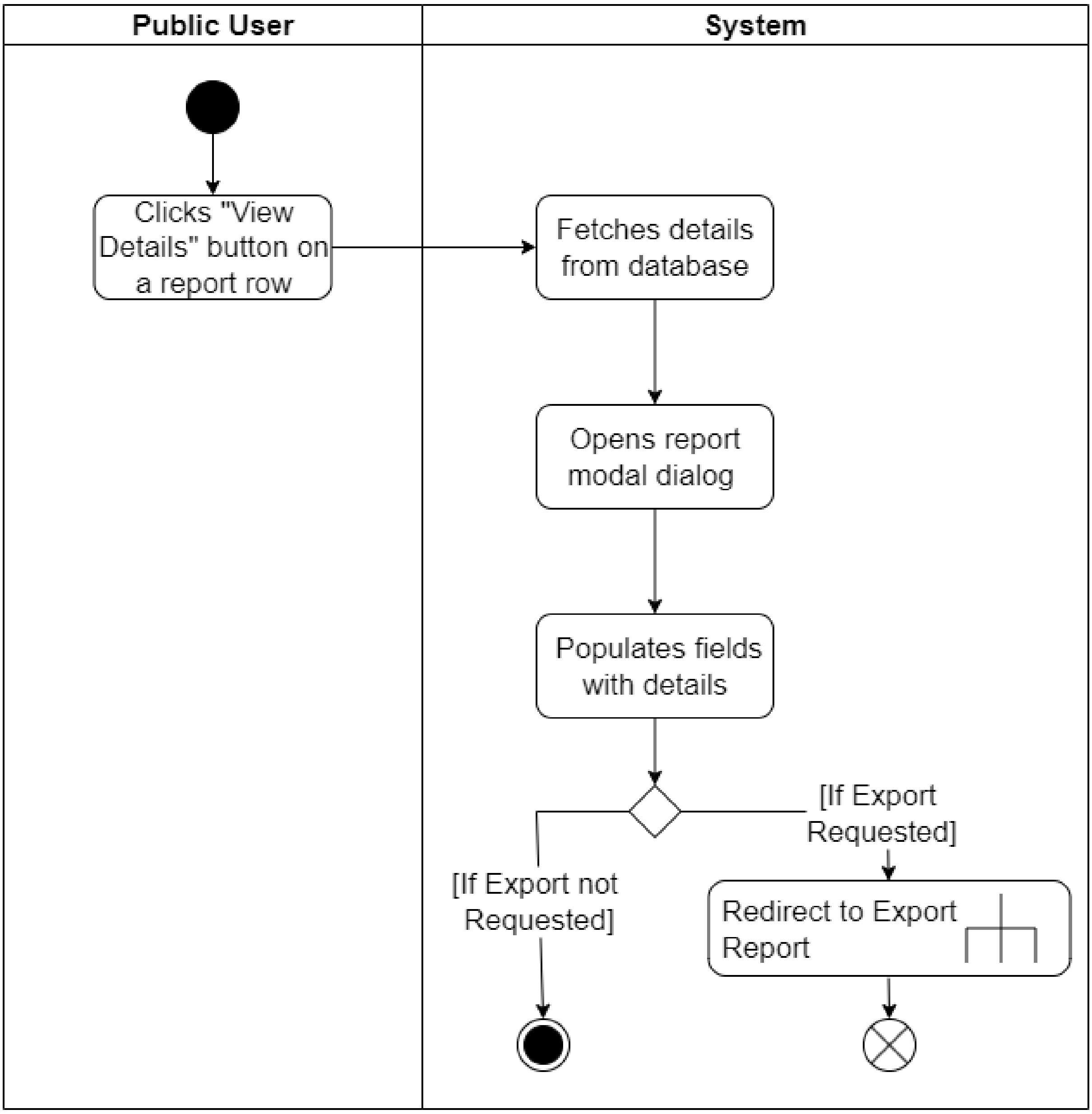


Figure 48: UC6_View Report Details

4.2.4.7 UC7_Export Report

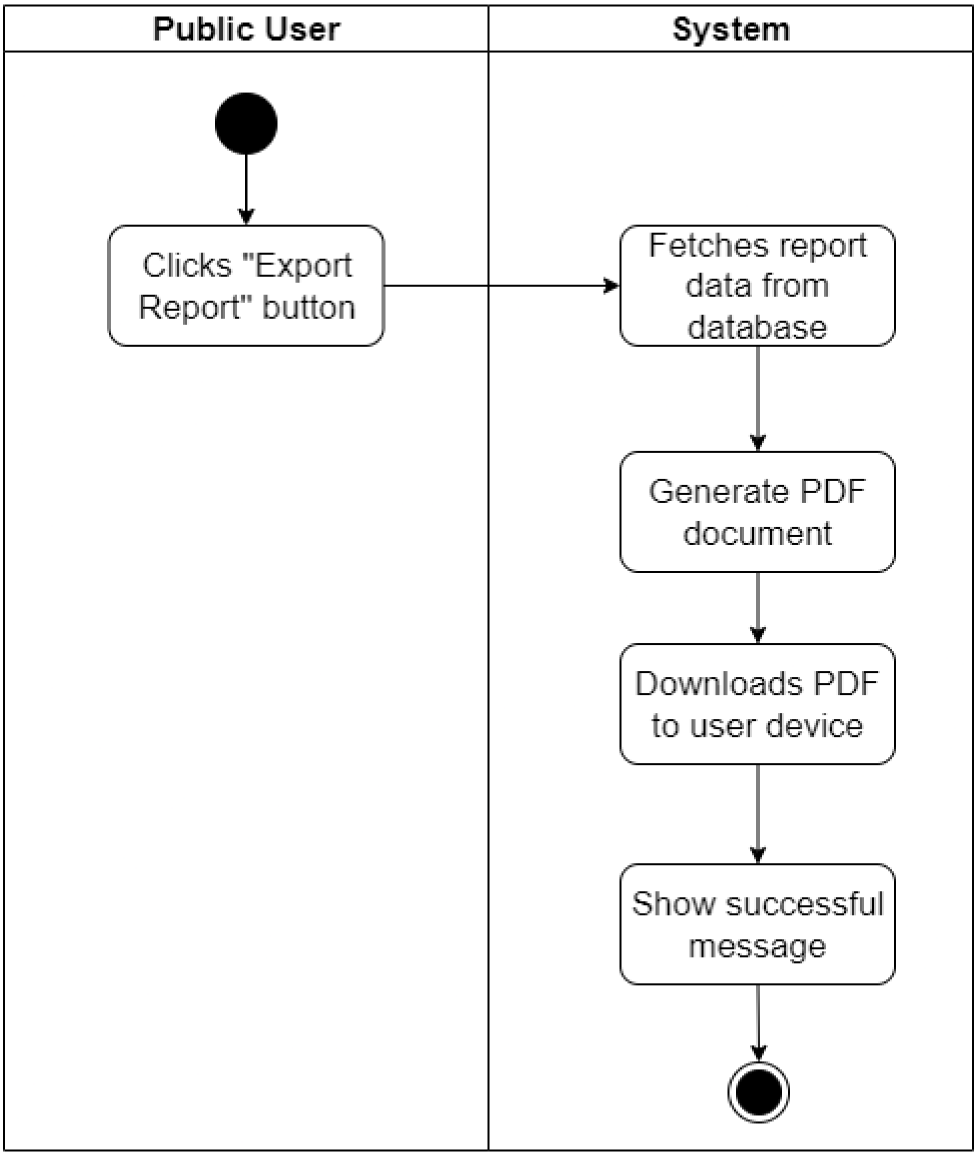


Figure 49: UC7_Exprot Report

4.2.4.8 UC8_View Profile Details

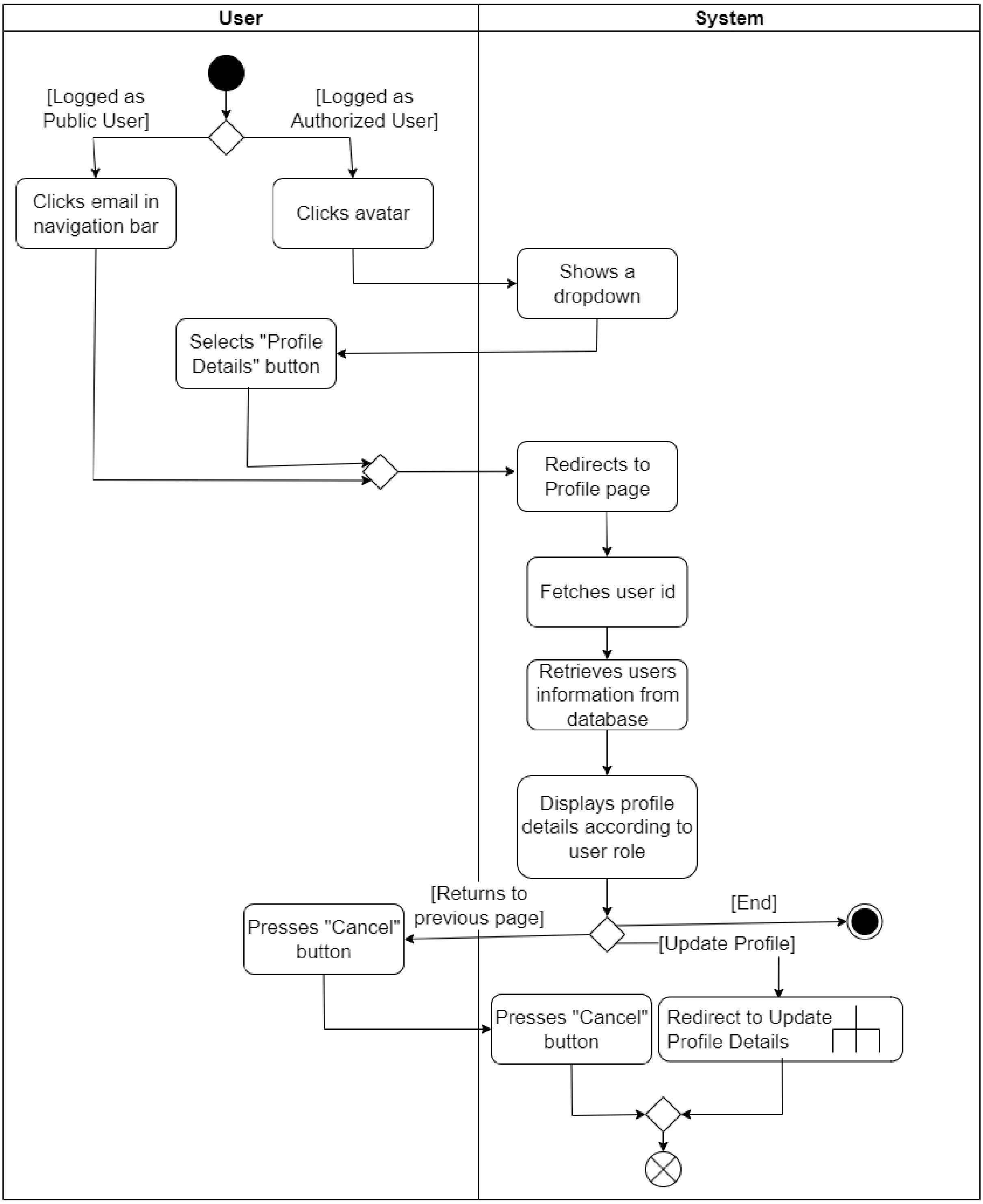


Figure 50: UC8_View Profile Details

4.2.4.9 UC9_Update Profile Details

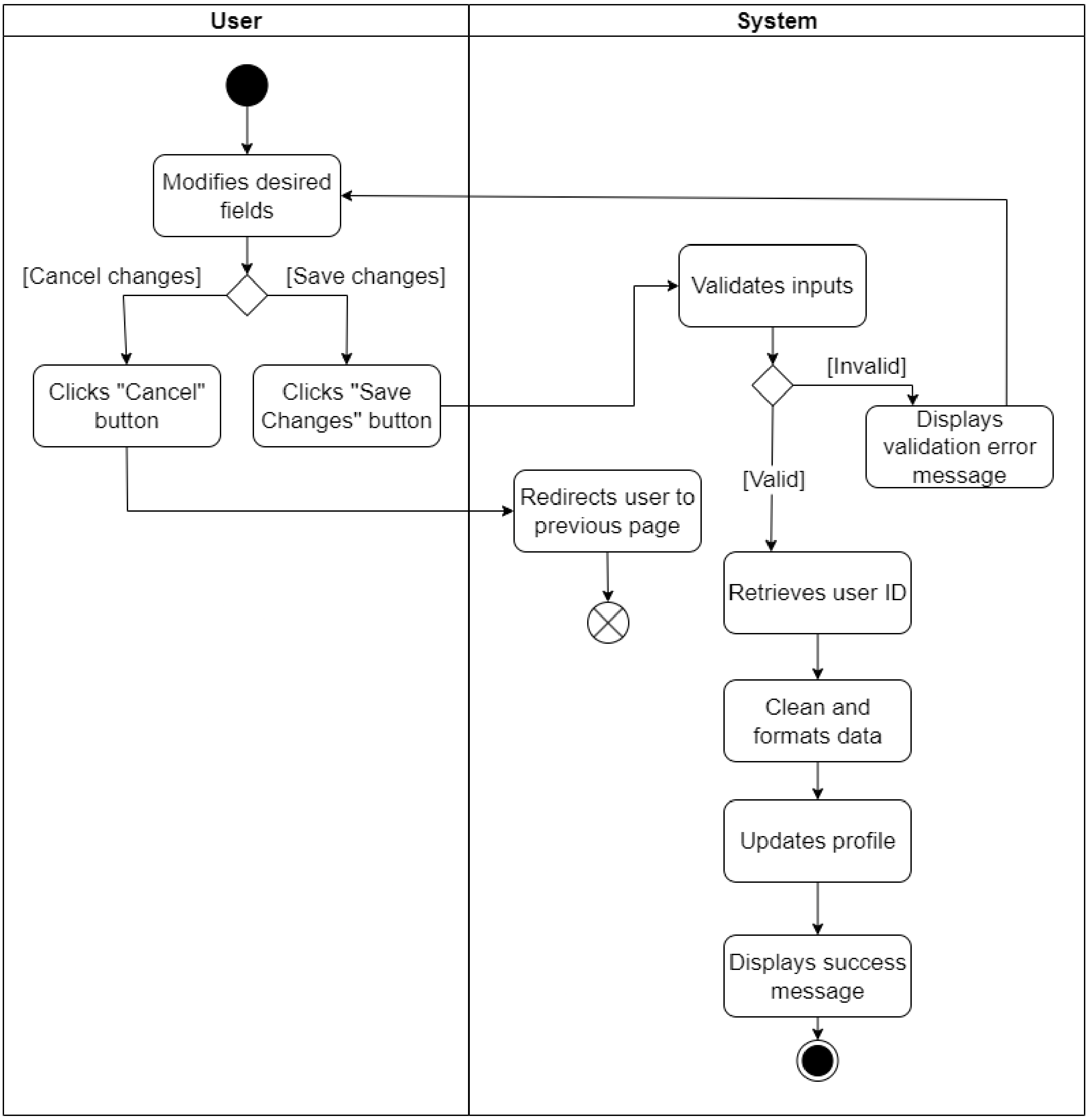


Figure 51: UC9_Update Profile Details

4.2.4.10 UC10_Logout

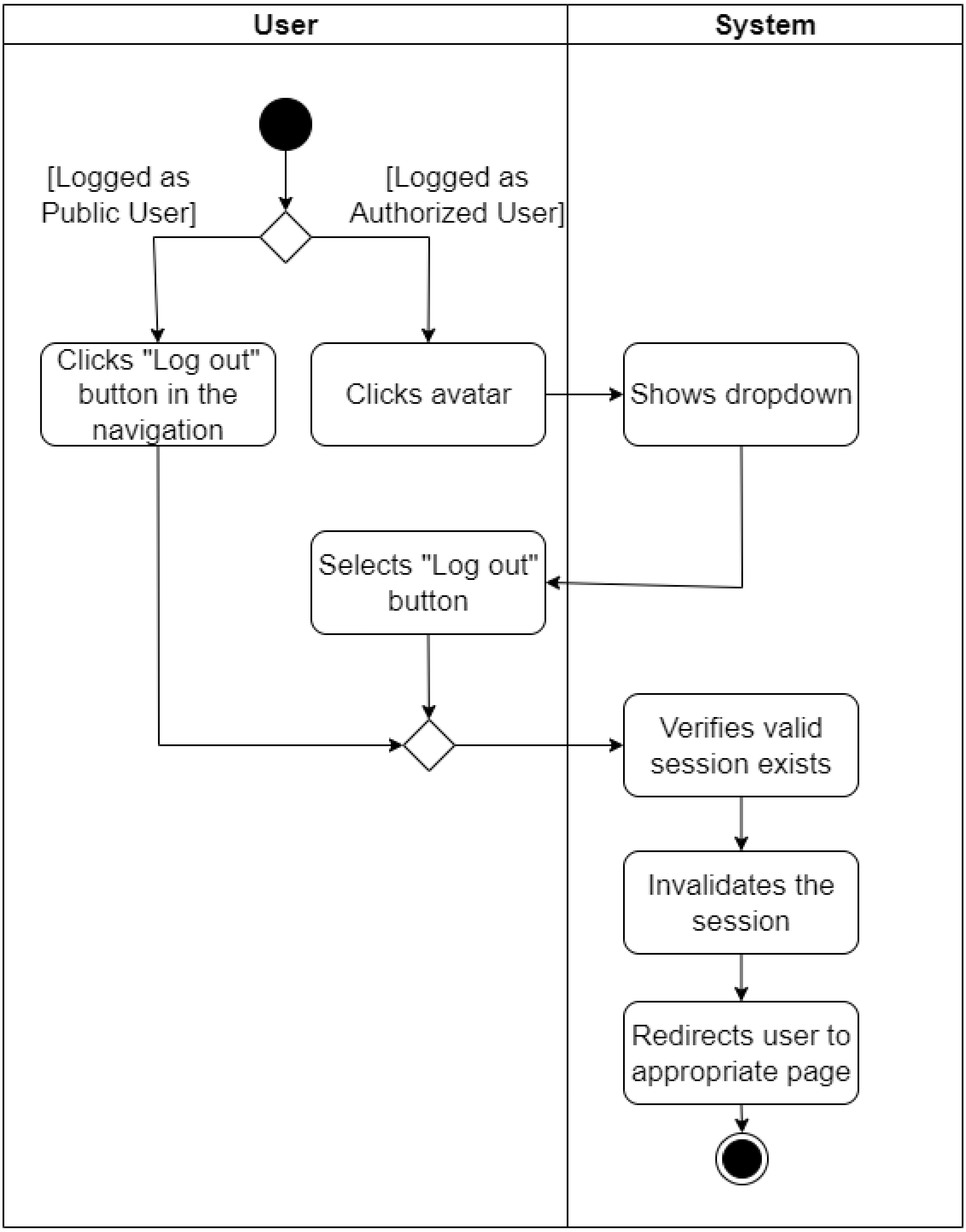


Figure 52: UC10_Logout

4.2.4.11 UC11_List Queries

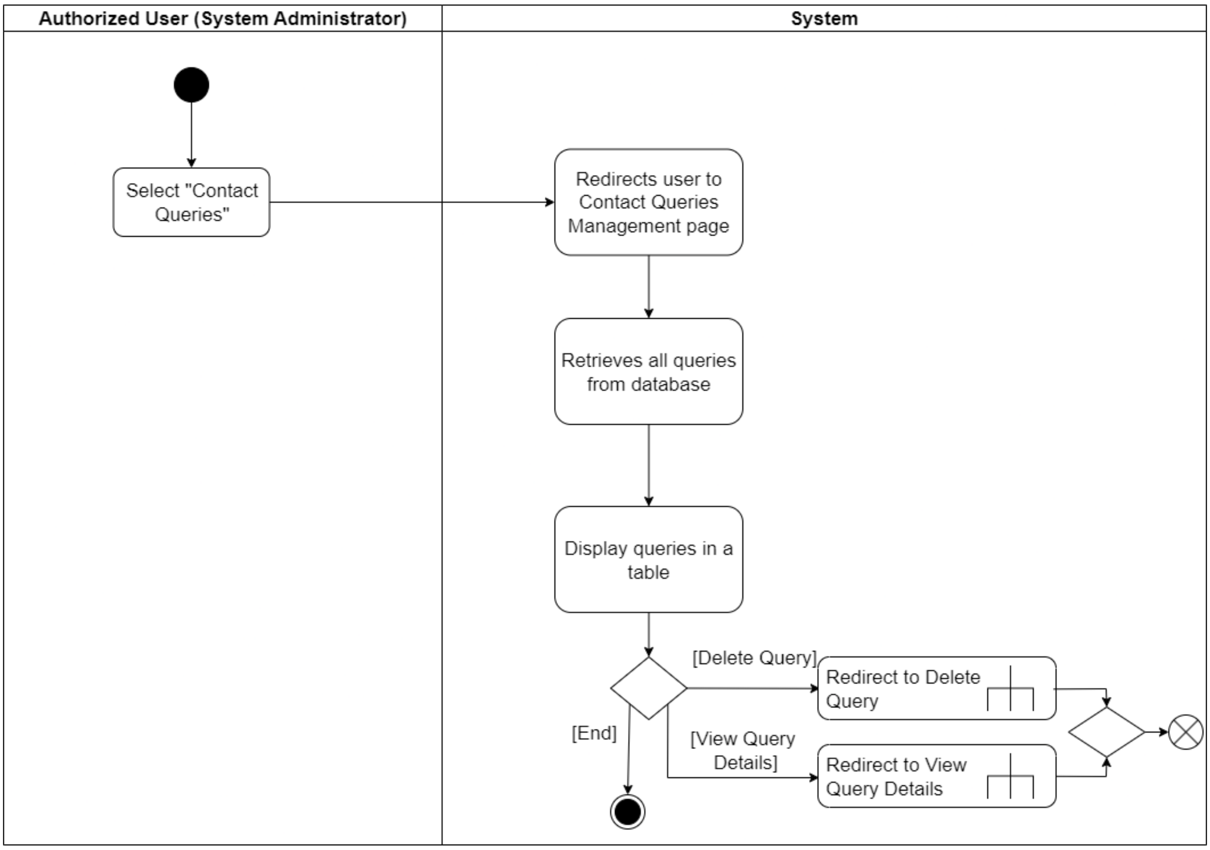


Figure 53: UC11_List Queries

4.2.4.12 UC12_View Query Details

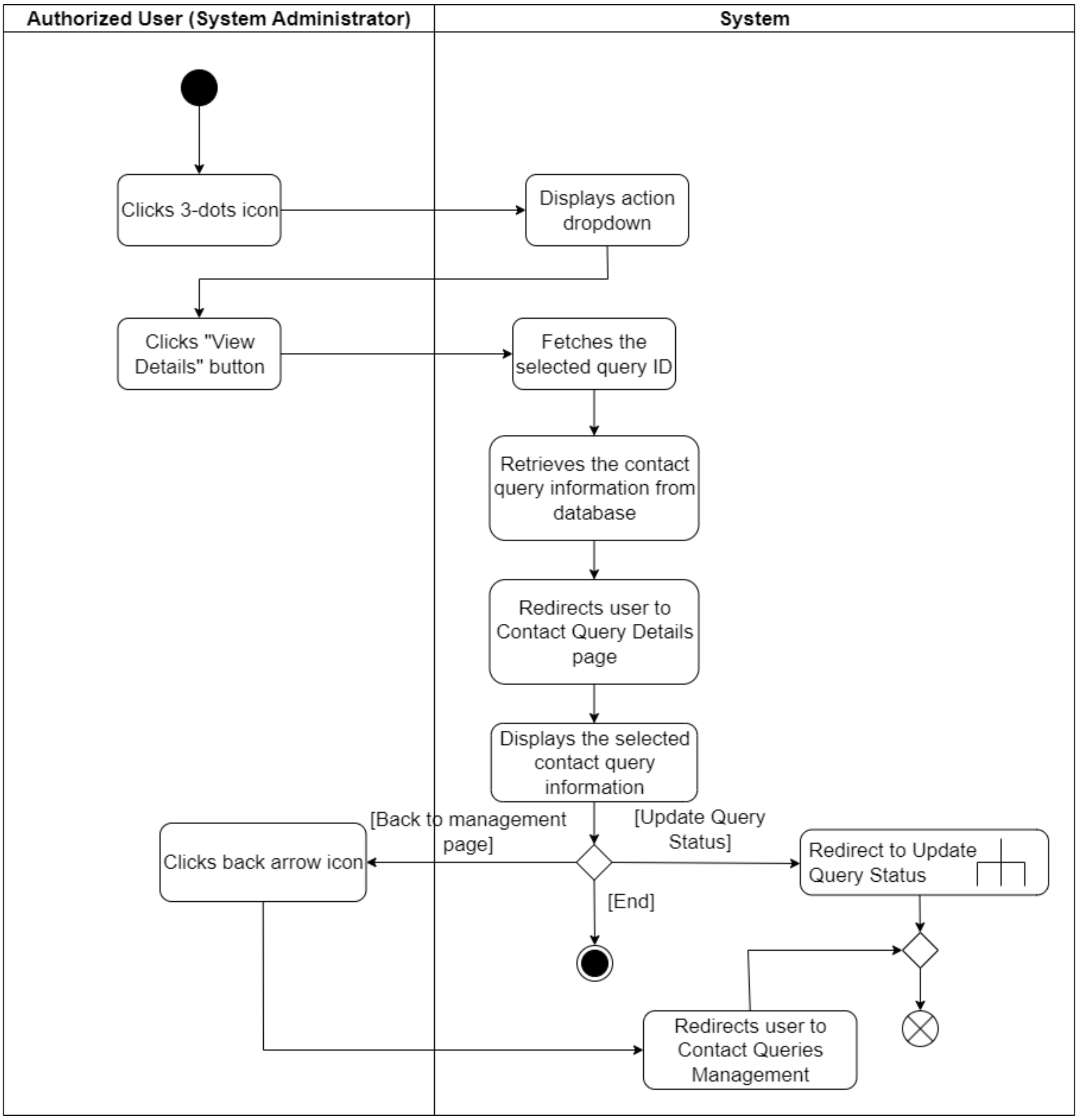


Figure 54:UC12_View Query Details

4.2.4.13 UC13_Update Query Status

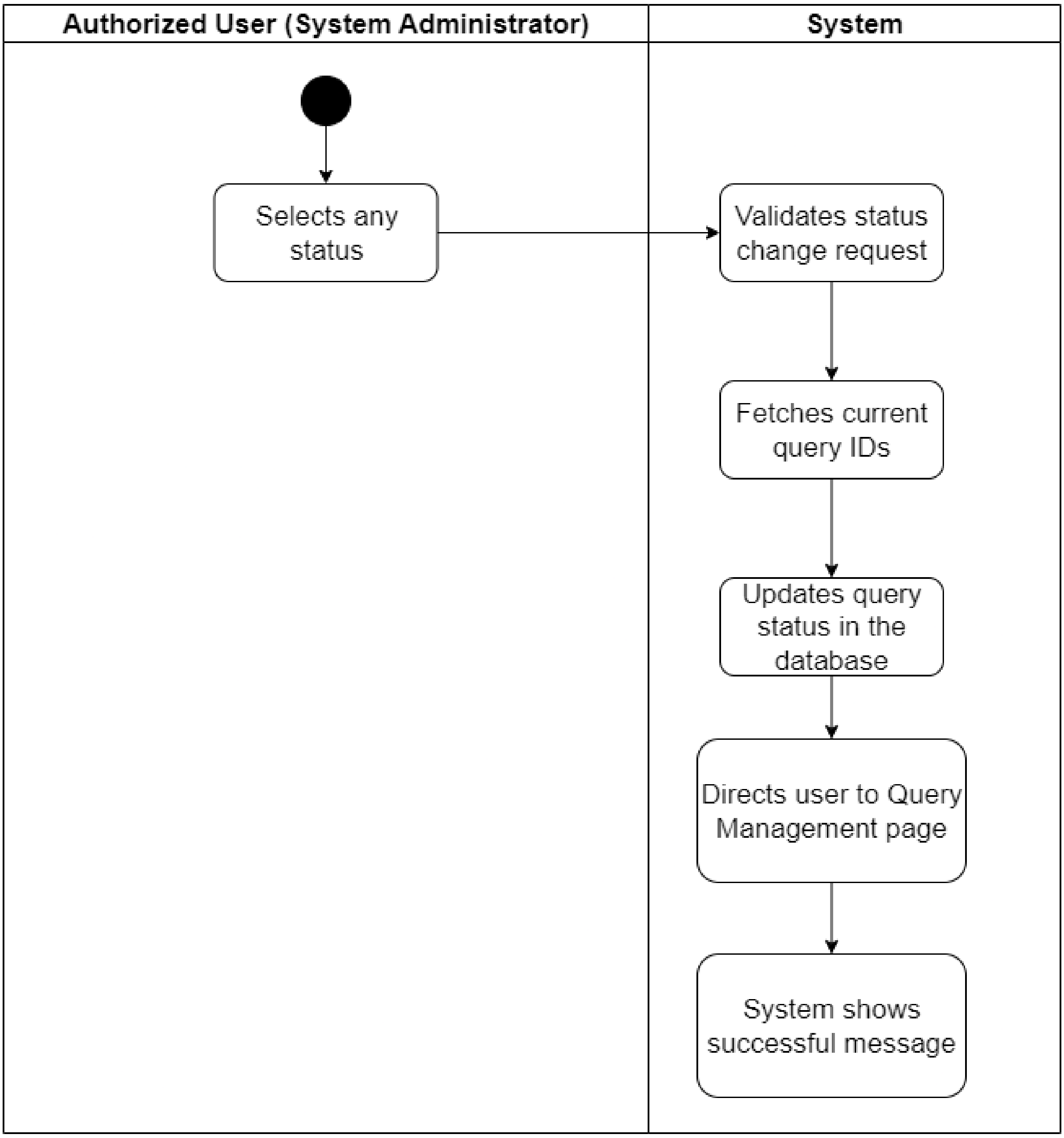


Figure 55: UC13_Update Query Status

4.2.4.14 UC14_Delete Query

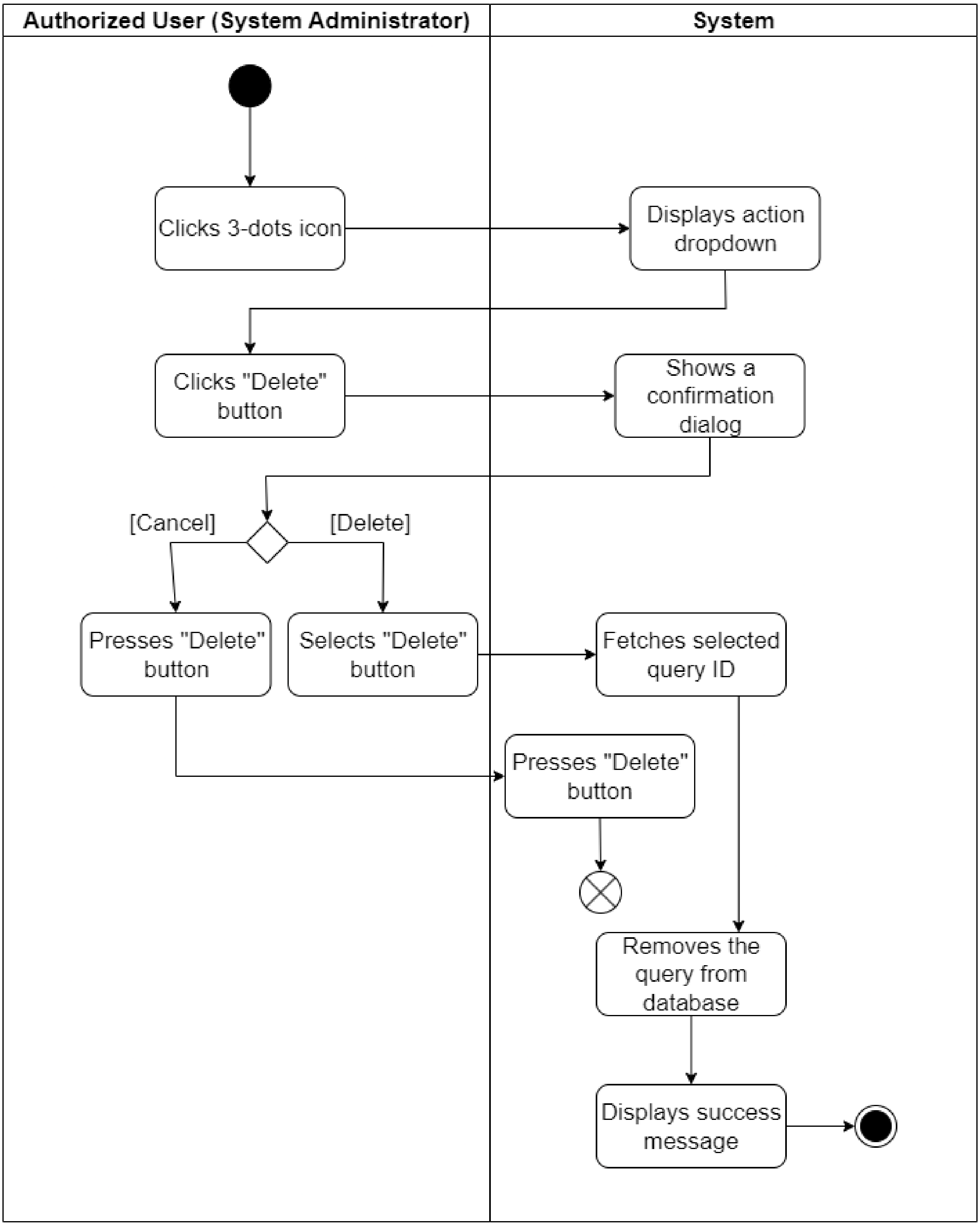


Figure 56: UC14_Delete Query

4.2.4.15 UC15_List Cases

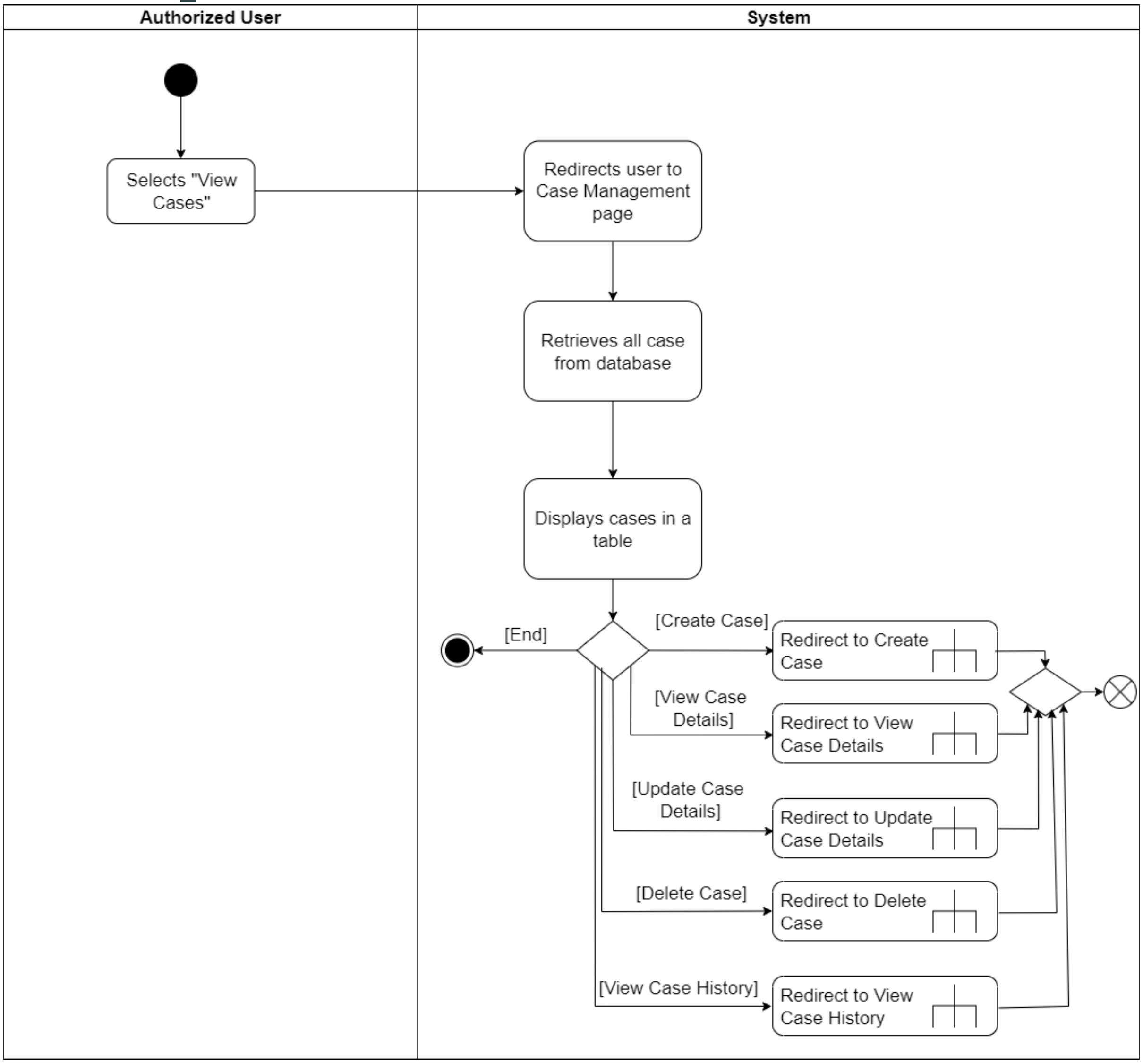


Figure 57: UC15_List Case

4.2.4.16 UC16_Create Case

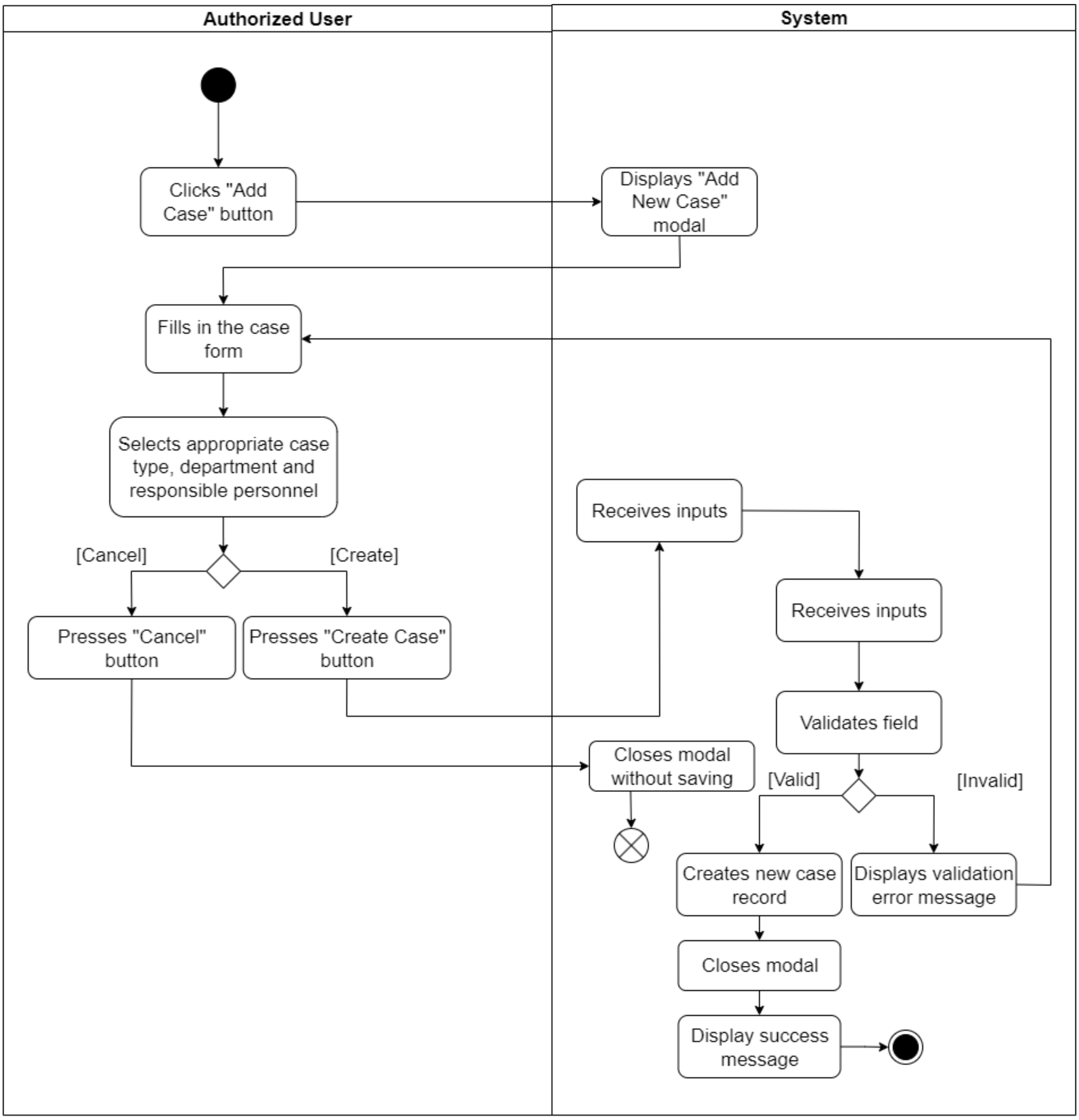


Figure 58: UC16_Create Case

4.2.4.17 UC17_ View Case Details

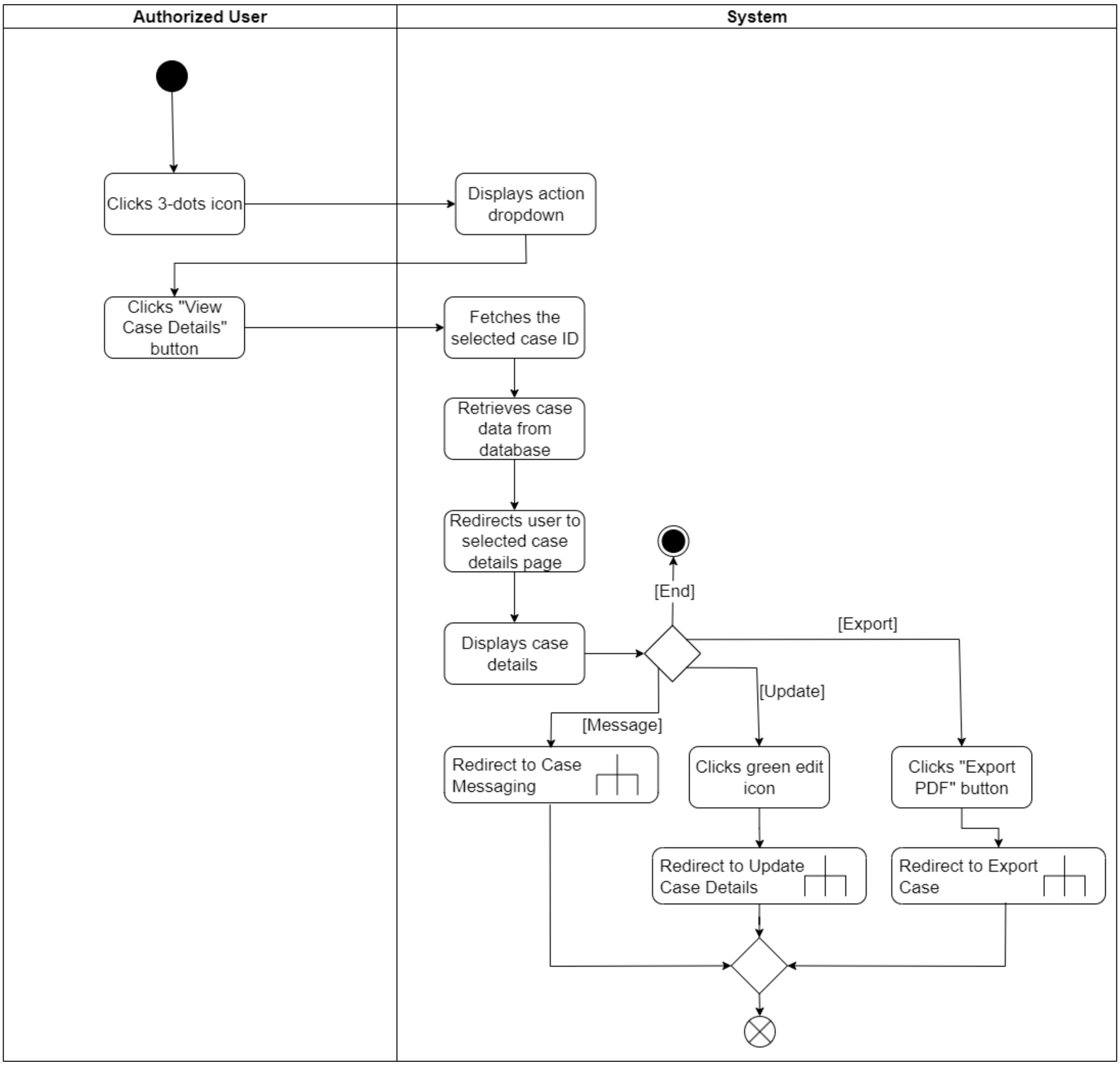


Figure 59: UC17_View Case Details

4.2.4.18 UC18_ Case Messaging

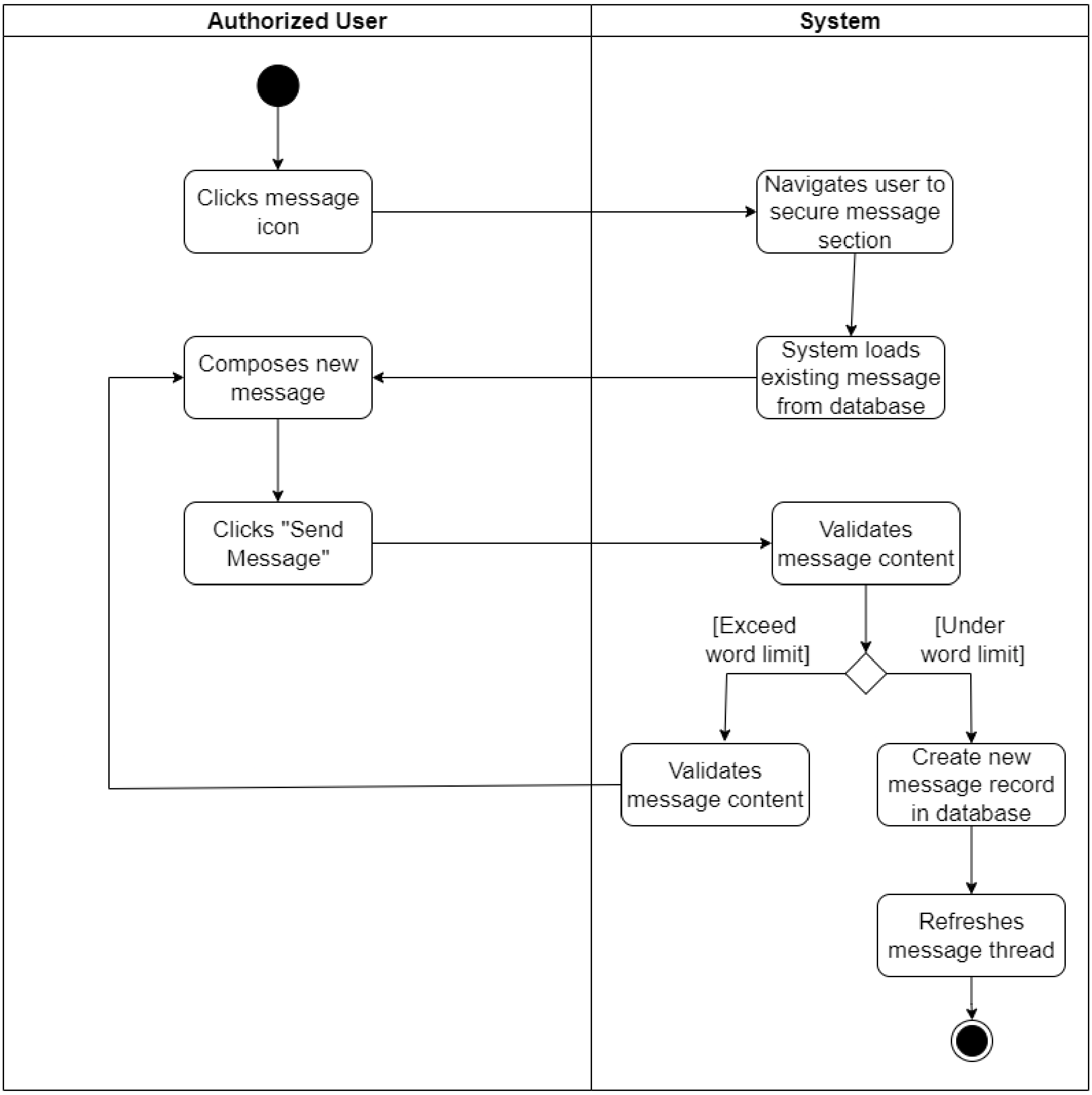


Figure 60: UC18_Case Messaging

4.2.4.19 UC19_ Update Case Details

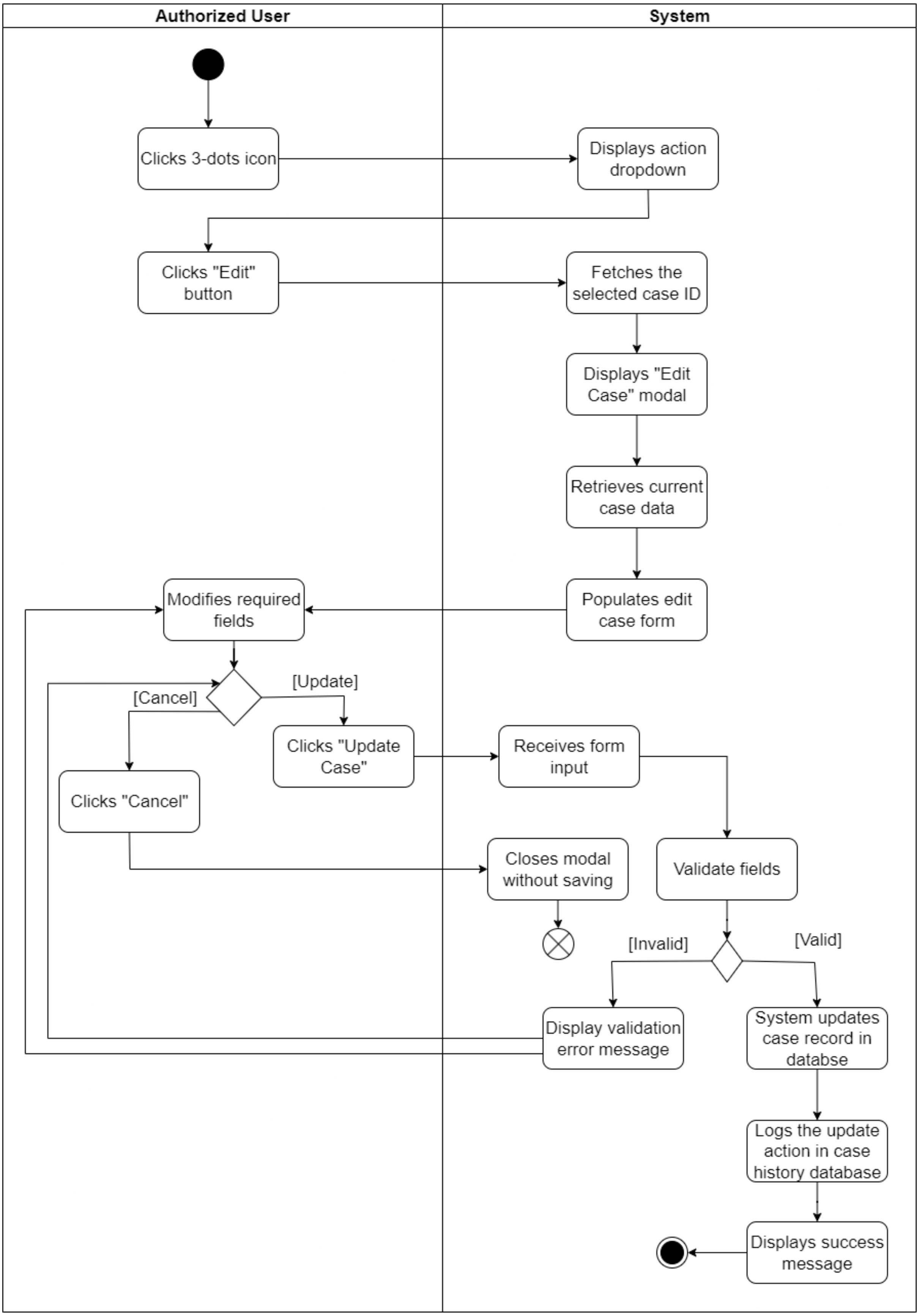


Figure 61: UC19_Update Case Details

4.2.4.20 UC20_ Delete Case

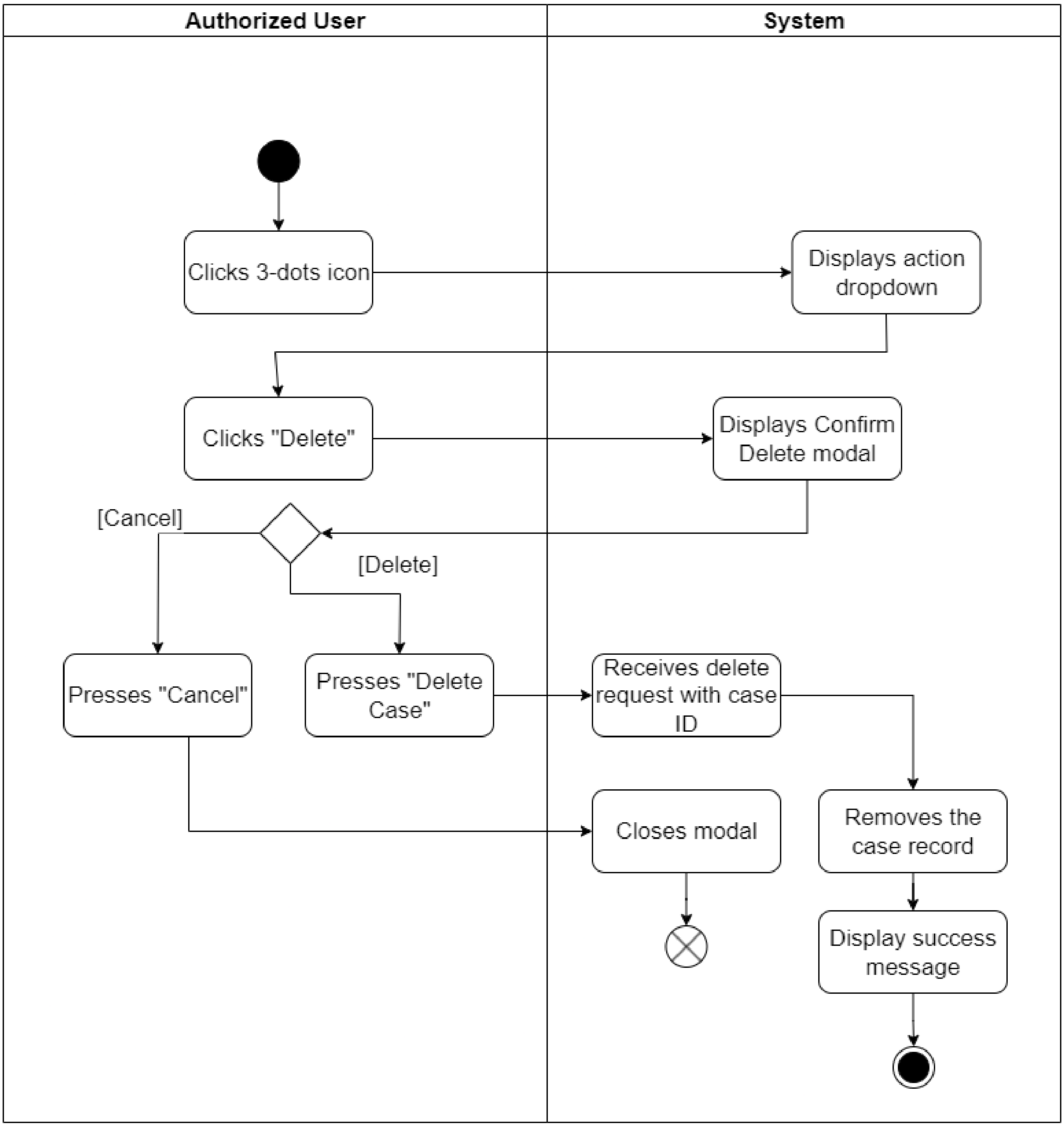


Figure 62: UC20_Delete Case

4.2.4.21 UC21_ View Case History

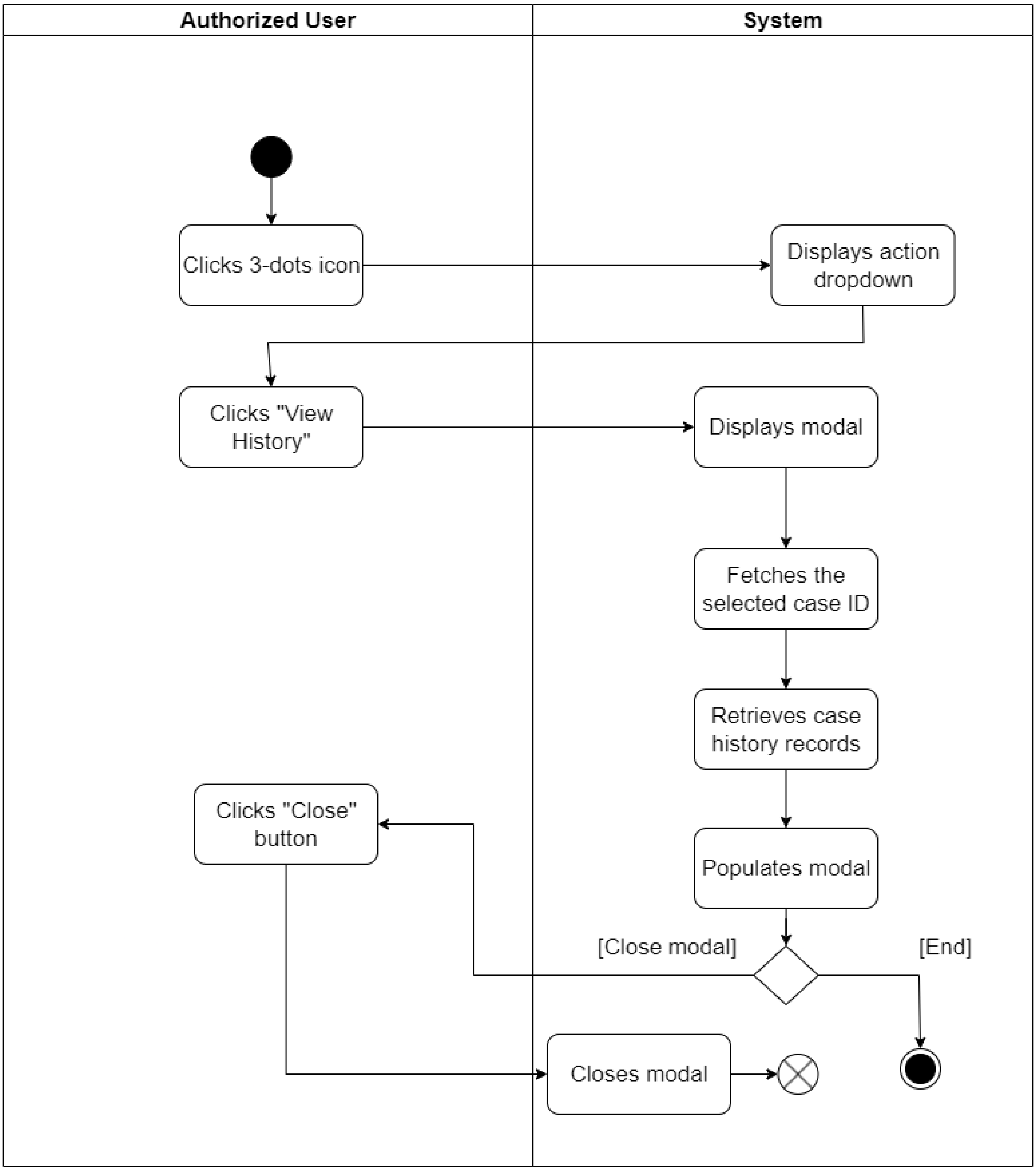


Figure 63: UC21_View Case History

4.2.4.22 UC22_ Export Case

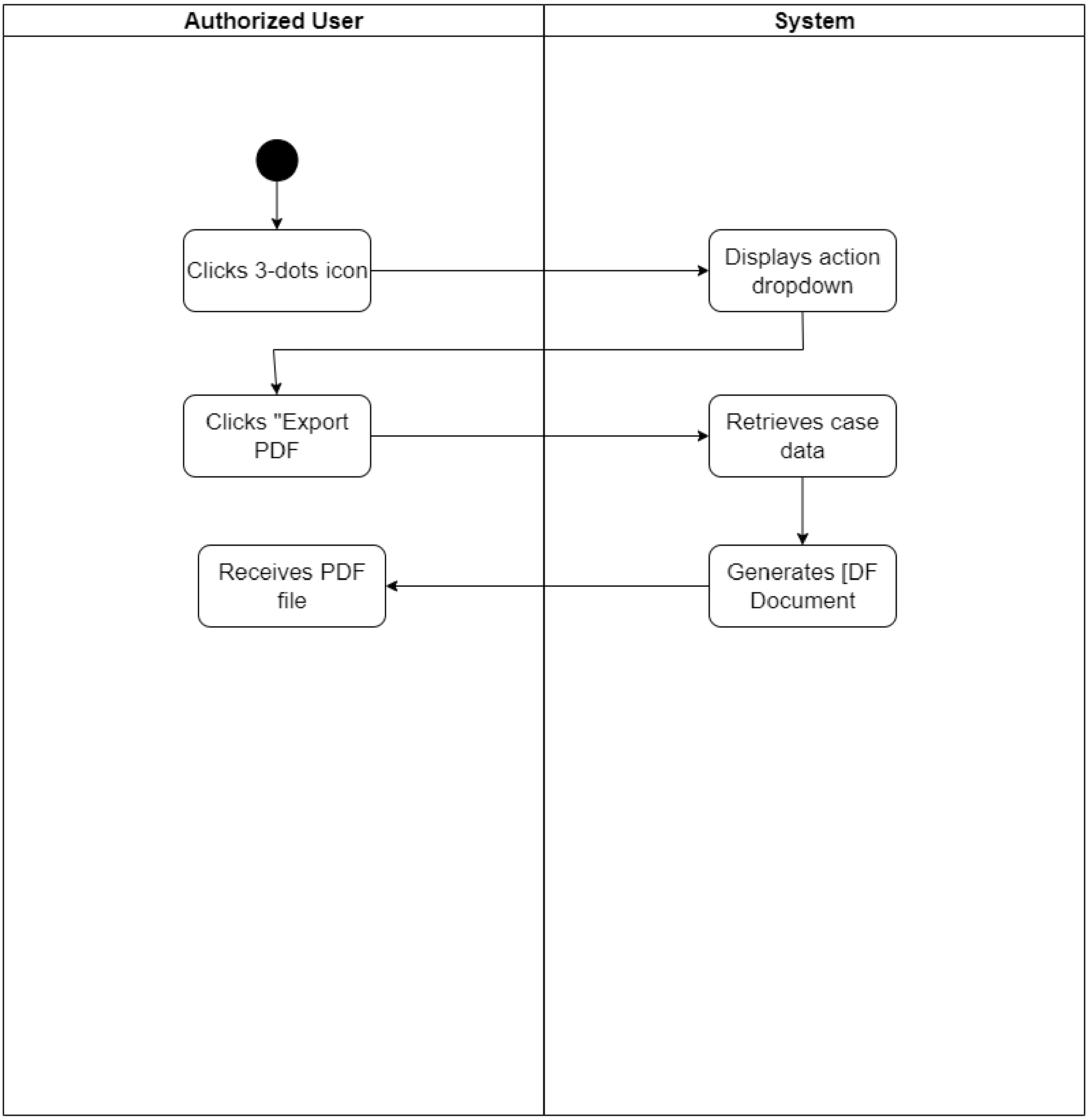


Figure 64: UC22_Export Case

4.2.4.23 UC23_ List Admins

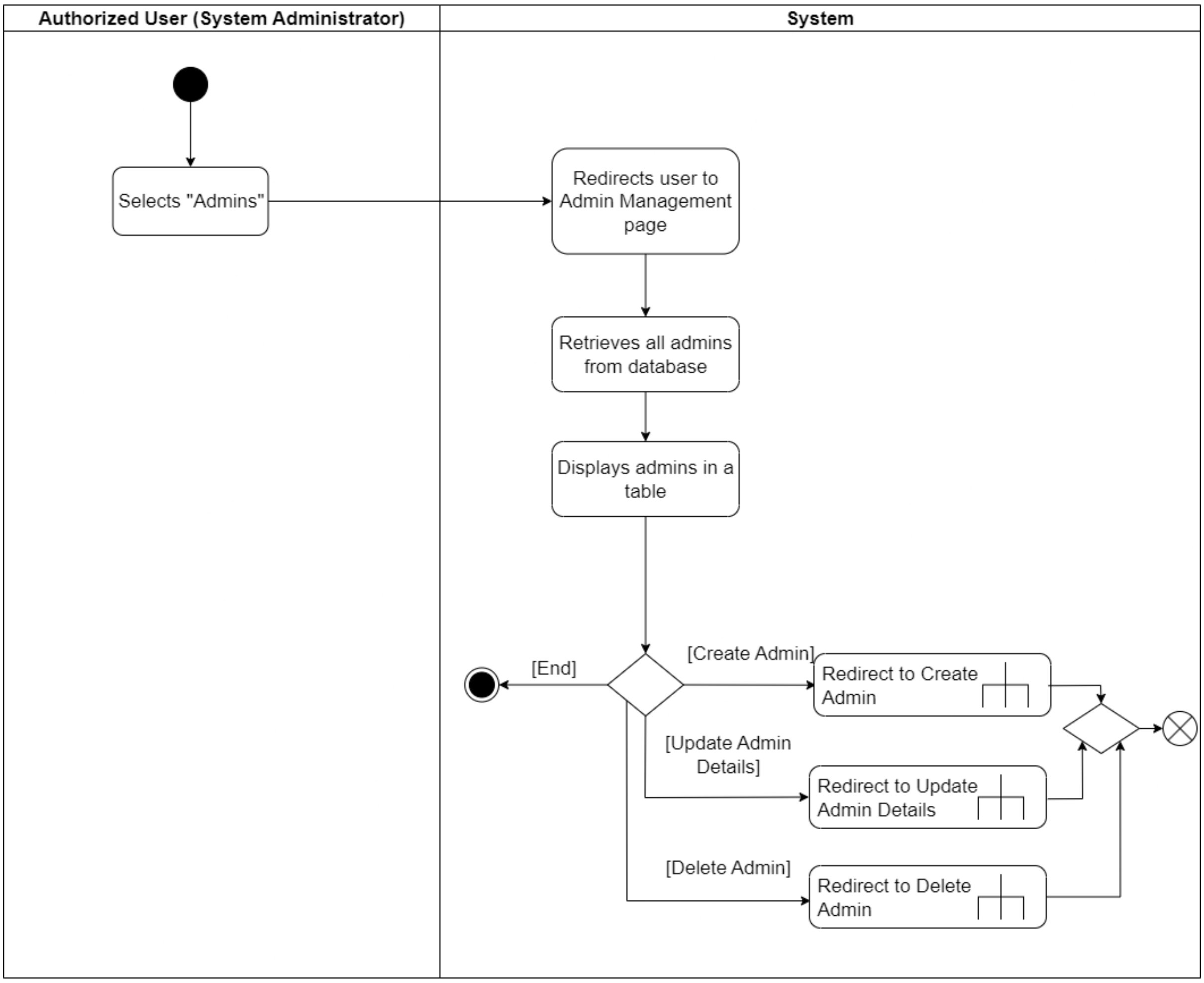


Figure 65: UC23_List Admins

4.2.4.24 UC24_ Create Admin

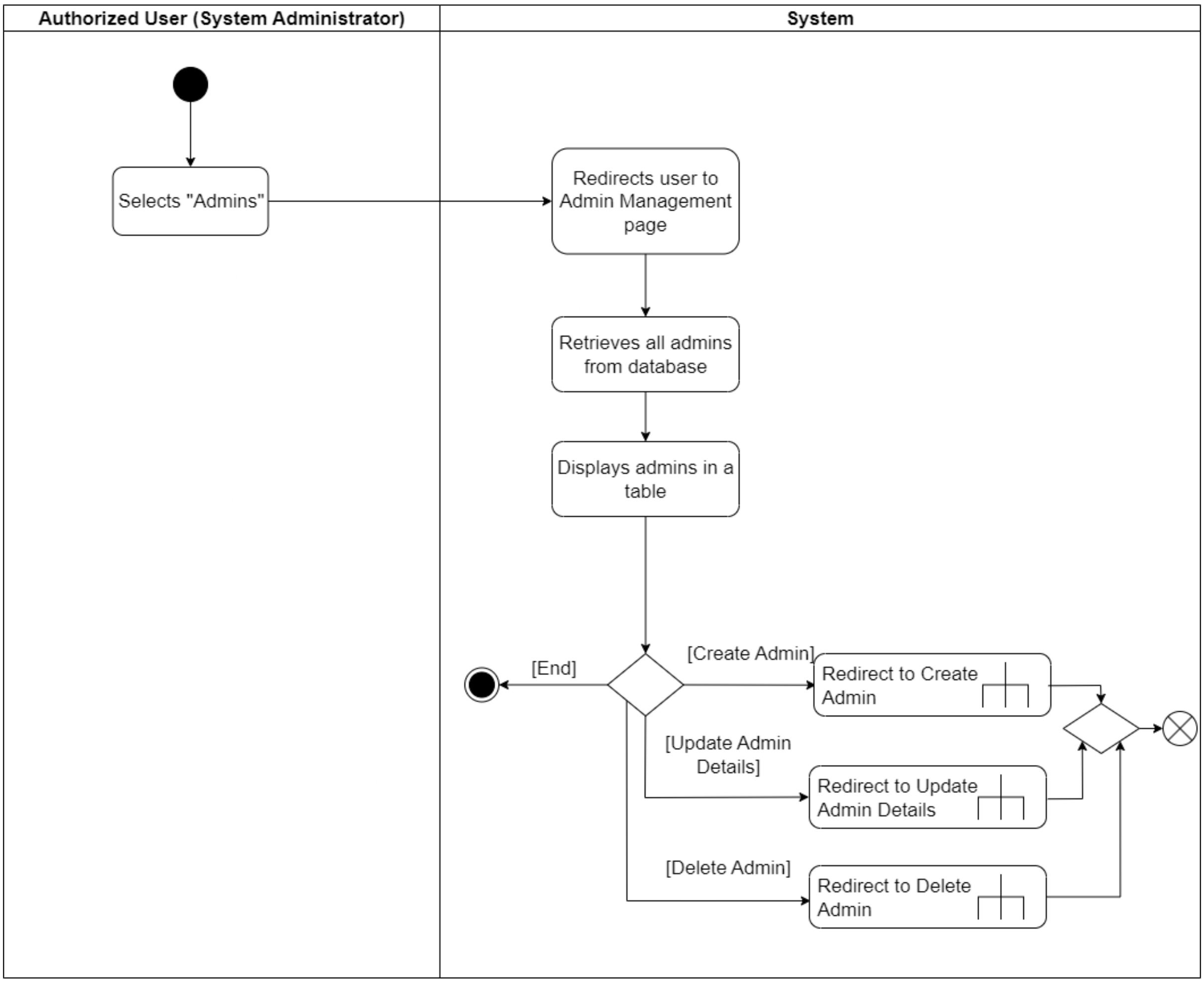


Figure 66: UC24_Create Admin

4.2.4.25 UC25_ Update Admin Details

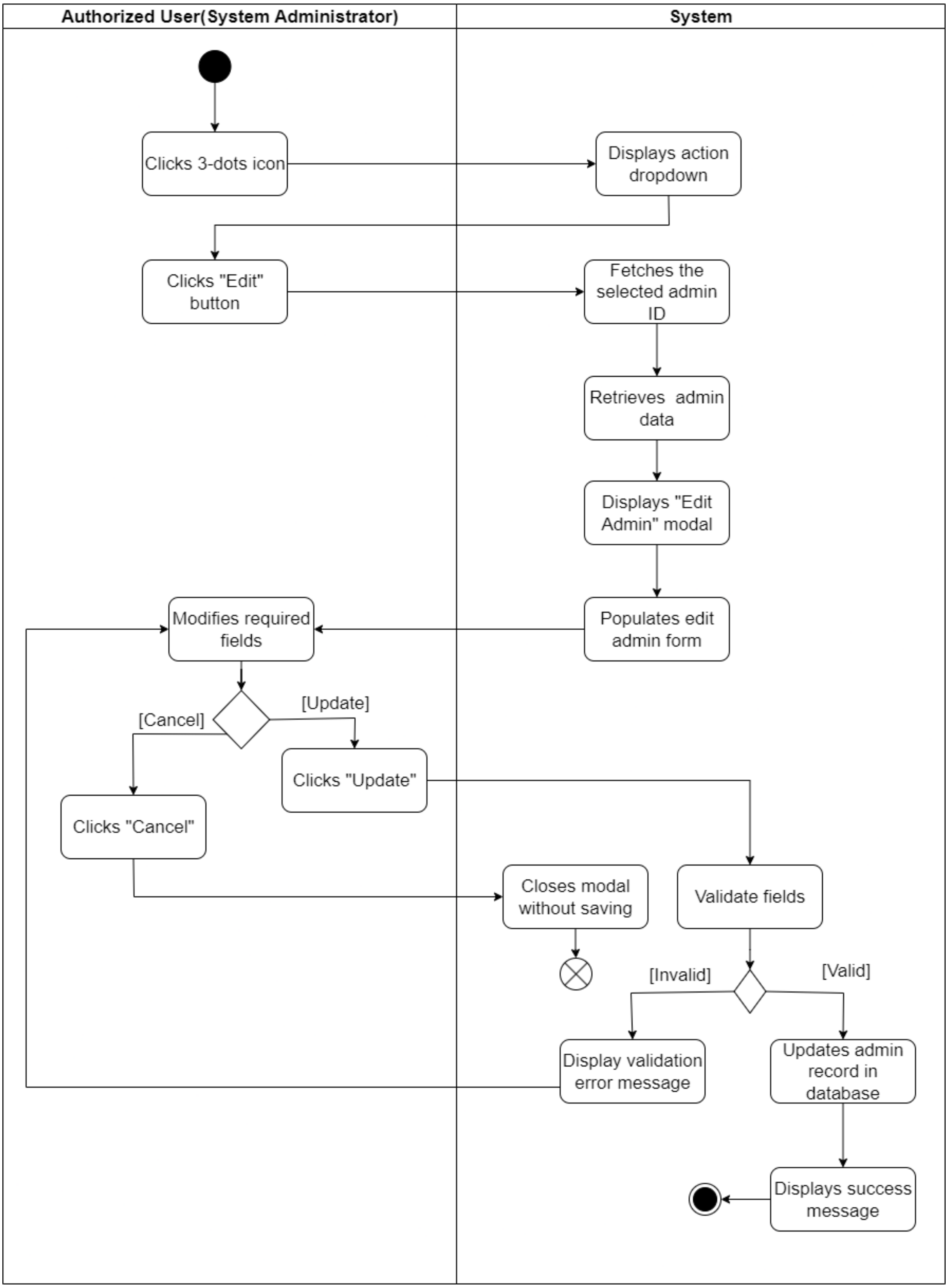


Figure 67: UC25_Update Admin Details

4.2.4.26 UC26_ Delete Admin

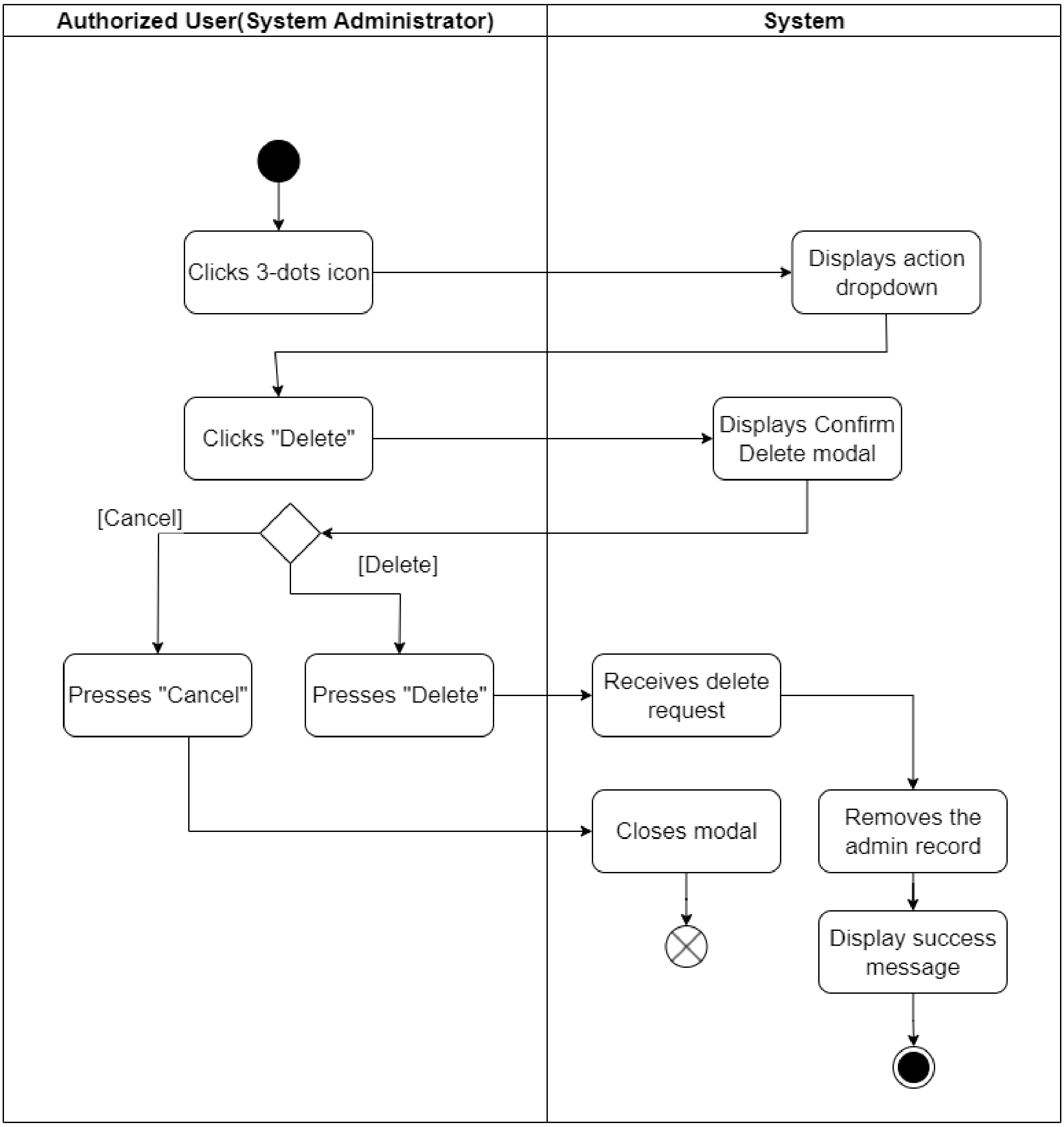


Figure 68: UC26_Delete Admin

4.2.4.27 UC27_ Export Reports

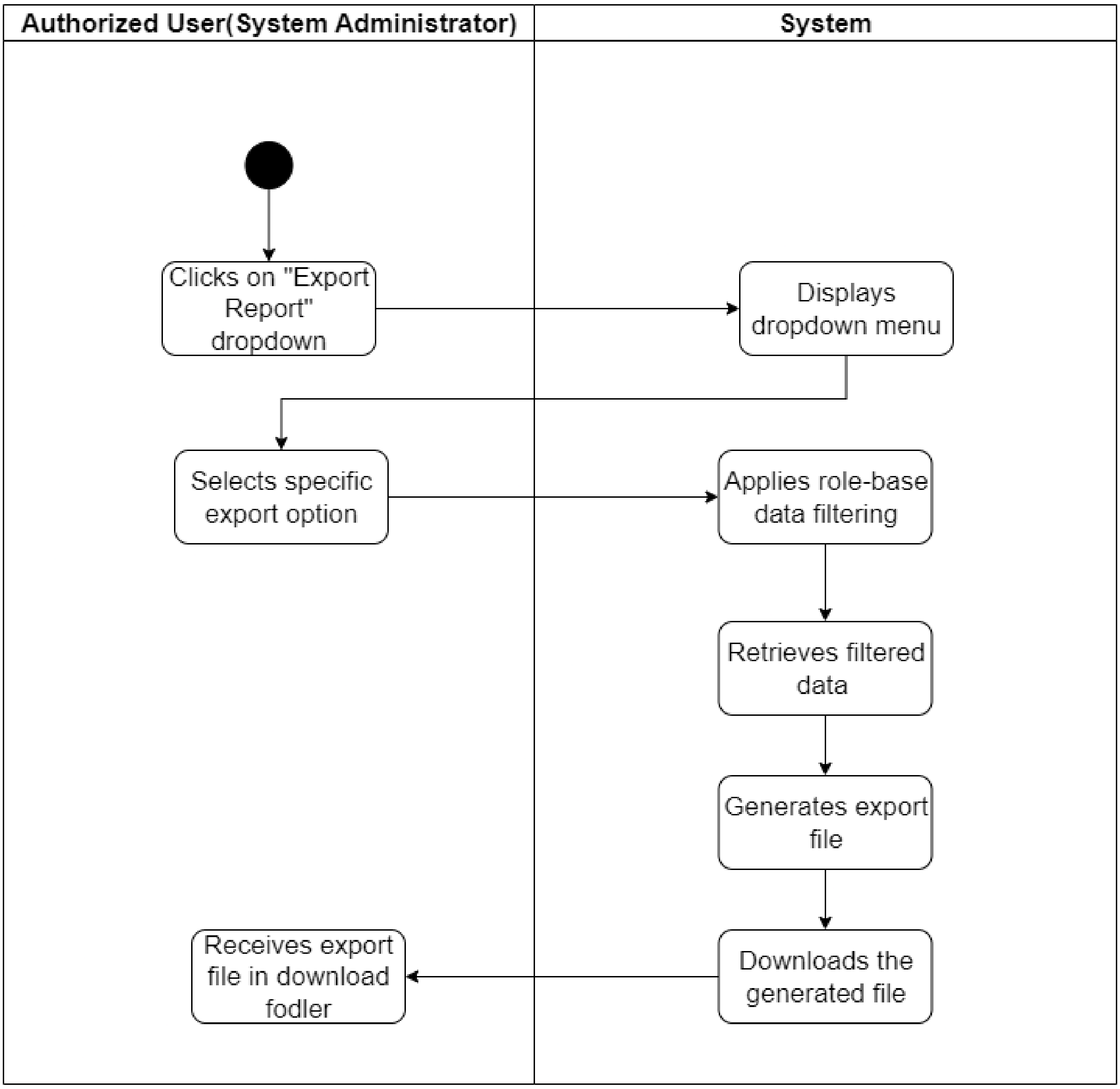


Figure 69: UC27_Export Reports

4.2.4.28 UC28_ List Public Users

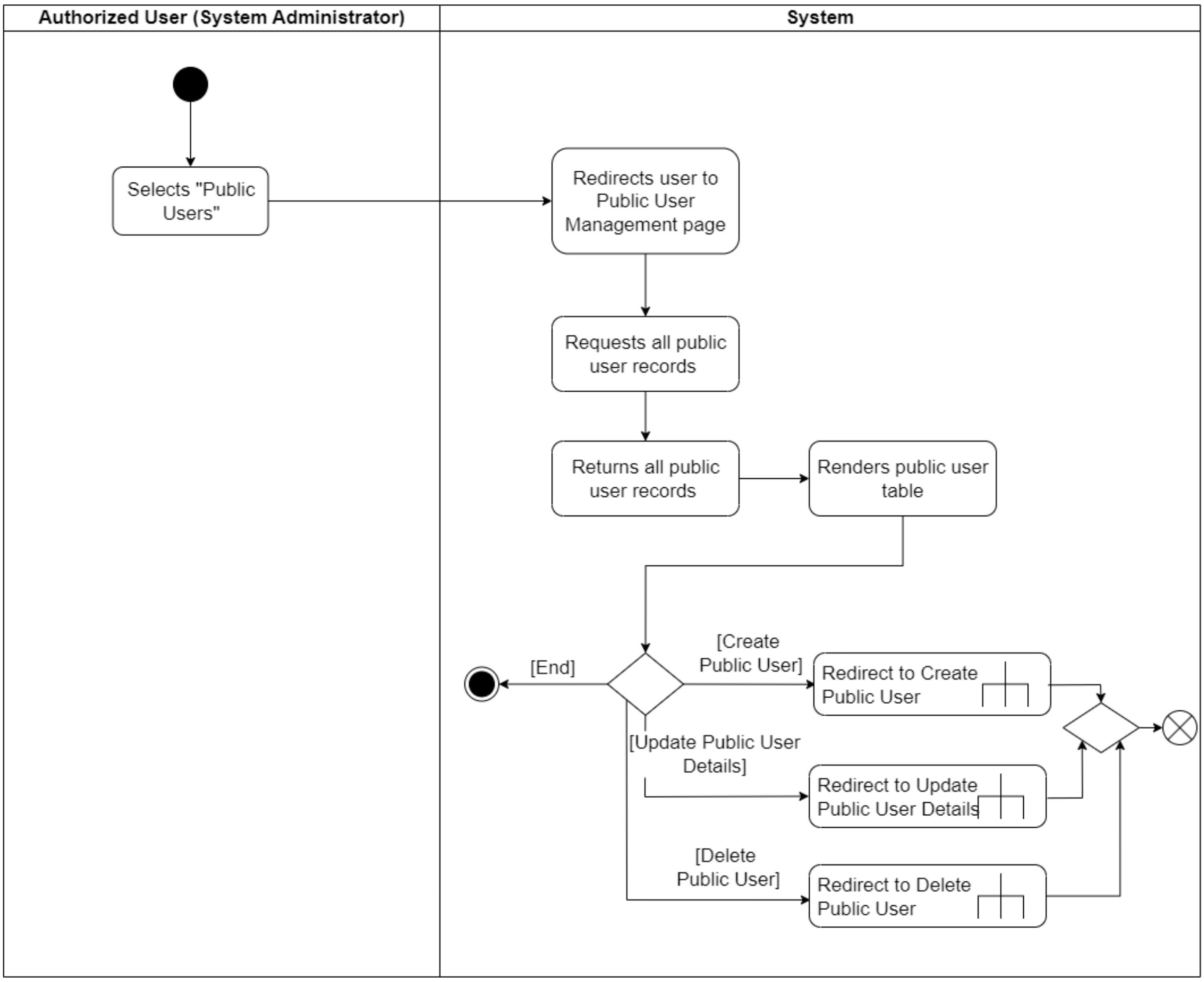


Figure 70: UC28_List Public Users

4.2.4.29 UC29_ Create Public User

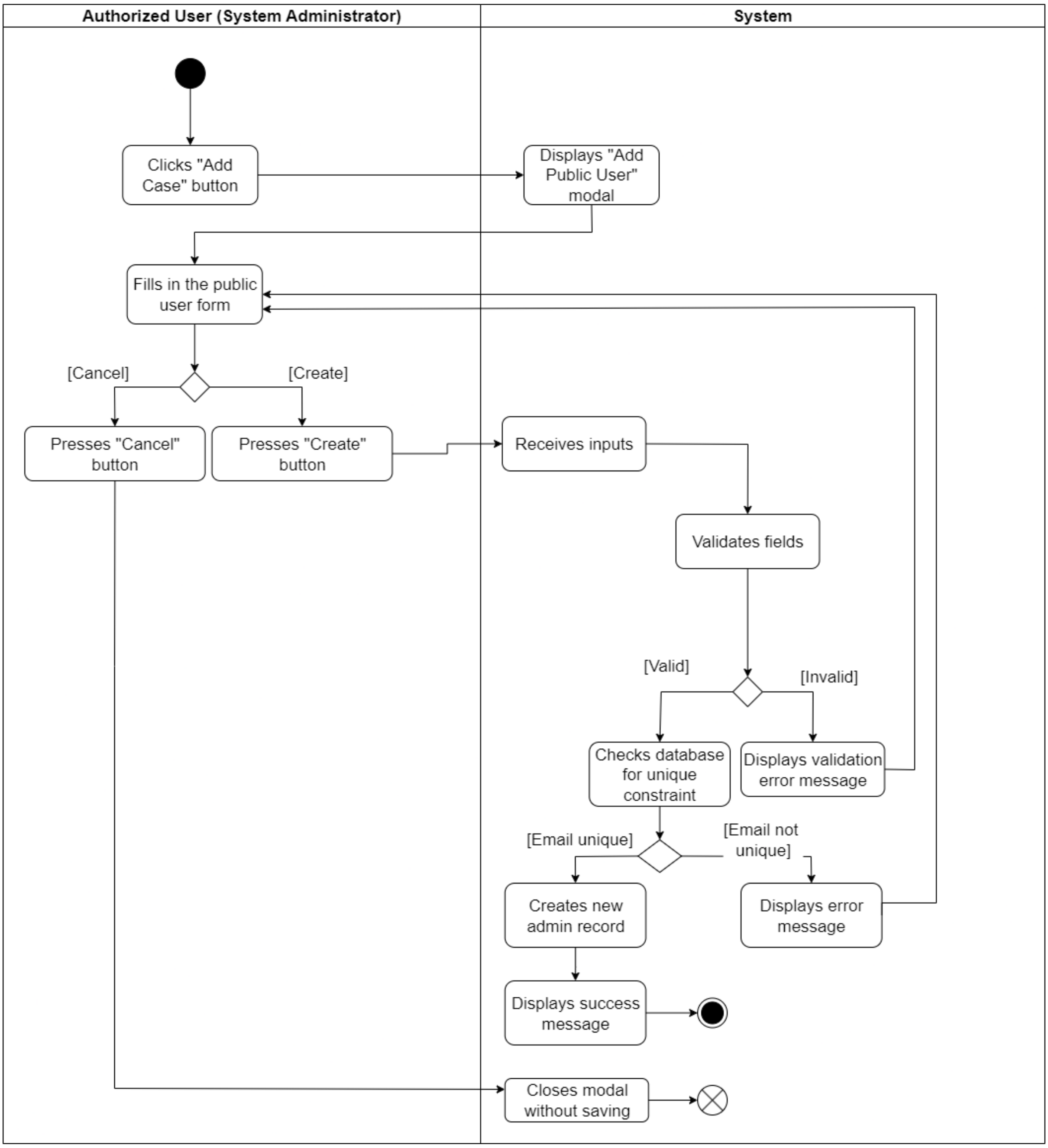


Figure 71: UC29_Create Public User

4.2.4.30 UC30_ Update Public User Details

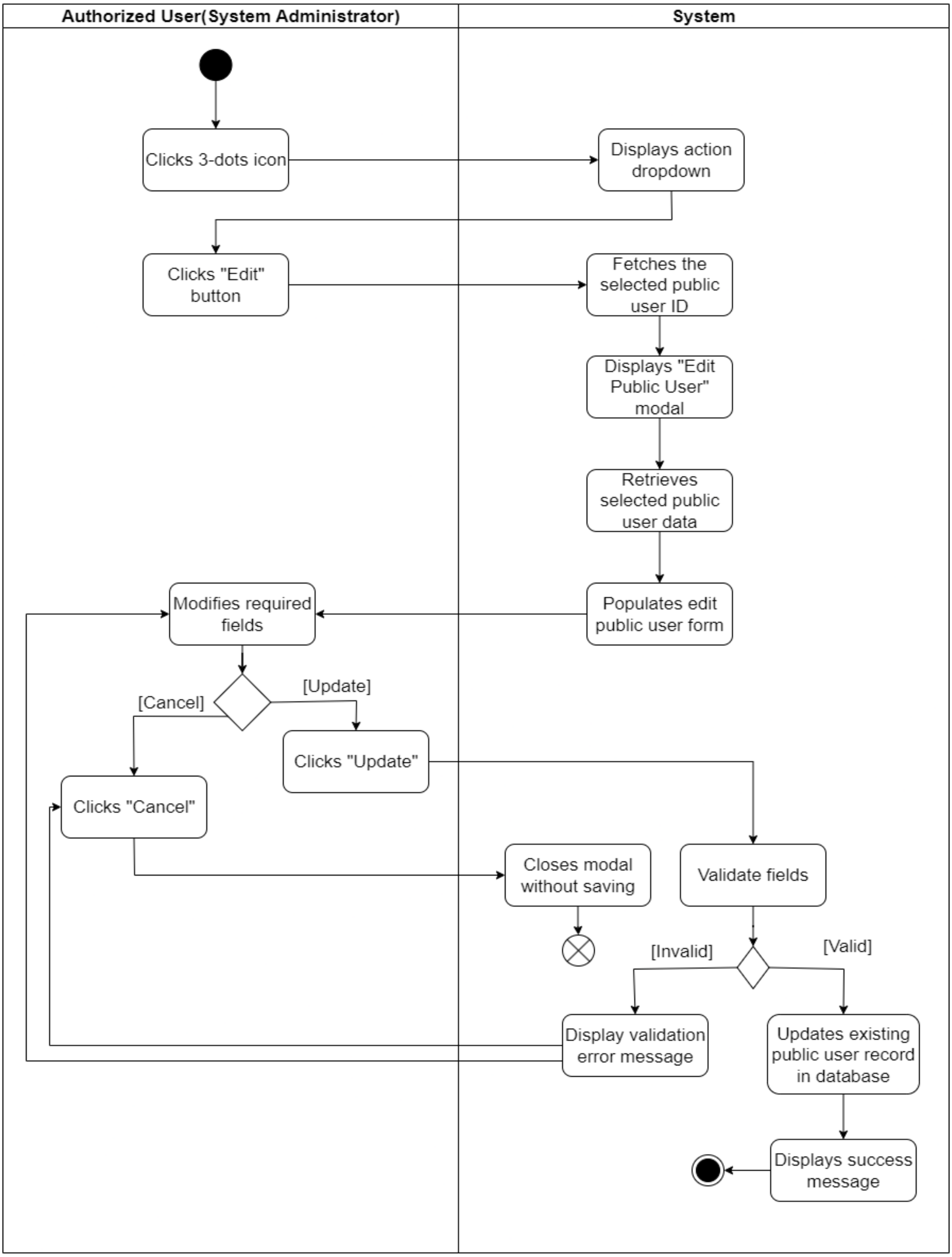


Figure 72: UC30_Update Public User Details

4.2.4.31 UC31_ Delete Public User

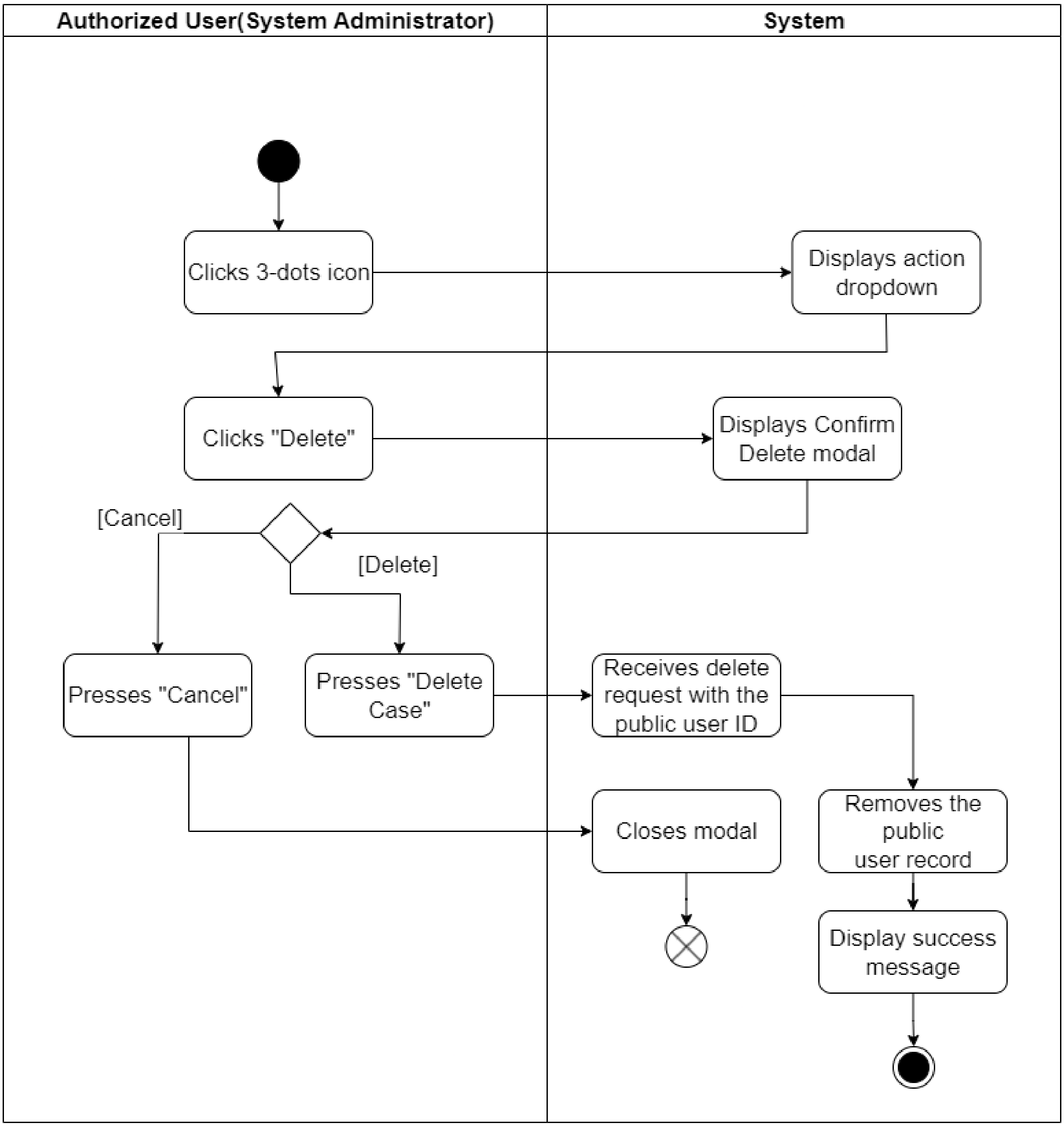


Figure 73: UC31_Delete Public User

4.2.4.32 UC32_ List Session Logs

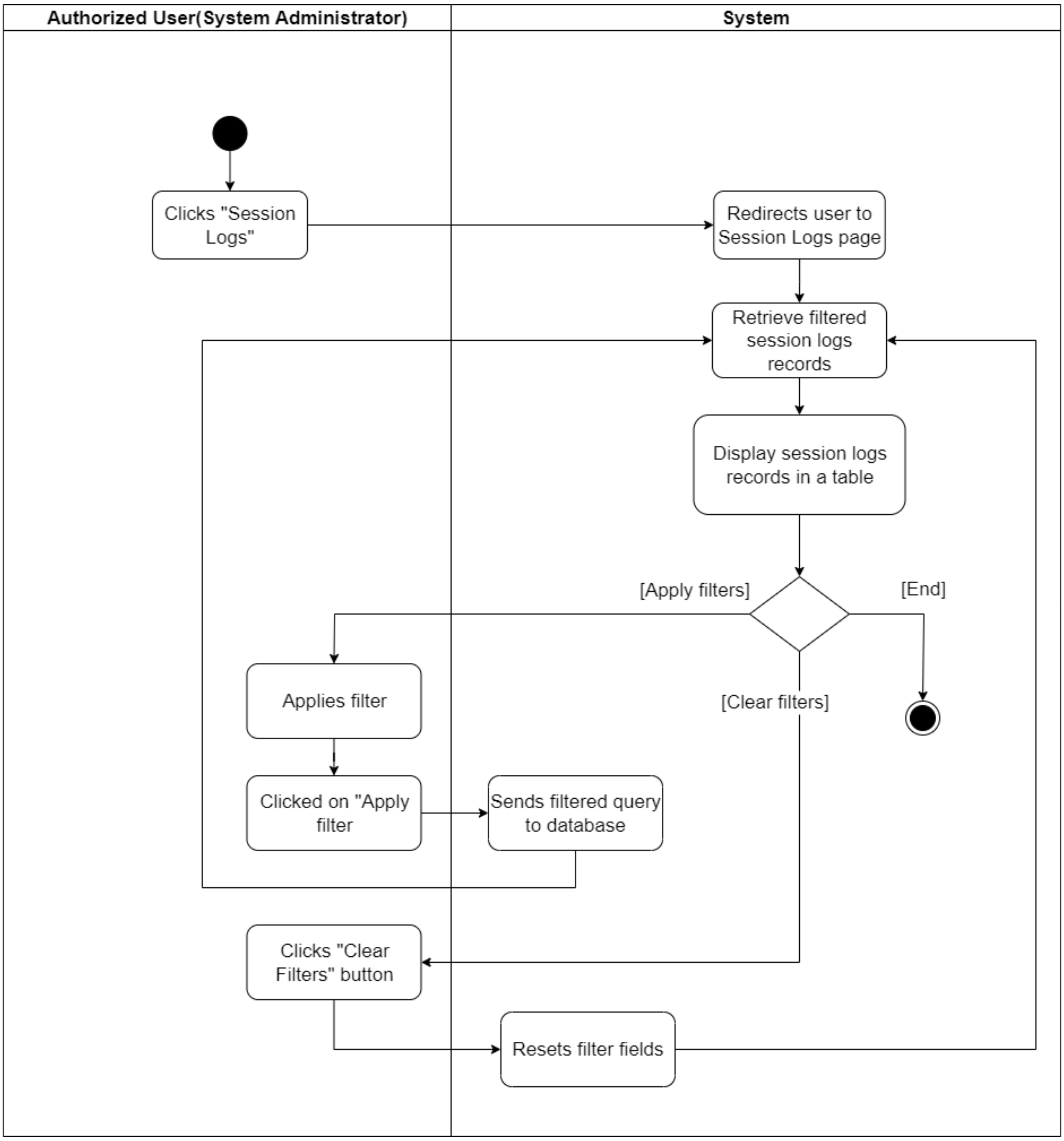


Figure 74: List Session Logs

4.2.4.33 UC33_ List Law Enforcements

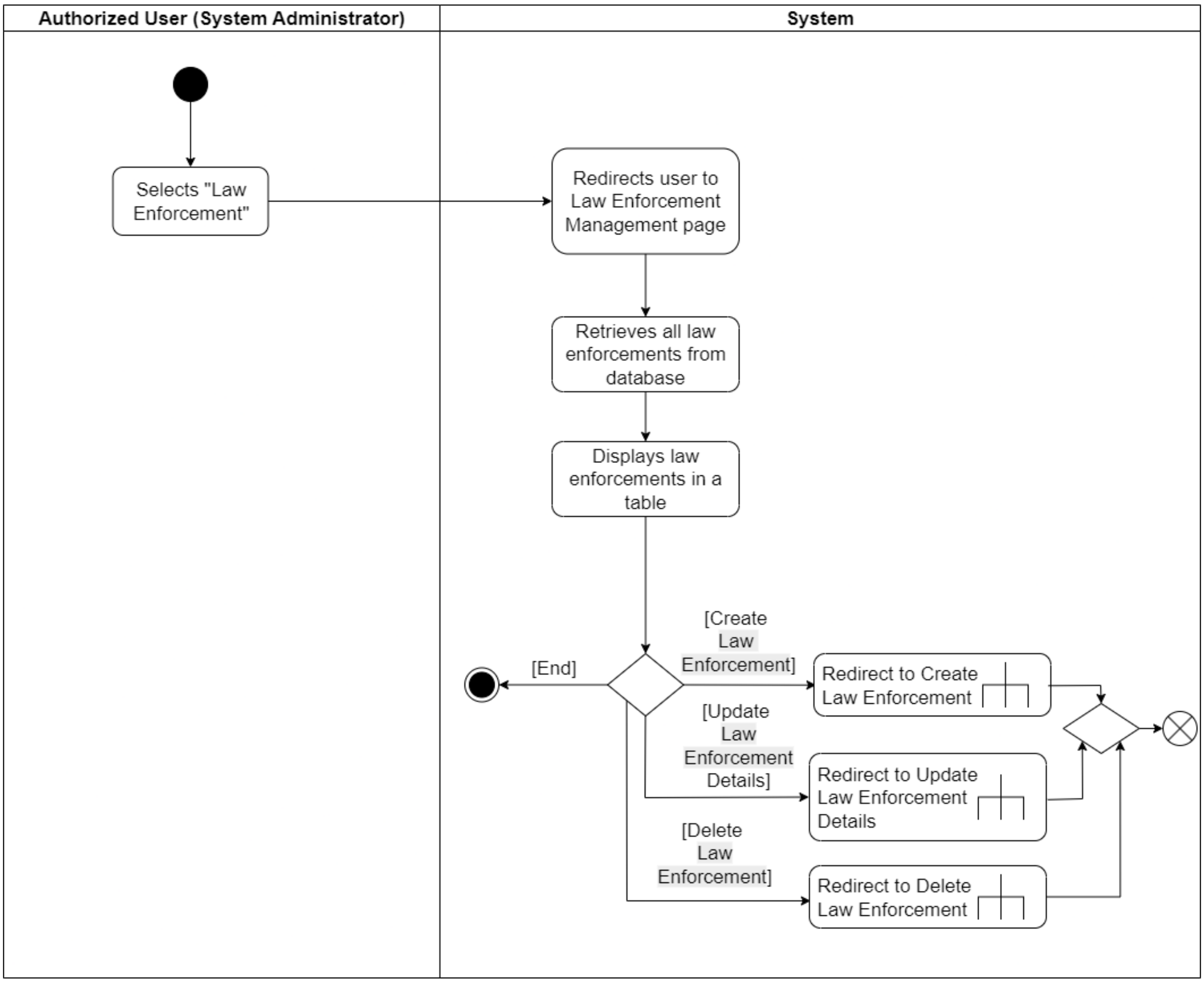


Figure 75: UC33_List Law Enforcement

4.2.4.34 UC34_ Create Law Enforcement

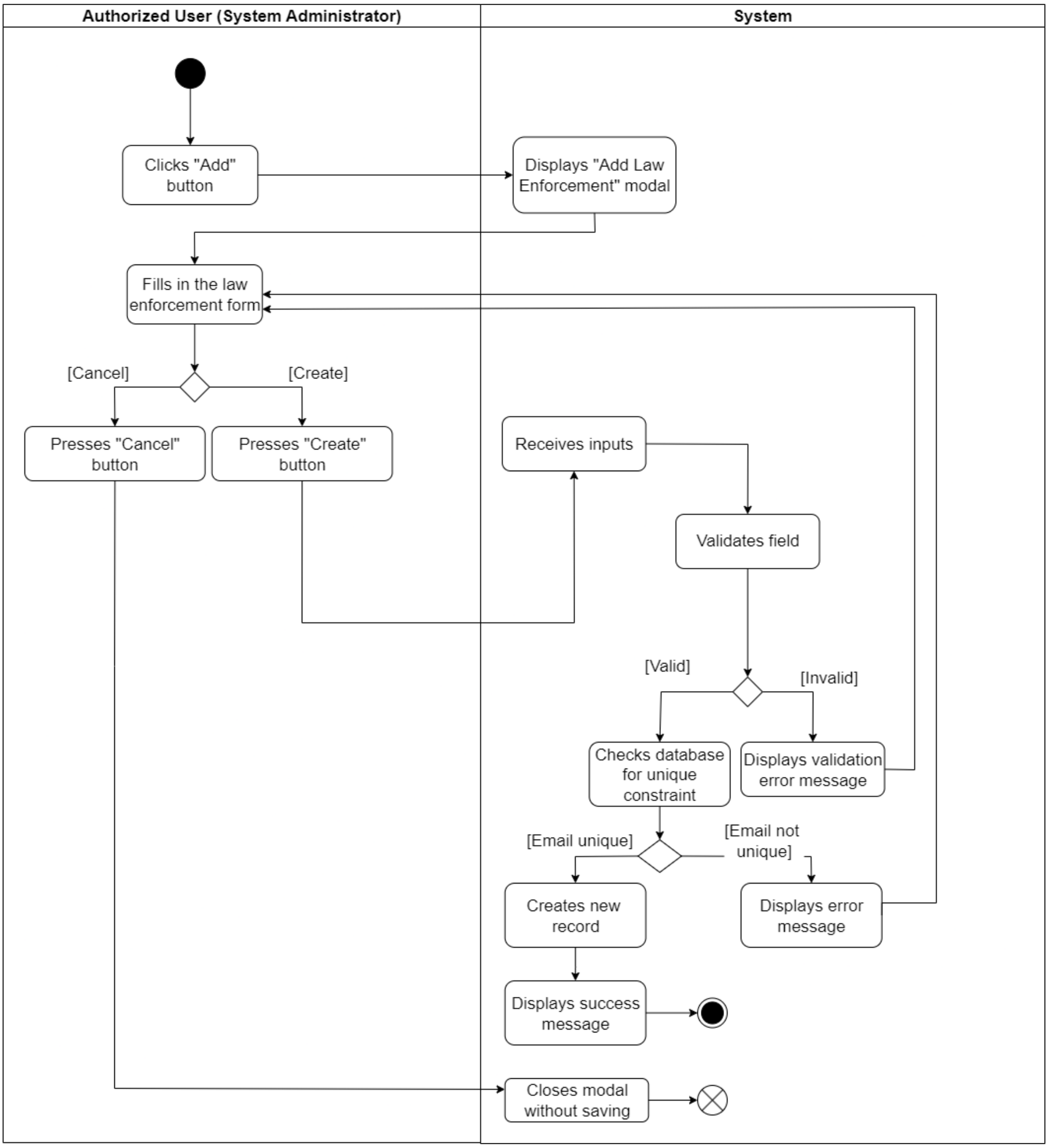


Figure 76: UC34_Create Law Enforcement

4.2.4.35 UC35_ Update Law Enforcement Details

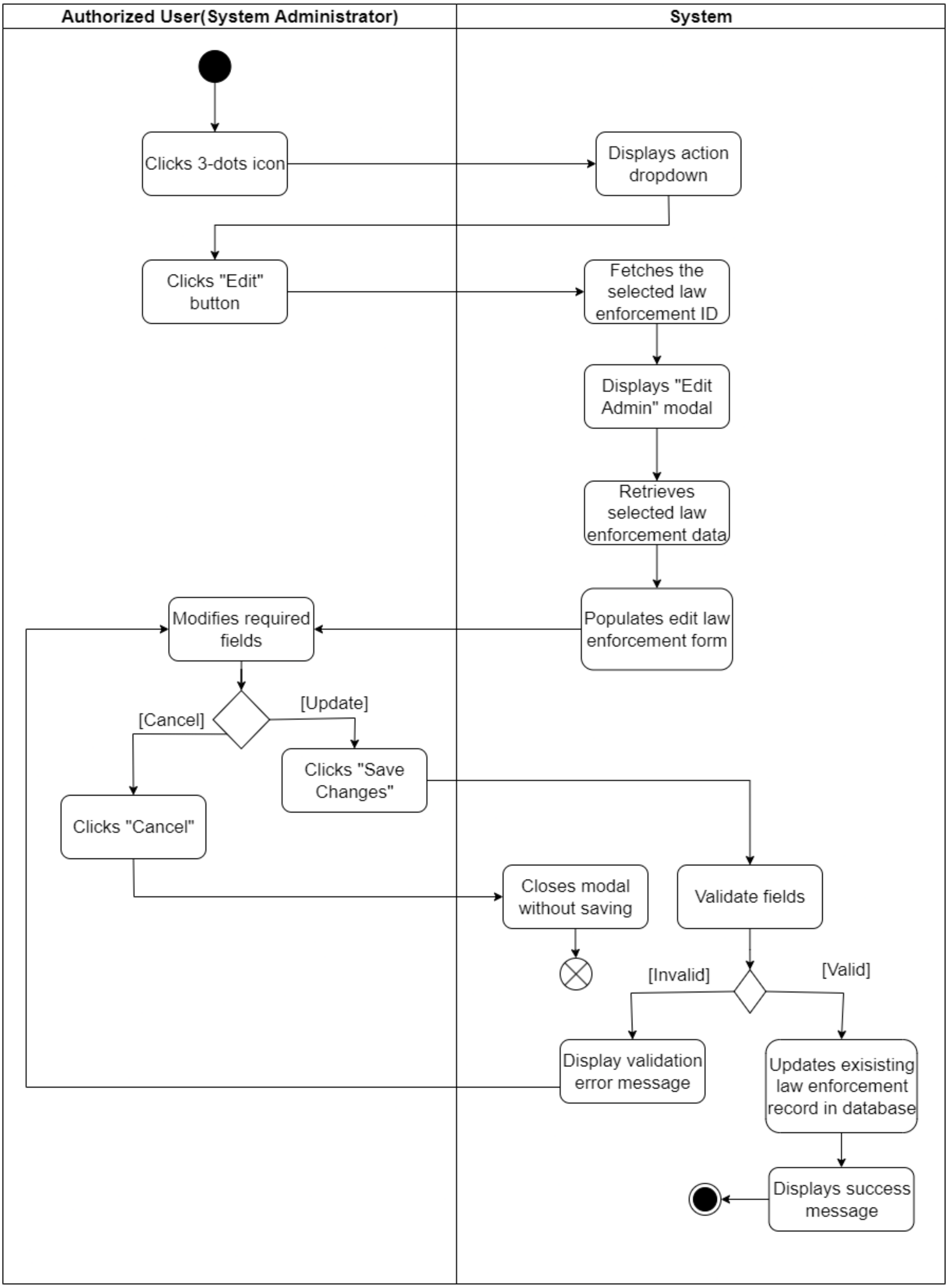


Figure 77: UC35_Update Law Enforcement Details

4.2.4.36 UC36_ Delete Law Enforcement

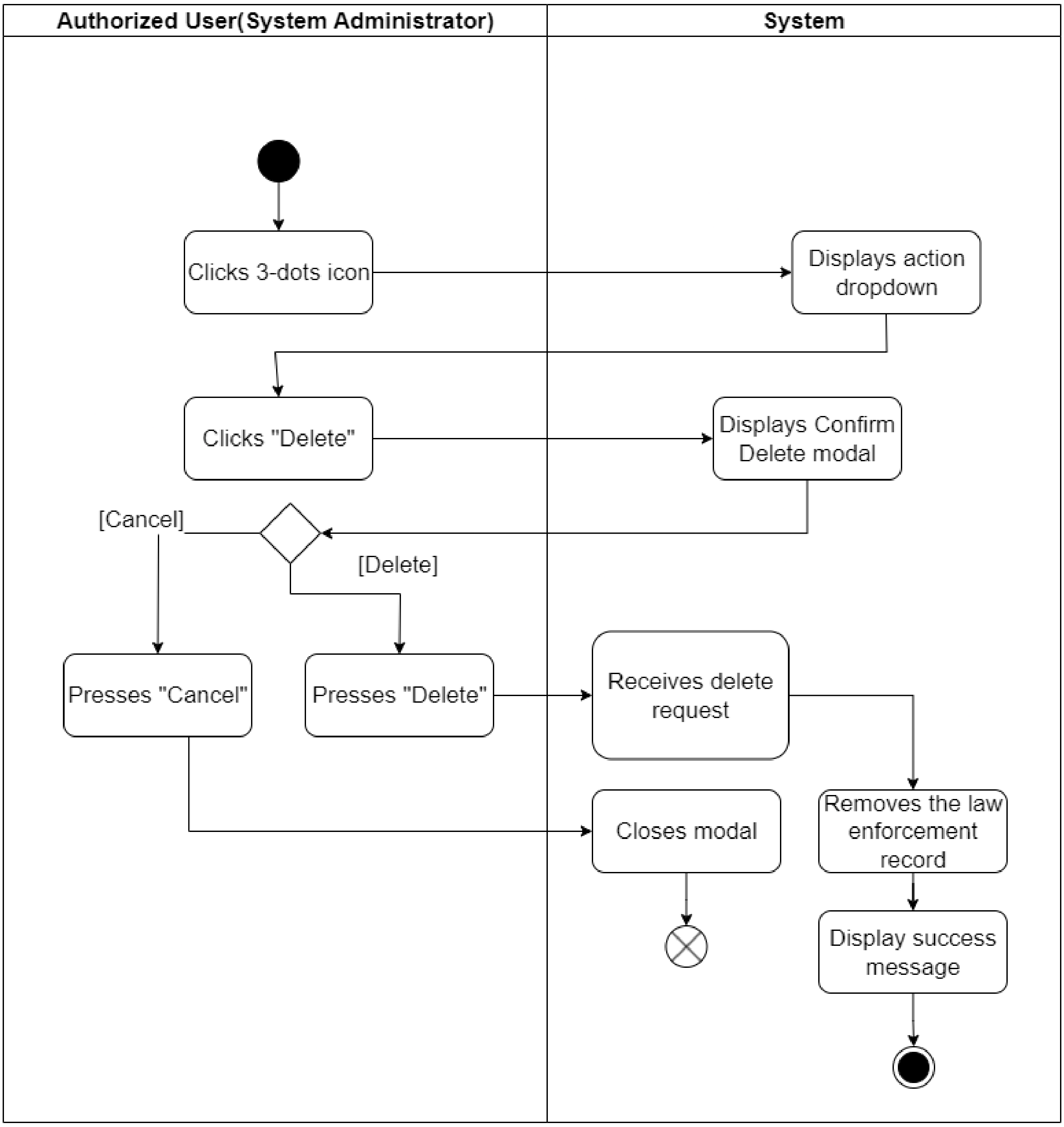


Figure 78: UC36_Delete Law Enforcement

4.2.4.37 UC37_ List Social Workers

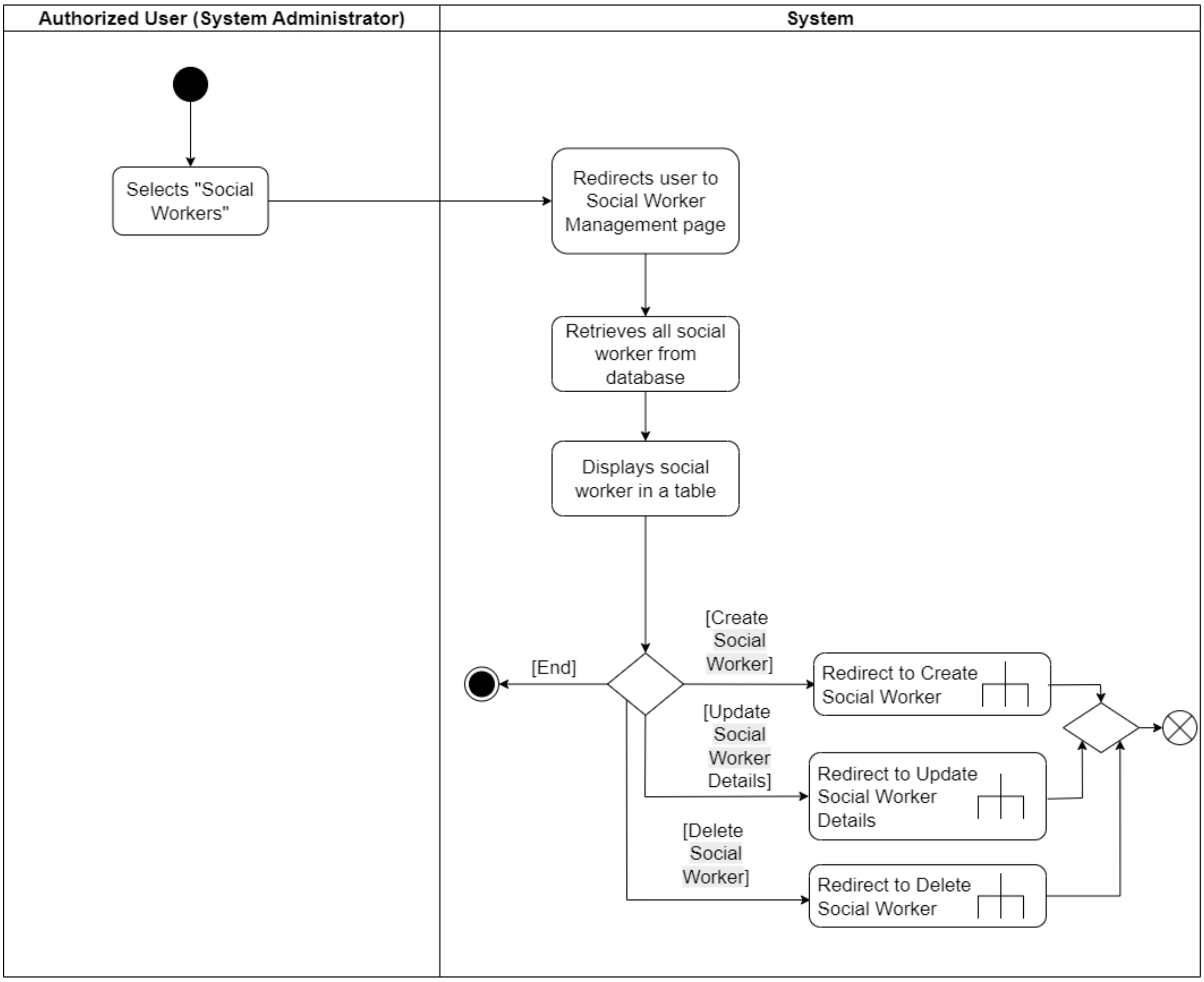


Figure 79: UC37_List Social Workers

4.2.4.38 UC38_ Create Social Worker

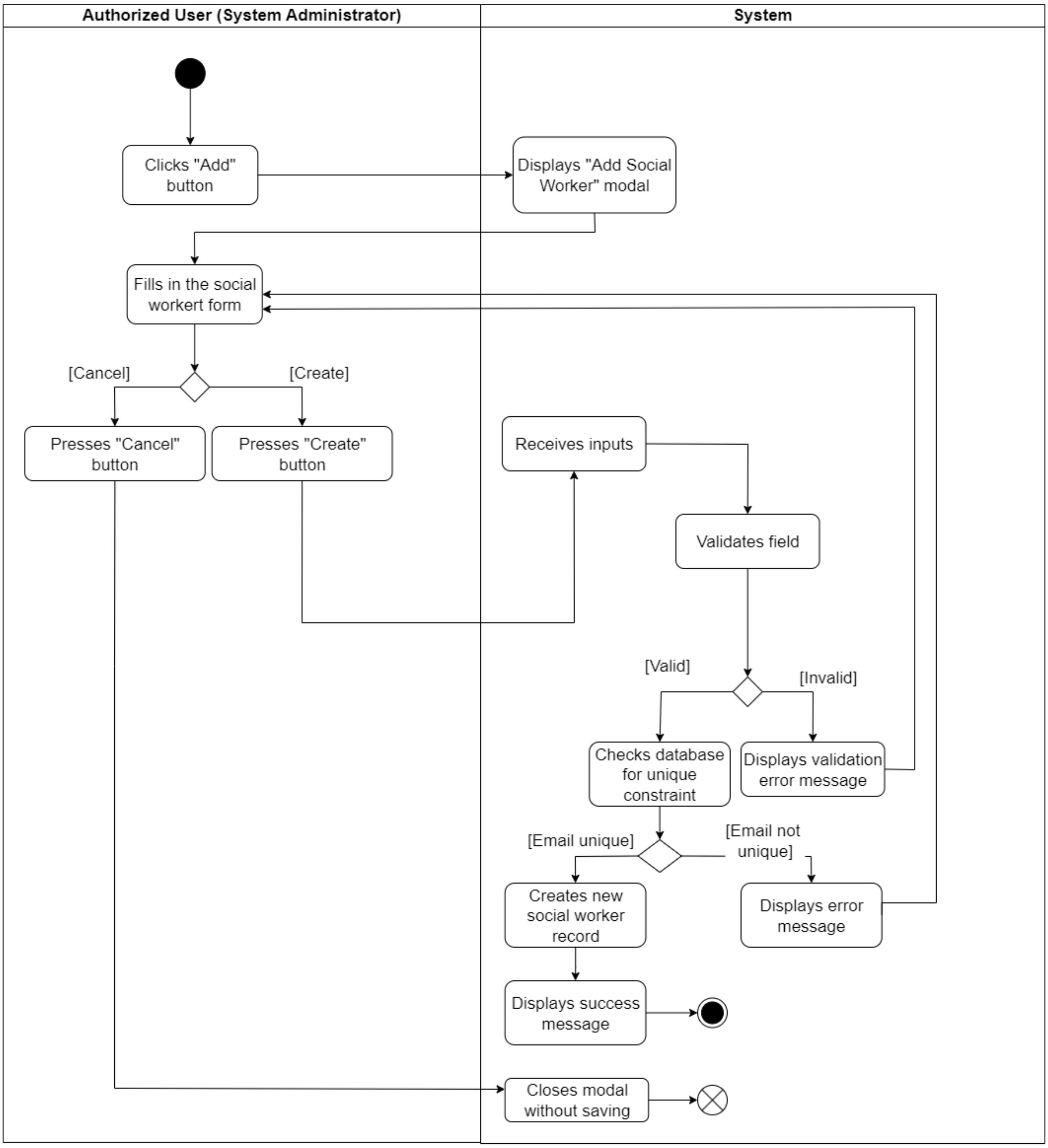


Figure 80: UC38_ List Social Worker

4.2.4.39 UC39_ Update Social Worker Details

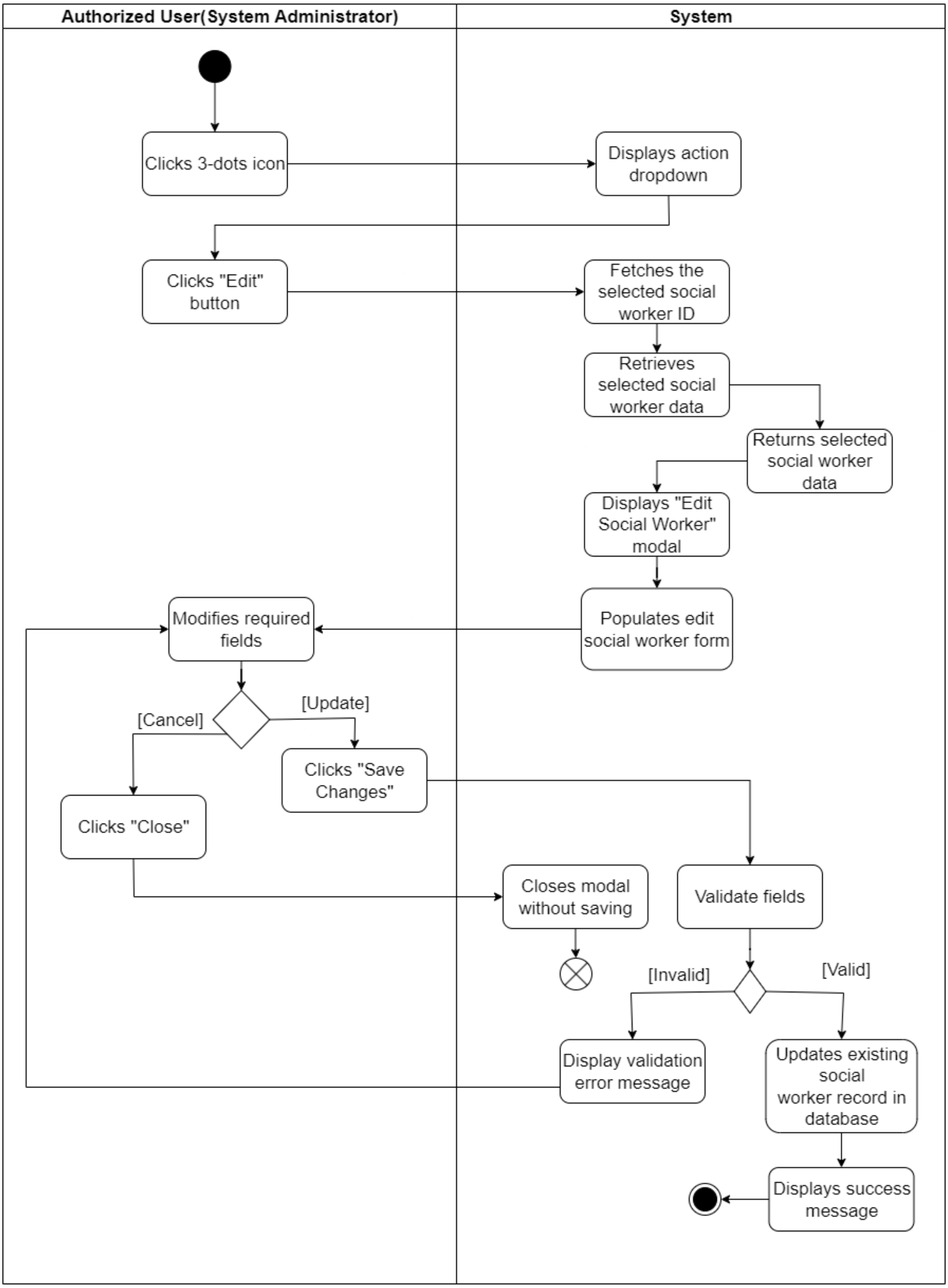


Figure 81: UC39_Update Social Worker Details

4.2.4.40 UC40_ Delete Social Worker

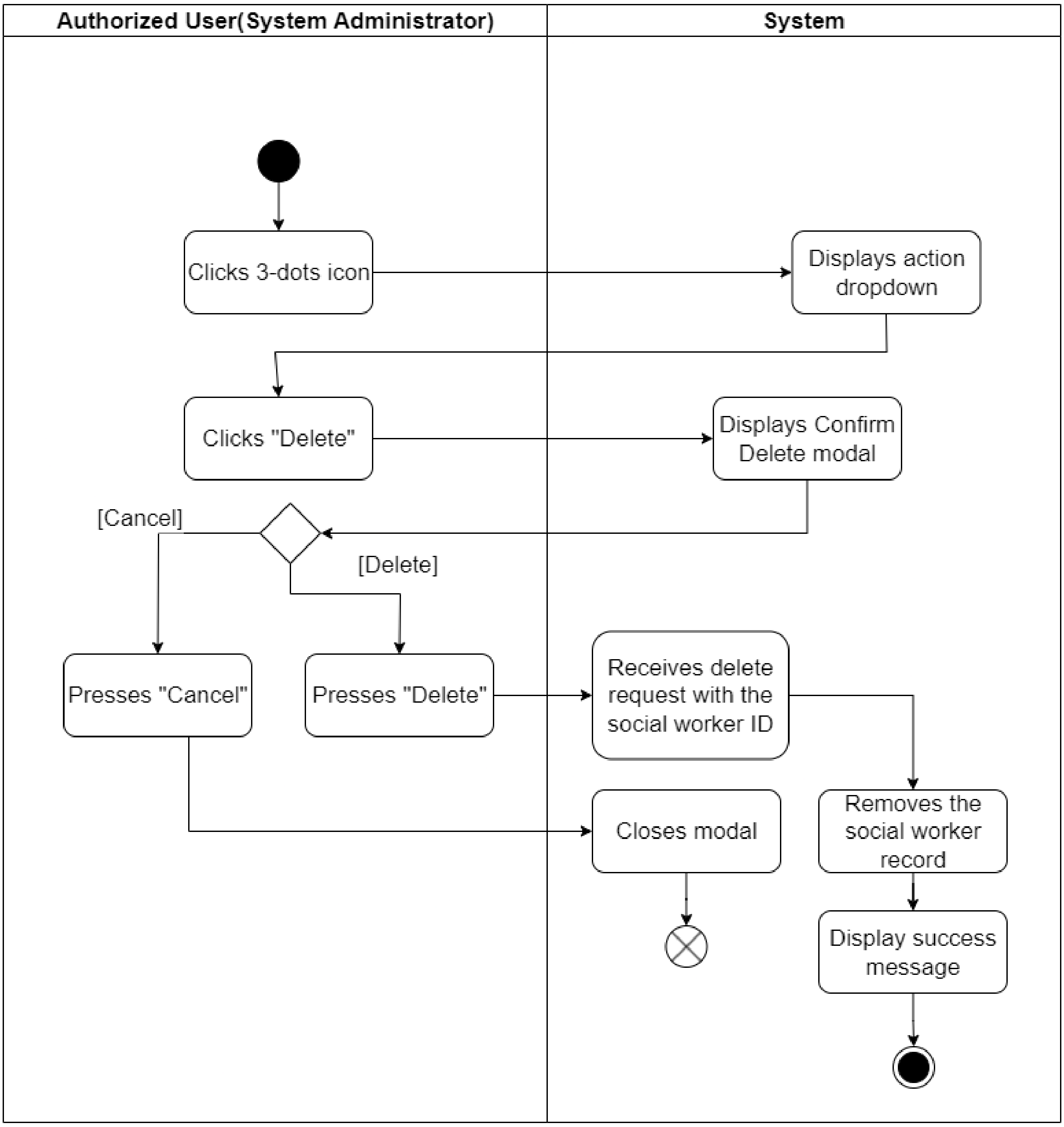


Figure 82: UC40_Delete Social Worker

4.2.4.41 UC41_ List Chief Welfare Officers

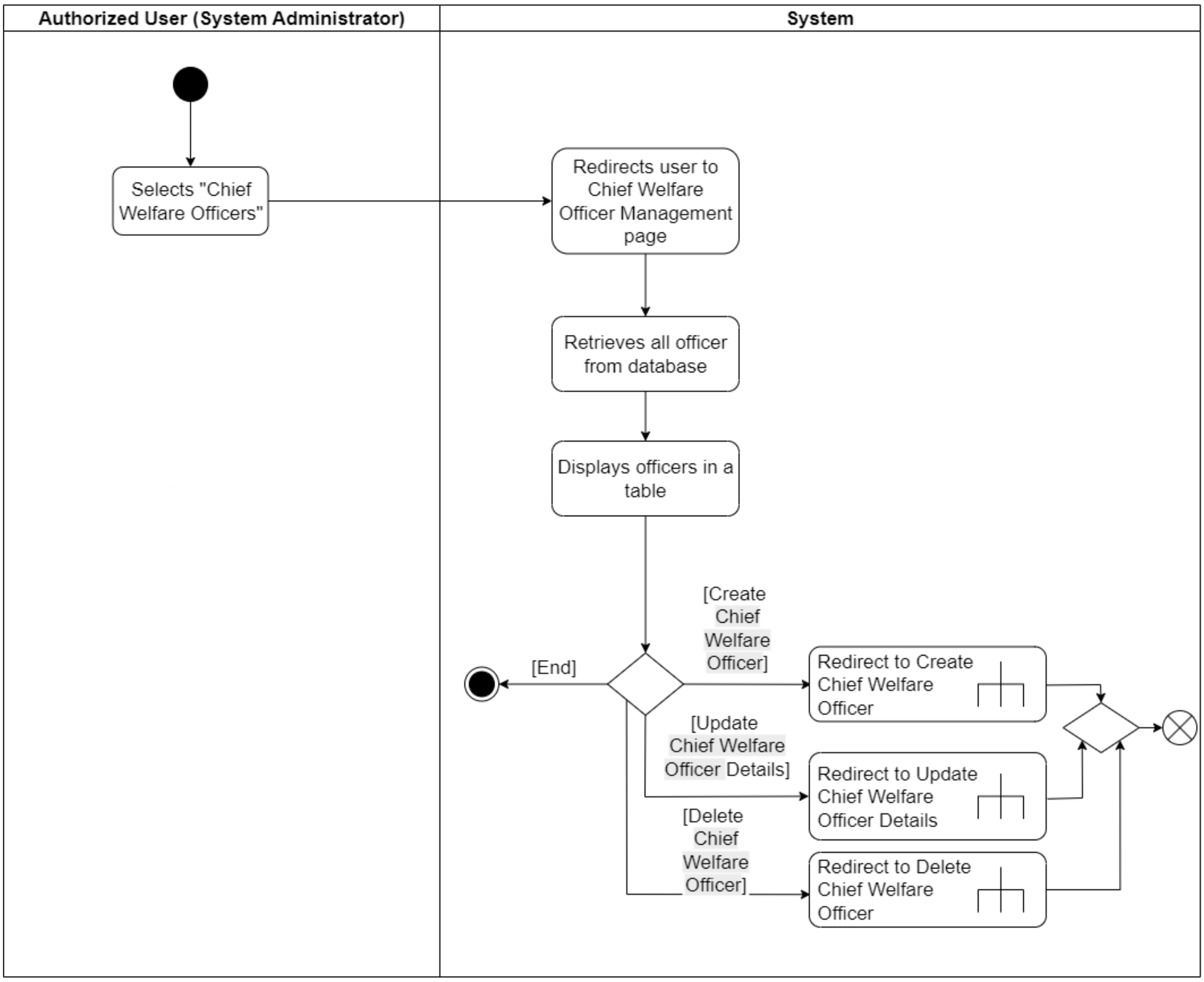


Figure 83: UC41_List Chief Welfare Officers

4.2.4.42 UC42_ Create Chief Welfare Officer

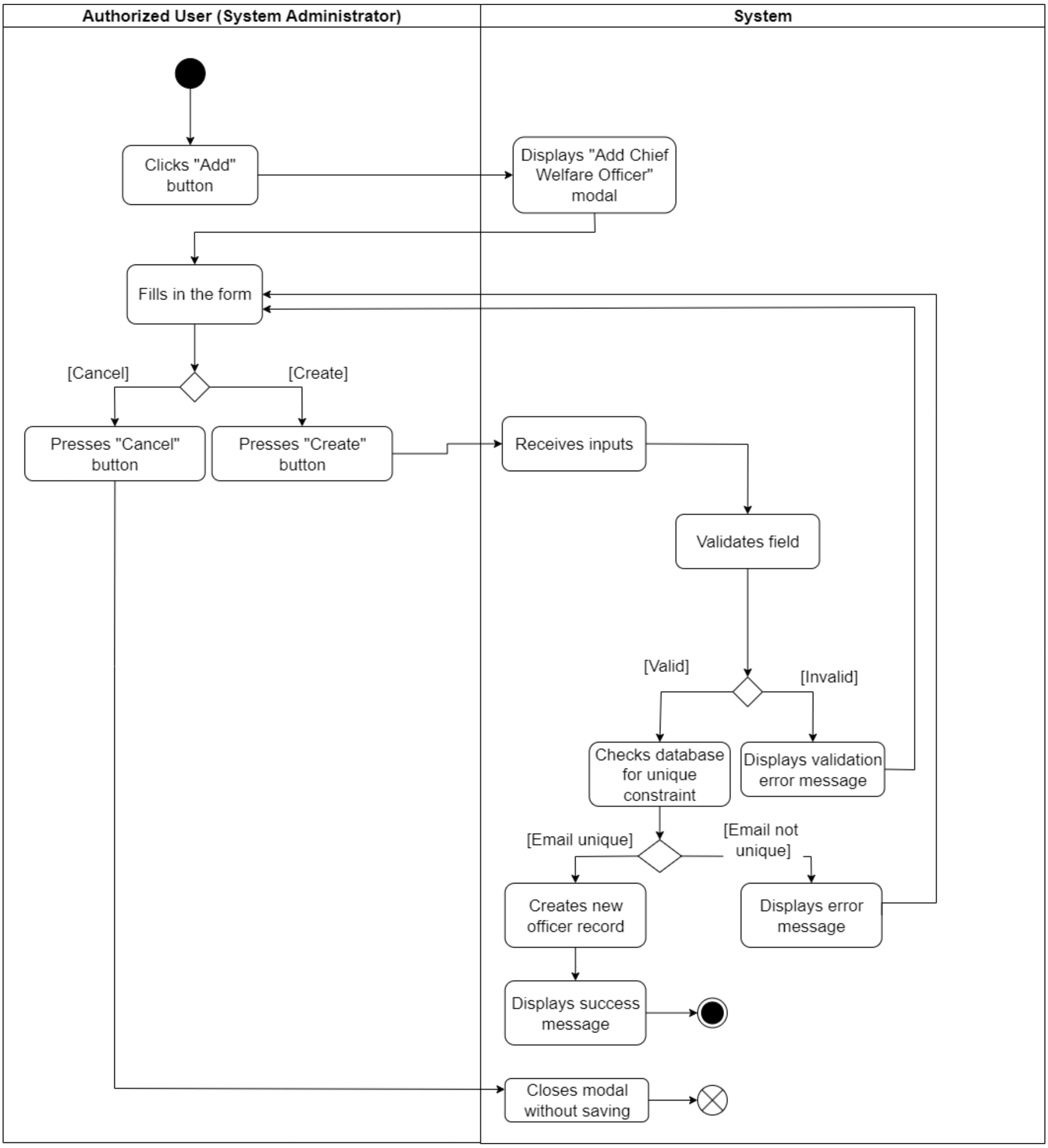


Figure 84: UC42_Create Chief Welfare Officer

4.2.4.43 UC43_ Update Chief Welfare Officer Details

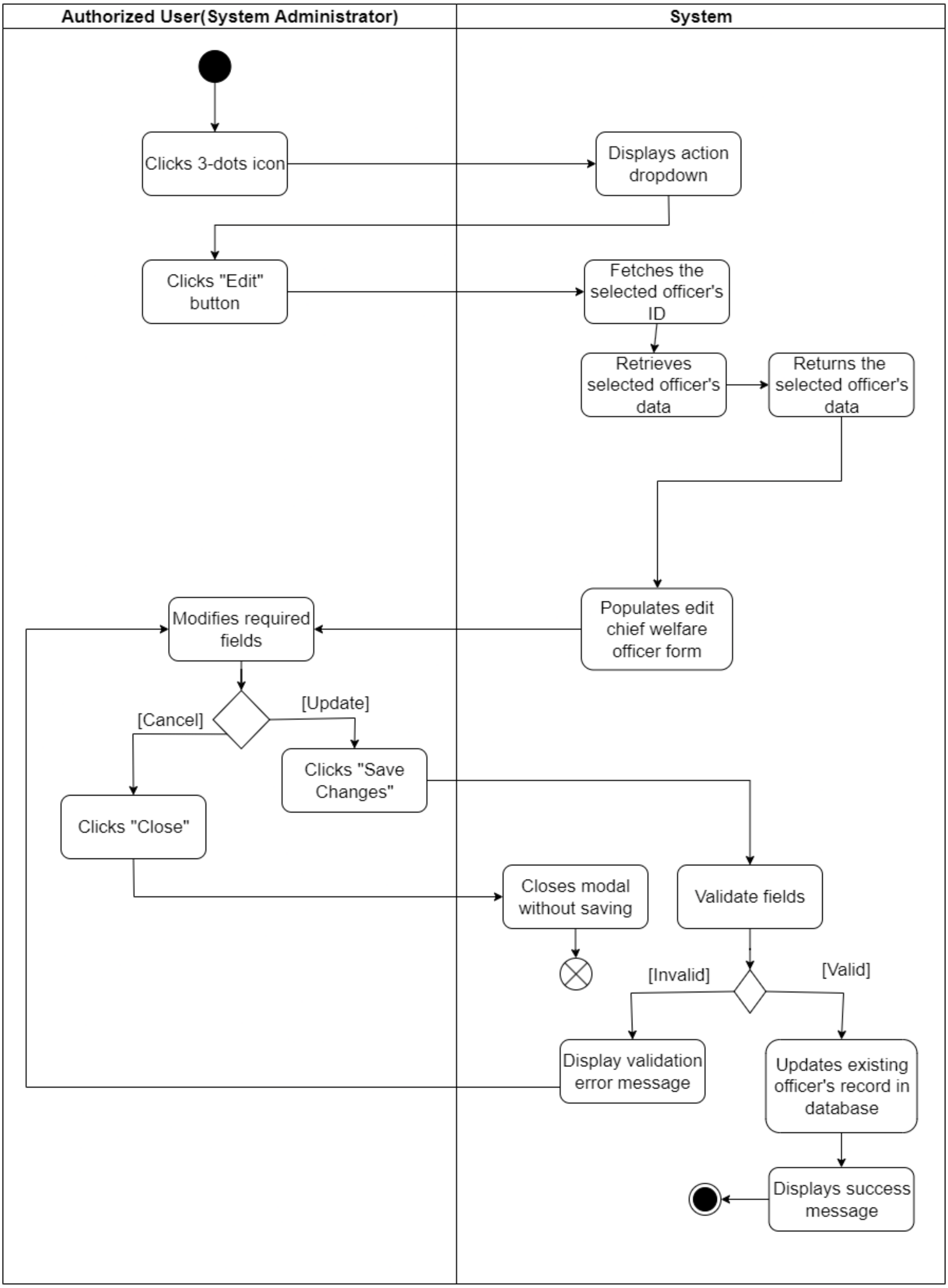


Figure 85: UC43_Update Chief Welfare Officer Details

4.2.4.44 UC44_ Delete Chief Welfare Officer

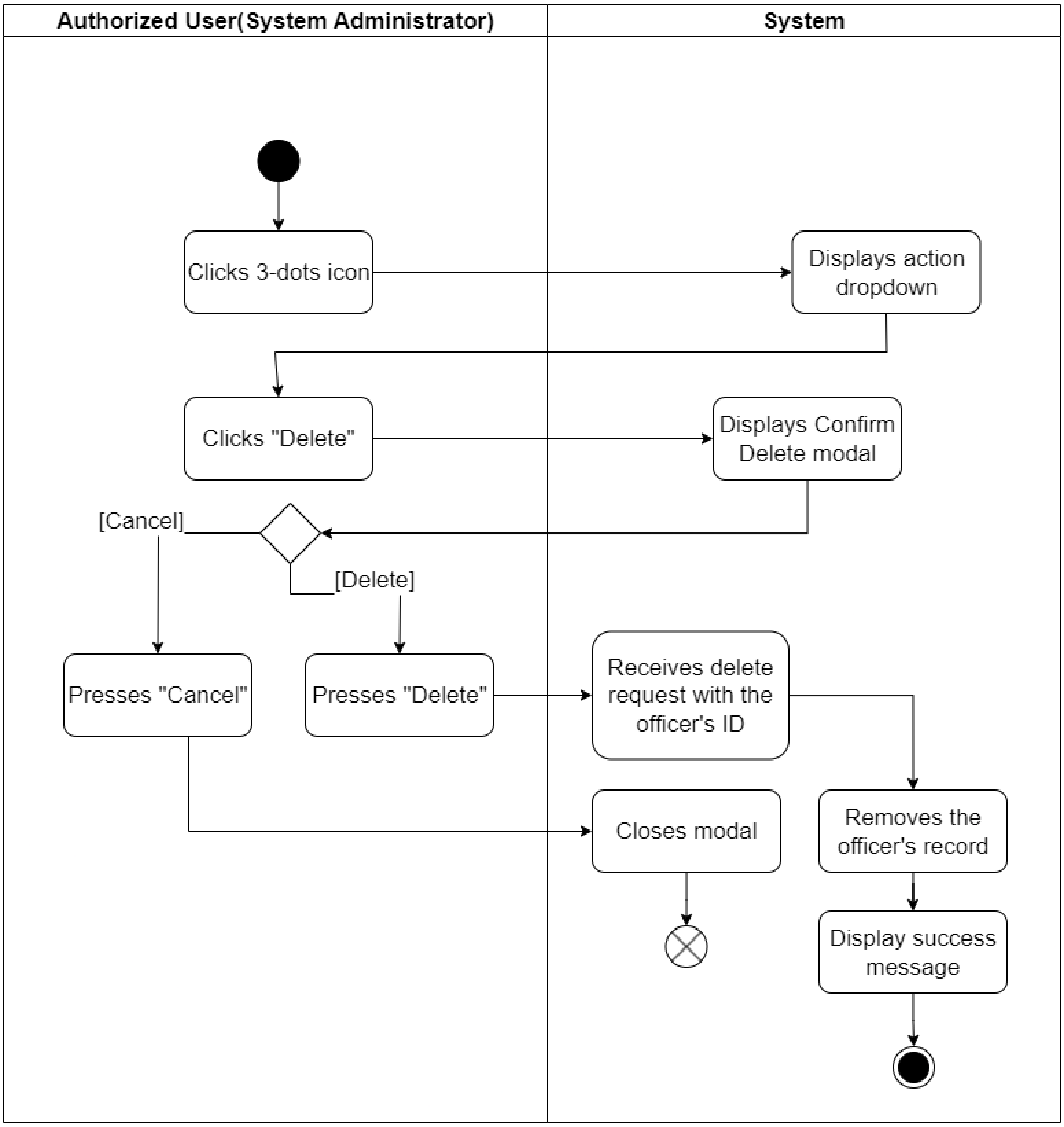


Figure 86: UC44_Delete Chief Welfare Officer

4.2.4.45 UC45_ List Healthcare Professionals

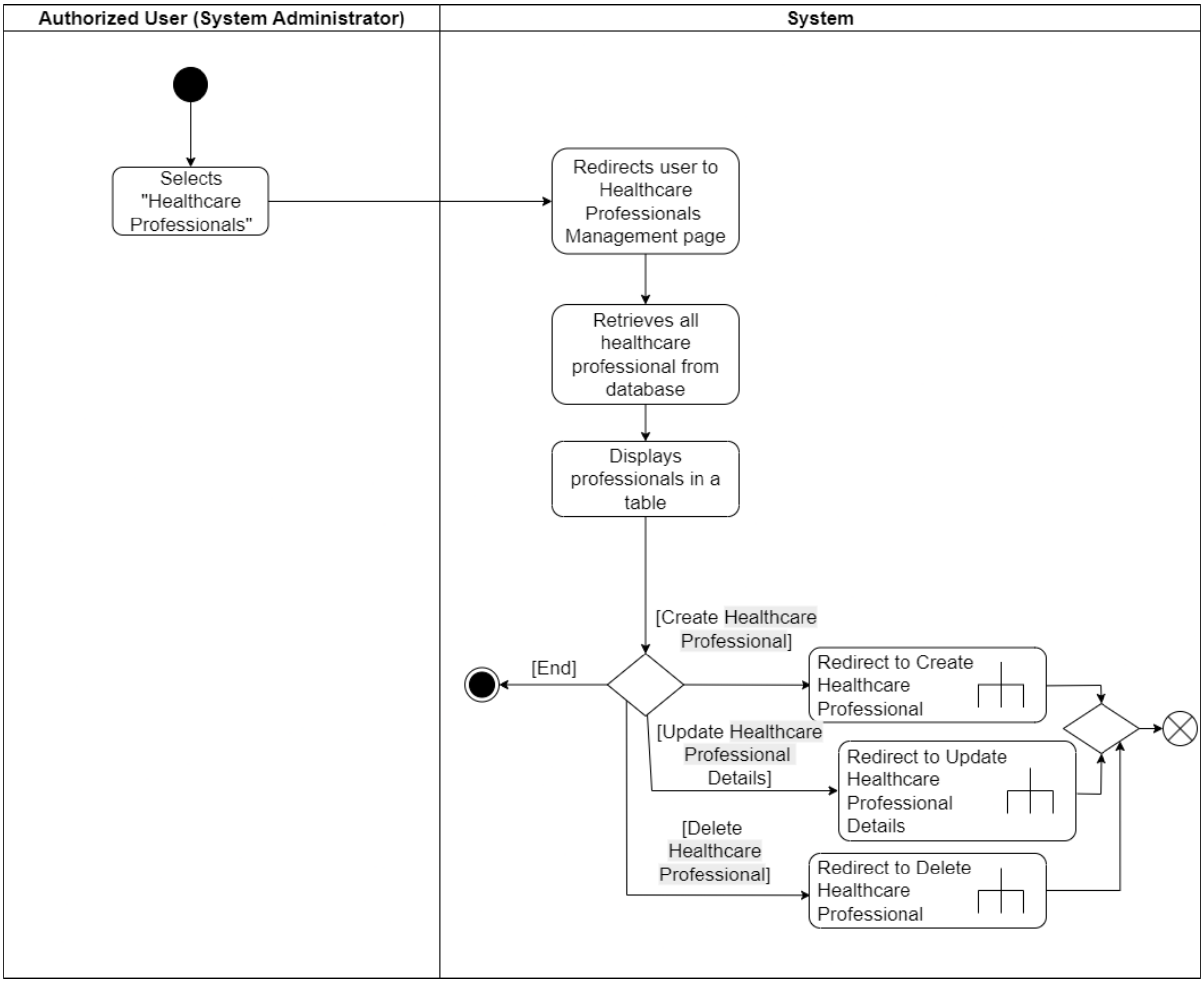


Figure 87: UC45_List Healthcare Professionals

4.2.4.46 UC46_ Create Healthcare Professional

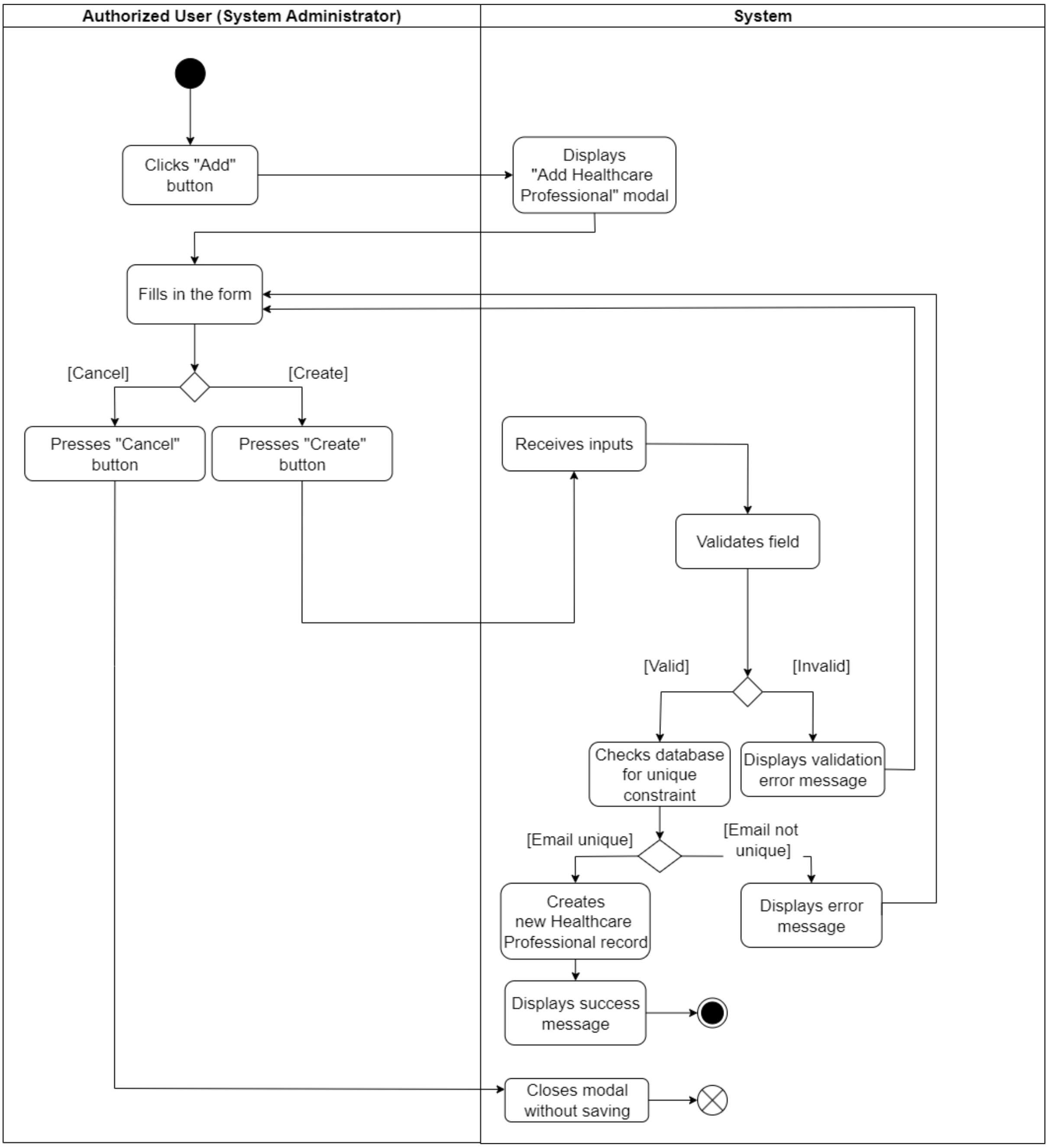


Figure 88: UC46_Create Healthcare Professional

4.2.4.47 UC47_ Update Healthcare Professional Details

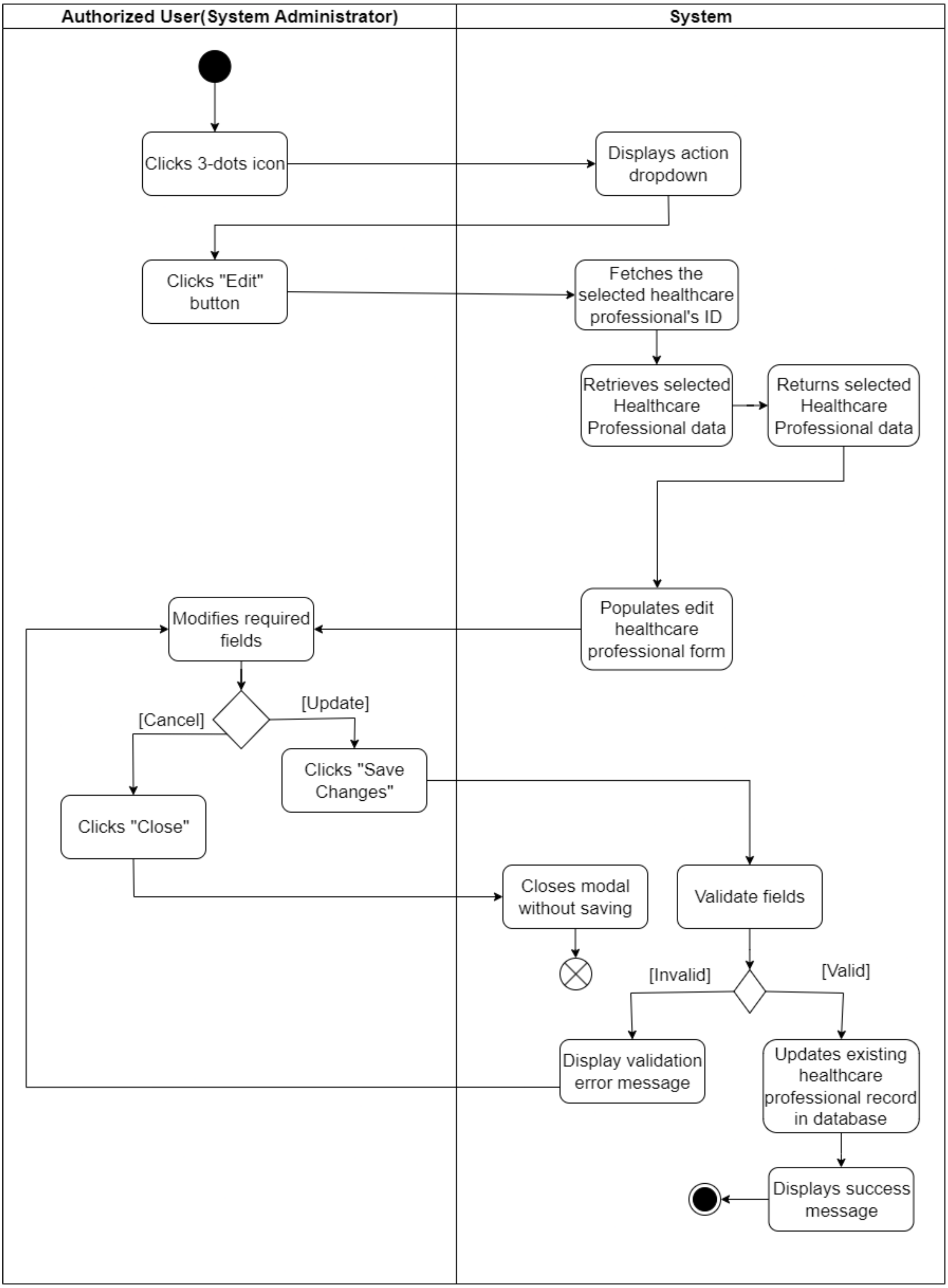


Figure 89: UC47_Update Healthcare Professional Details

4.2.4.48 UC48_ Delete Healthcare Professional

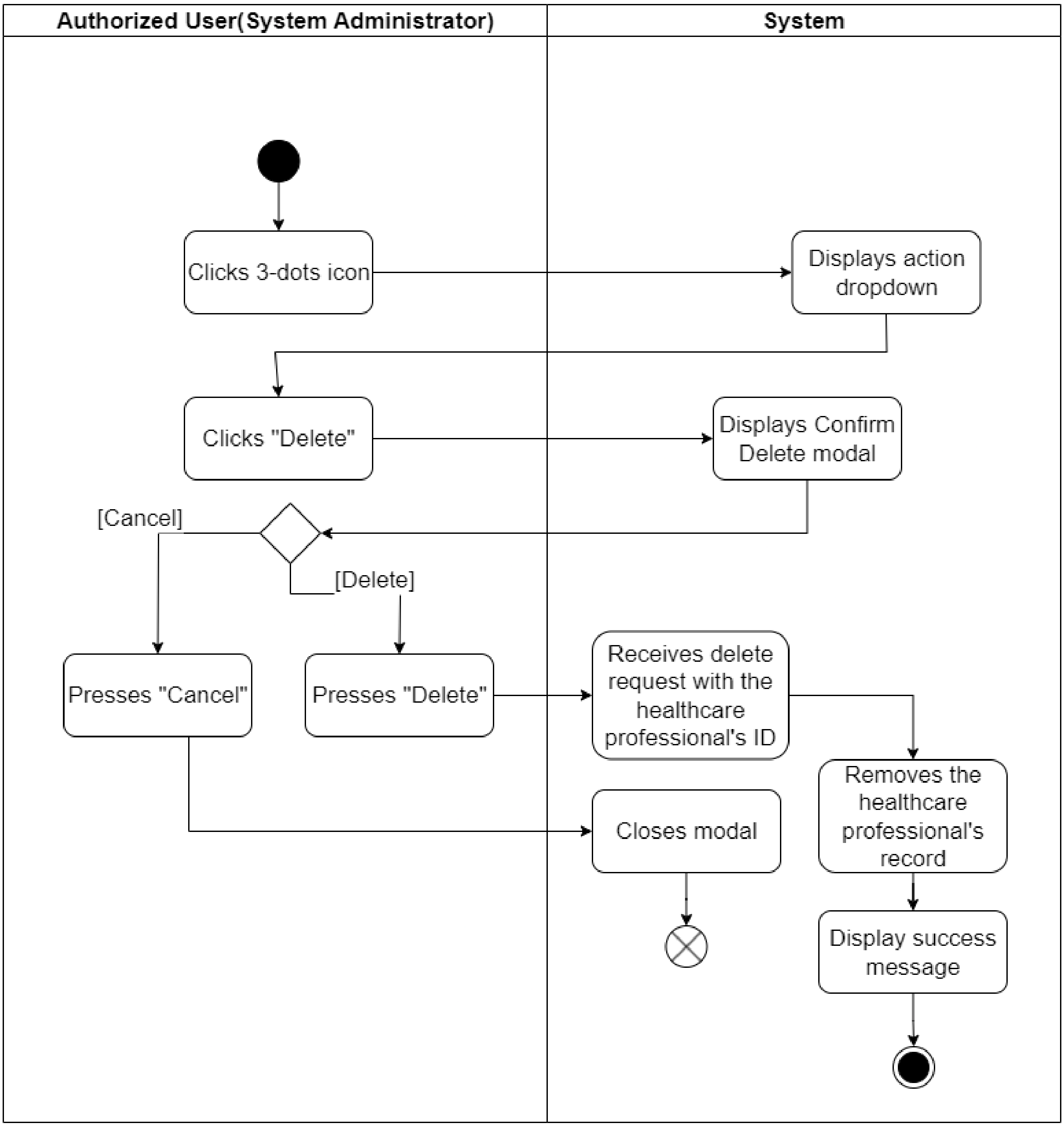


Figure 90: UC48_Delete Healthcare Professional

4.2.4.49 UC49_ List Permissions

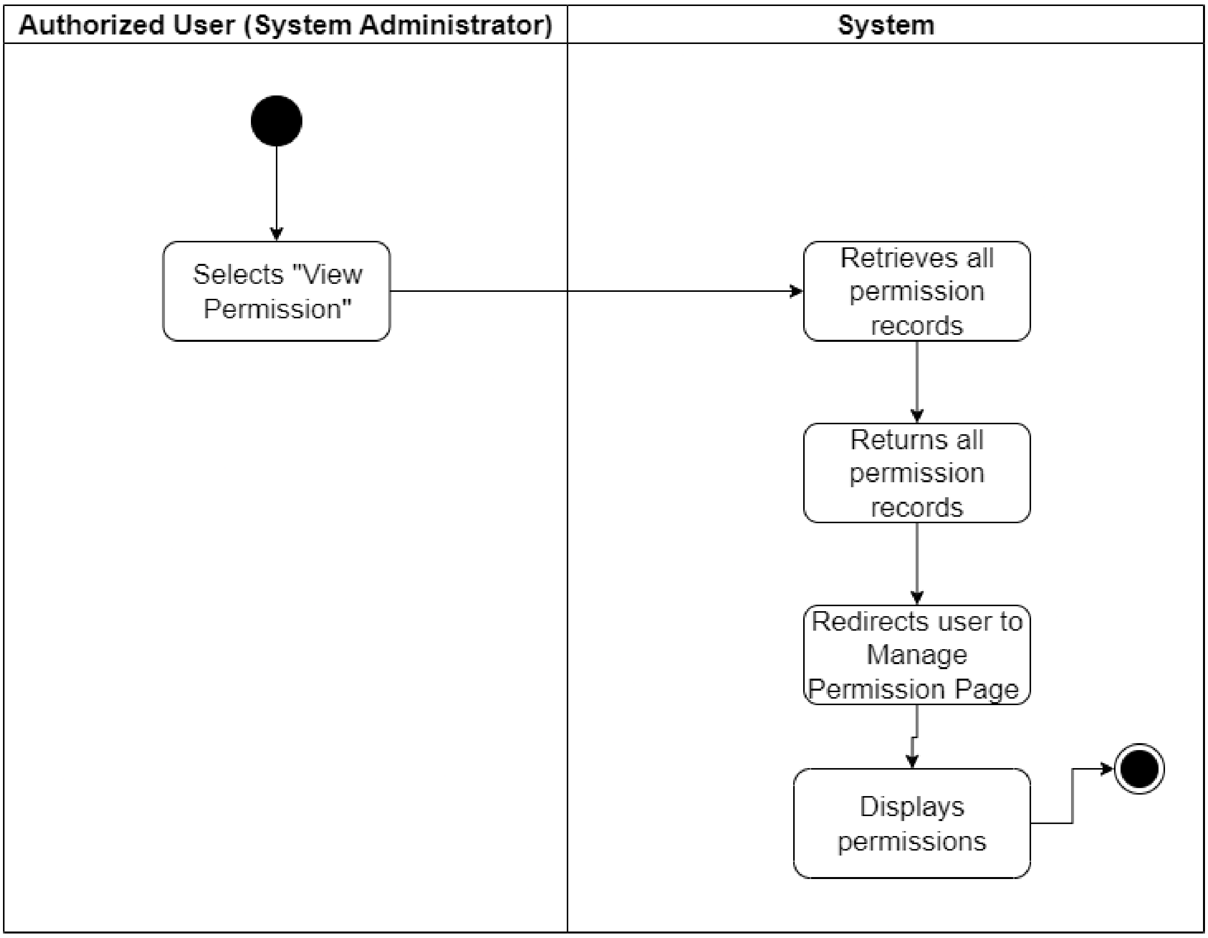


Figure 91: UC49_List Permissions

4.2.4.50 UC50_ List Roles

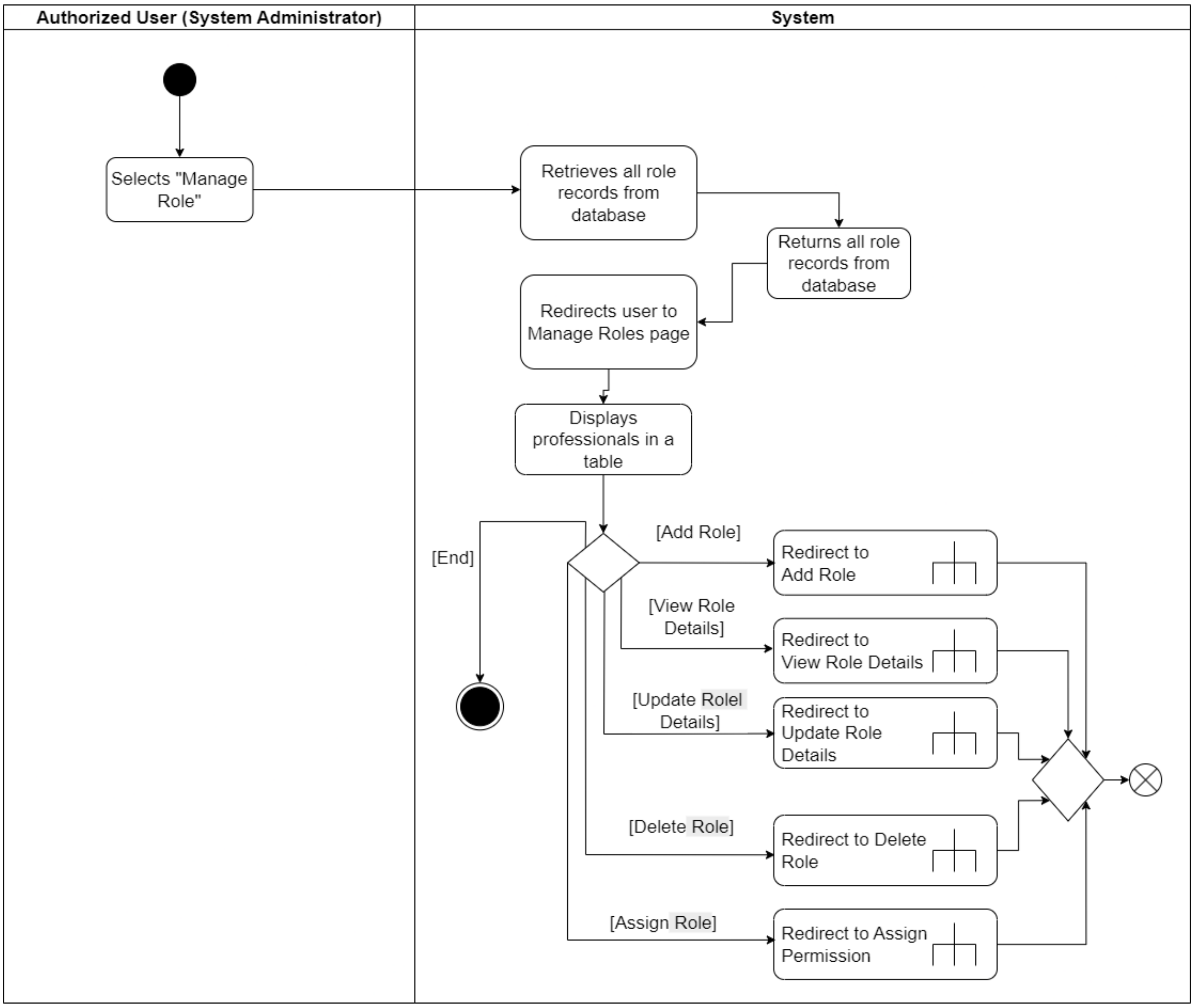


Figure 92: UC50_List Roles

4.2.4.51 UC51_ Add Role

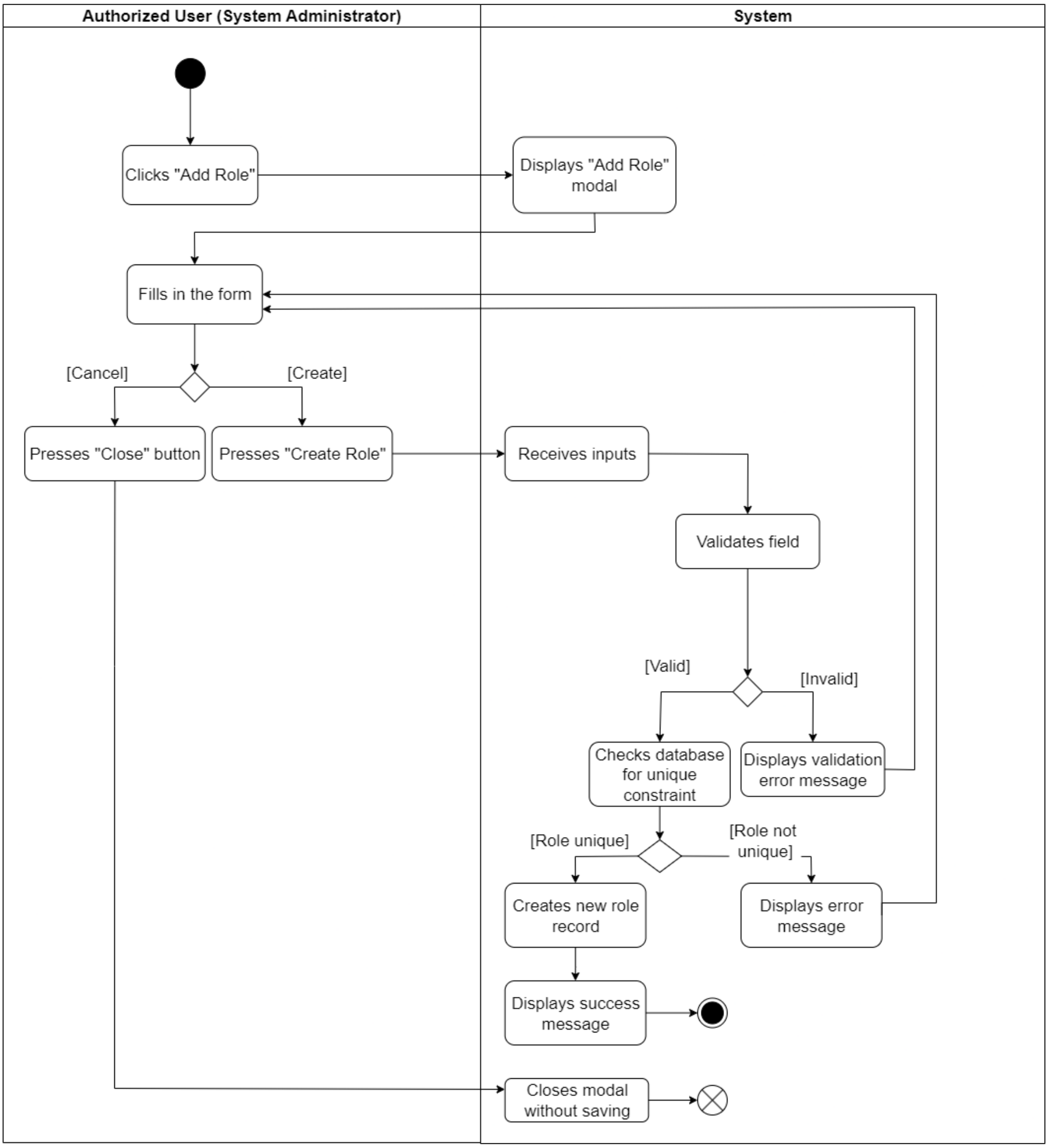


Figure 93: UC51_Add Role

4.2.4.52 UC52_ View Role Details

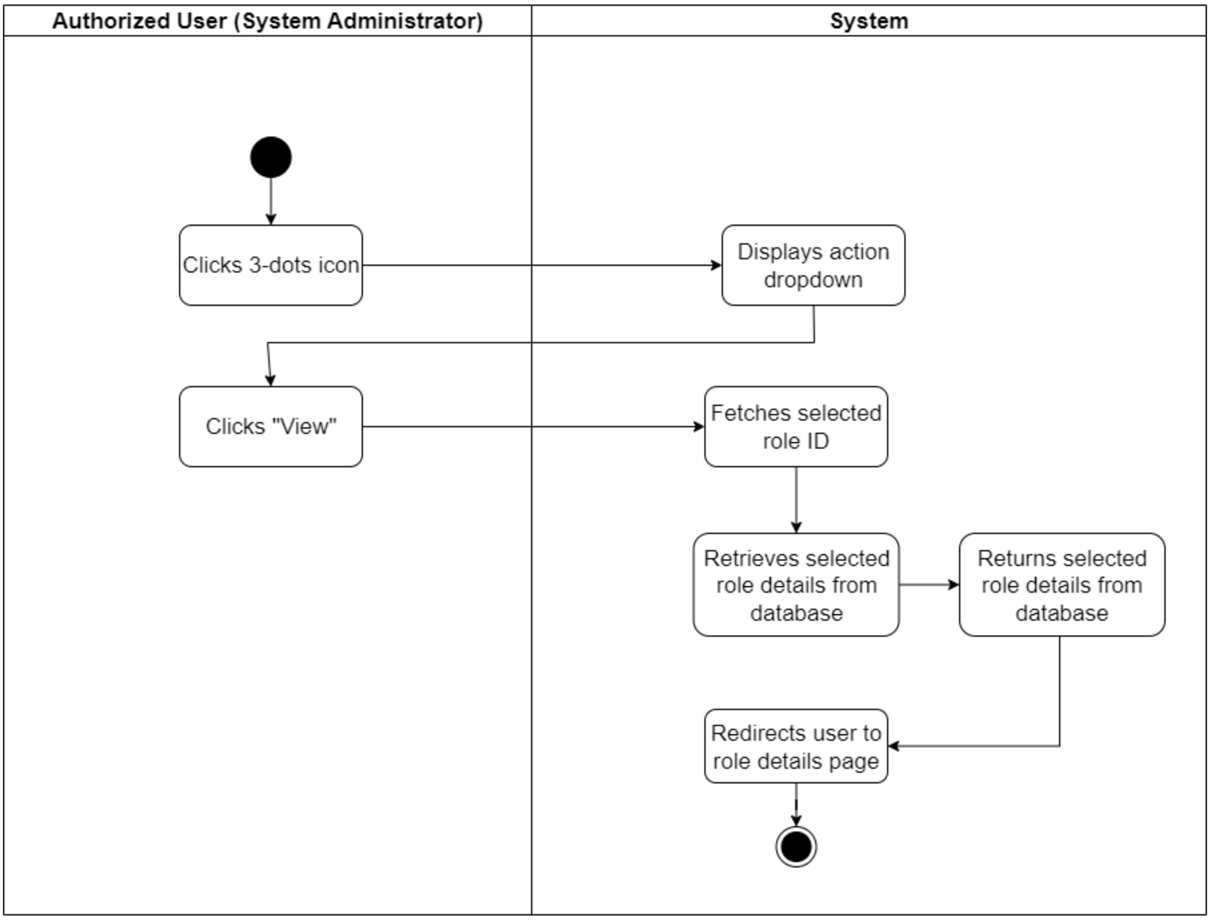


Figure 94: UC52_View Role Details

4.2.4.53 UC53_ Update Role Details

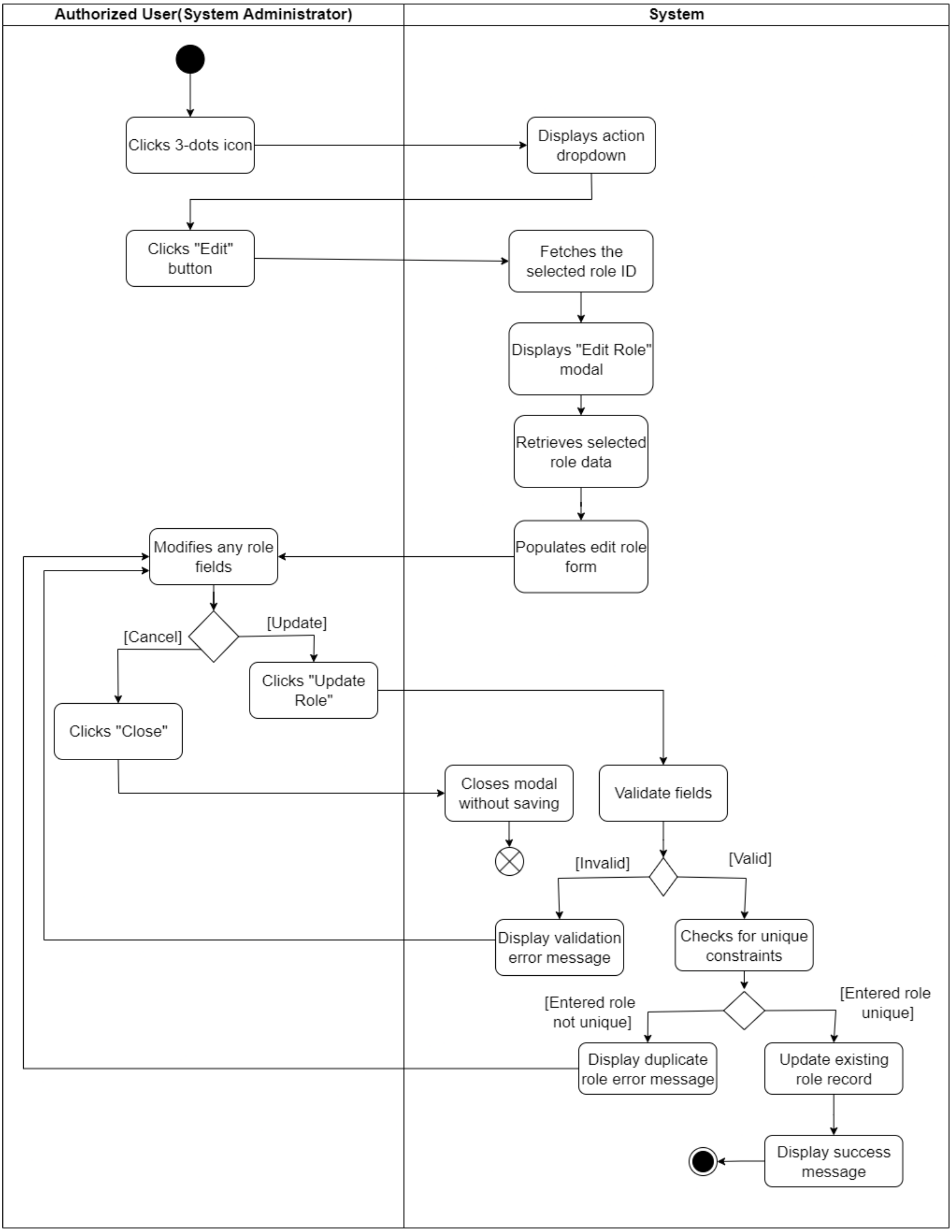


Figure 95: UC53_Update Role Details

4.2.4.54 UC54_ Delete Role

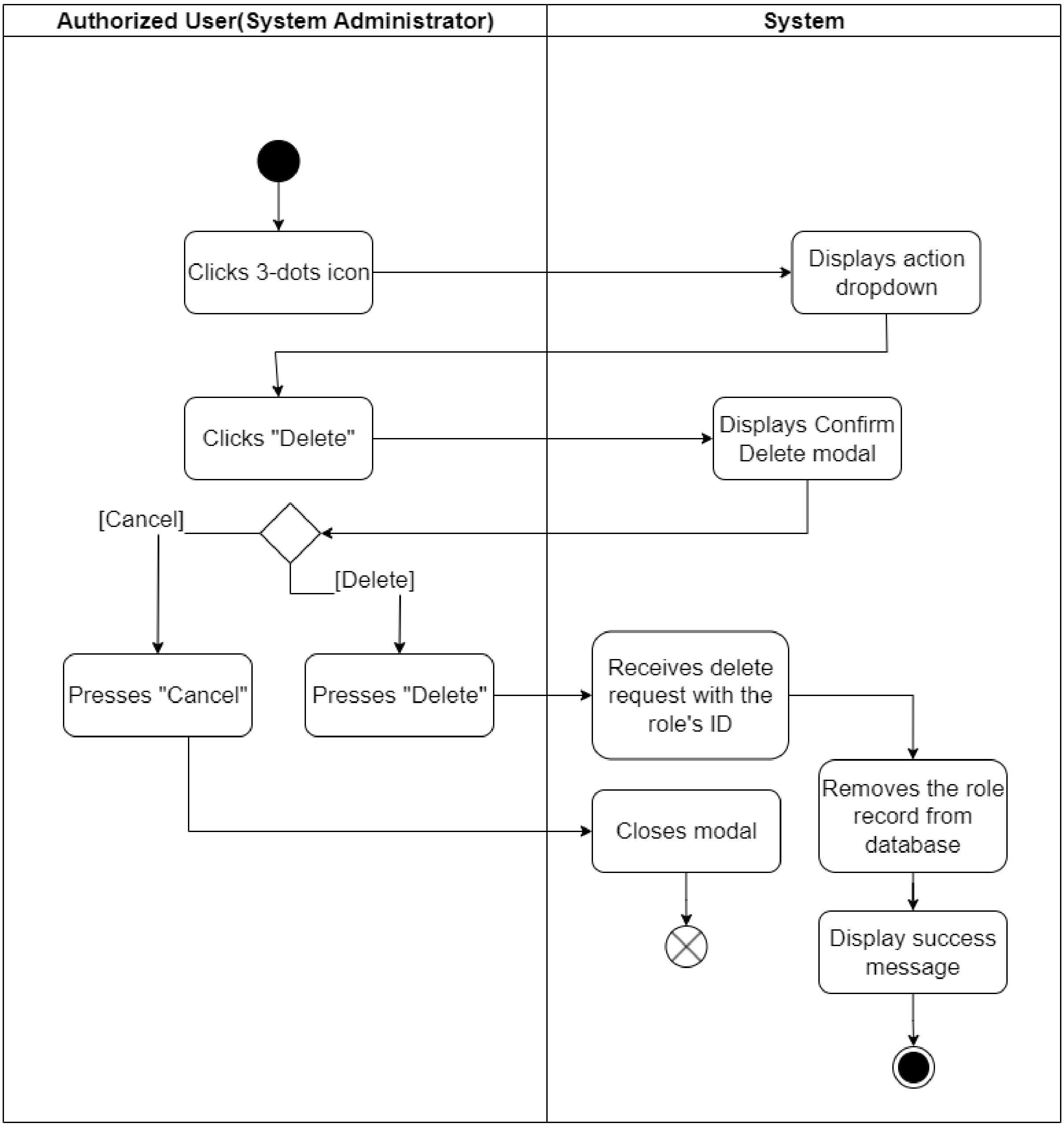


Figure 96: UC54_Delete Role

4.2.4.55 UC55_ Assign Permissions

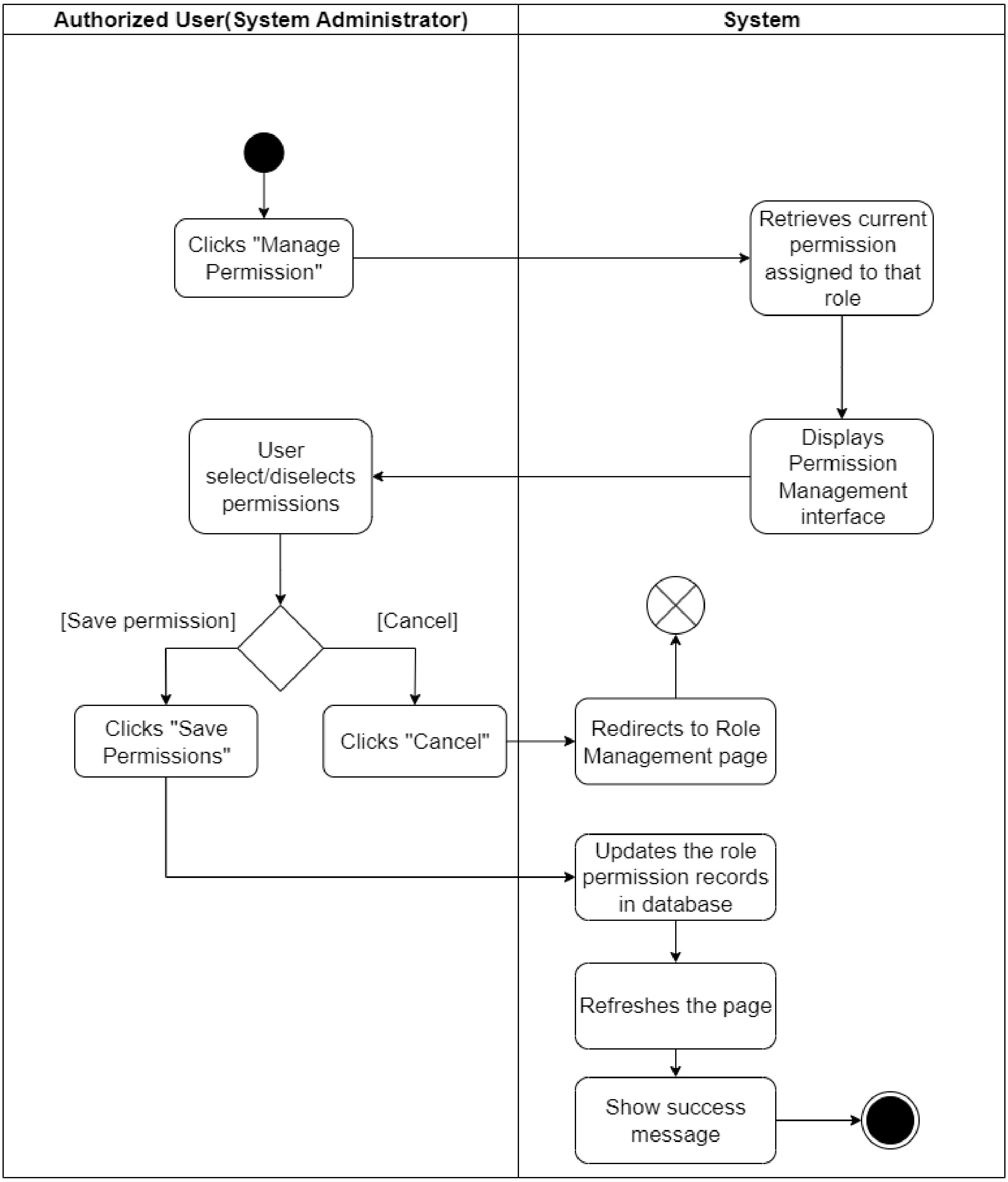


Figure 97: UC55_Assign Permission

4.2.4.56 UC56_Create Query

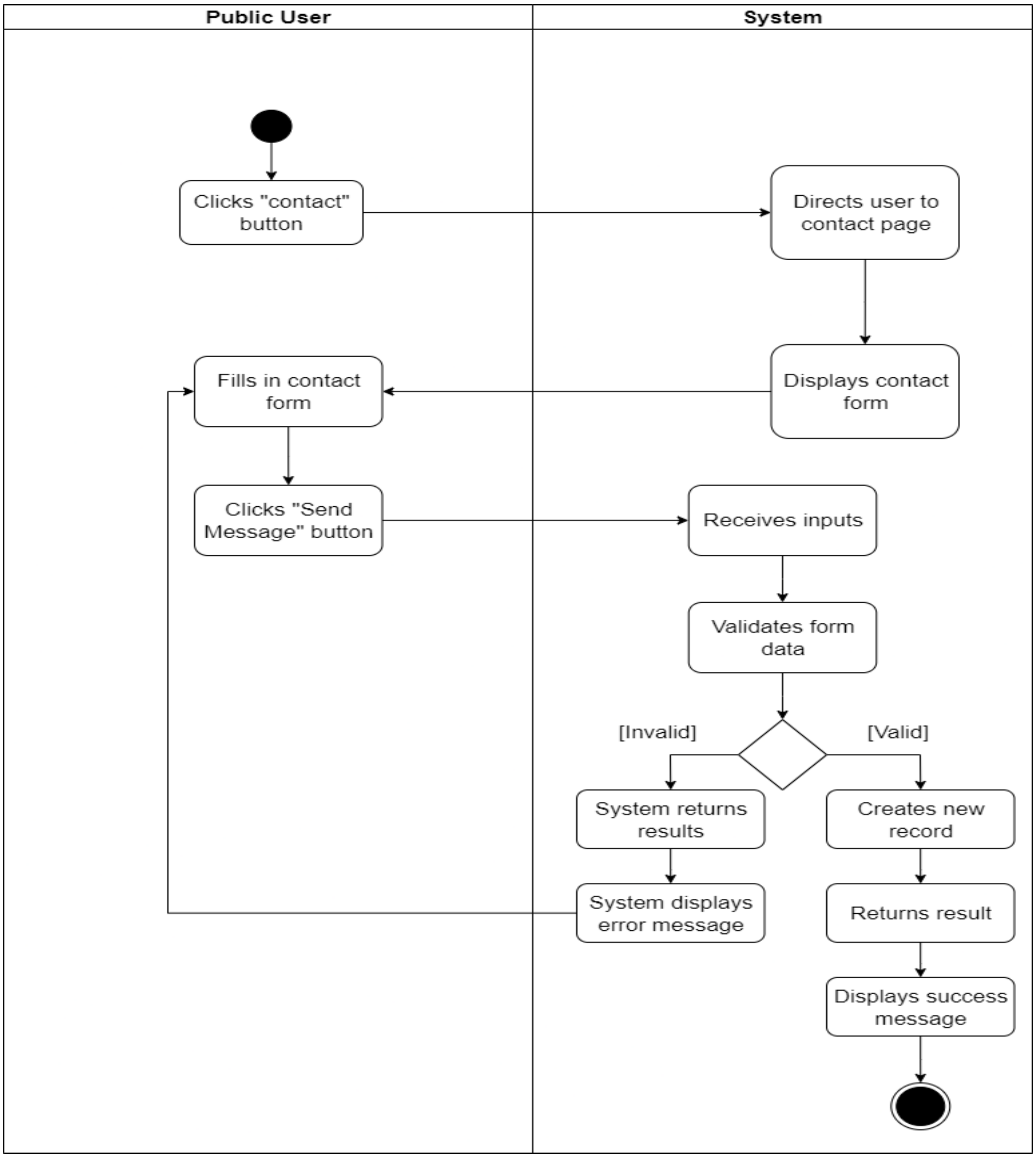


Figure 98: UC56_Create Query

4.2.5 Sequence Diagram

4.2.5.1 Register Account

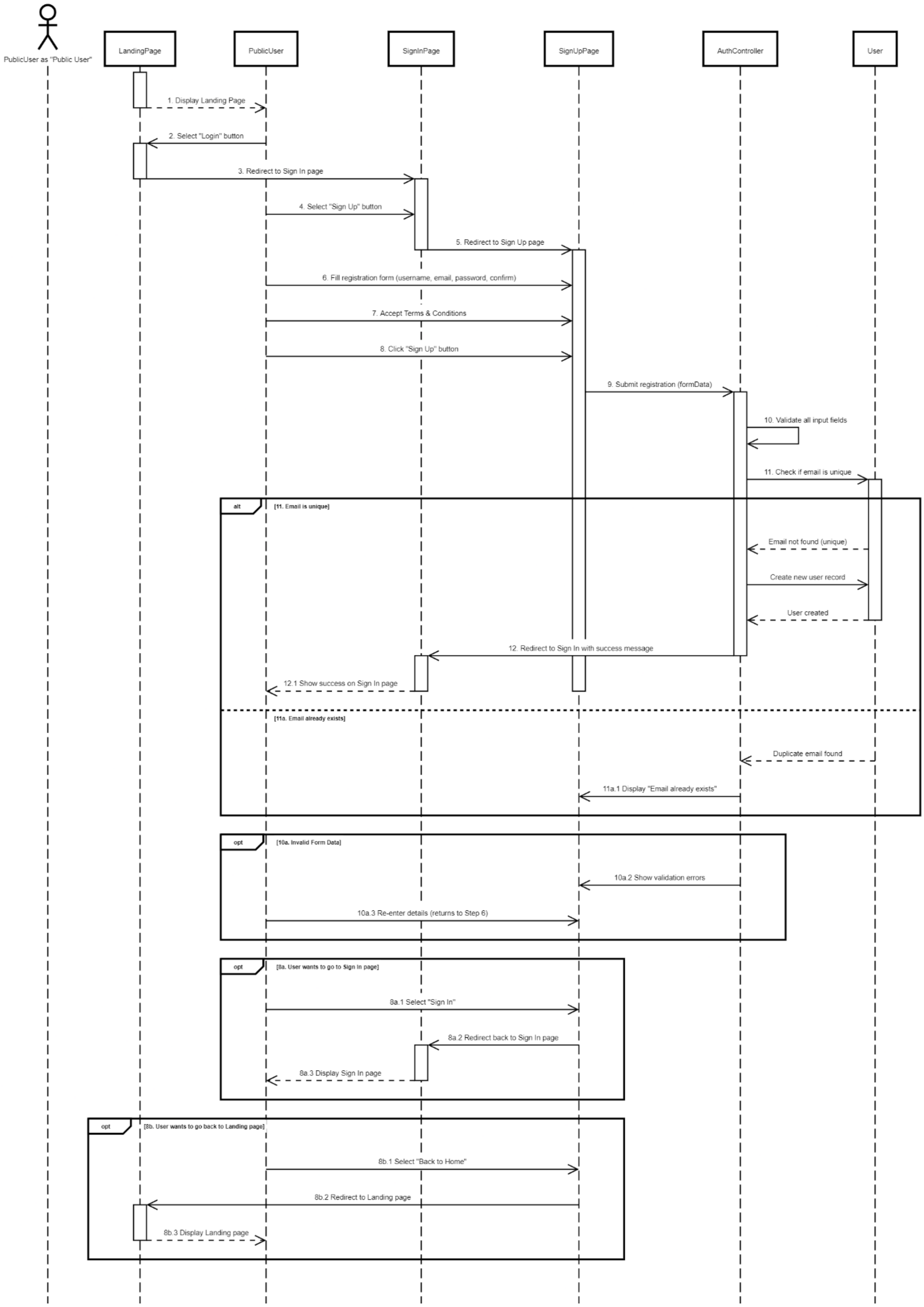


Figure 99: Register Account

4.2.5.2 Login

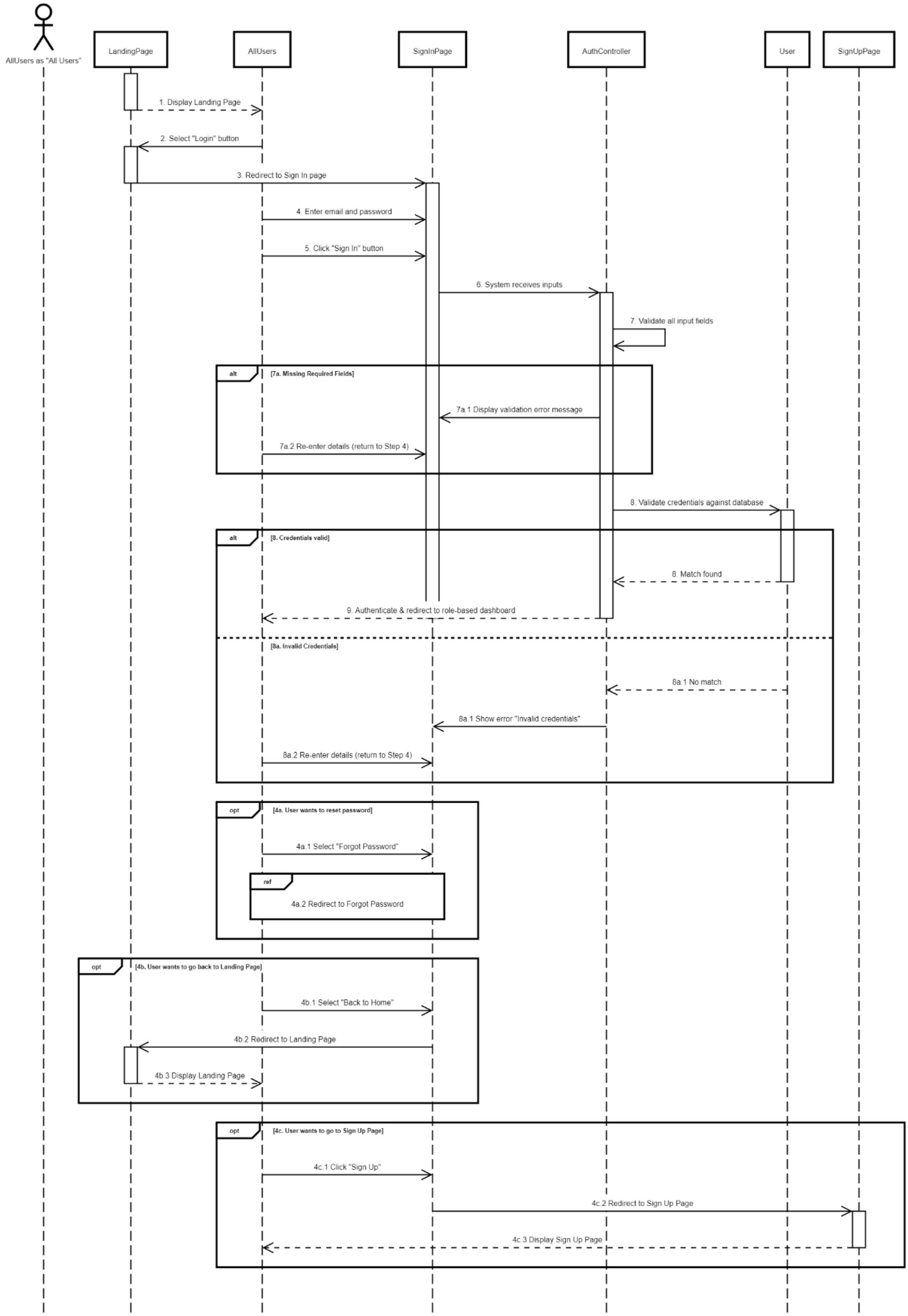

AllUsers as "All Users"
LandingPage
AllUsers
SignInPage
AuthController
User
SignUpPage
1. Display Landing Page
2. Select "Login" button
3. Redirect to Sign In page
4. Enter email and password
5. Click "Sign In" button
6. System receives inputs
7. Validate all input fields
alt
[7a. Missing Required Fields]
7a.1 Display validation error message
7a.2 Re-enter details (return to Step 4)
8. Validate credentials against database
alt
[8. Credentials valid]
8. Match found
9. Authenticate & redirect to role-based dashboard
[8a. Invalid Credentials]
8a.1 No match
8a.1 Show error "Invalid credentials"
8a.2 Re-enter details (return to Step 4)
opt
[4a. User wants to reset password]
4a.1 Select "Forgot Password"
ref
4a.2 Redirect to Forgot Password
opt
[4b. User wants to go back to Landing Page]
4b.1 Select "Back to Home"
4b.2 Redirect to Landing Page
4b.3 Display Landing Page
opt
[4c. User wants to go to Sign Up Page]
4c.1 Click "Sign Up"
4c.2 Redirect to Sign Up Page
4c.3 Display Sign Up Page

Figure 100: Login

4.2.5.3 Forgot Password

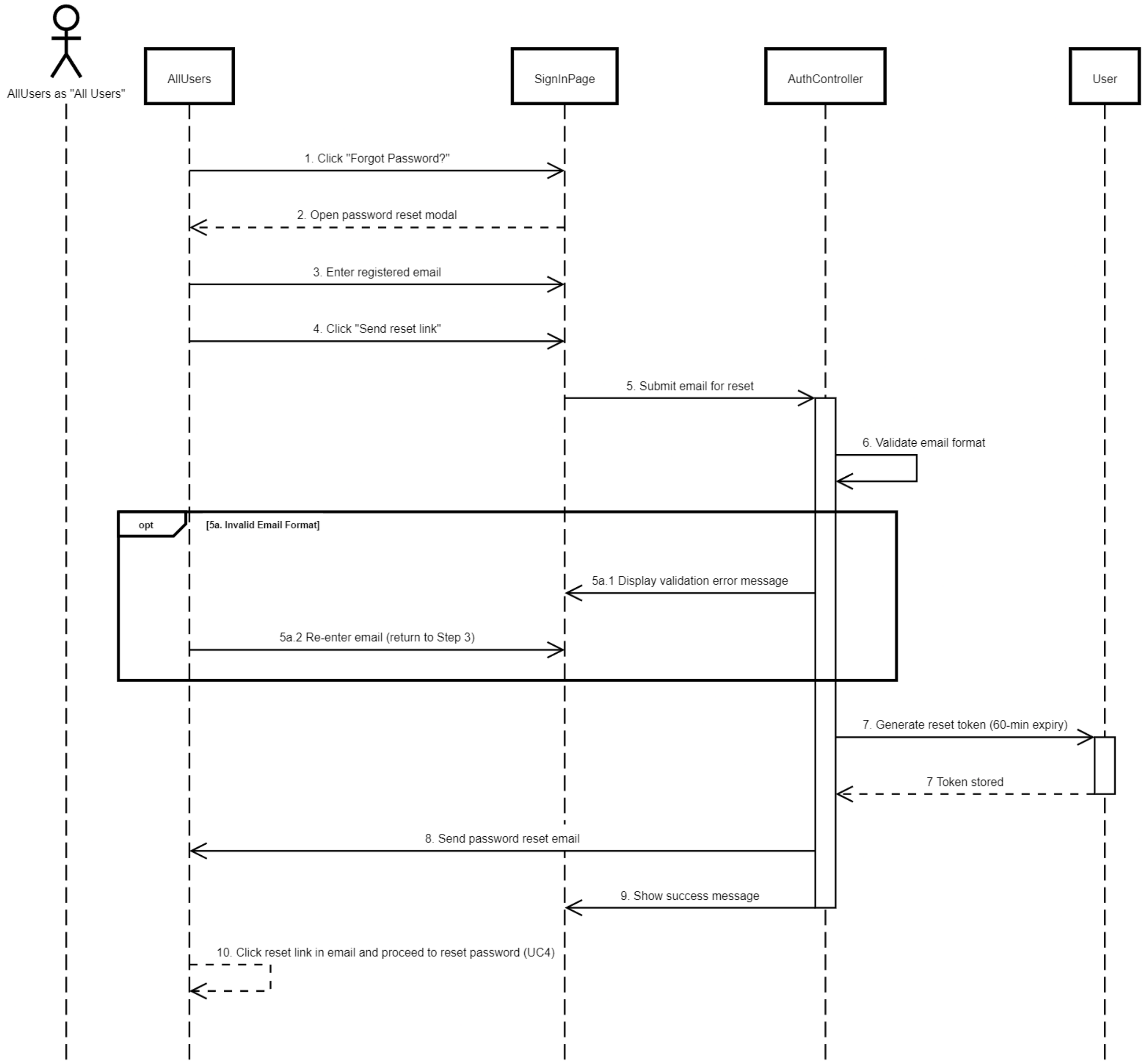


Figure 101: Forgot Password

4.2.5.4 Create Report

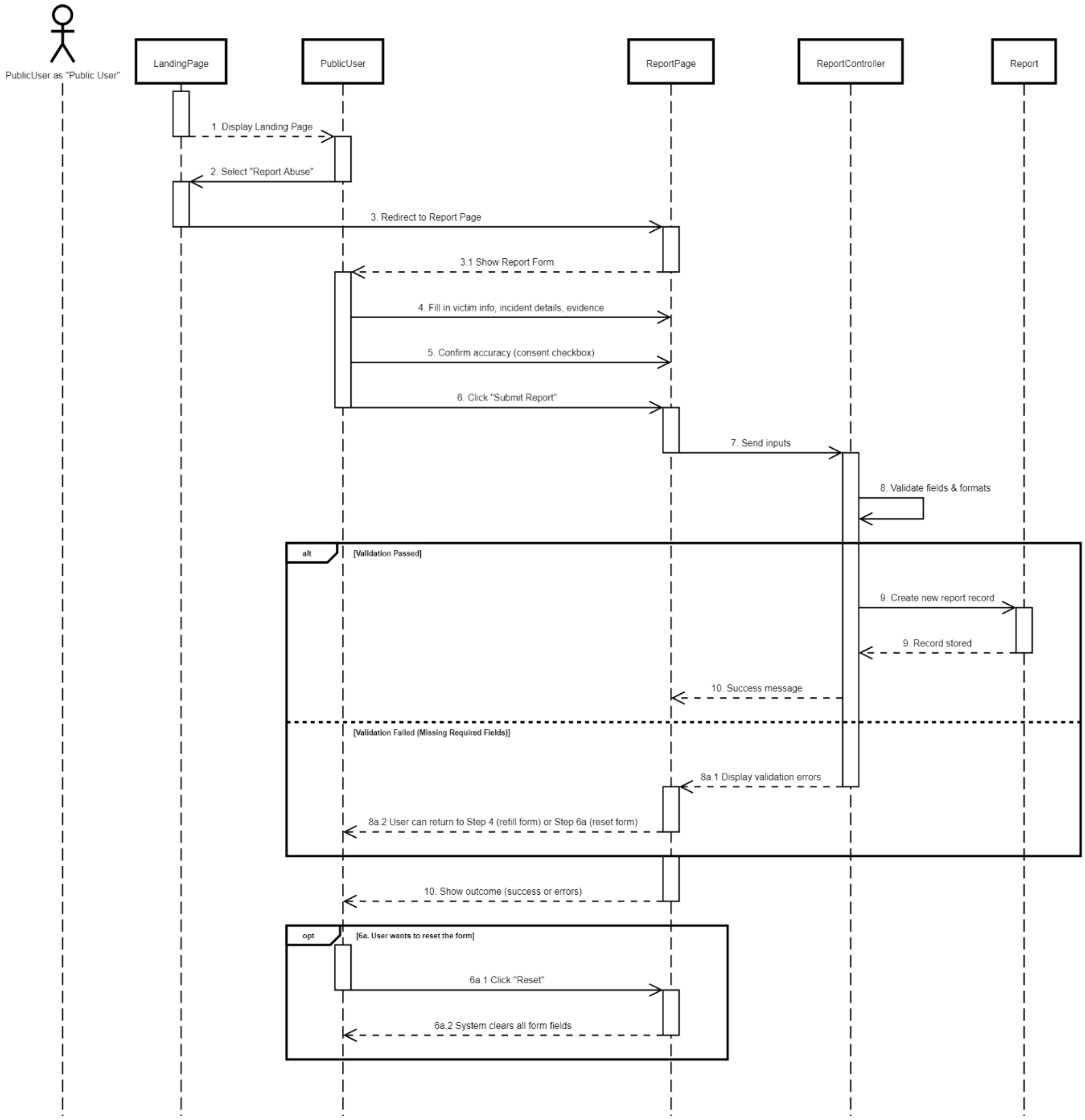


Figure 102: Create Report

4.2.5.5 List Reports

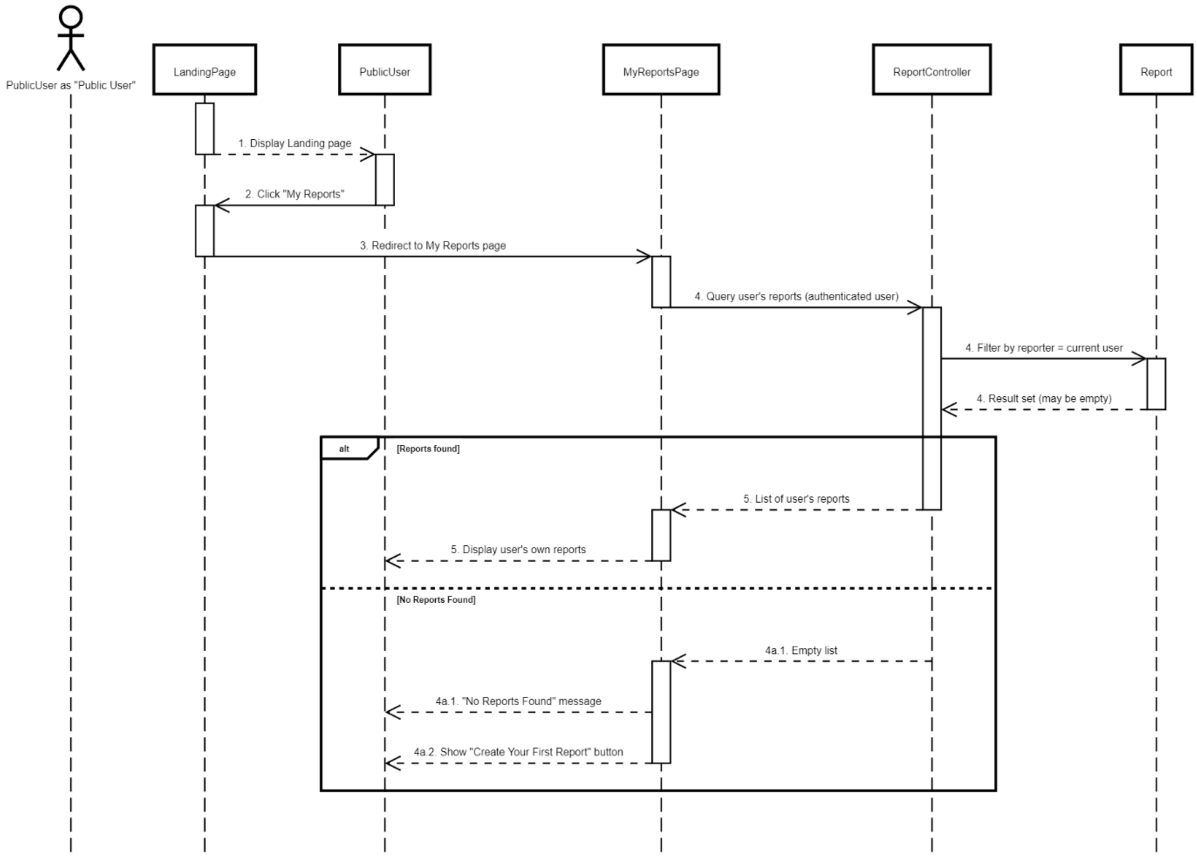


Figure 103: List Reports

4.2.5.6 View Report Details

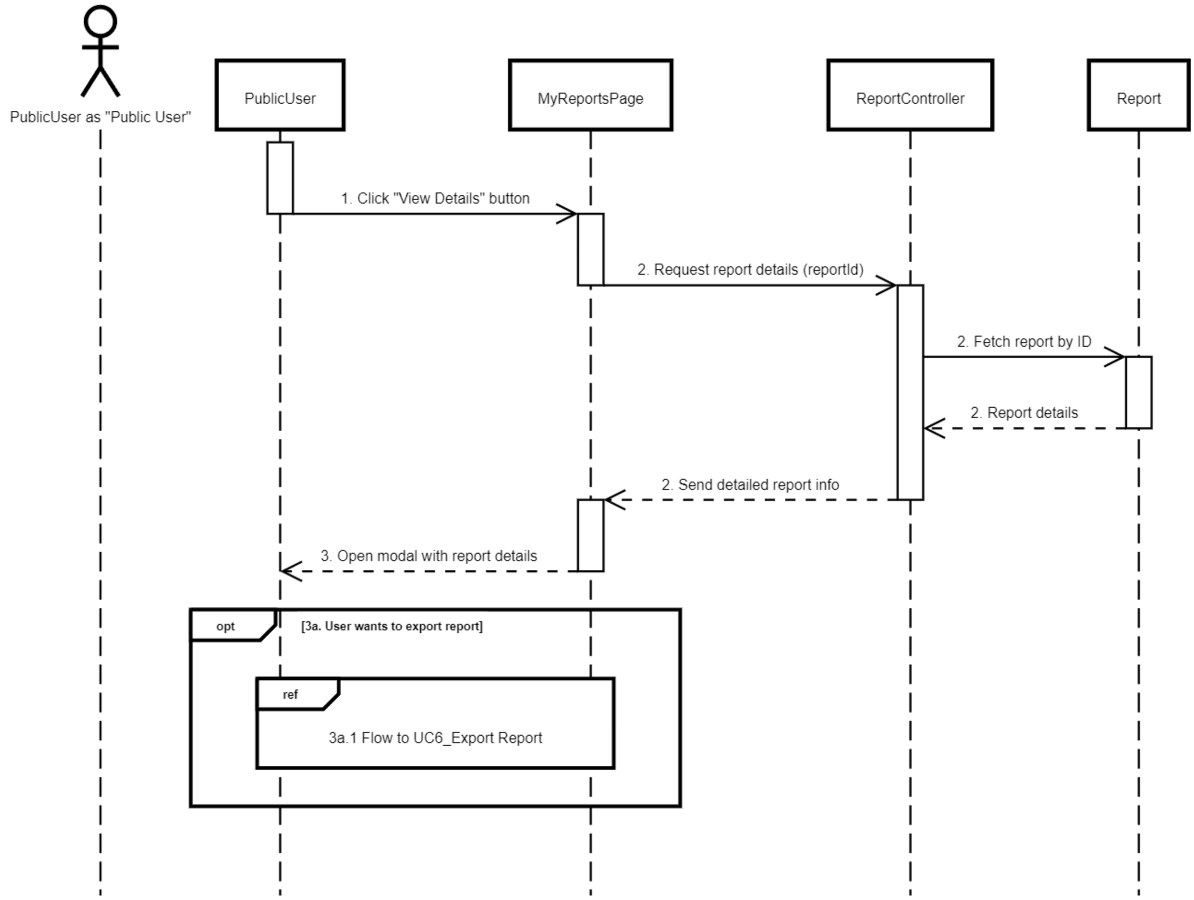


Figure 104: View Report Details

4.2.5.7 Export Report

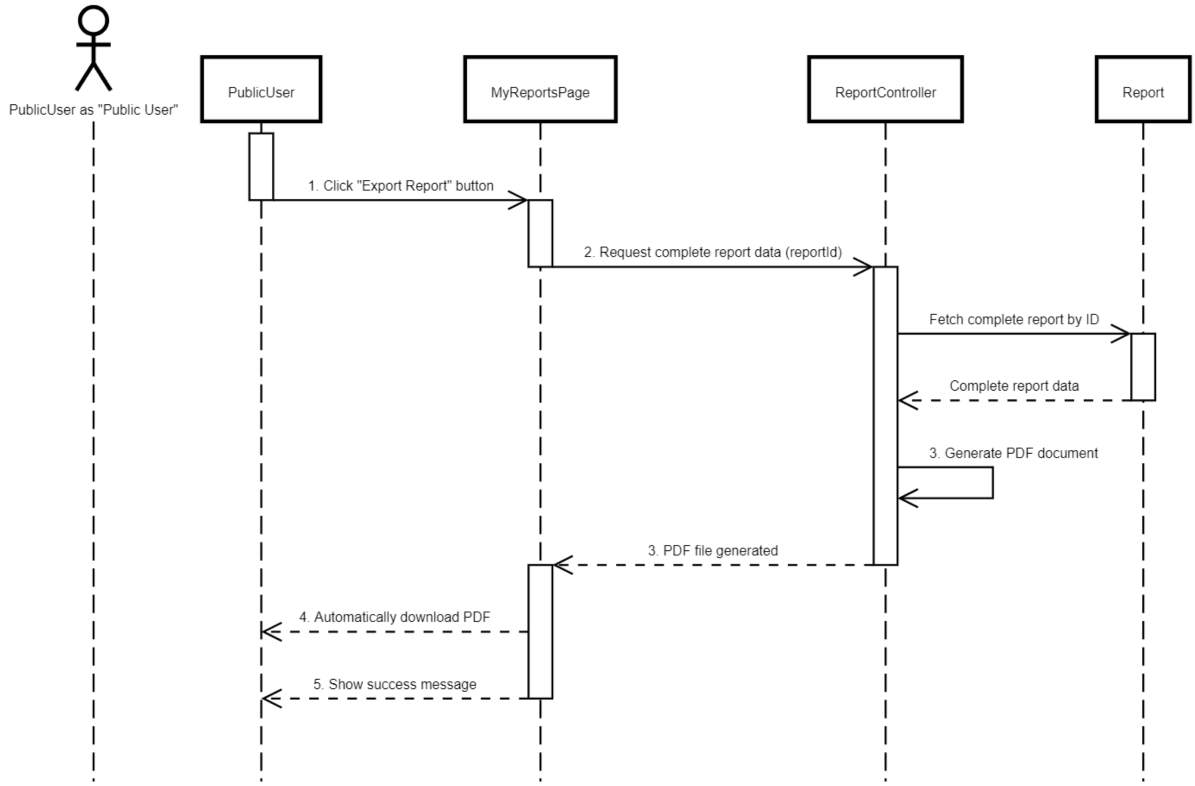


Figure 105: Export Report

4.2.5.8 View Profile Details

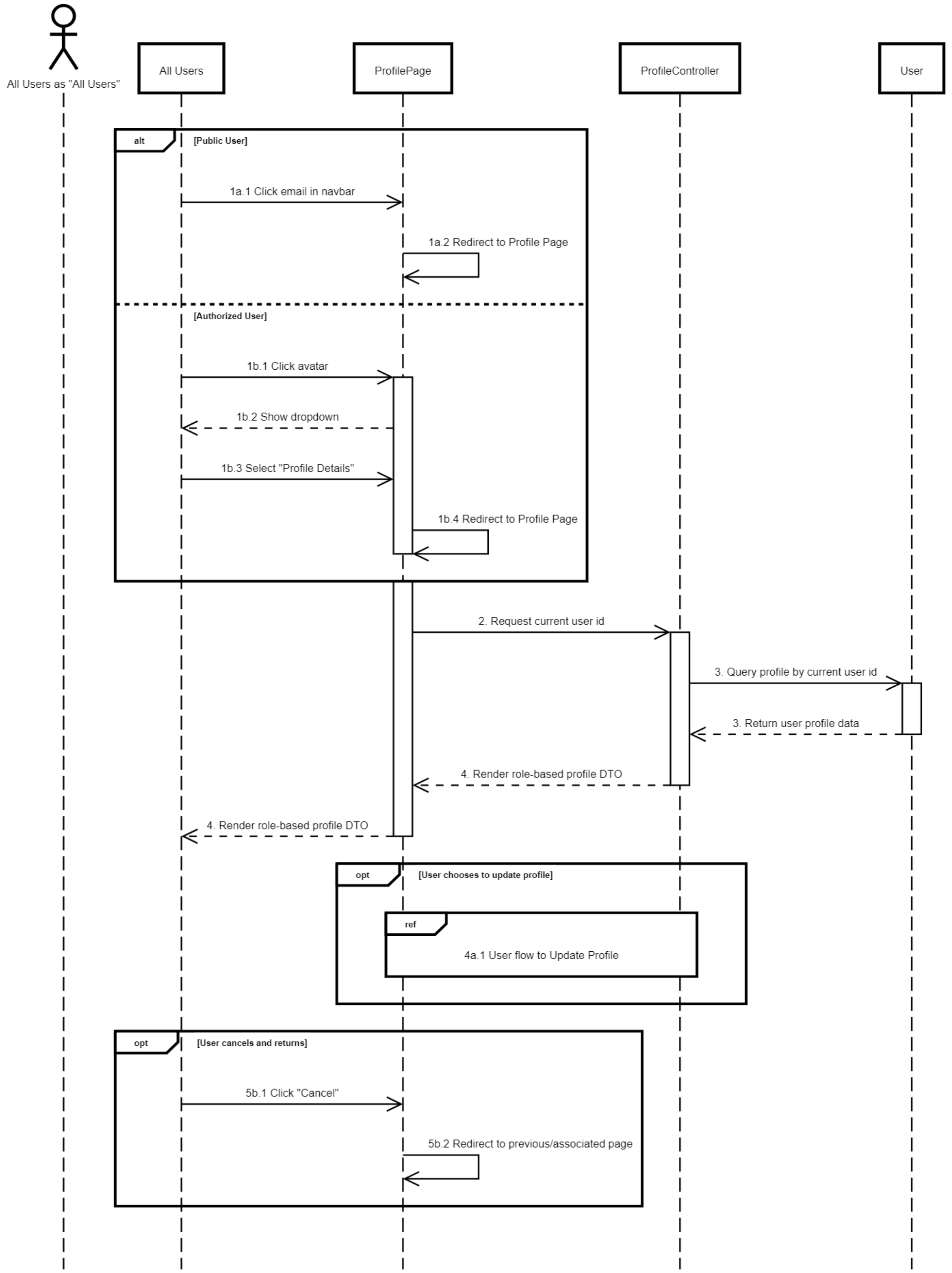


Figure 106: View Profile Details

4.2.5.9 Update Profile Details

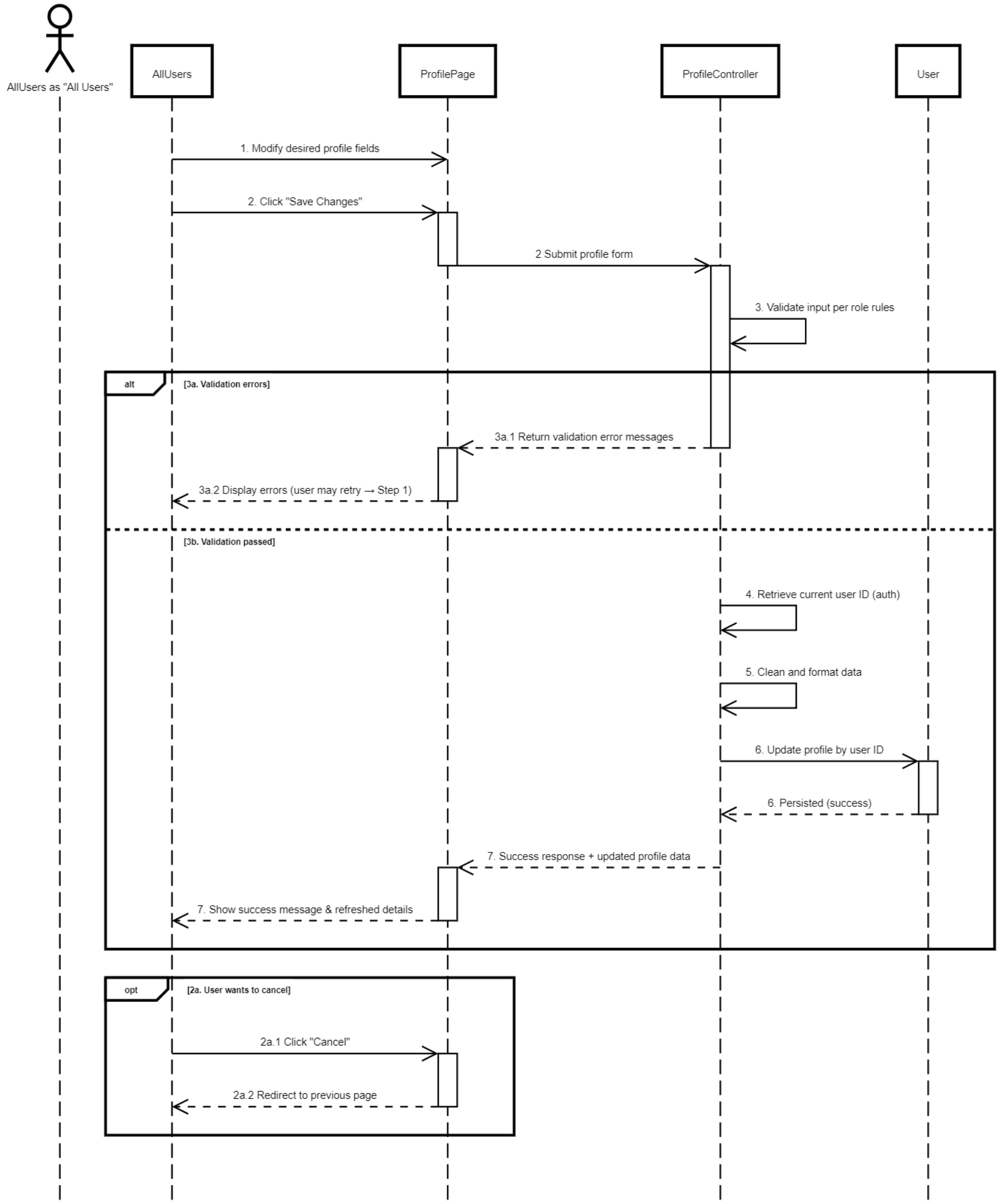


Figure 107: Update Profile Details

4.2.5.10 Logout

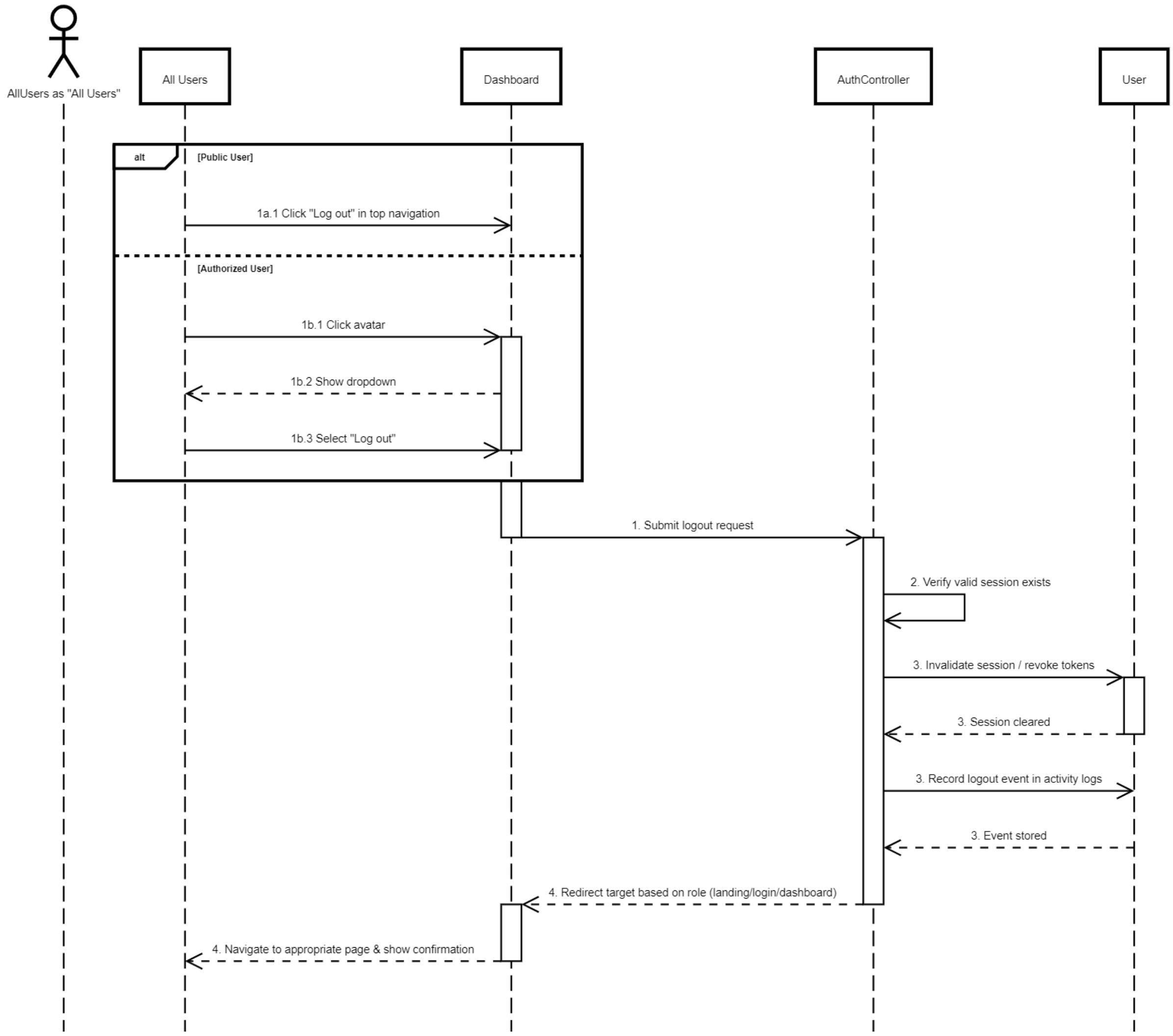


Figure 108: Logout

4.2.5.11 List Queries

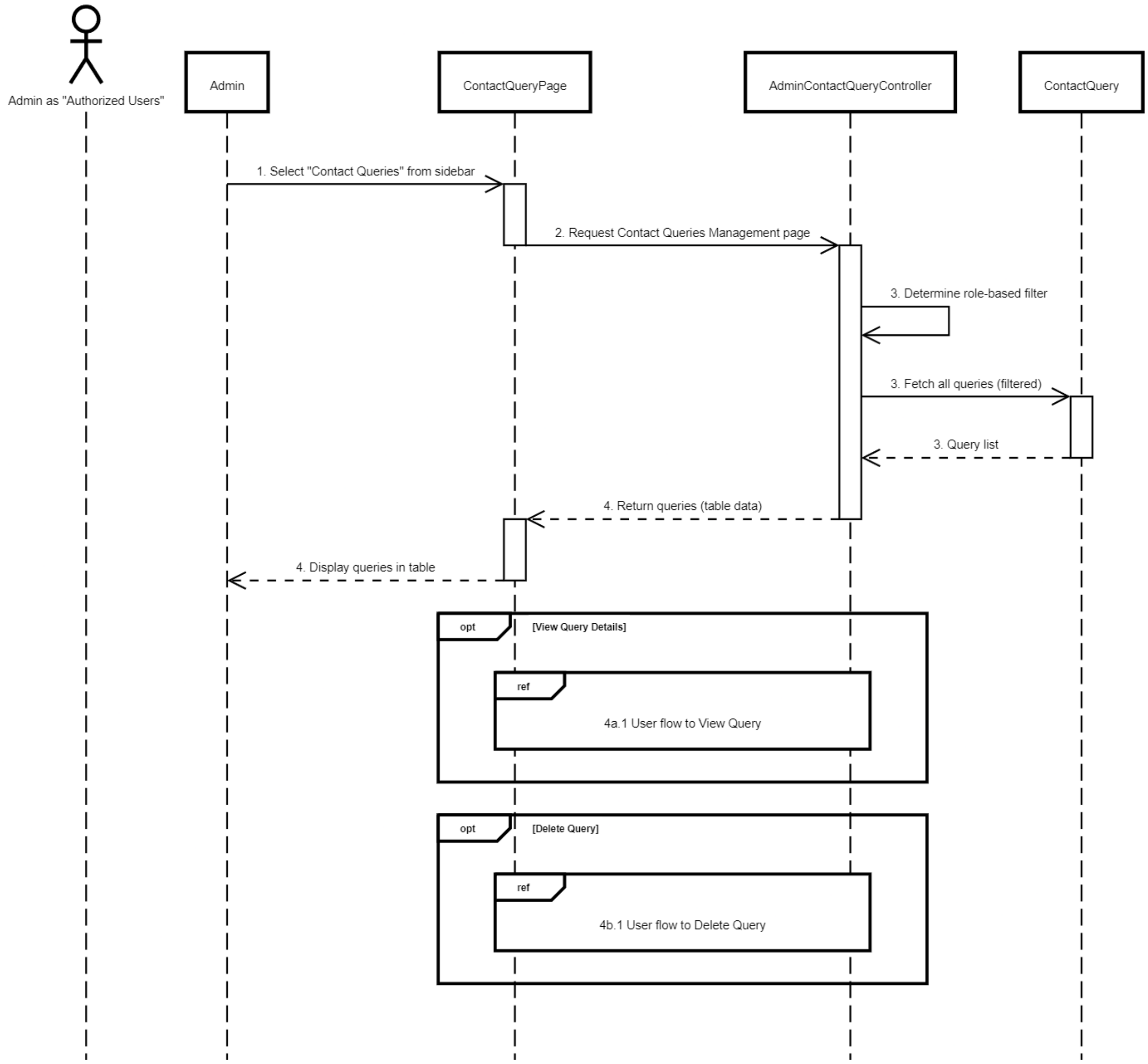


Figure 109: List Queries

4.2.5.12 View Query Details

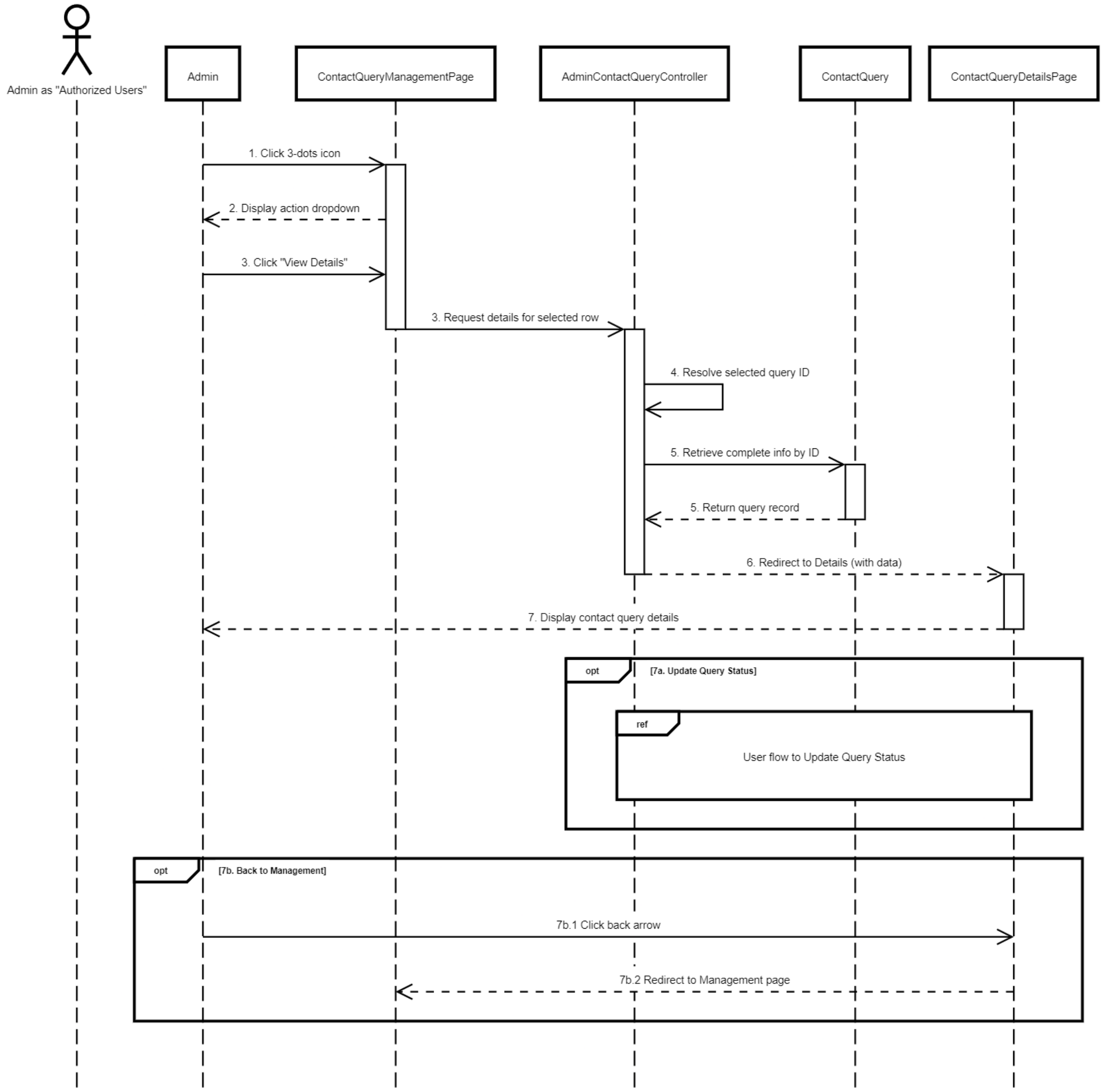


Figure 110: View Query Details

4.2.5.13 Update Query Status

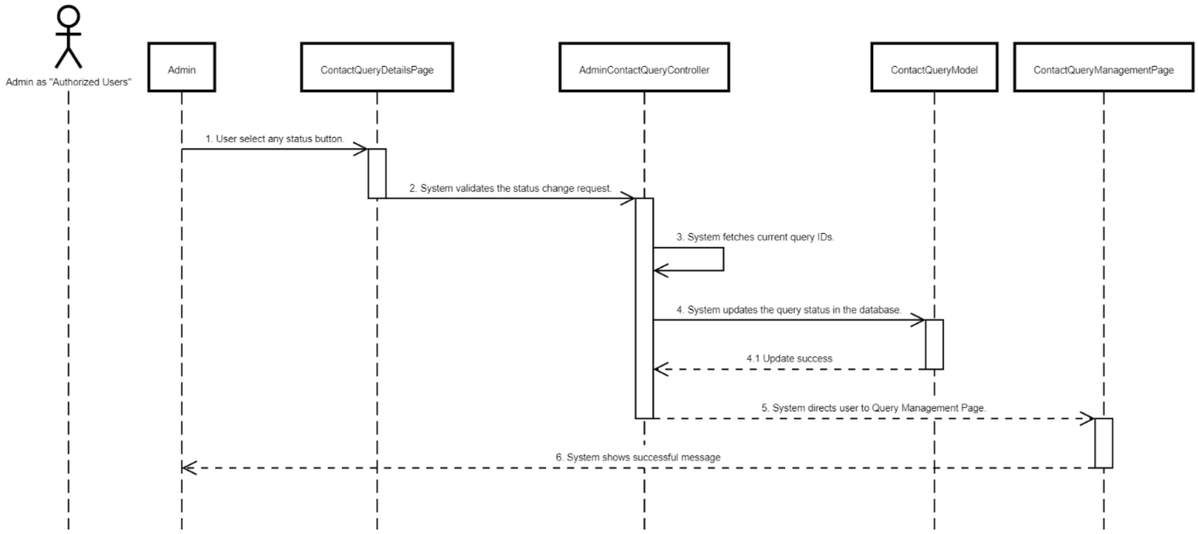


Figure 111: Update Query Status

4.2.5.14 Delete Query

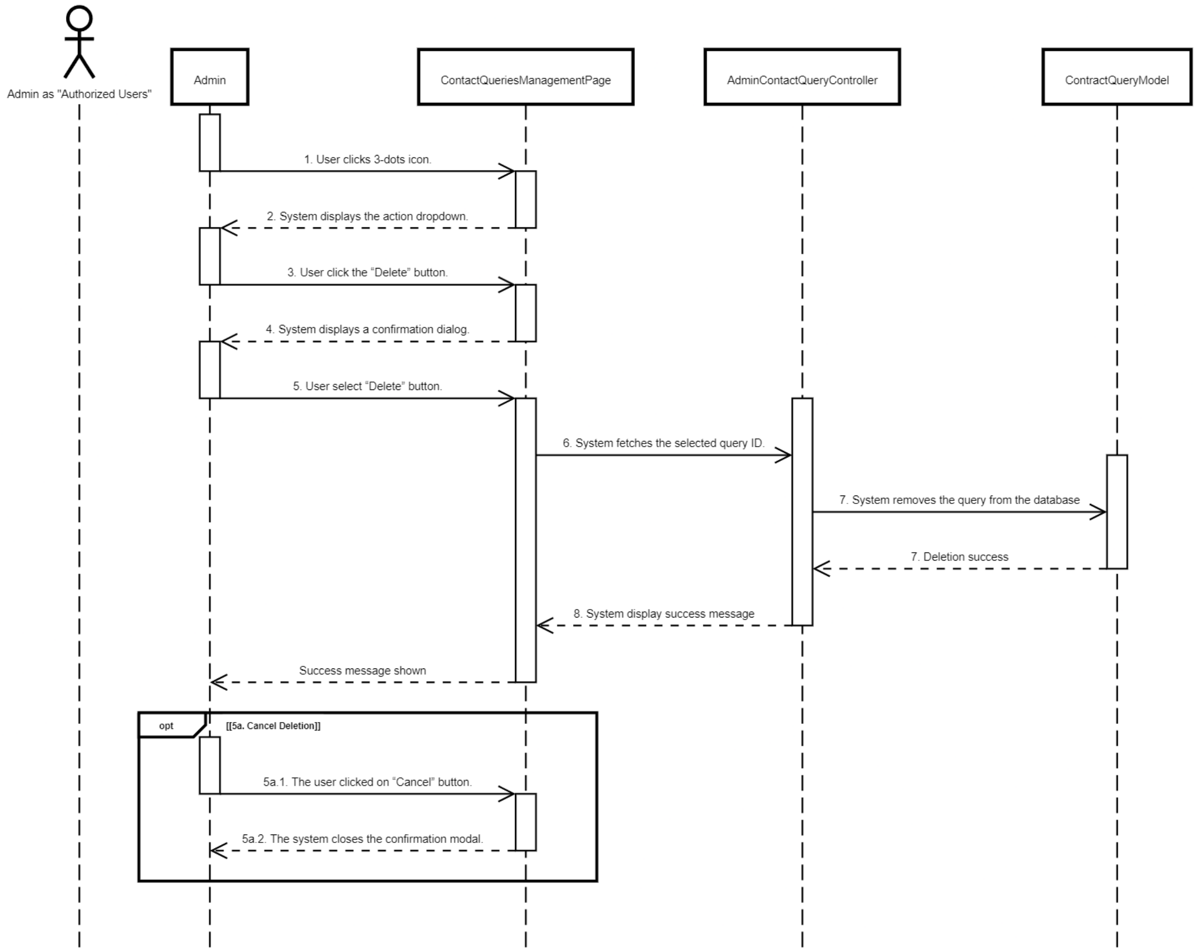


Figure 112: Delete Query

4.2.5.15 List Cases

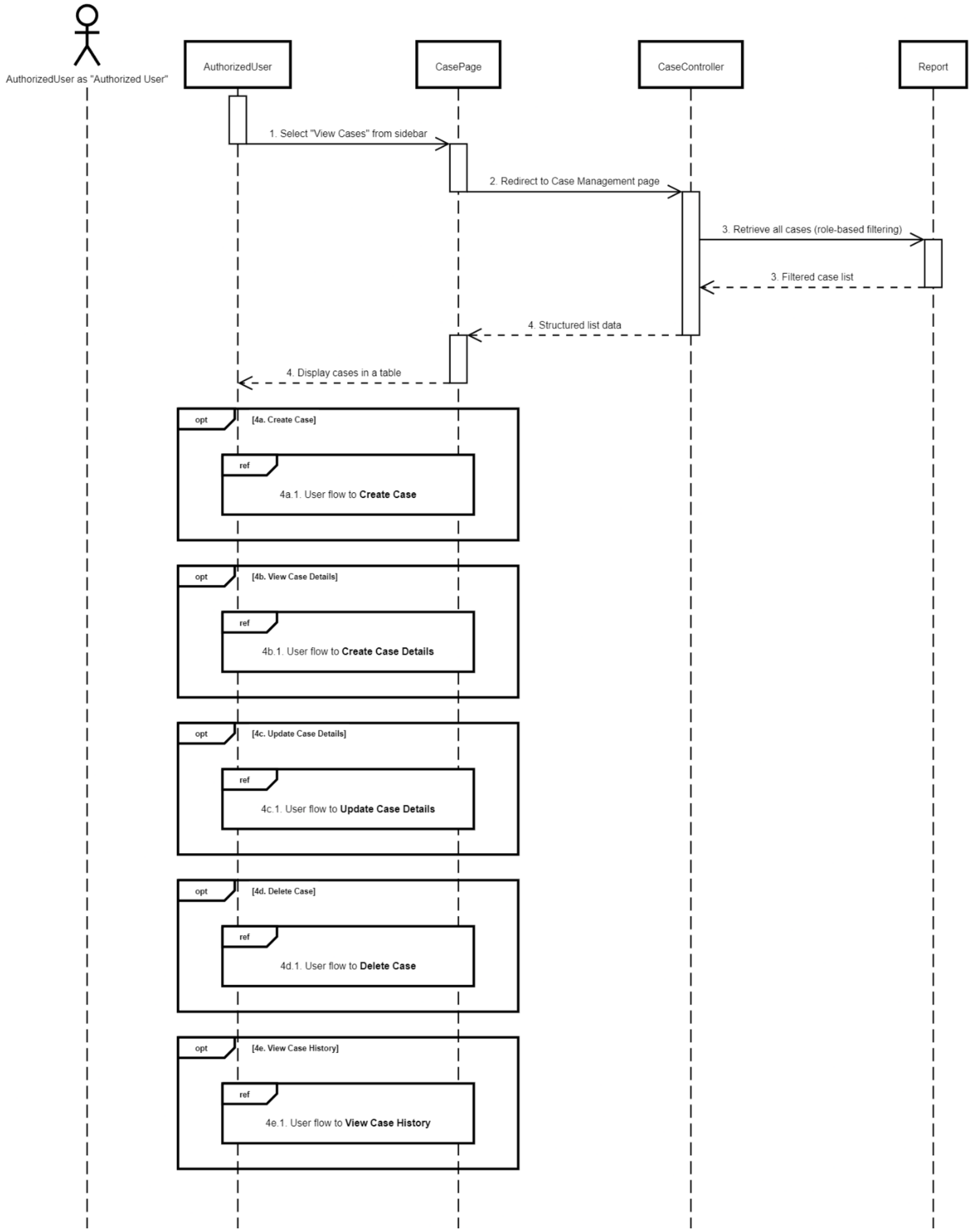


Figure 113: List Cases

4.2.5.16 Create Case

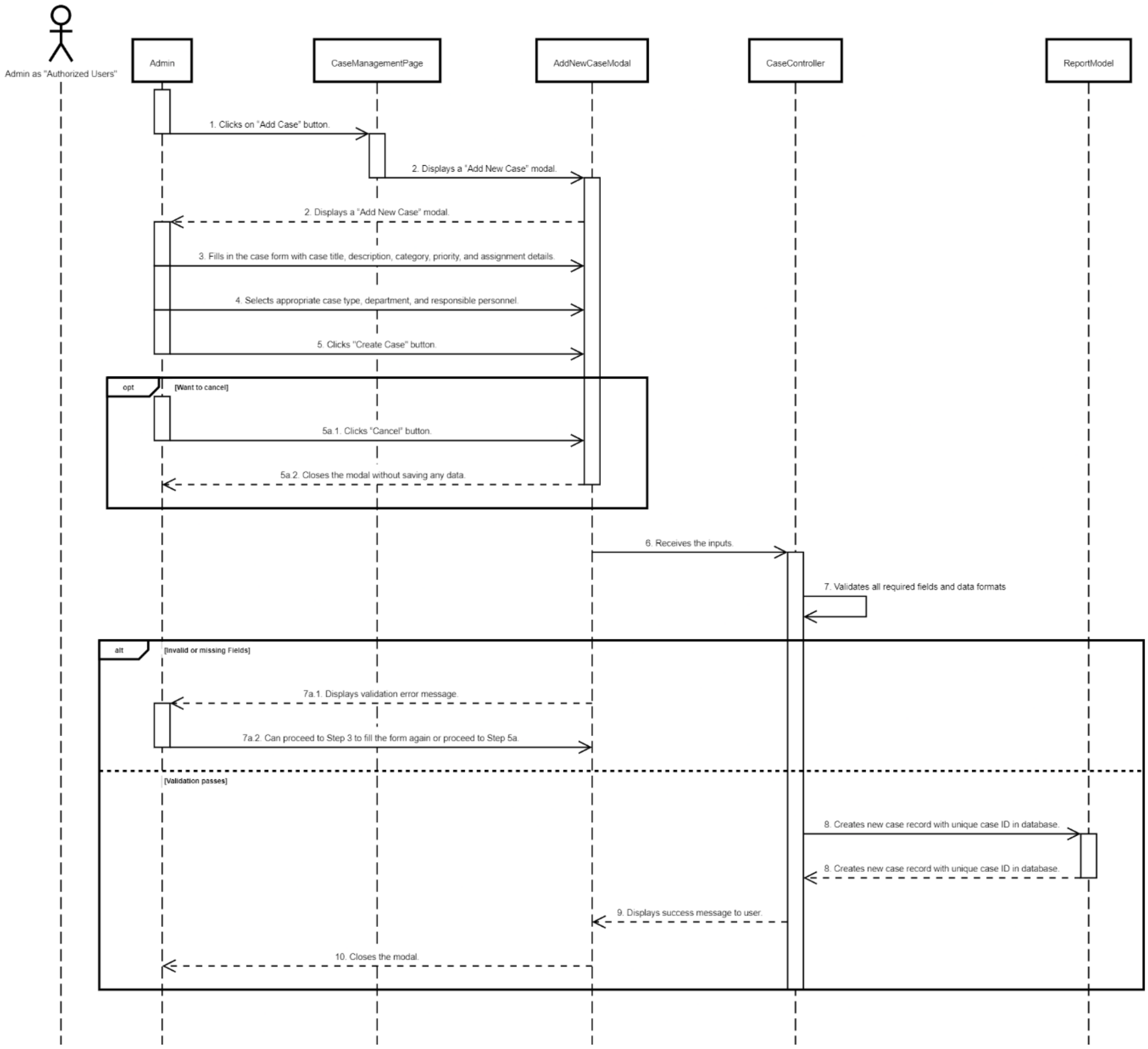


Figure 114: Create Case

4.2.5.17 View Case Details

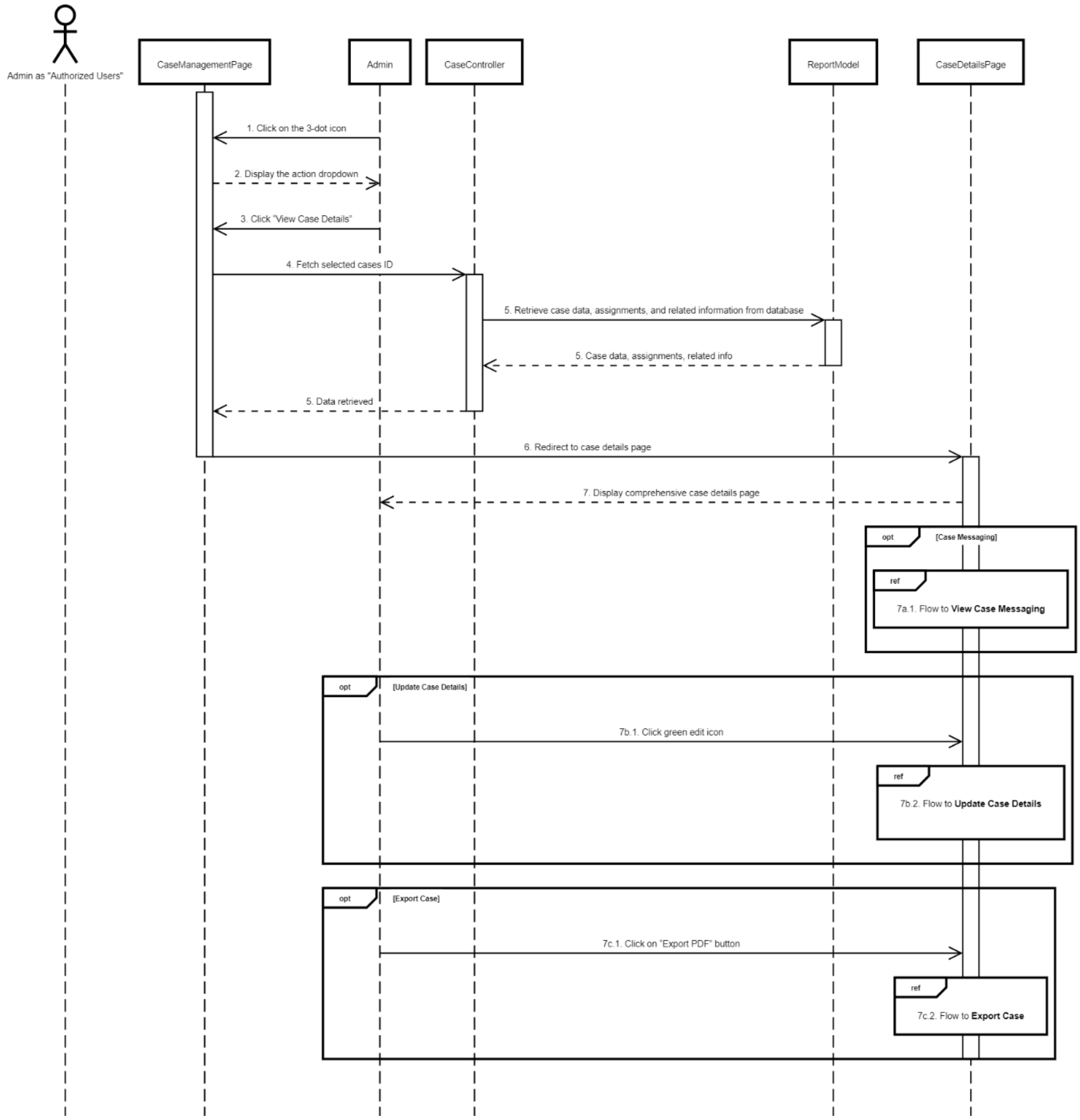


Figure 115: View Case Details

4.2.5.18 Case Messaging

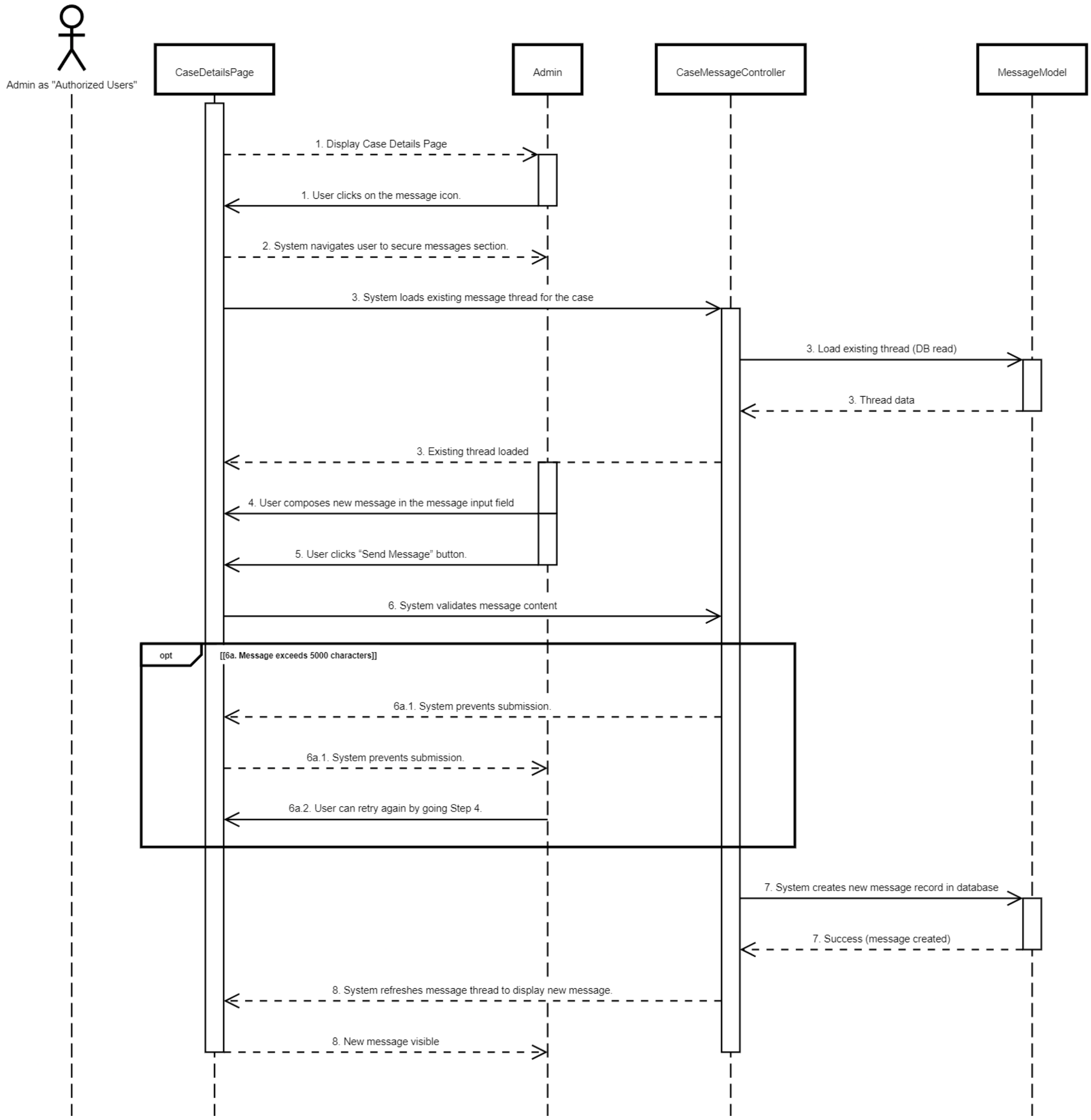


Figure 116: Case Messaging

4.2.5.19 Update Case Details

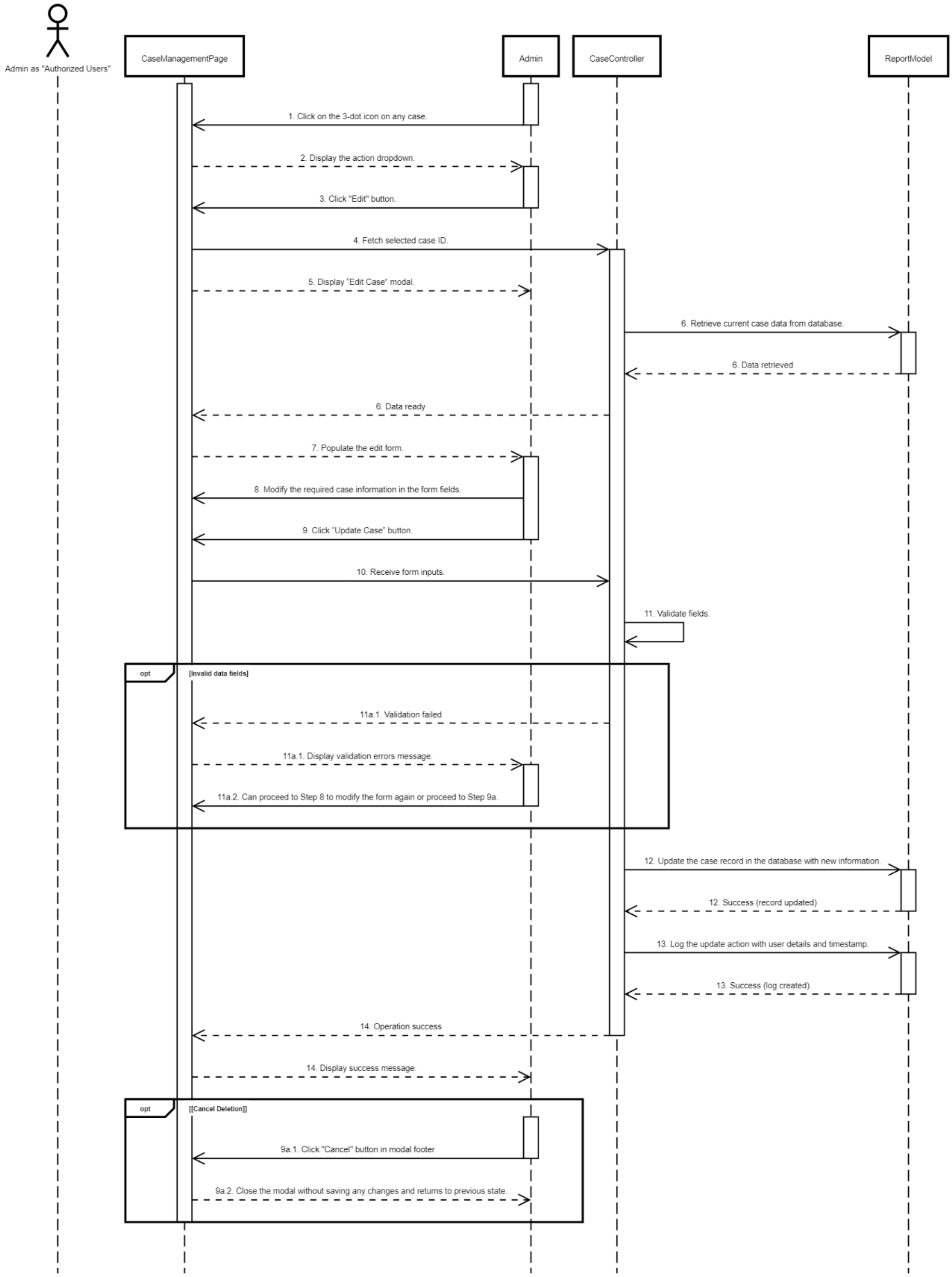


Figure 117: Update Case Details

4.2.5.20 Delete Case

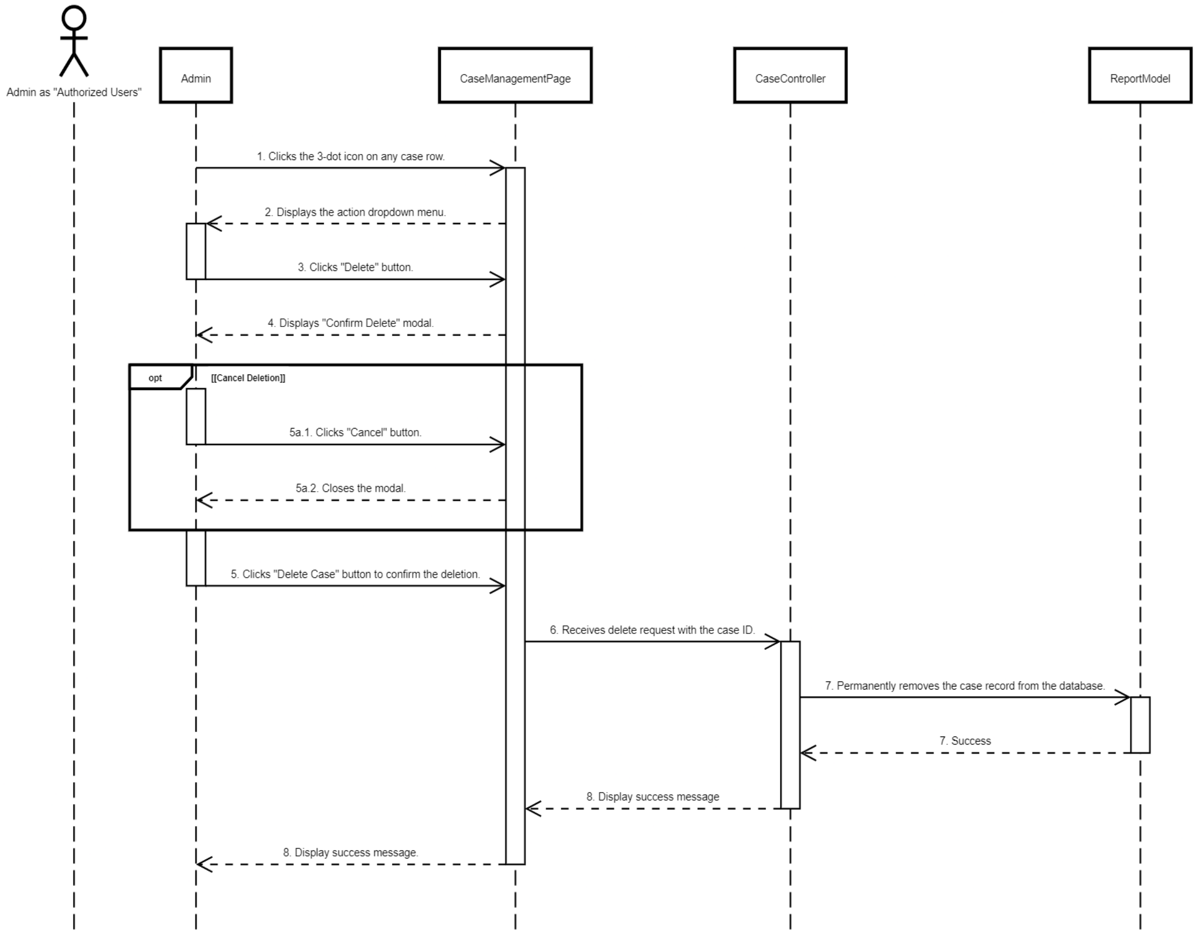


Figure 118: Delete Case

4.2.5.21 View Case History

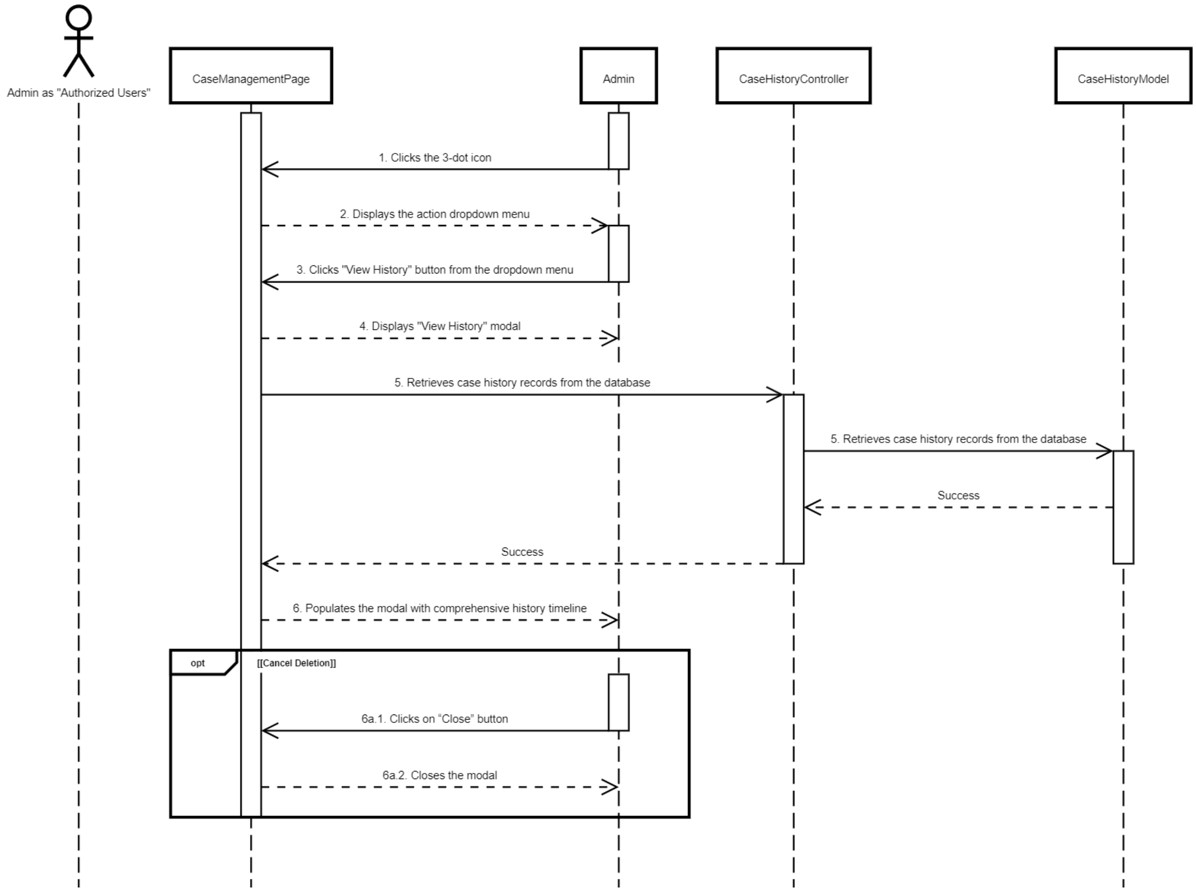


Figure 119: View Case History

4.2.5.22 Export Case

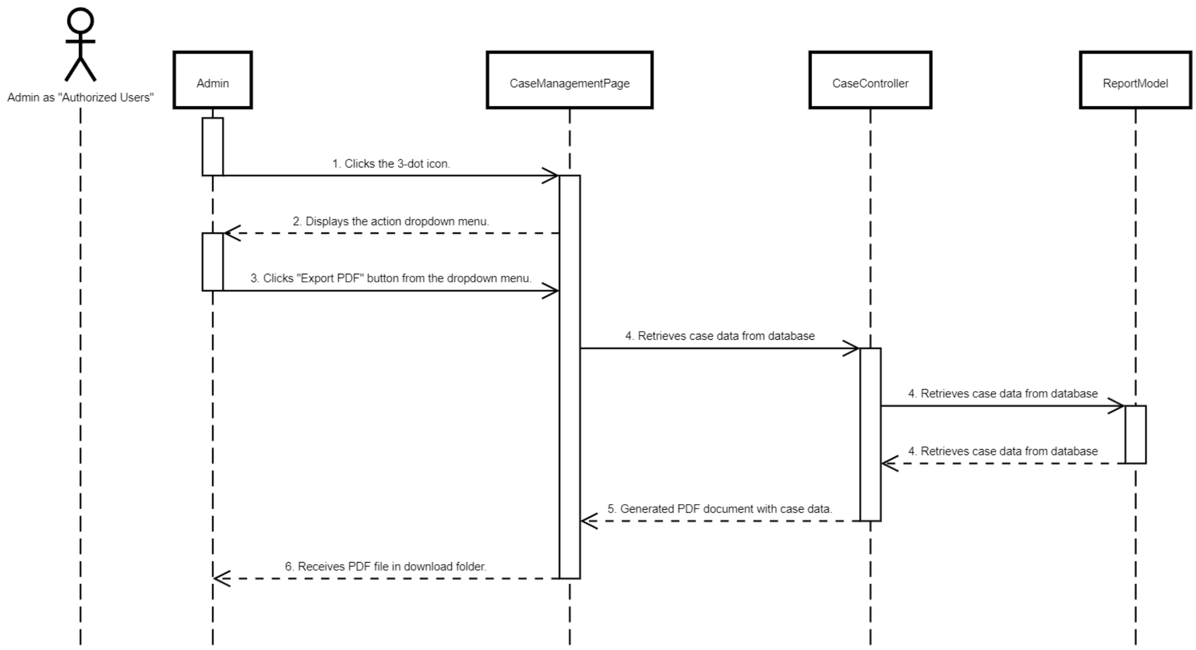


Figure 120: Export Case

4.2.5.23 List Admins

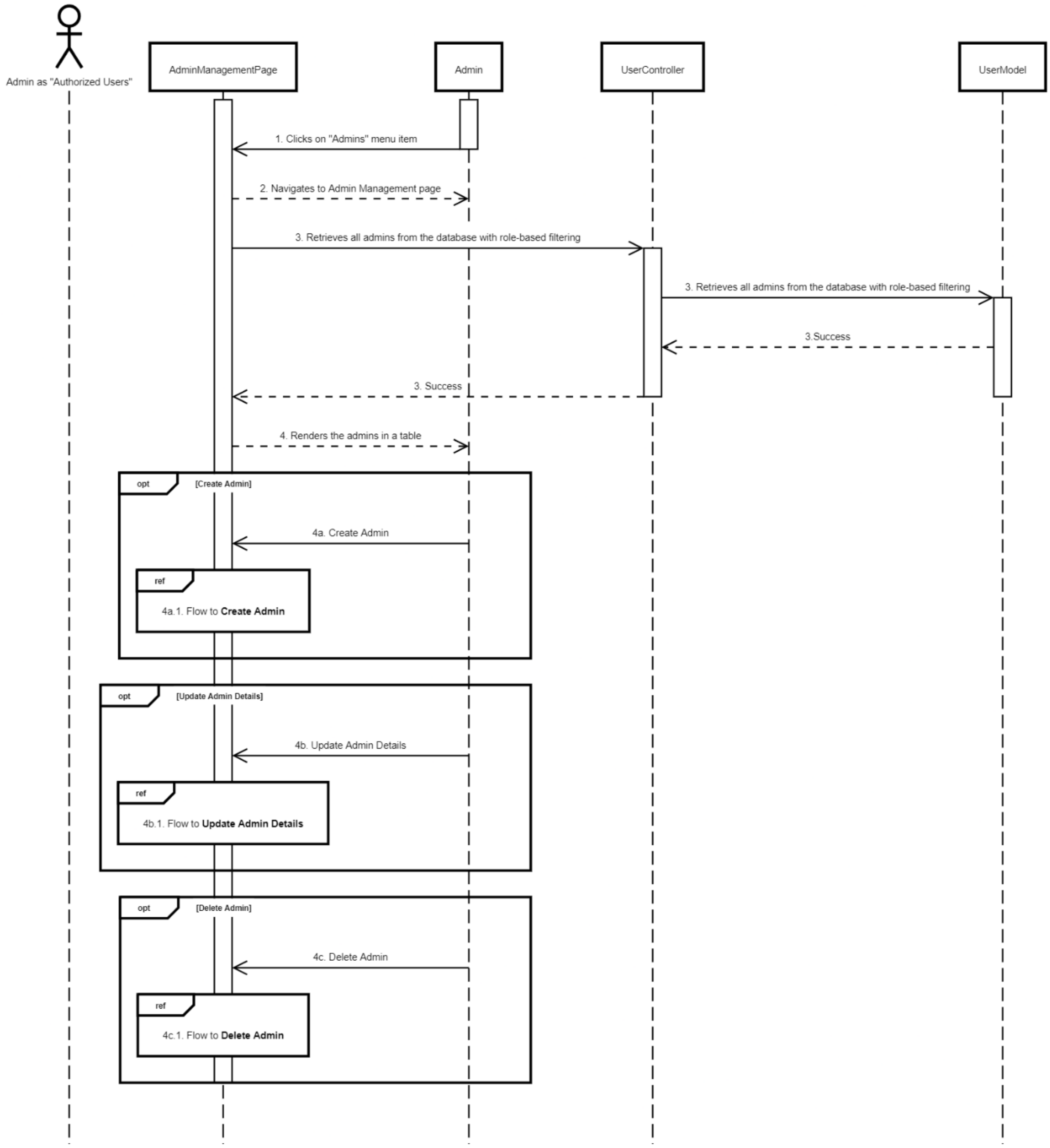


Figure 121: List Admins

4.2.5.24 Create Admin

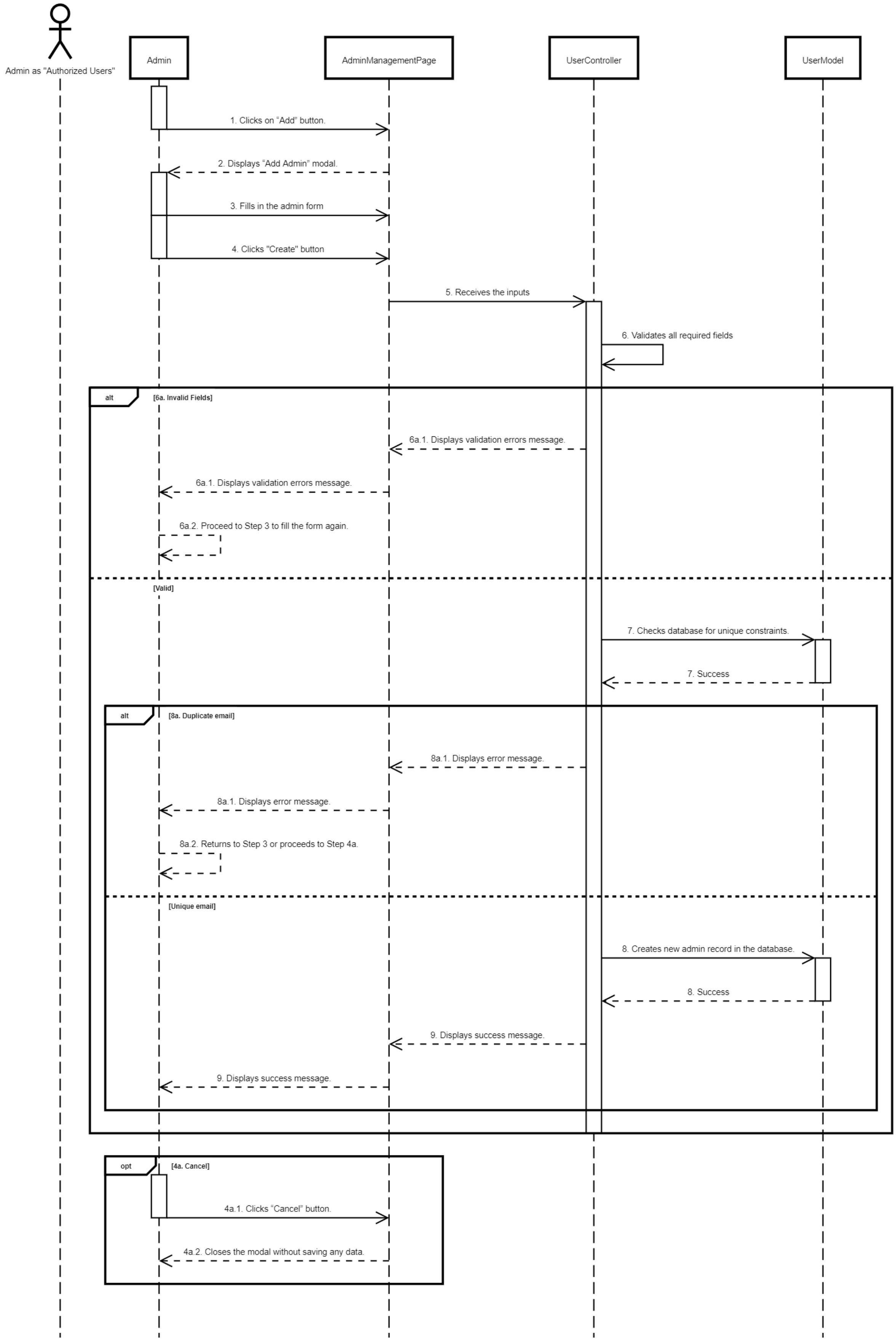


Figure 122: Create Admin

4.2.5.25 Update Admin Details

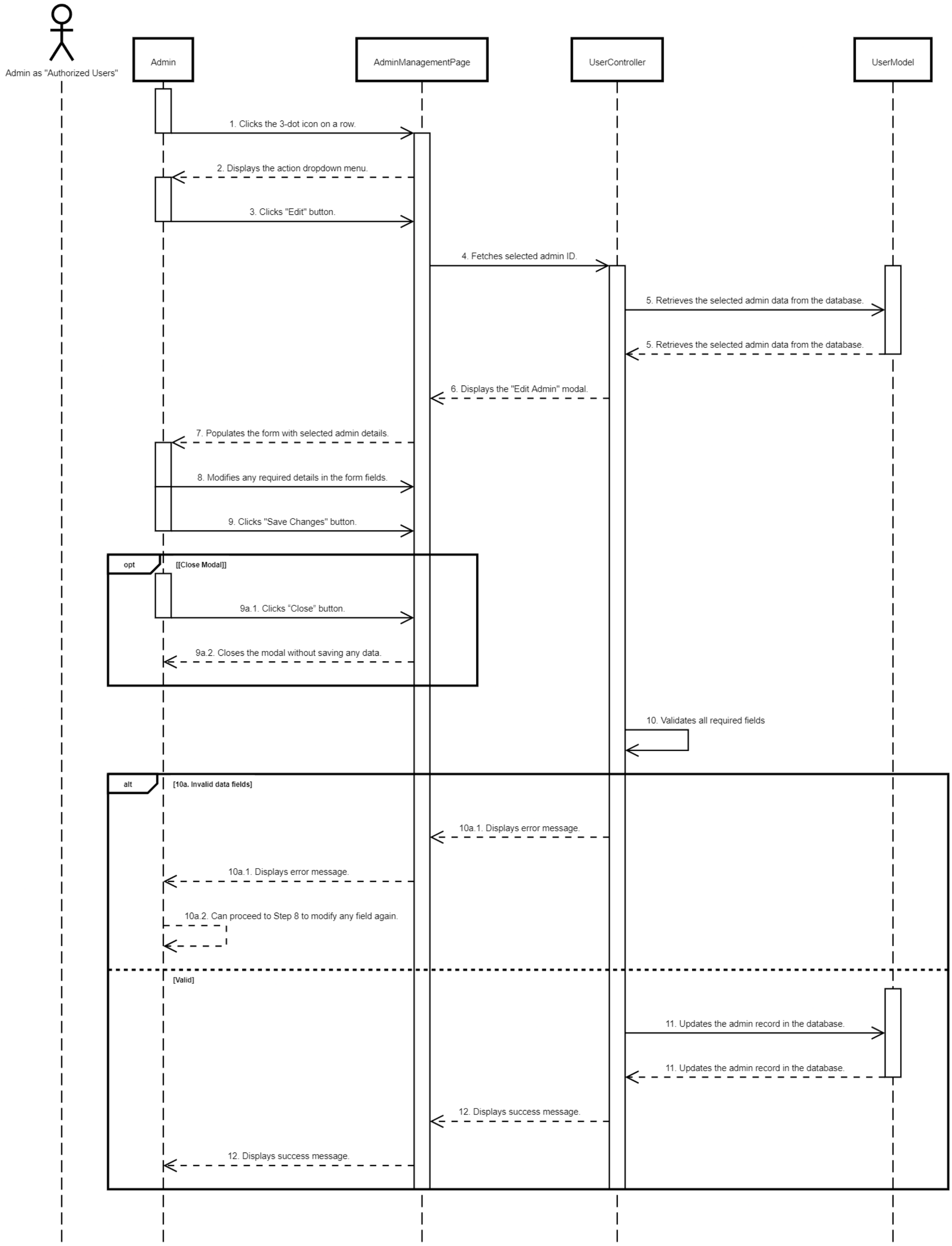


Figure 123: Update Admin Details

4.2.5.26 Delete Admin

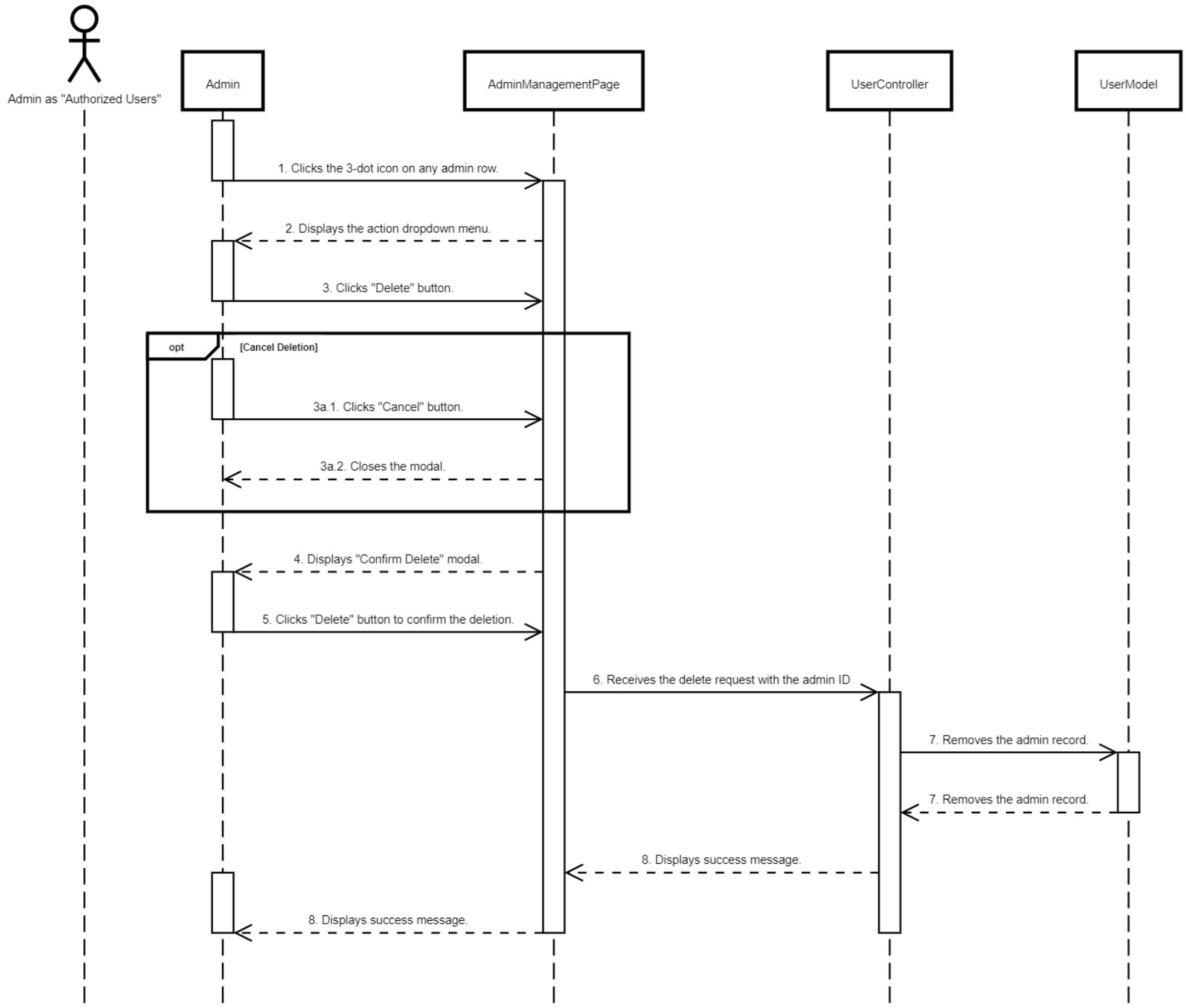


Figure 124: Delete Admin

4.2.5.27 Export Reports

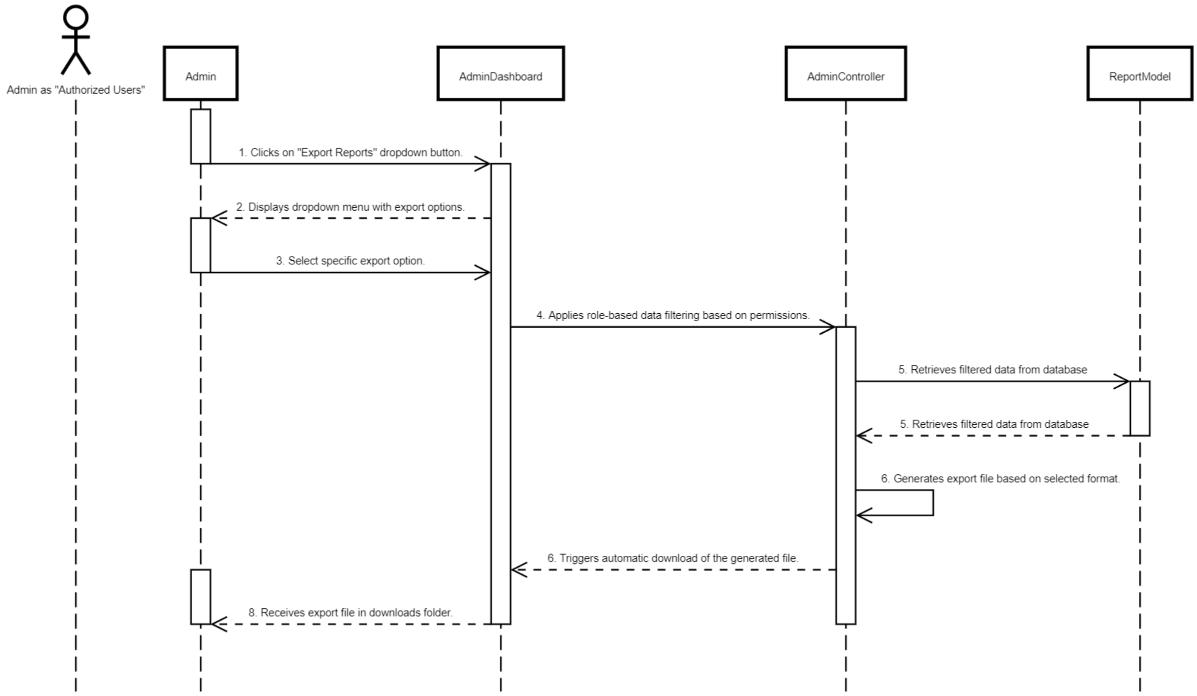


Figure 125: Export Reports

4.2.5.28 List Public Users

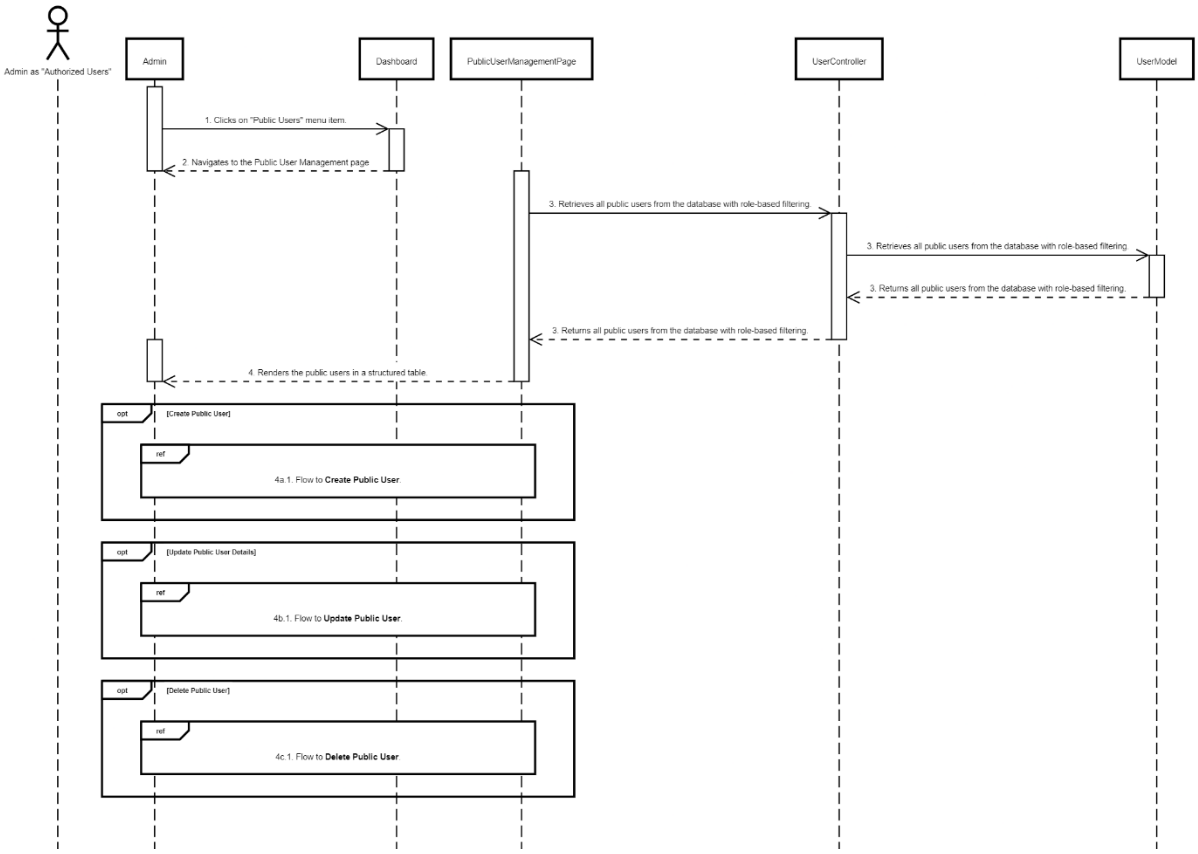


Figure 126: List Public Users

4.2.5.29 Create Public User

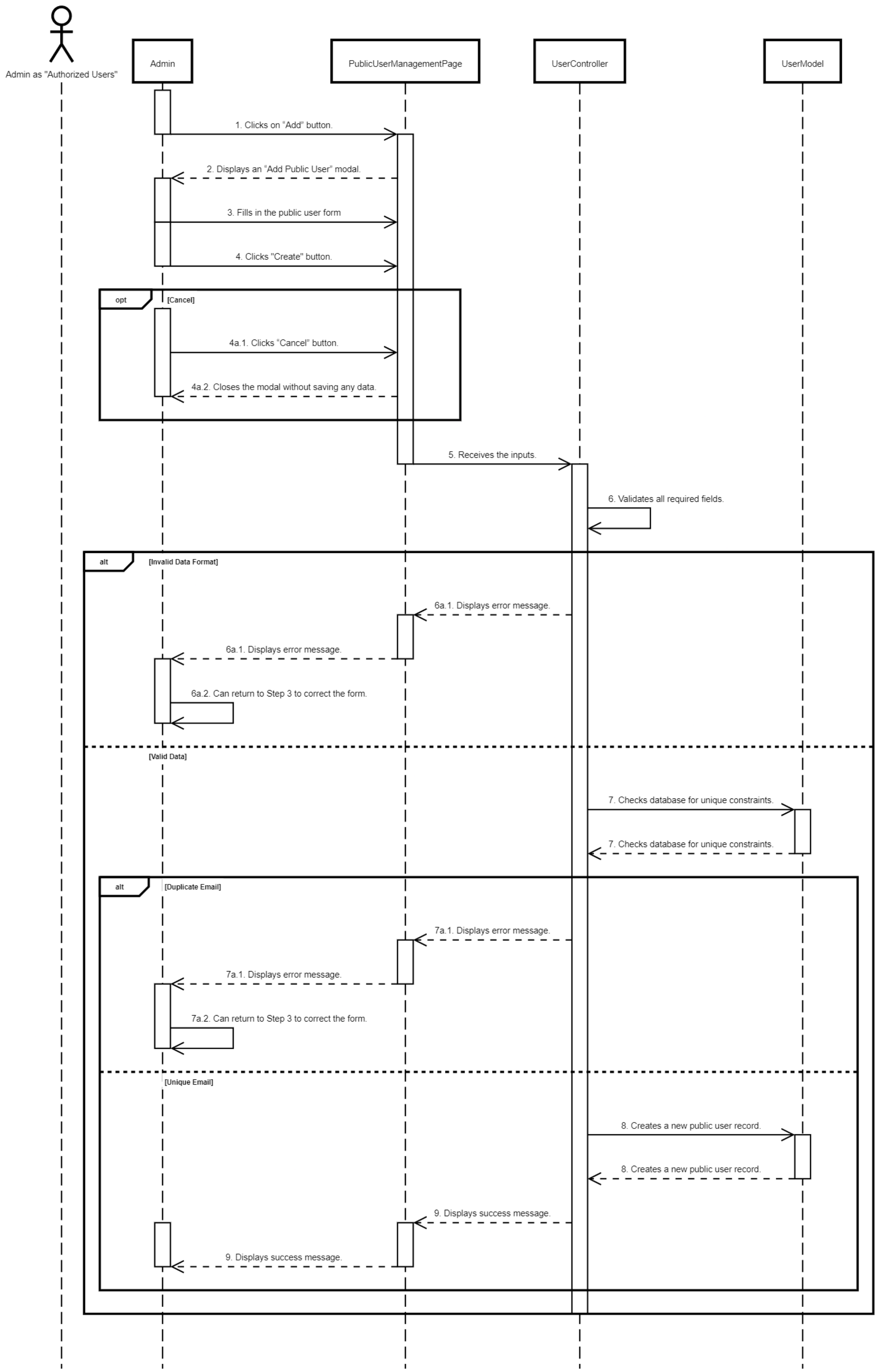

Admin as "Authorized Users"
Admin
PublicUserManagementPage
UserController
UserModel
1. Clicks on "Add" button.
2. Displays an "Add Public User" modal.
3. Fills in the public user form
4. Clicks "Create" button.
opt
[Cancel]
4a.1. Clicks "Cancel" button.
4a.2. Closes the modal without saving any data.
5. Receives the inputs.
6. Validates all required fields.
alt
[Invalid Data Format]
6a.1. Displays error message.
6a.1. Displays error message.
6a.2. Can return to Step 3 to correct the form.
[Valid Data]
7. Checks database for unique constraints.
7. Checks database for unique constraints.
alt
[Duplicate Email]
7a.1. Displays error message.
7a.1. Displays error message.
7a.2. Can return to Step 3 to correct the form.
[Unique Email]
8. Creates a new public user record.
8. Creates a new public user record.
9. Displays success message.
9. Displays success message.

Figure 127: Create Public User

4.2.5.30 Update Public User Details

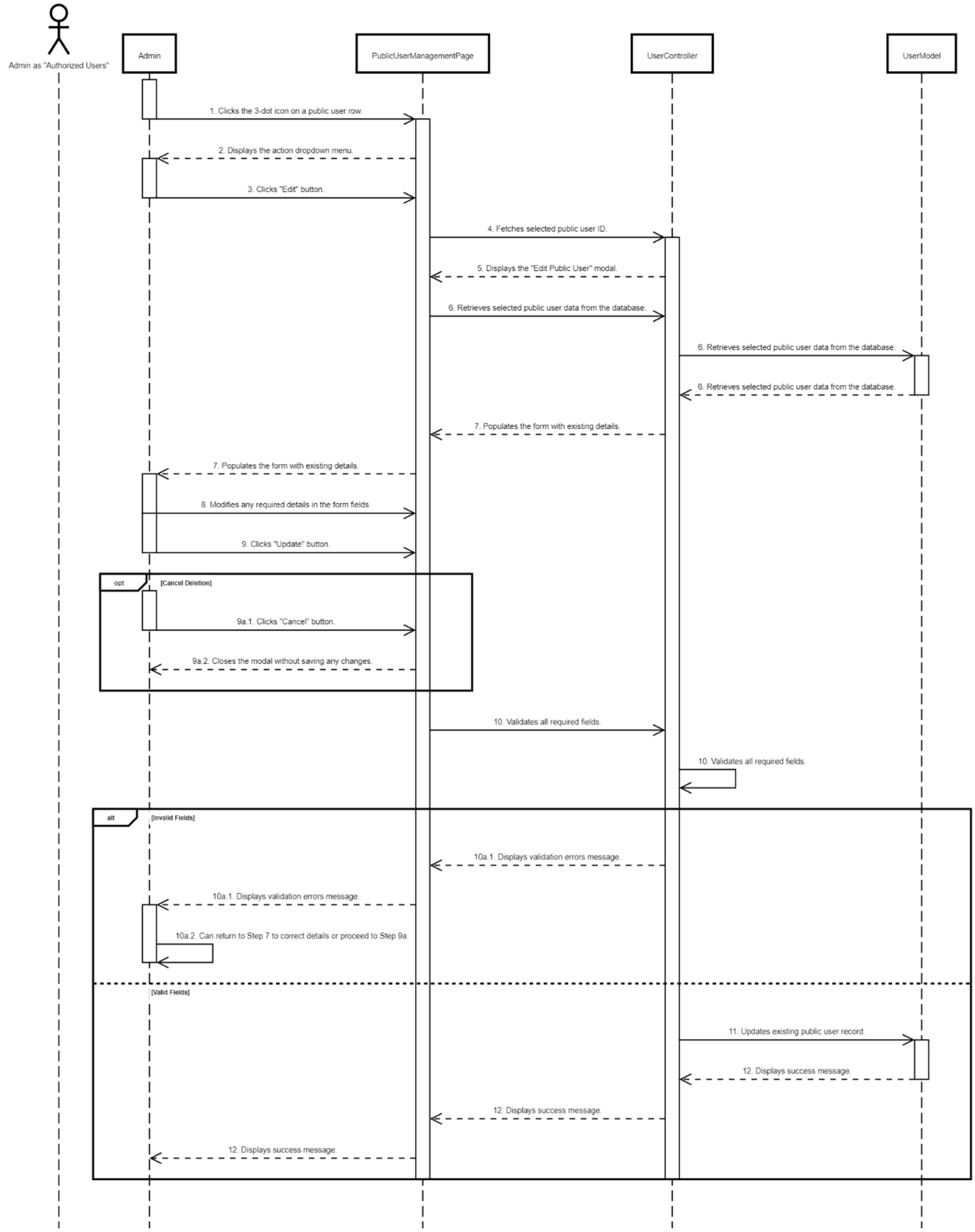


Figure 128: Update Public User Details

4.2.5.31 Delete Public User

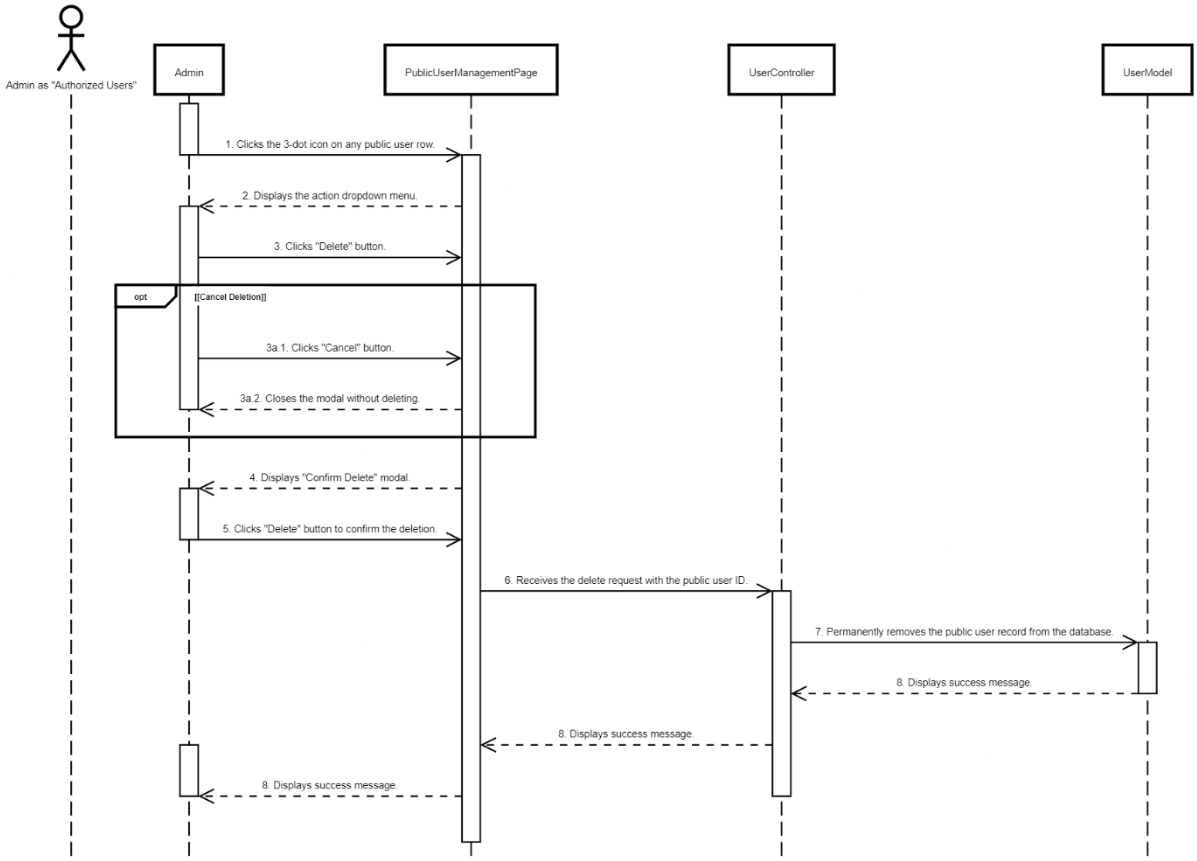


Figure 129: Delete Public User

4.2.5.32 List Session Logs

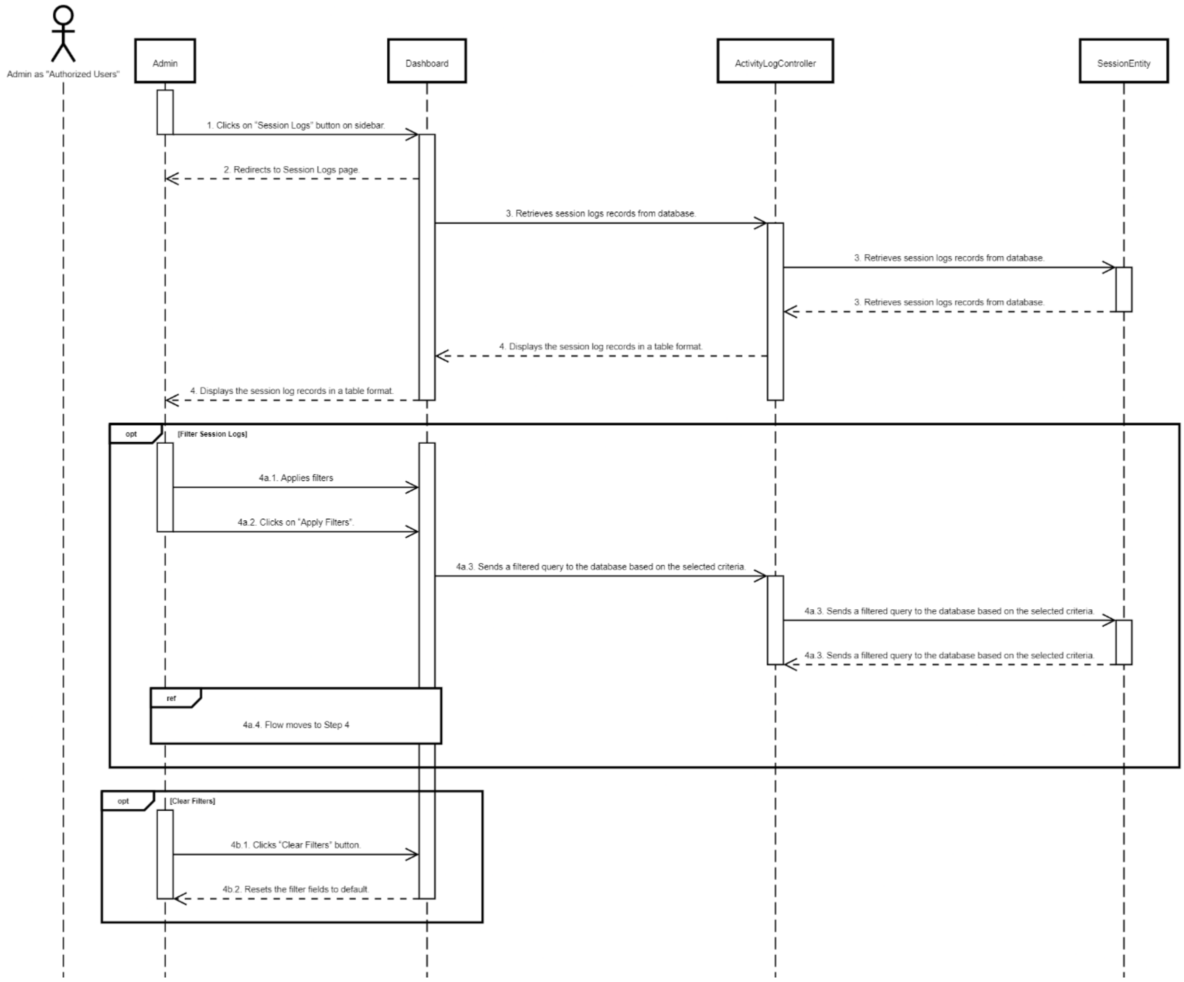


Figure 130: List Session Logs

4.2.5.33 List Law Enforcements

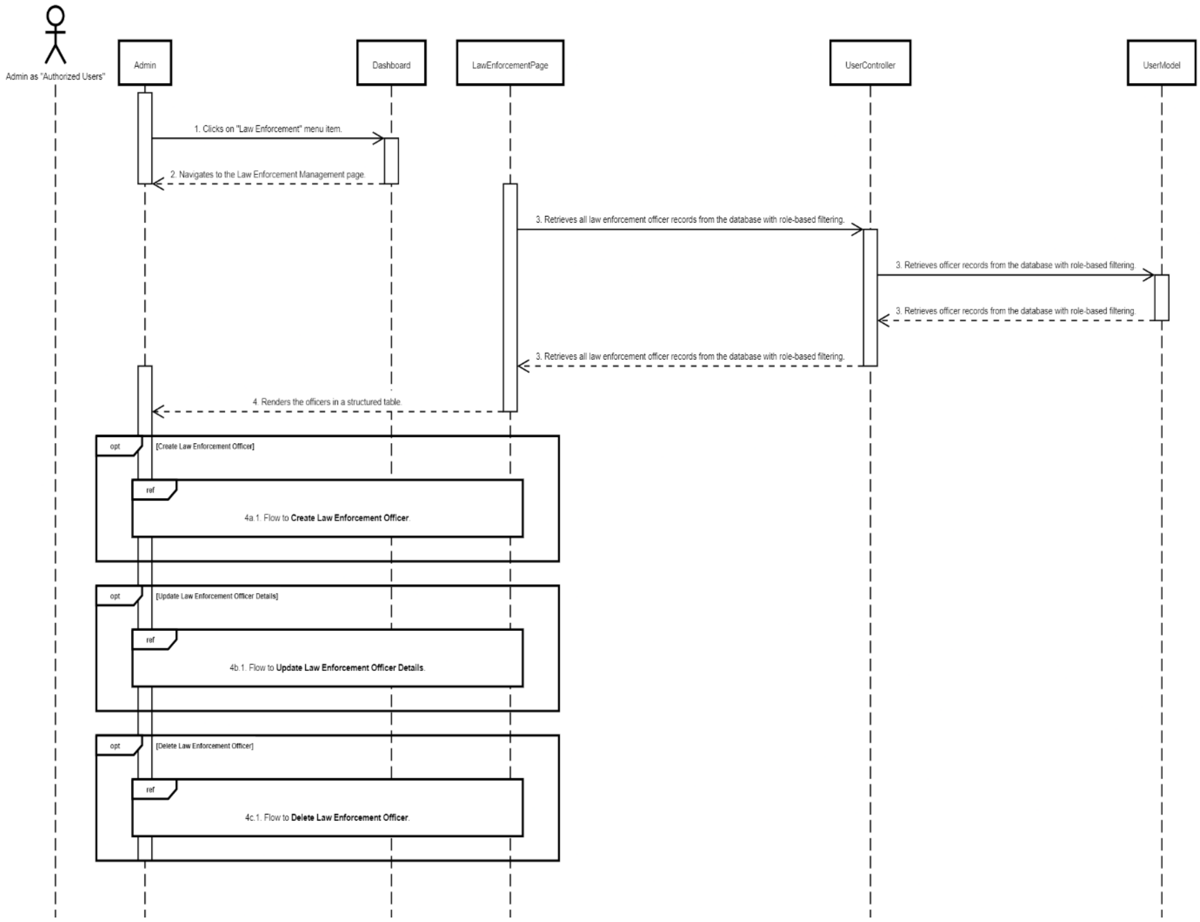


Figure 131: List Law Enforcements

4.2.5.34 Create Law Enforcement

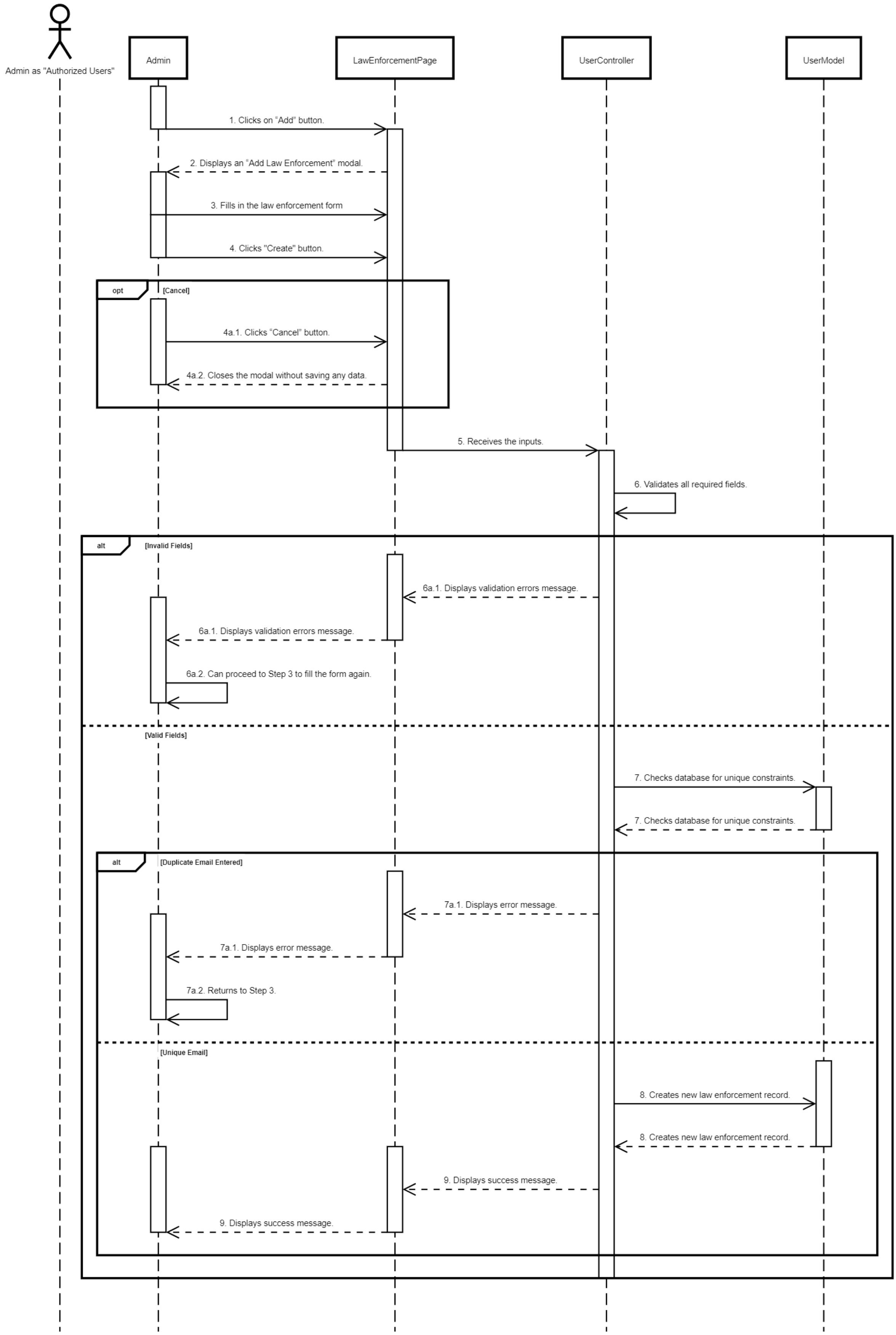


Figure 132: Create Law Enforcement

4.2.5.35 Update Law Enforcement Details

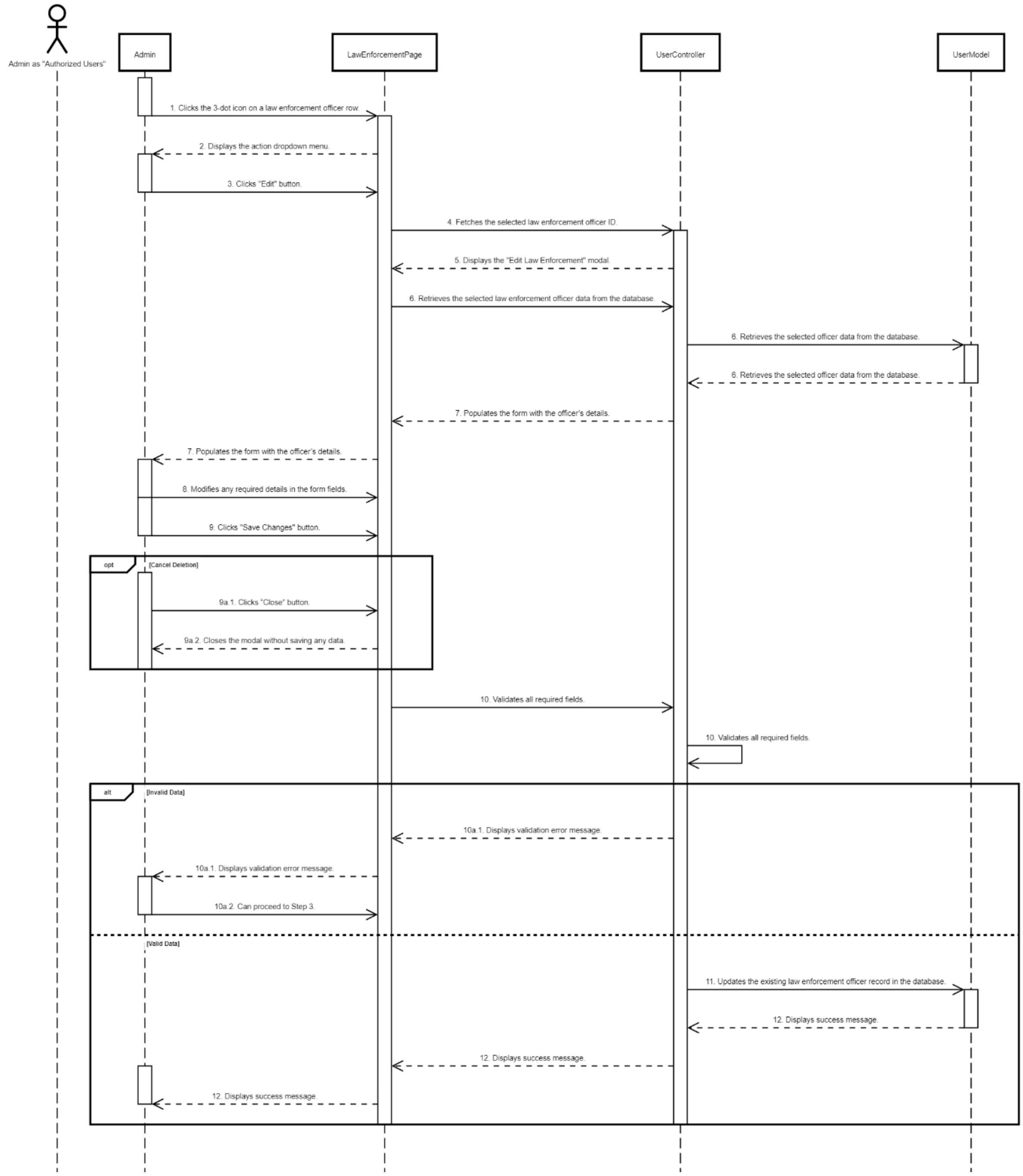


Figure 133: Update Law Enforcement Details

4.2.5.36 Delete Law Enforcement

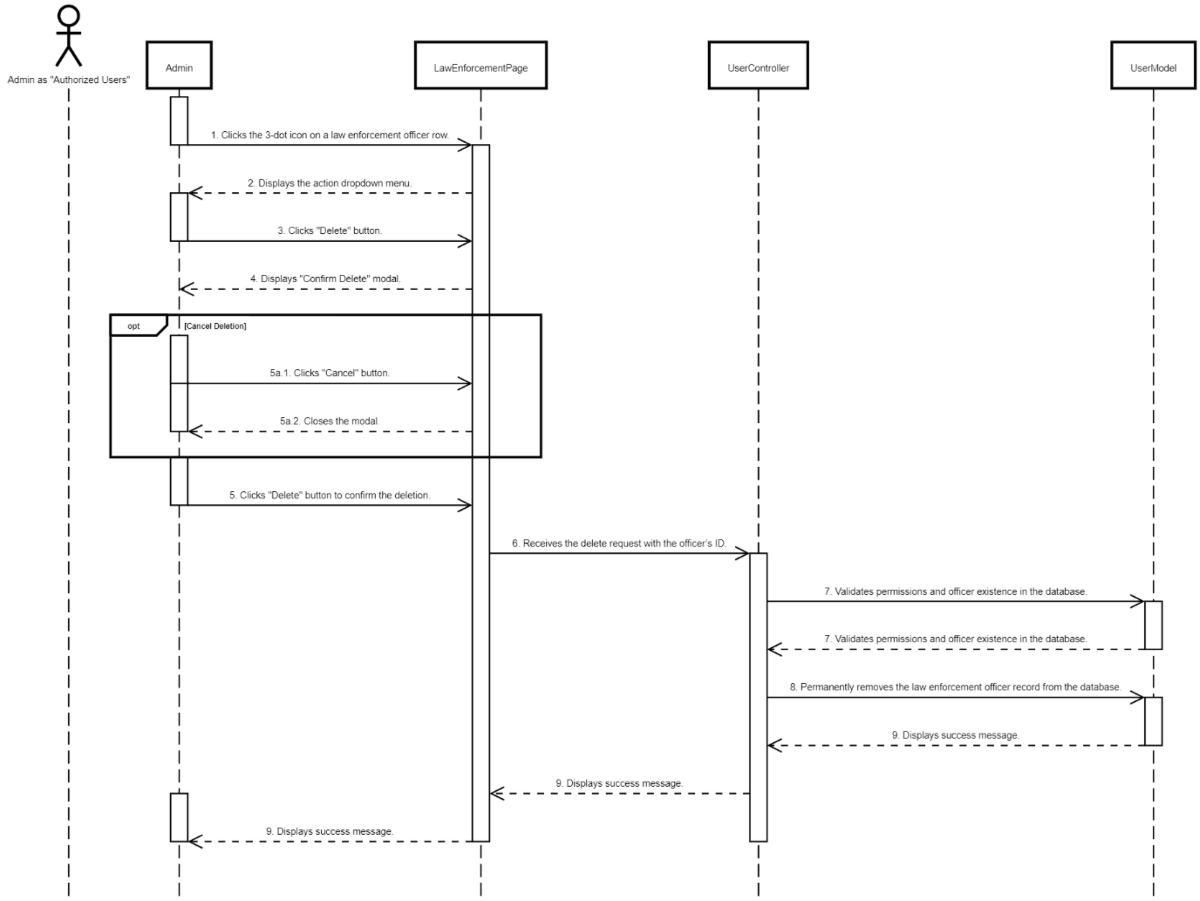


Figure 134: Delete Law Enforcement

4.2.5.37 List Social Workers

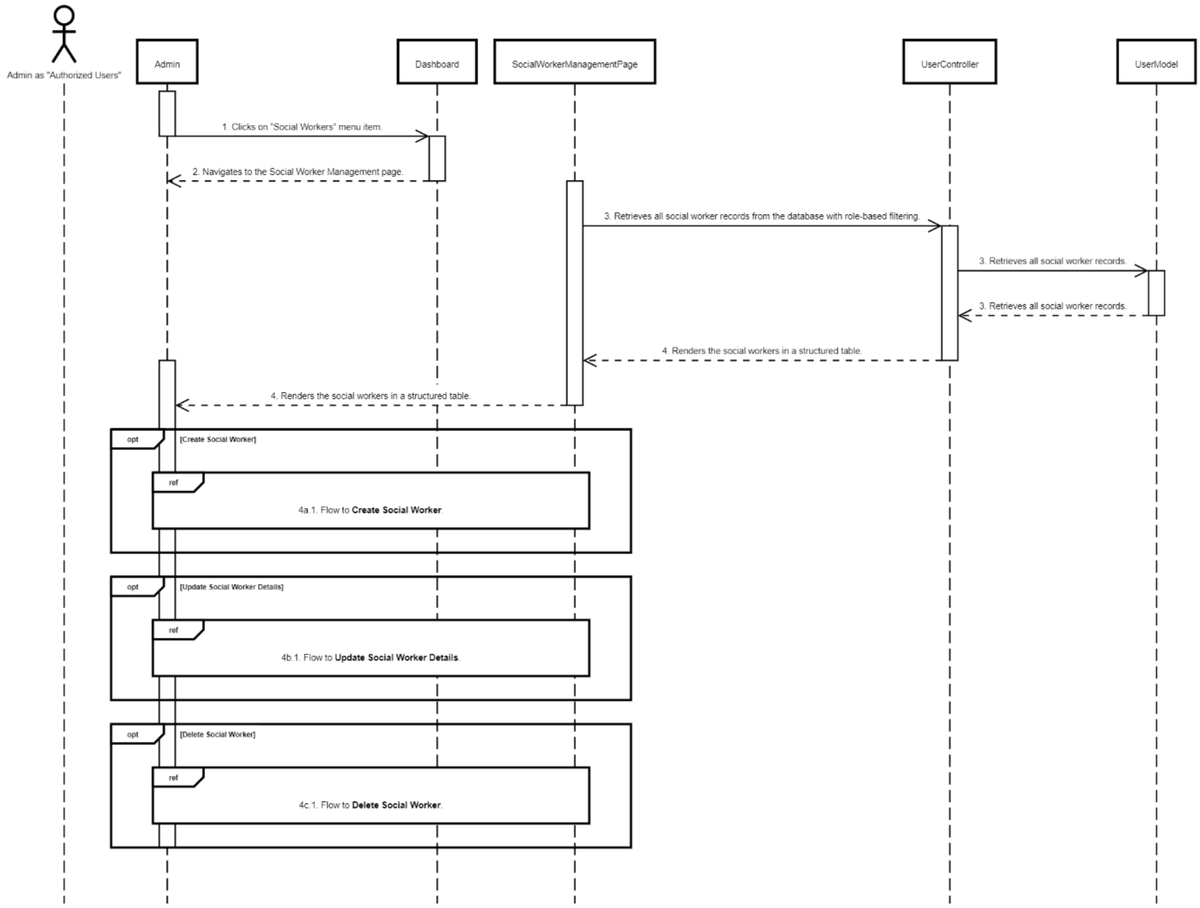


Figure 135: List Social Workers

4.2.5.38 Create Social Worker

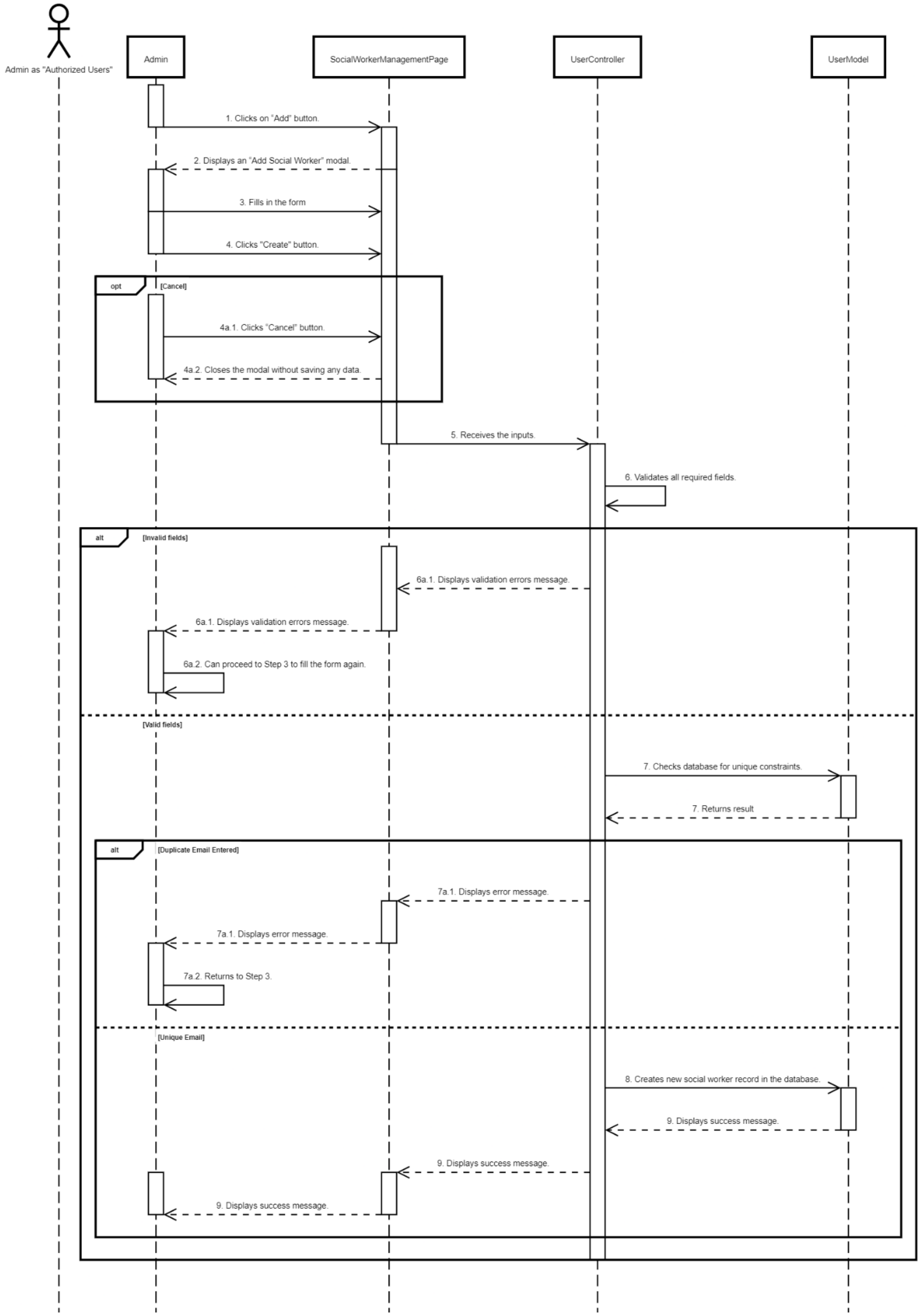


Figure 136: Create Social Worker

4.2.5.39 Update Social Worker Details

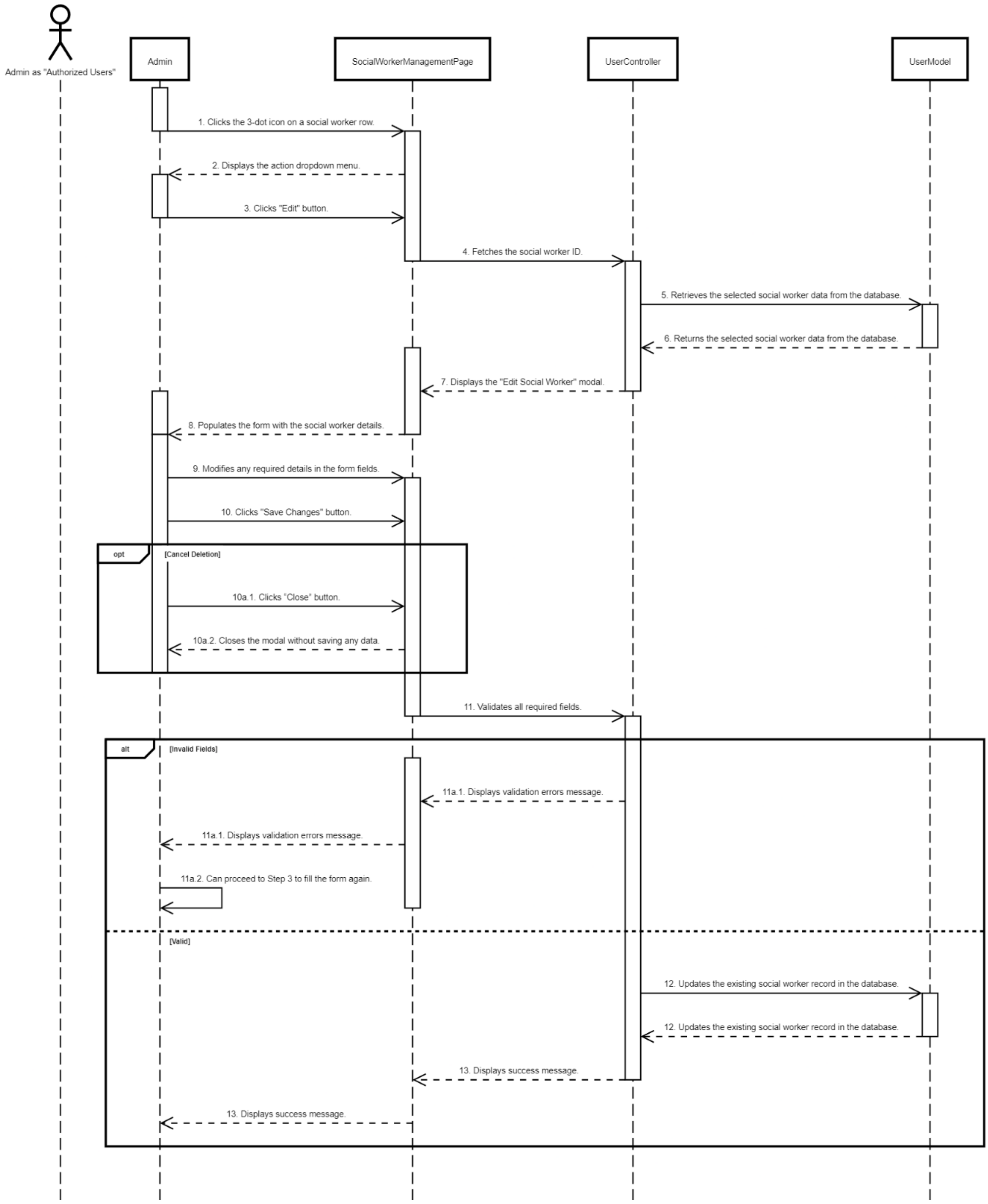


Figure 137: Update Social Worker

4.2.5.40 Delete Social Worker

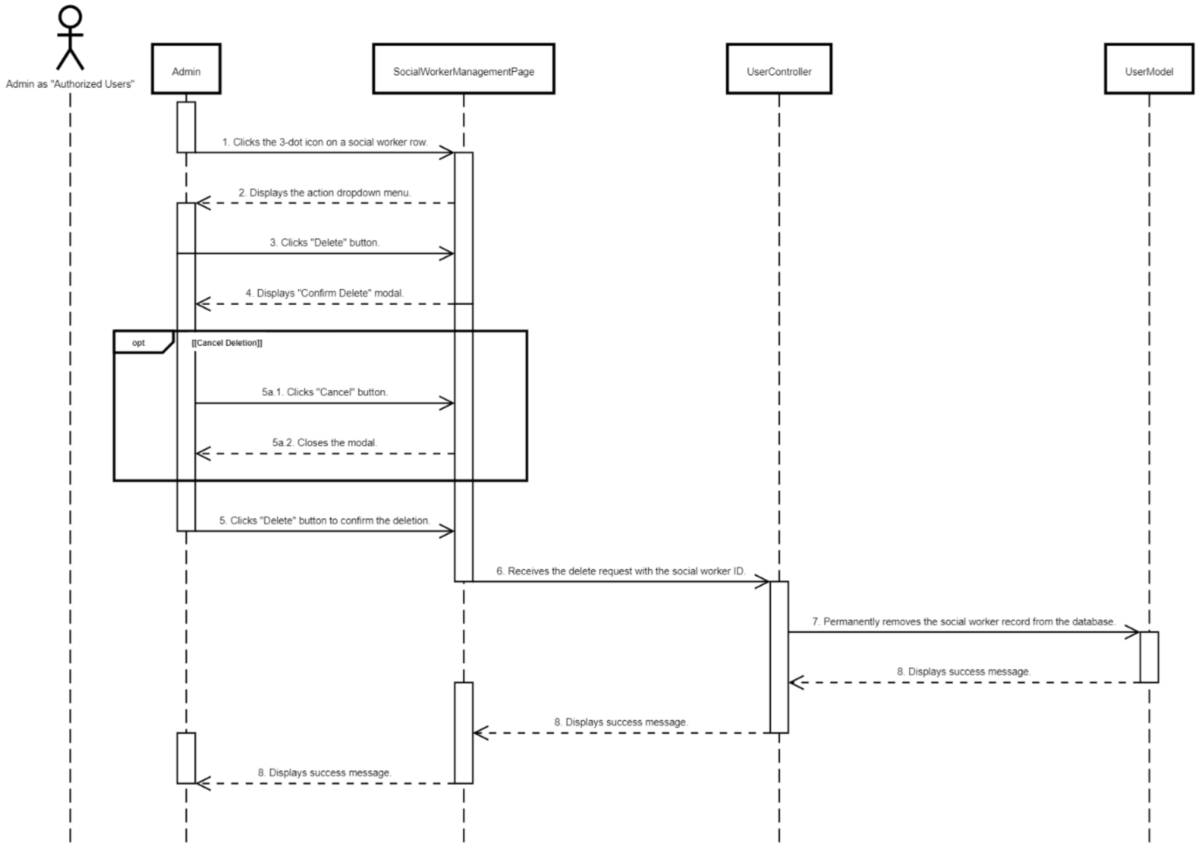


Figure 138: Delete Social Worker

4.2.5.41 List Chief Welfare Officers

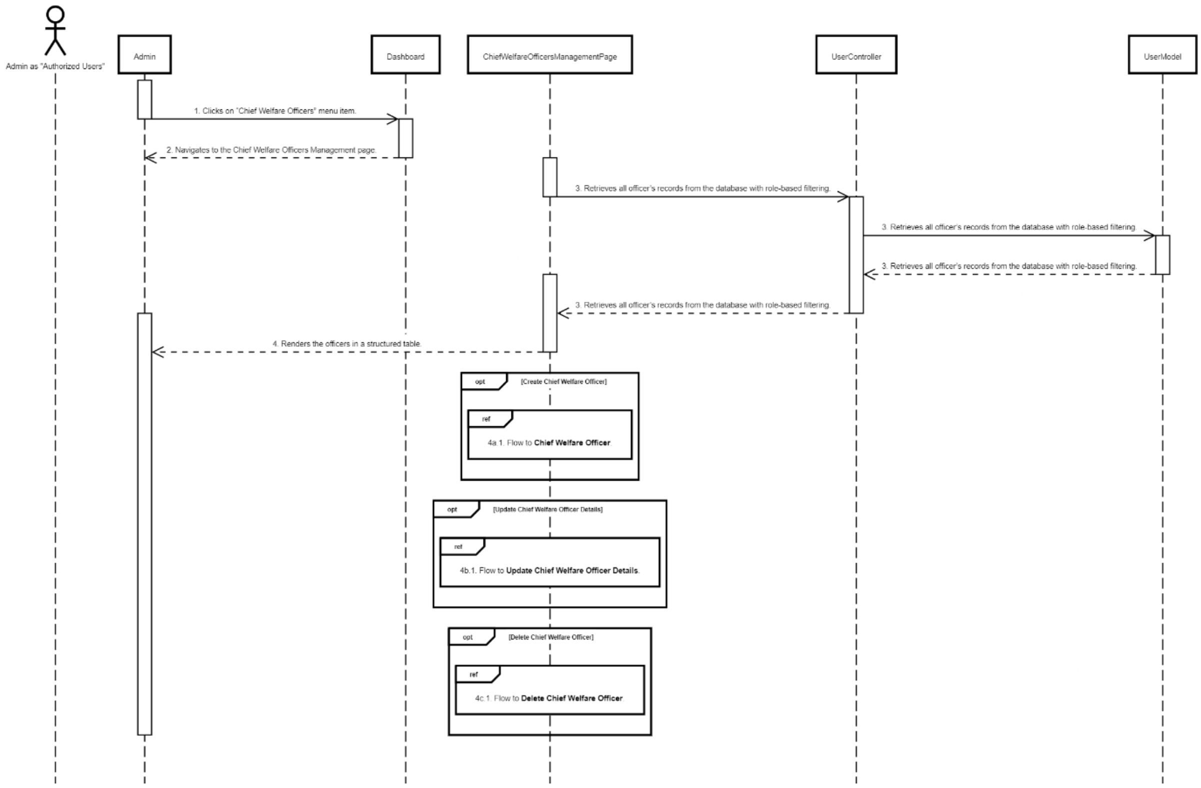


Figure 139: List Chief Welfare Officers

4.2.5.42 Create Chief Welfare Officer

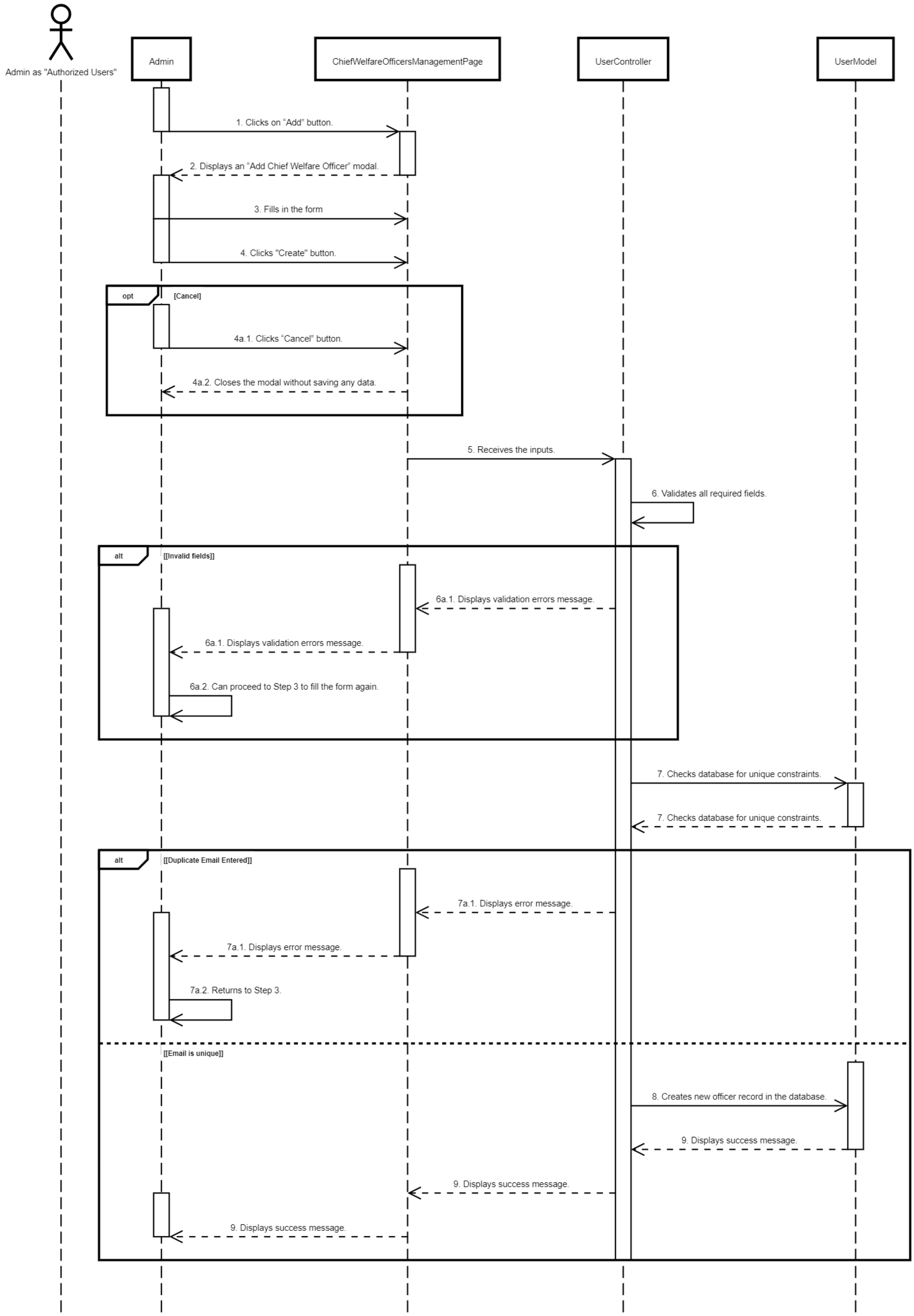
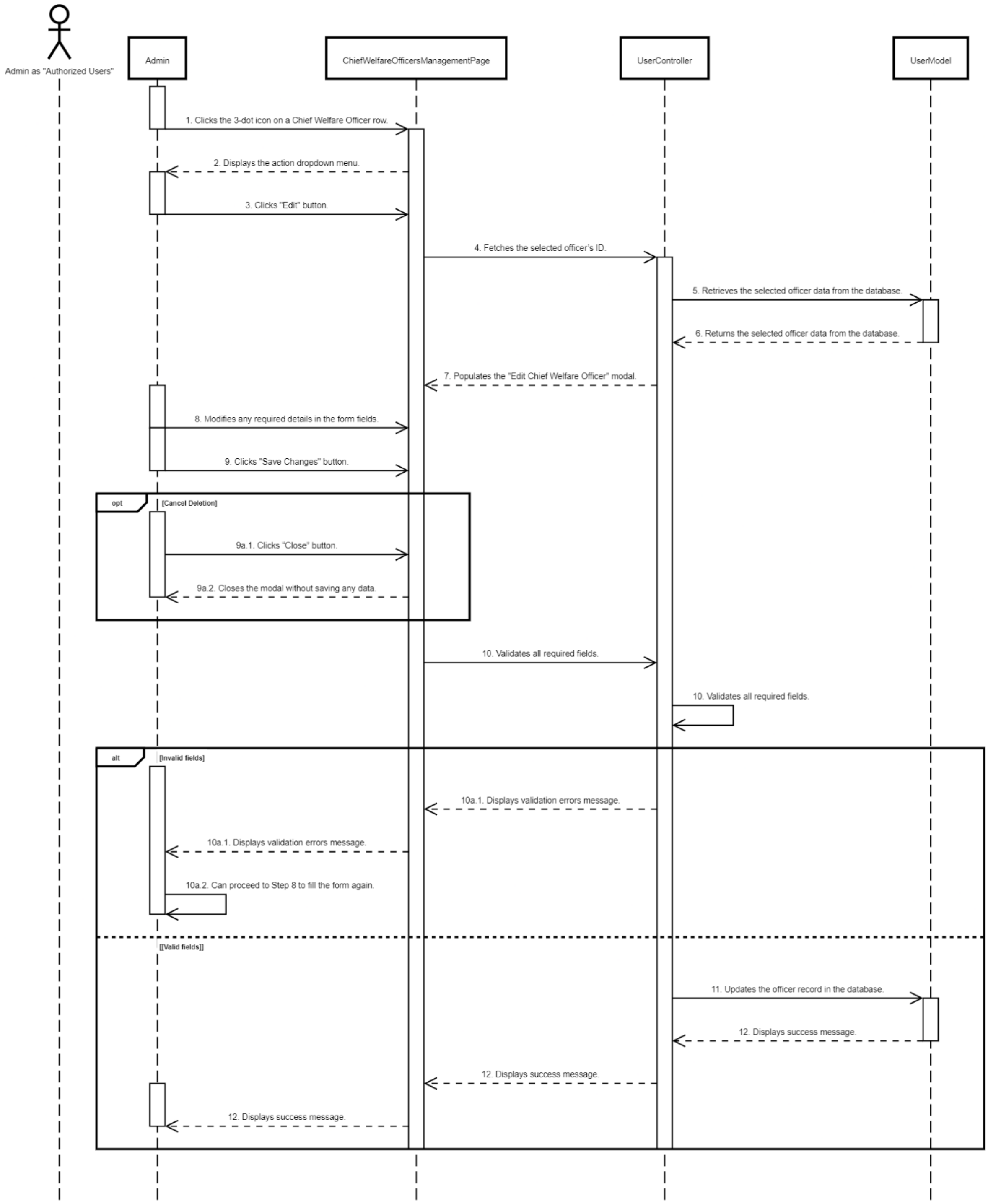


Figure 140: Create Chief Welfare Officer

4.2.5.43 Update Chief Welfare Officer Details



4.2.5.44 Delete Chief Welfare Officer

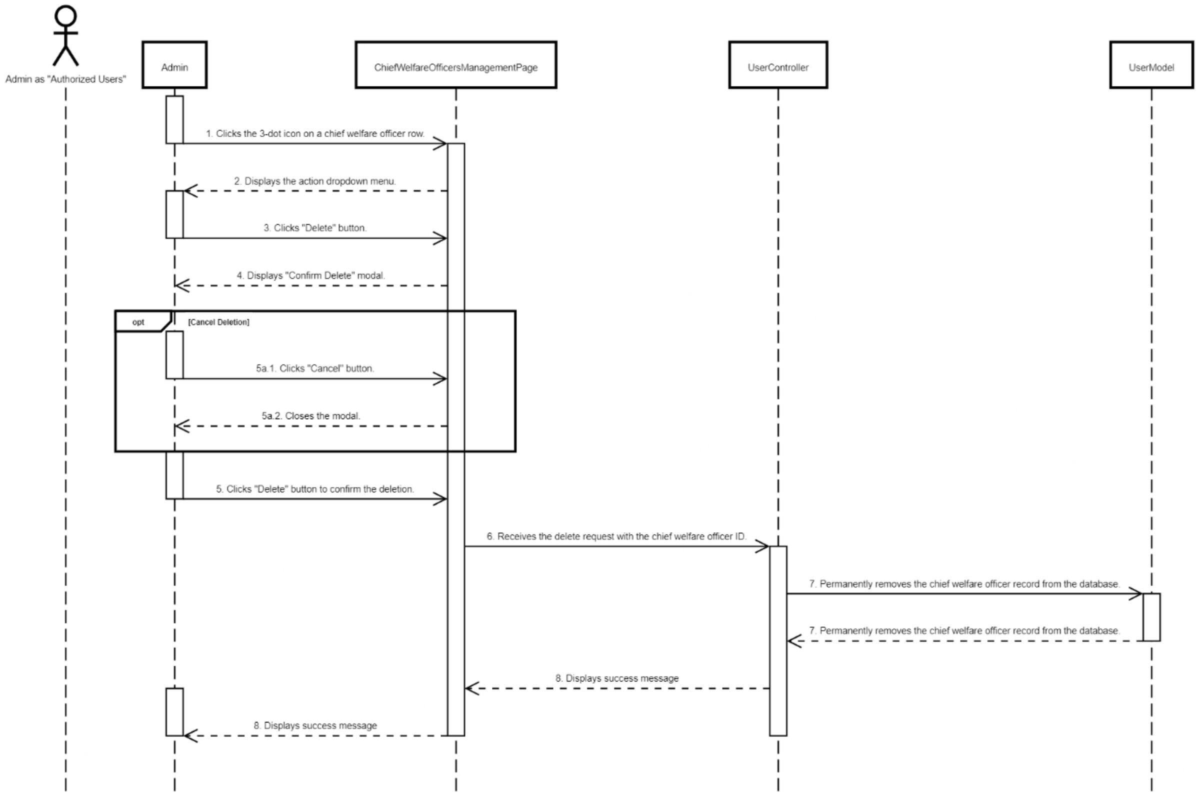


Figure 141: Delete Chief Welfare Officer

4.2.5.45 List Healthcare Professionals

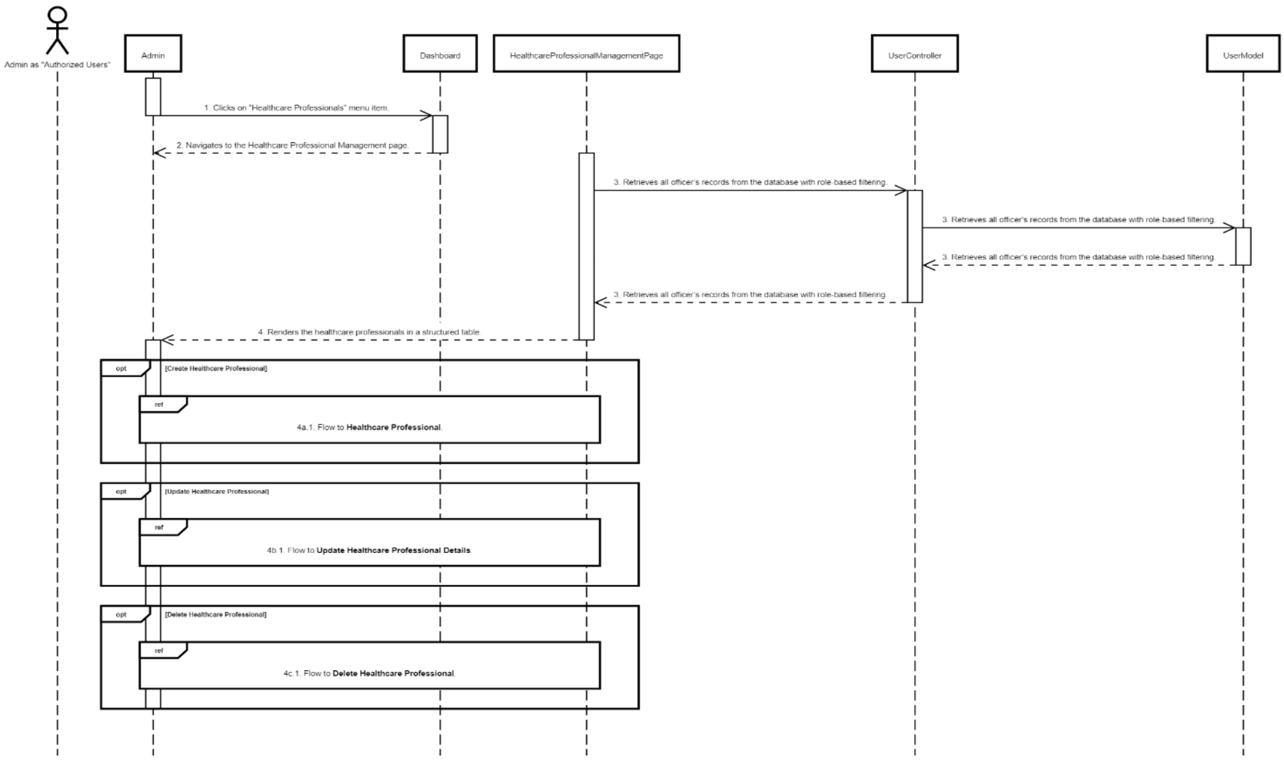


Figure 142: List Healthcare Professionals

4.2.5.46 Create Healthcare Professional

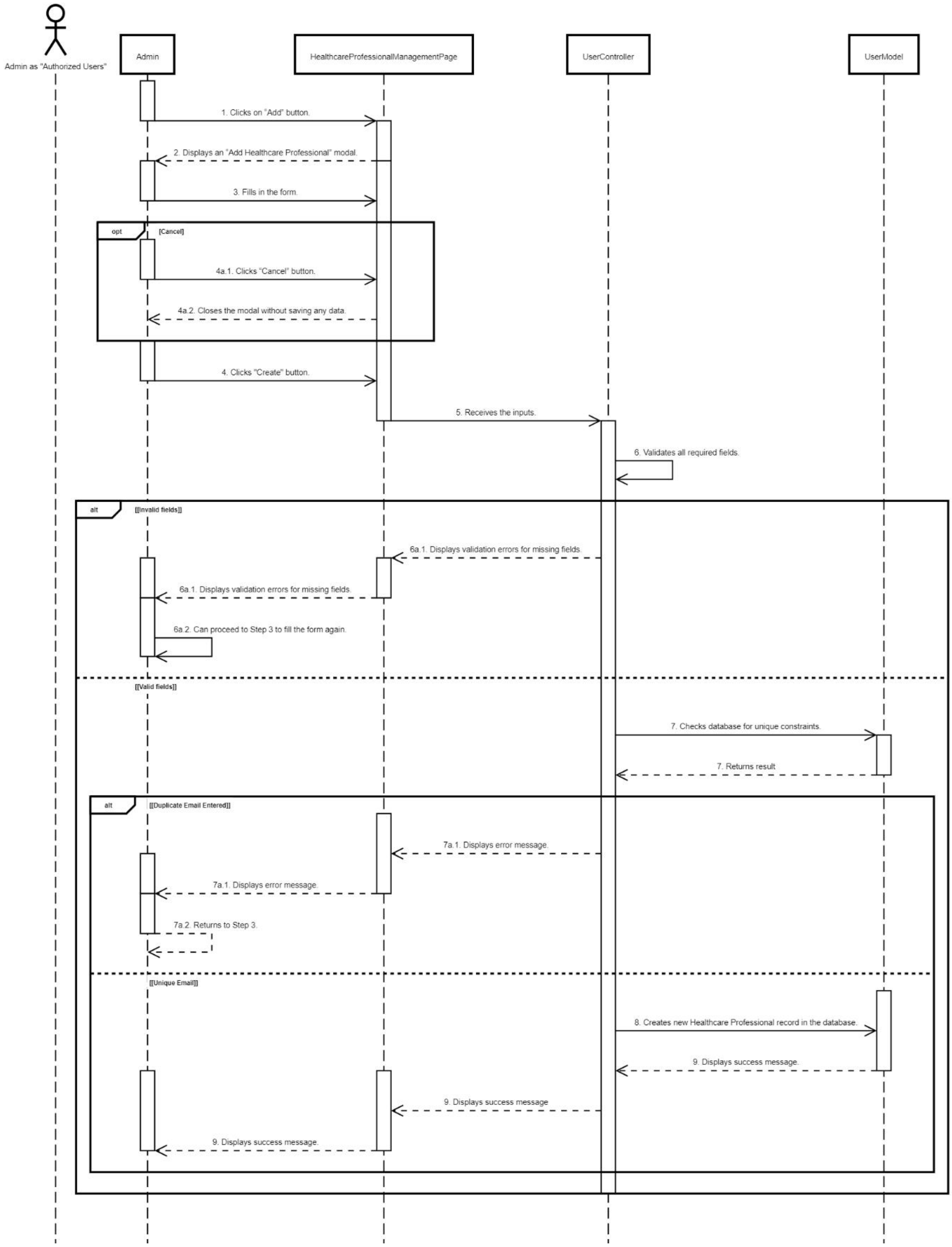


Figure 143: Create Healthcare Professional

4.2.5.47 Update Healthcare Professional Details

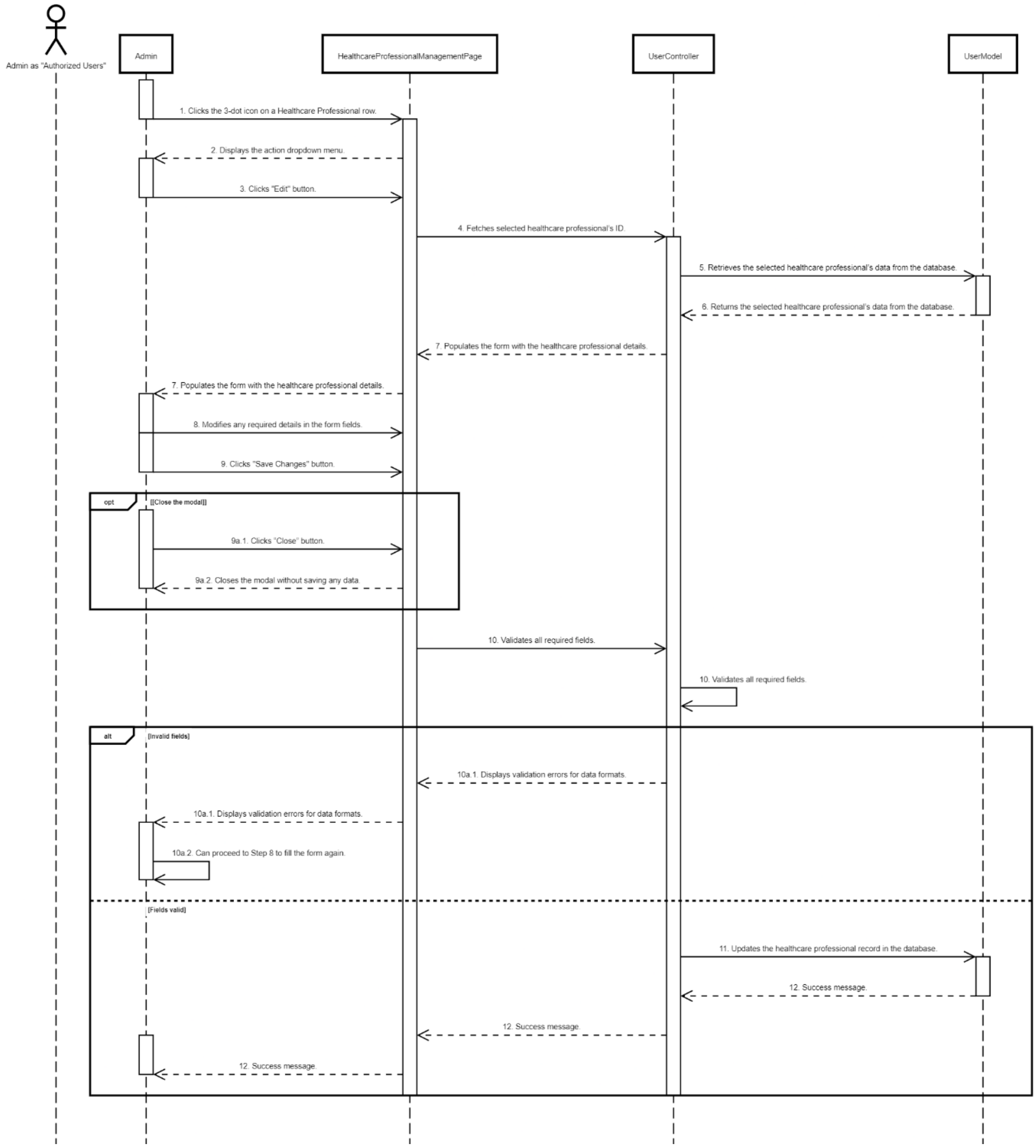


Figure 144: Update Healthcare Professional Details

4.2.5.48 Delete Healthcare Professional

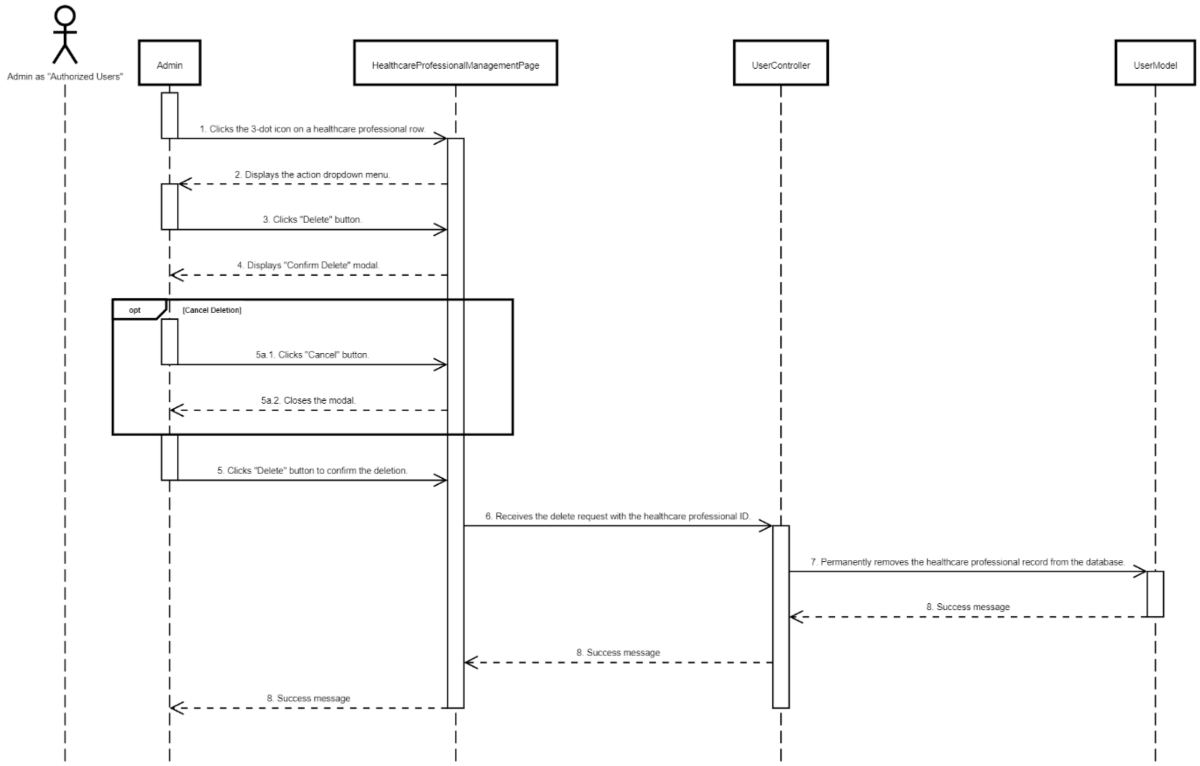


Figure 145: Delete Healthcare Professional

4.2.5.49 List Permissions

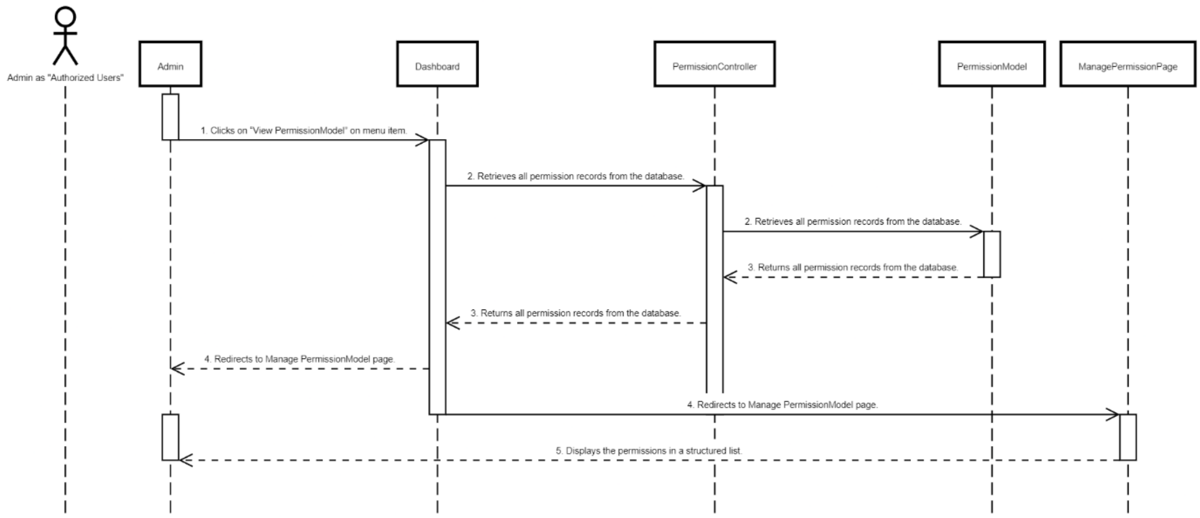


Figure 146: List Permissions

4.2.5.50 List Roles

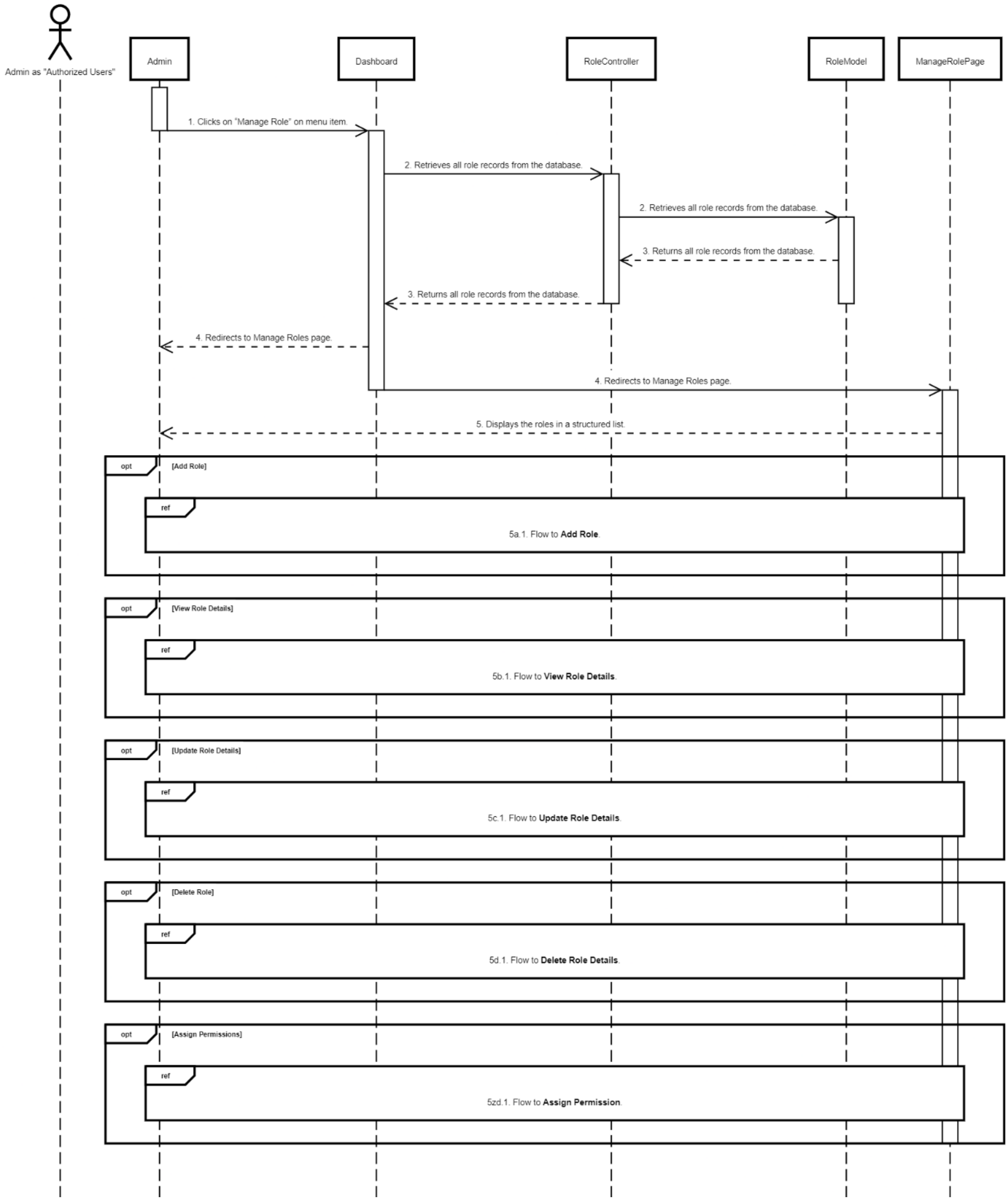


Figure 147: List Roles

4.2.5.51 Add Role

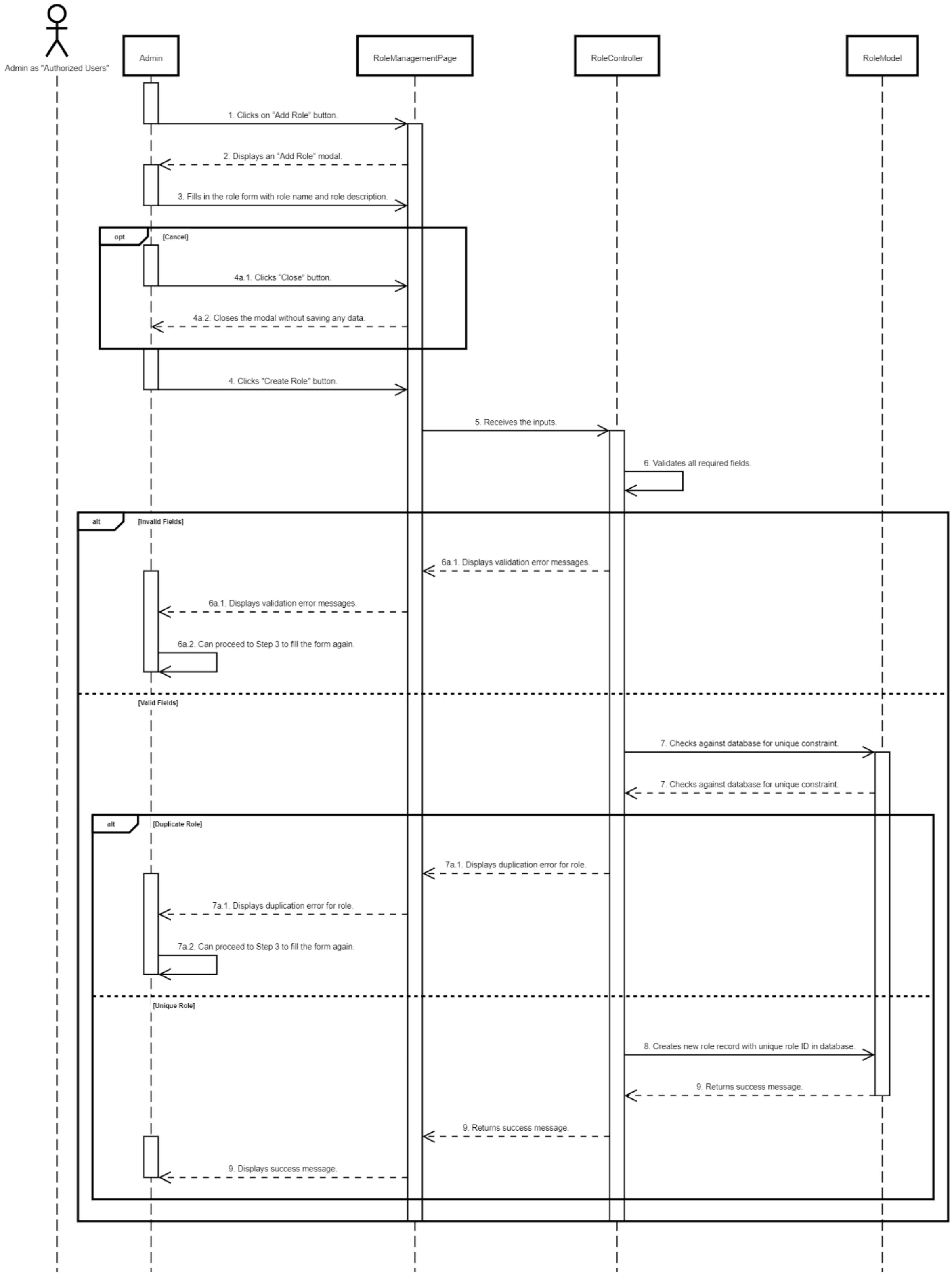


Figure 148: Add Role

4.2.5.52 View Role Details

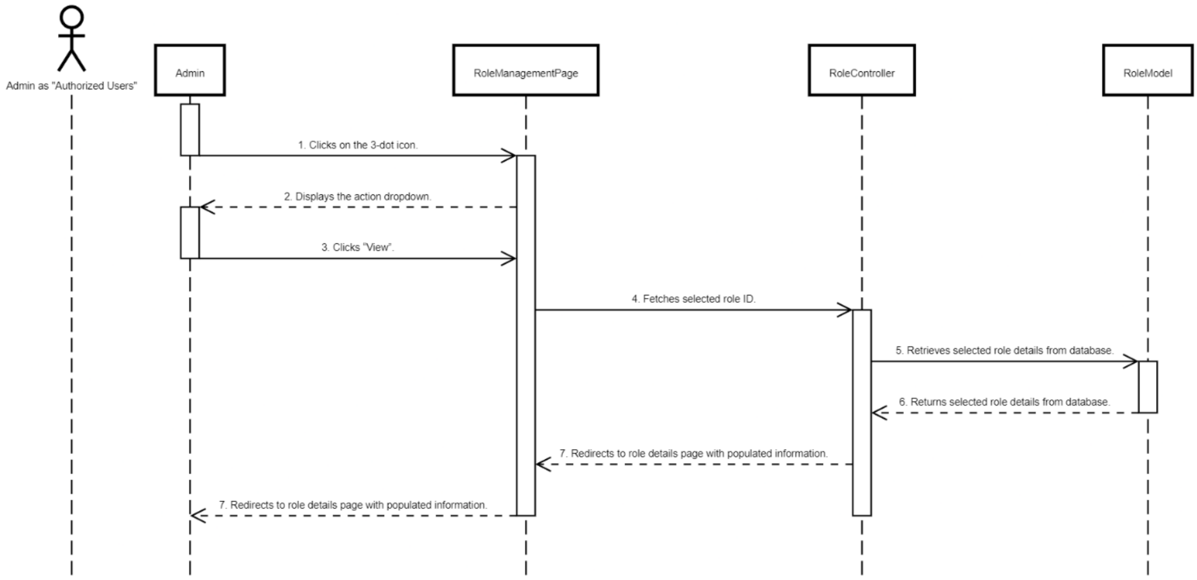


Figure 149: View Role Details

4.2.5.53 Update Role Details

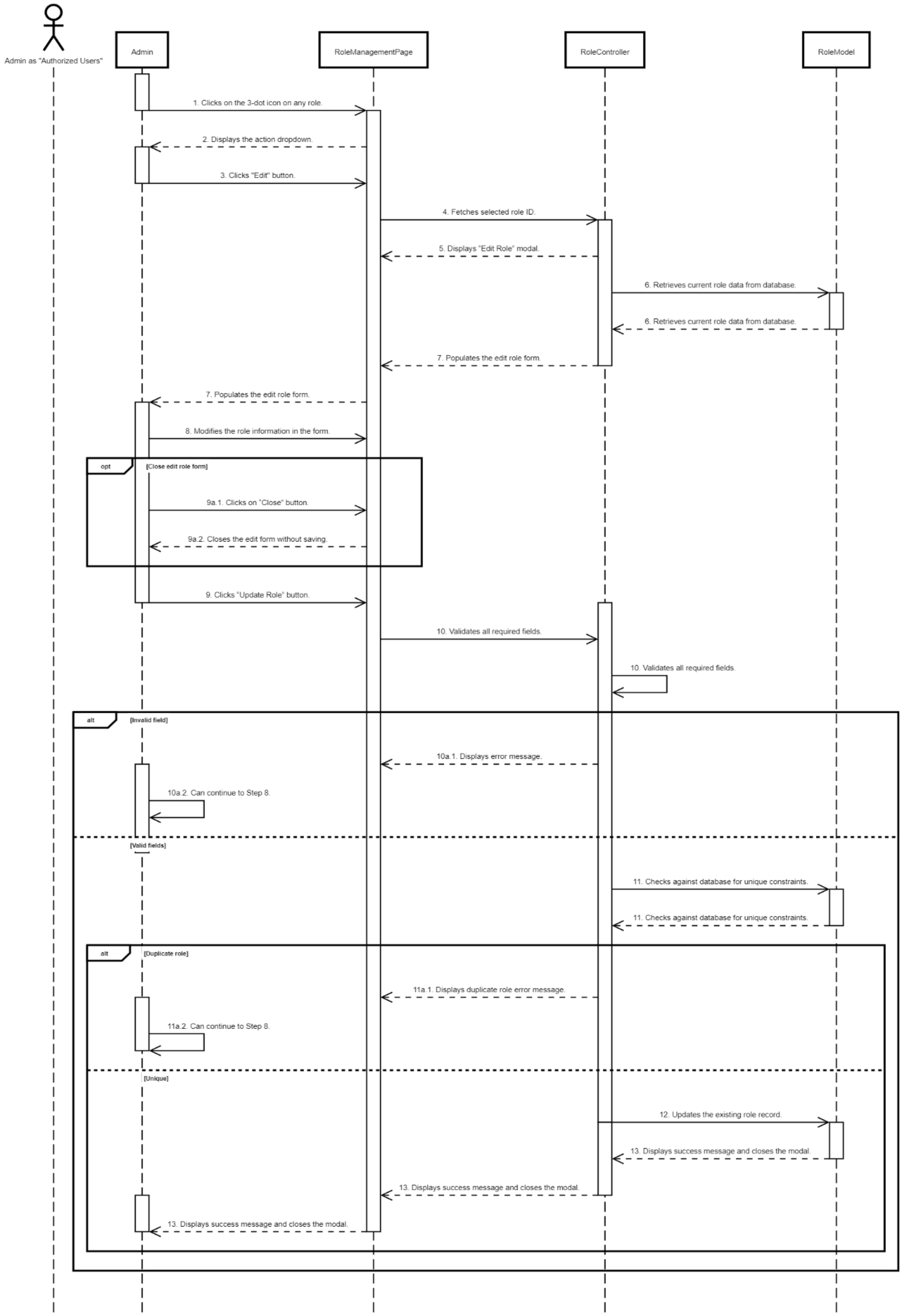


Figure 150: Update Role Details

4.2.5.54 Delete Role

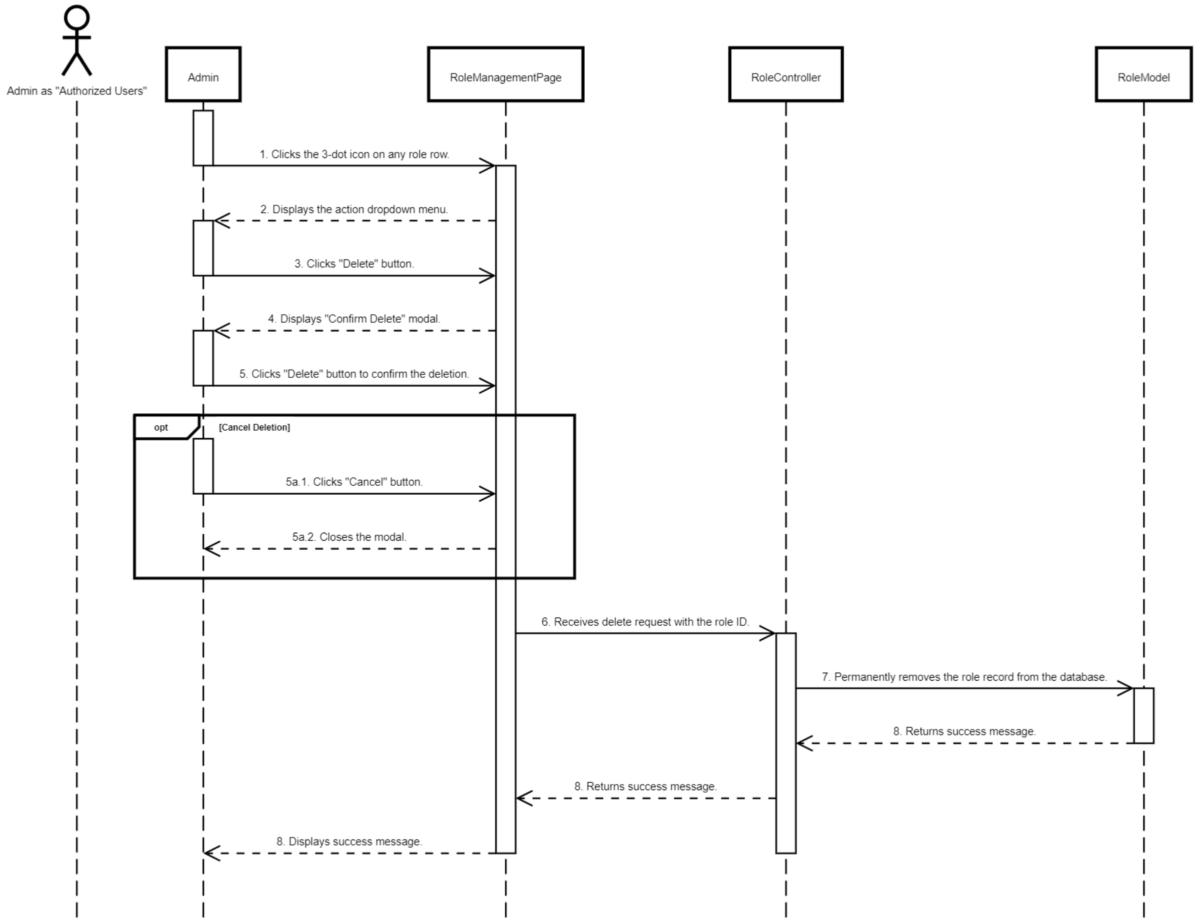


Figure 151: Delete Role

4.2.5.55 Assign Permissions

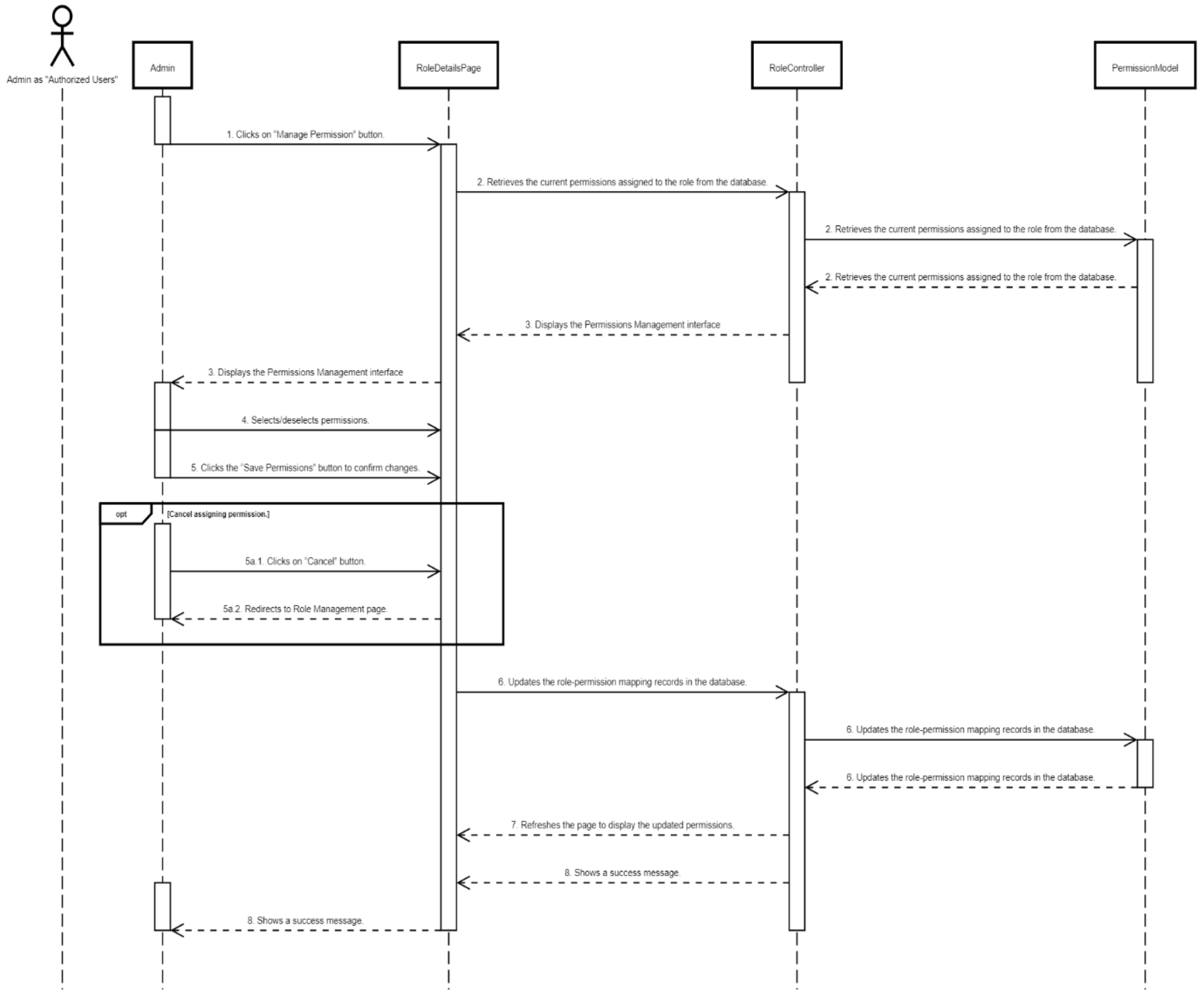


Figure 152: Assign Permissions

4.2.5.56 Create Query

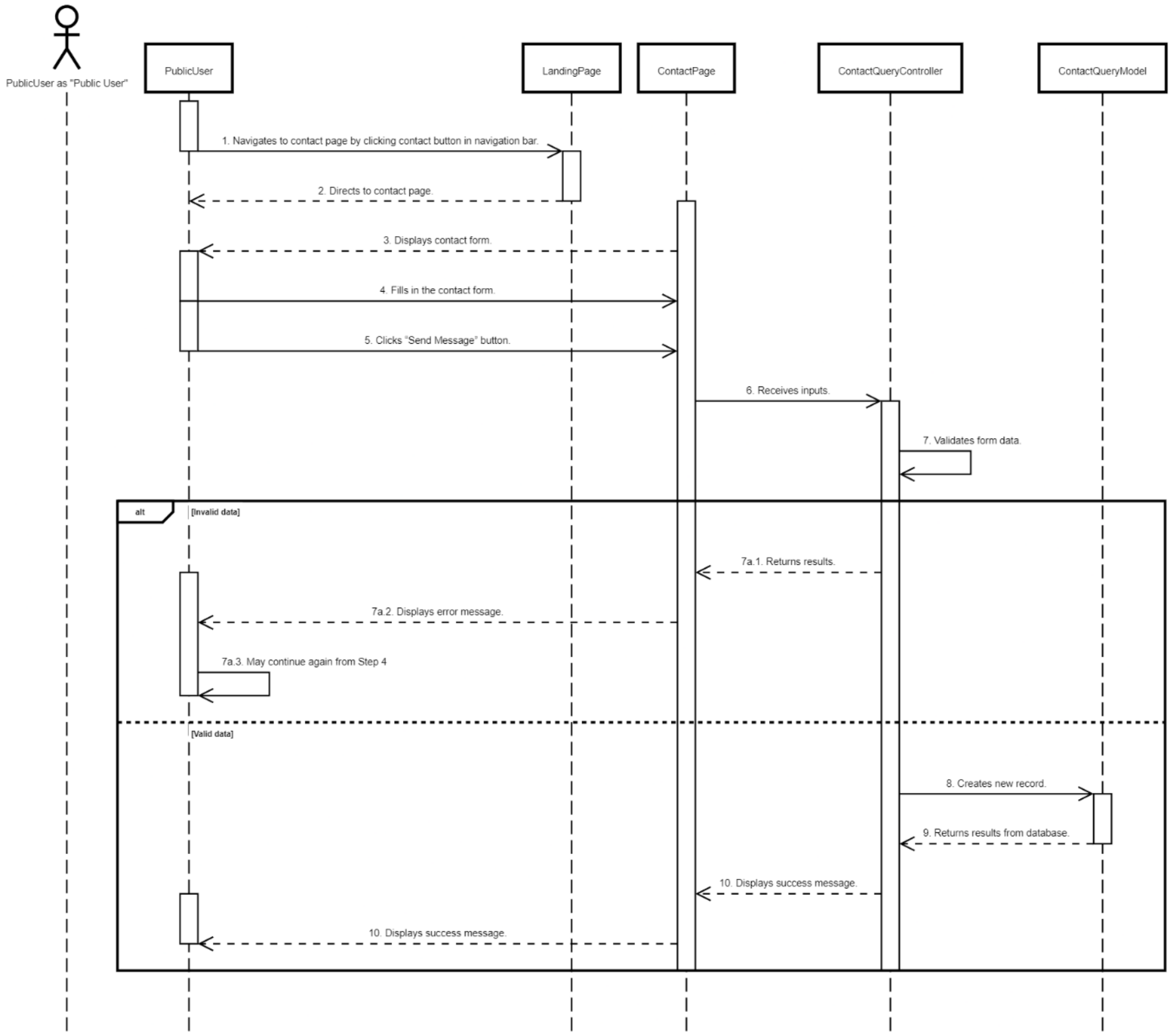


Figure 153: Create Query

4.3 Database Design

4.3.1 Entity Relationship Diagram (ERD)

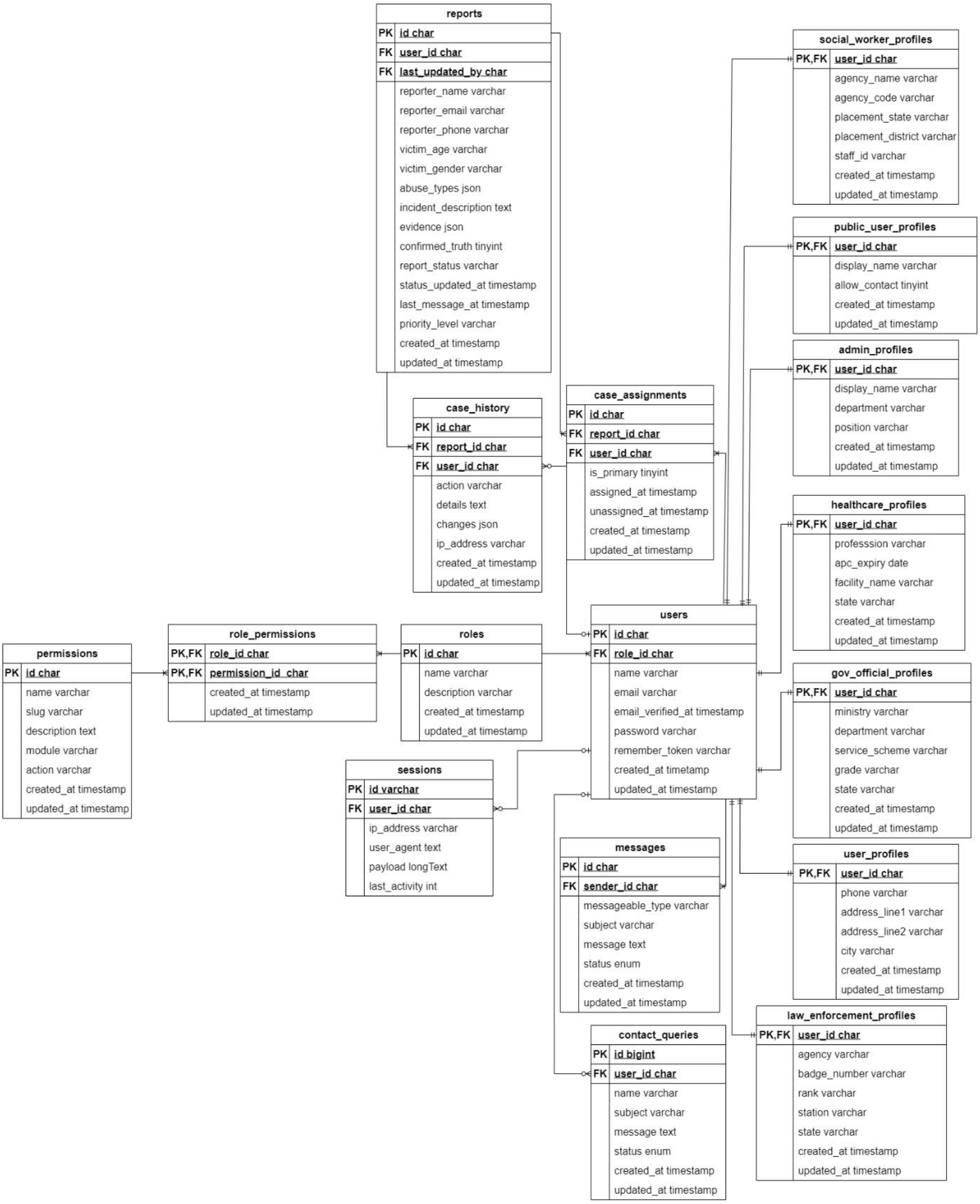


Figure 154: SinDa Entity Relationship Diagram

4.3.2 Database Schema

4.3.2.1 users Table

Name of table: users

Description: Core user authentication and profile table storing user account information and role assignments.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each user.
char	role_id	Foreign Key, Nullable	References roles, set to null on deletion.
varchar	name	Not Null	Full name of the user.
varchar	email	Unique, Not Null	User's email address.
timestamp	email_verified_at	Nullable	Timestamp when email was verified.
varchar	Password	Not Null	Hashed password for authentication.
varchar	remember_token	Nullable	Token for "remember me" functionality.
timestamp	created_at	Not Null	Record creation timestamp.
timestamp	updated_at	Not Null	Record last update timestamp.

Table 65: users table

4.3.2.2 roles Table

Name of table: roles

Description: Defines different user roles within the application, controlling permissions and access levels.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each role.
varchar	name	Foreign Key, Nullable	The name of the role.
varchar	description	Nullable	A brief description of what the role entails.

timestamp	created_at	Not Null	Timestamp when the role was created.
timestamp	updated_at	Not Null	Timestamp when the role was updated.

Table 66: roles table

4.3.2.3 role_permissions Table

Name of table: role_permissions

Description: A pivot table that establishes a many-to-many relationship between roles and permissions, defining which permissions are granted to each role.

Data Type	Field Name	Constraints	Description
char	role_id	Primary Key, Foreign Key	References roles. Part of the composite primary keys.
char	name	Primary Key, Foreign Key	References permissions. Part of the composite primary keys
timestamp	created_at	Not Null	Timestamp when the role was created.
timestamp	updated_at	Not Null	Timestamp when the role was updated.

Table 67: role_permissions table

4.3.2.4 permissions Table

Name of table: permissions

Description: Defines specific permissions that can be granted to roles, controlling granular access to application features and resources.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each permission.
varchar	name	Not Null	The name of permission
varchar	slug	Unique, Not Null	URL-friendly identifier for the permission.

text	description	Nullable	Detailed description of what the permission allows.
varchar	module	Not Null	The module/area this permission belongs to (e.g., 'users', 'cases', 'reports').
varchar	action	Not Null	The action this permission allows (e.g., 'view', 'create', 'edit', 'delete').
timestamp	created_at	Not Null	Timestamp when the role was created.
timestamp	updated_at	Not Null	Timestamp when the role was updated.

Table 68: permissions table

4.3.2.5 sessions Table

Name of table: sessions

Description: Stores information about active user session.

Data Type	Field Name	Constraints	Description
varchar	id	Primary Key	Unique identifier for the session.
char	user_id	Foreign Key, Nullable	The name of permission
varchar	slug	Unique, Not Null	Refers to the users table. Helps to identifies authenticated user associated with the session.
varchar	ip_address	Nullable	The IP address from which the session originated (IPv4/IPv6).
text	user_agent	Nullable	The user agent string of the client (browser, device information).
longtext	payload	Not Null	Encrypted or serialized session data

int	last_activity	Not Null, Indexed	Unix timestamp of the last activity for this session

Table 69: sessions Table

4.3.2.6 reports Table

Name of table: reports table

Description: Stores incident reports submitted by users.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each report.
char	user_id	Foreign Key, Nullable	Refer to users table. Optional link to registered user.
varchar	reporter_name	Nullable	Name of the person reporting the incident.
varchar	reporter_email	Nullable	Email address of the reporter.
varchar	reporter_phone	Nullable	Phone number of the reporter.
varchar	victim_age	Not Null	Age of the victim.
varchar	victim_gender	Not Null, Indexed	Gender of the victim.
longtext	abuse_types	Nullable	Array of abuse types selected.
text	incident_description	Not Null	Detailed description of the incident.
text	incident_location	Not Null	Location where the incident occurred.
date	incident_date	Not Null	Date when the incident occurred.
varchar	suspected_abuser	Nullable	Information about the suspected abuser.
longtext	evidence	Nullable	Array of evidence files uploaded.
tinyint	confirmed_truth	Not Null, Default: false	Whether the truth of the report has been confirmed.

varchar	report_status	Not Null, Default: “Submitted”	Status of the report.
char	last_updated_by	Foreign Key, Nullable	User who last updated the report.
timestamp	status_updated_by	Nullable	When the status was last updated.
timestamp	last_message_at	Nullable	Timestamp of the last message in this case.
varchar	Priority_level	Not Null, Default: ‘Medium’	Priority level of the case.
uimestamp	created_at	Not Null	Timestamp when the report was created.
uimestamp	updated_at	Not Null	Timestamp when the report was last updated.

Table 70: reports Table

4.3.2.7 case_history Table

Name of table: case_history

Description: Records a chronological log of actions and changes related to specific reports, including who performed the action and from where. This table provides an audit trail for case management.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each history entry.
char	report_id	Foreign Key	Refer to reports table. Links the history entry to a specific report.
char	user_id	Foreign Key, Nullable	Refers to users table. Identifies the user who performed the action.

varchar	action	Not Null	Describes the type of action performed. For instance, created, updated, status_changed and assigned.
text	details	Nullable	Detailed description or notes about the action.
longtext	changes	Nullable	Stores a JSON object representing the specific changes made.
varchar	ip_address	Nullable	The IP address from which the action was initiated.
varchar	user_agent	Nullable	The user agent string of the client that performed the action.
timestamp	created_at	Not Null	Timestamp when the history entry was created.
timestamp	updated_at	Not Null	Timestamp when the history entry was last updated.

Table 71: case_history Table

4.3.2.8 case_assignments Table

Name of table: case_assignments

Description: A pivot table that manages the assignment of reports to specific users, indicating who is responsible for a particular case. It also tracks the primary assignee and assignment timestamps.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each case assignment record.
char	report_id	Foreign Key	Refer to reports table. Links the assignment to a specific report.
char	user_id	Foreign Key	References users table. Links the assignment to a specific user.

tinyint	Is_primary	Not Null, Default: false	Indicates if this user is the primary assignee for the report.
timestamp	assigned_at	Not Null, Default: current_timestamp	Timestamp when the report was unassigned from the user.
timestamp	created_at	Not Null	Timestamp when the assignment record was created.
timestamp	updated_at	Not Null	Timestamp when the assignment record was last updated.

Table 72: case_assignments Table

4.3.2.9 messages Table

Name of table: messages

Description: A pivot table that manages the assignment of reports to specific users, indicating who is responsible for a particular case. It also tracks the primary assignee and assignment timestamps.

Data Type	Field Name	Constraints	Description
char	id	Primary Key	Unique identifier for each message.
varchar	messageable_type	Not Null	The type of related model for polymorphic relationship.
char	messageable_id	Not Null	The ID of the related model instance for polymorphic relationship.
char	sender_id	Foreign Key	Refer to users table. Helps to identifies the user who sent the message.
text	body	Not Null	The content of the message.
longtext	attachments	Nullable	Array of attachment files.

timestamp	created_at	Not Null	Timestamp when the assignment record was created.
timestamp	updated_at	Not Null	Timestamp when the assignment record was last updated.

Table 73: messages table

4.3.2.10 contact_queries Table

Name of table: messages

Description: Stores contact form submissions and inquiries from users, including both guest submissions and registered user queries.

Data Type	Field Name	Constraints	Description
char	id	Primary Key, Auto Increment	Unique identifier for each contact query.
varchar	name	Not Null	Name of the person submitting the query.
varchar	email	Not Null	Email address of the person submitting the query.
varchar	subject	Not Null	Subject line of the contact query.
text	message	Not Null	The content/message of the contact query.
longtext	attachments	Not Null	Array of attachment files.
char	user_id	Foreign Key, Nullable	Refer to users table. Optional link to registered user.
enum	updated_at	Not Null, Default: “Pending”	Status of the query ('pending', 'in_progress', 'resolved').
timestamp	created_at	Not Null	Timestamp when the query was submitted.
timestamp	Updated_at	Not Null	Timestamp when the query was last updated.

Table 74: contact_queries Table

4.3.2.11 user_profiles Table

Name of table: messages

Description: Stores contact form submissions and inquiries from users, including both guest submissions and registered user queries.

Data Type	Field Name	Constraints	Description
char	user_id	Primary Key, Foreign Key	Refer to users table. One to one relationship with users table.
varchar	phone	Nullable	Phone number of the user.
varchar	address_line1	Nullable	First line of the user's address.
varchar	address_line2	Nullable	Second line of the user's address.
varchar	city	Nullable	City where the user is located.
varchar	postcode	Nullable, Indexed	Postal/ZIP code of the user's location.
varchar	state	Nullable, Indexed	State/Province where the user is located.
varchar	avatar_path	Nullable	File path of the user's profile avatar image.
timestamp	created_at	Not Null	Timestamp when the profile was created.
timestamp	updated_at	Not Null	Timestamp when the profile was last updated.

Table 75: user_profiles Table

4.3.2.12 admin_profiles Table

Name of table: messages

Description: Stores contact form submissions and inquiries from users, including both guest submissions and registered user queries.

Data Type	Field Name	Constraints	Description

char	user_id	Primary Key, Foreign Key	Refer to users table. One to one relationship with users table.
varchar	display_name	Not Null	Display name for the admin user.
varchar	department	Nullable	Department where the admin works.
varchar	position	Nullable	Position of the Admin.
timestamp	created_at	Not Null	Timestamp when the admin profile was created.
timestamp	updated_at	Not Null	Timestamp when the admin profile was last updated.

Table 76: admin_profiles Table

4.3.2.13 social_worker_profiles Table

Name of table: social_worker_profiles

Description: Stores additional profile information specifically for users with social worker roles, including agency details, placement information, and staff identification.

Data Type	Field Name	Constraints	Description
char	user_id	Primary Key, Foreign Key	Refer to user table. One to one relationship with users table.
varchar	agency_name	Nullable	Name of the social work agency.
varchar	agency_code	Nullable	Internal reference code for the agency.
varchar	placement_state	Nullable, Indexed	State where the social worker is placed
varchar	placement_district	Nullable	District where the social worker is placed.
varchar	staff_id	Nullable	Staff identification number for social worker.
timestamp	created_at	Not Null	Timestamp when the social worker profile was submitted.

timestamp	updated_at	Not Null	Timestamp when the social worker profile was last updated.

Table 77: social worker profile Table

4.3.2.14 healthcare_profiles Table

Name of table: social_worker_profiles

Description: Stores additional profile information specifically for users with social worker roles, including agency details, placement information, and staff identification.

Data Type	Field Name	Constraints	Description
char	user_id	Primary Key, Foreign Key	Refer to user table. One to one relationship with users table.
varchar	Profession	Not Null, Default 'doctor'	Healthcare profession (e.g., 'doctor', 'nurse', 'therapist').
date	apc_expiry	Nullable	Annual Practising Certificate expiry date.
varchar	facility_name	Nullable	Name of the healthcare facility.
varchar	state	Nullable, Indexed	State where the healthcare facility is located.
timestamp	created_at	Not Null	Timestamp when the healthcare profile was created.
timestamp	updated_at	Not Null	Timestamp when the healthcare profile was last updated.

Table 78: healthcare_profiles Table

4.3.2.15 law_enforcement_profiles Table

Name of table: law_enforcement_profiles

Description: Stores additional profile information specifically for users with law enforcement roles.

Data Type	Field Name	Constraints	Description

char	user_id	Primary Key, Foreign Key	Refer to user table. One to one relationship with users table.
varchar	agency	Nullable, Indexed	Law Enforcement agency.
varchar	badge_number	Nullable, Indexed	Badge/identification number of the officer.
varchar	rank	Nullable	Officer rank (e.g., 'Insp.', 'Sgt.').
varchar	station	Nullable	Police station or enforcement station (e.g., 'IPD Petaling Jaya').
varchar	state	Nullable, Indexed	State where the officer is stationed
timestamp	created_at	Not Null	Timestamp when the law enforcement profile was created.
timestamp	updated_at	Not Null	Timestamp when the law enforcement profile was last updated.

Table 79: law_enforcement_profiles Table

4.3.2.16 gov_official profiles Table

Name of table: gov_official profiles

Description: Stores profile details for the government child welfare officers.

Data Type	Field Name	Constraints	Description
char	user_id	Primary Key, Foreign Key	Refer to user table. One to one relationship with users table.
varchar	ministry	Nullable	Ministry name (e.g., KPWKM, KPM).
varchar	department	Nullable	Department name (e.g., JKM, JPNIN).
varchar	service_scheme	Nullable	Government service scheme (e.g., M, N, FA).
varchar	grade	Nullable	Grade (e.g., M41, N29).
varchar	state	Nullable, Indexed	State where the officers is stationed.

timestamp	created_at	Not Null	Timestamp when the profile was created.
timestamp	updated_at	Not Null	Timestamp when the profile was last updated.

Table 80: gov_official profiles Table

4.3.2.17 public_user_profiles Table

Name of table: public_user_profiles

Description: Stores profile details for public users, including optional display name and contact consent.

Data Type	Field Name	Constraints	Description
char	user_id	Primary Key, Foreign Key	Refer to user table. One to one relationship with users table.
varchar	display_name	Nullable	Optional public display name.
tinyint	allow_contact	Not Null, Default: false	Whether the user consents to being contacted.
timestamp	created_at	Not Null	Timestamp when the profile was created.
timestamp	updated_at	Not Null	Timestamp when the profile was last updated.

Table 81: public_user_profiles Table

4.4 Interface Design (Wireframes)

4.4.1 Landing Page

Figure 155: Landing Page

Above is the landing page wireframe for the system. Starting with the navigation links located on the upper part of the webpage, it allows quick access to About, How It Works, and Login pages. In the main field, the site mission is presented with two buttons, the first is to allow submission of the anonymous reports, and the second is for the registered users login and track their cases. In the area just below this, the page describes characteristics like confidentiality, real time case tracking, and interagency collaboration, to which are added short explanations on how the system functions, its role in supporting agencies, and the possible effects of reports.

This page also has a section showing the alignment with the United Nations Sustainable Development Goal16, which underlines the system's contribution to the elimination of abuse and the strengthening of accountability. The footer, which is also included, provides the contact options and copyright details, making sure that the user knows precisely how to contact the support for help.

4.4.2 Sign In Page

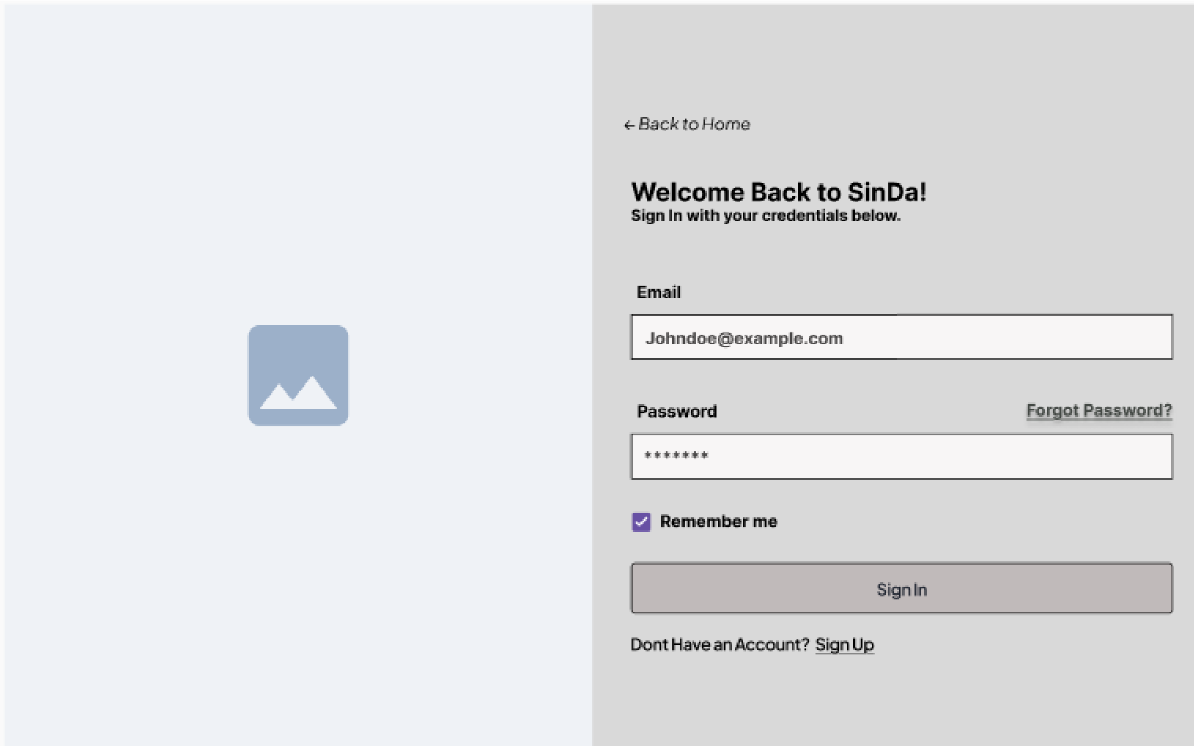


Figure 156: Sign in Page

This wireframe represents the Sign In page of the proposed system, where users log in with their registered credentials to access personalized features such as tracking reports. It includes fields for email and password, a “Forgot Password?” link for account recovery, and a “Remember me” checkbox that allows users to stay logged in on their device for convenience. For first-time users, a “Sign Up” option is available, while a link to return to the main page ensures easy navigation. The main action button, “Sign In,” validates the entered details and grants access to the system. These design choices improve usability and accessibility by supporting password recovery, reducing repeated logins, and separating account creation from the login process to provide a clear and user-friendly experience.

4.4.3 Reset Your Password Modal

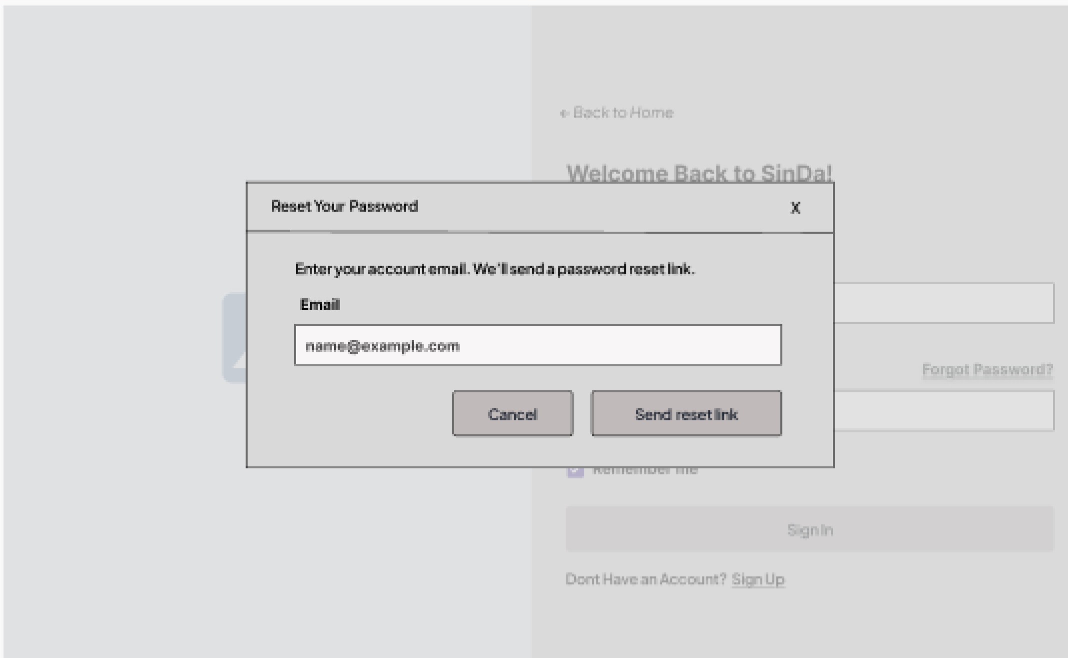


Figure 157: Reset Your Password modal

This wireframe shows the Reset Your Password modal, which enables users to recover account access by entering their registered email address. It provides a single input field for the email and two action buttons: "Send reset link" to trigger a password reset email and "Cancel" to close the modal without making changes. The design ensures a simple and focused recovery process, minimizing user confusion by only requesting essential information, while the two-button choice offers both convenience and control.

4.4.4 Reset Your Password Page

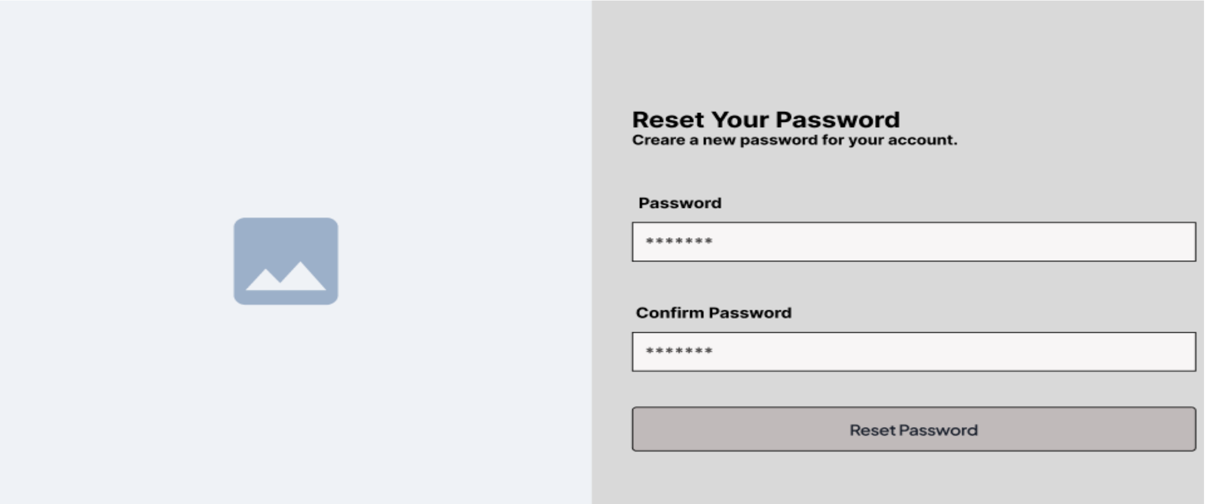


Figure 158: Reset Your Password Page

This wireframe illustrates the Reset Password page, where users can create a new password after requesting a reset link. It includes two fields: one for entering the new password and another to confirm it, ensuring accuracy. A single action button, Reset Password, finalizes the process and updates the user’s credentials.

4.4.5 Create Account Page

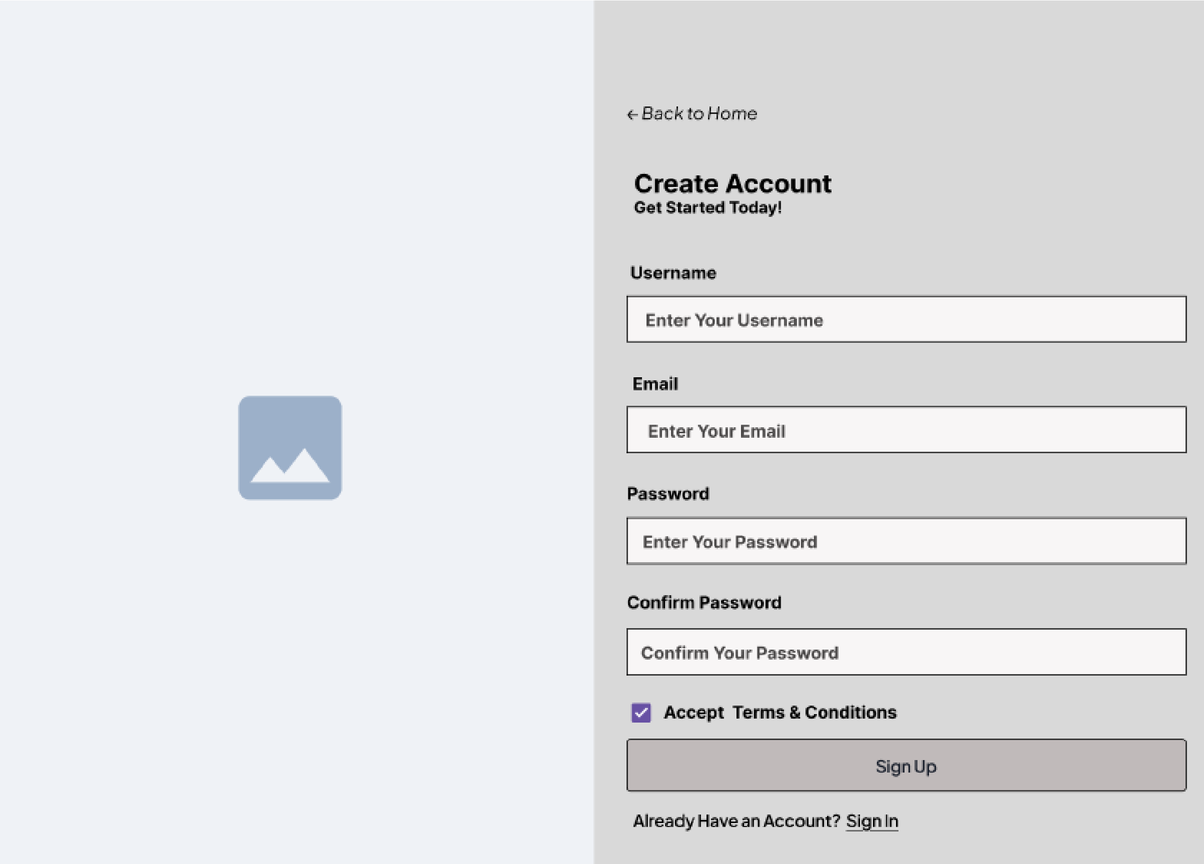


Figure 159: Sign Up Page

This wireframe shows the Create Account page, where new public user can register for access to the SinDa system. The form includes fields for entering a username, email, password, and confirming the password to ensure accuracy. Besides that, the user must check a box paired with a note that states their acceptance of the terms and conditions. At the bottom of the page, there is a link to the Sign In page for active users together with a Sign Up button to officially complete the registration process.

4.4.6 Contact Us Page

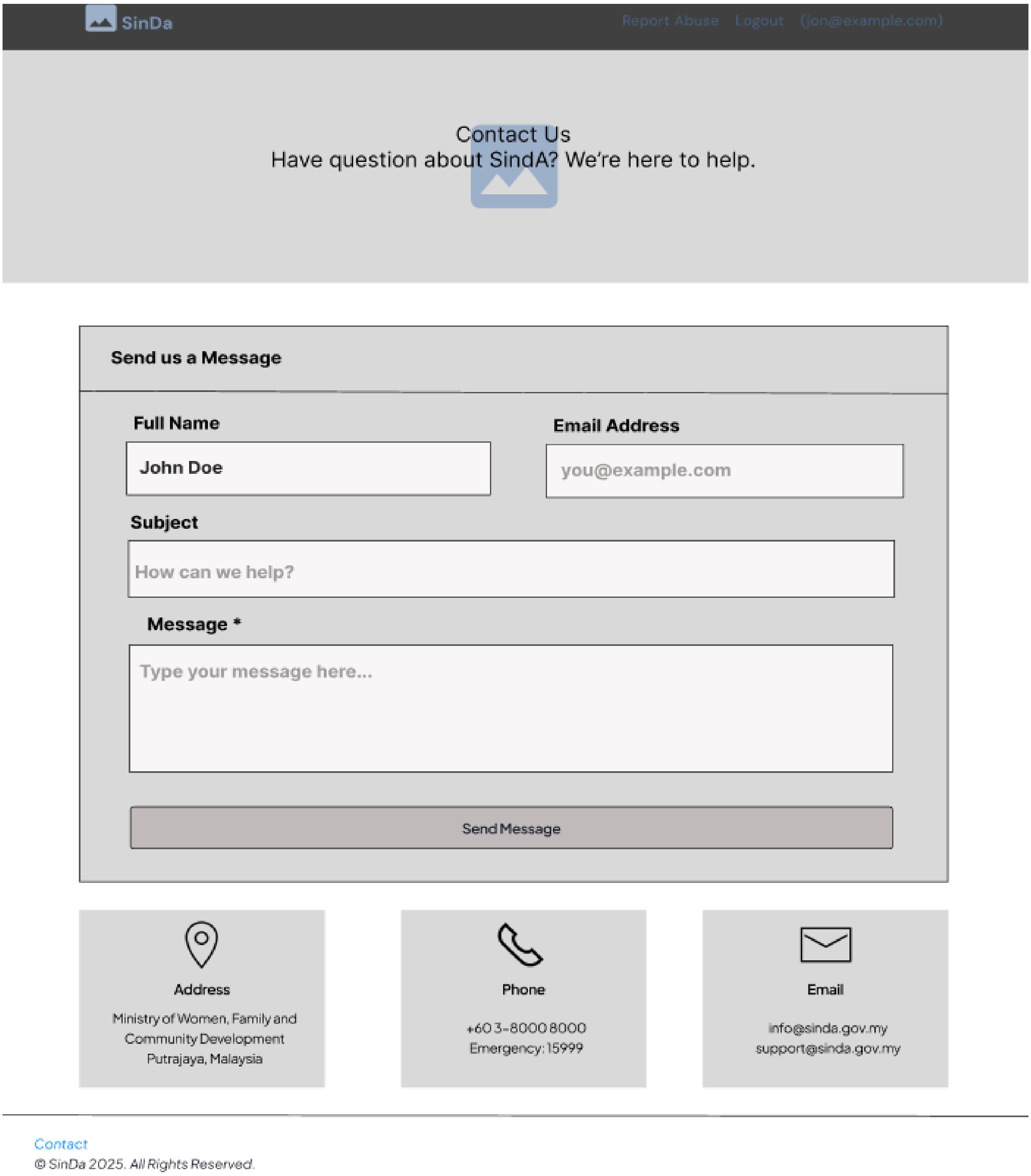


Figure 160: Contact Us Page

The Contact Us page, where users may ask questions of the system administrator directly, is represented by this wireframe. Key information such as complete name, email address, subject, and message content is gathered by the form, which also includes a text field for in-depth questions. After finishing, users may click the submit Message button to submit their message. Other contact details, including the ministry's location, phone numbers, and email addresses, are included beneath the form, providing consumers with extra avenues for contacting the ministry for assistance.

4.4.7 Report Page

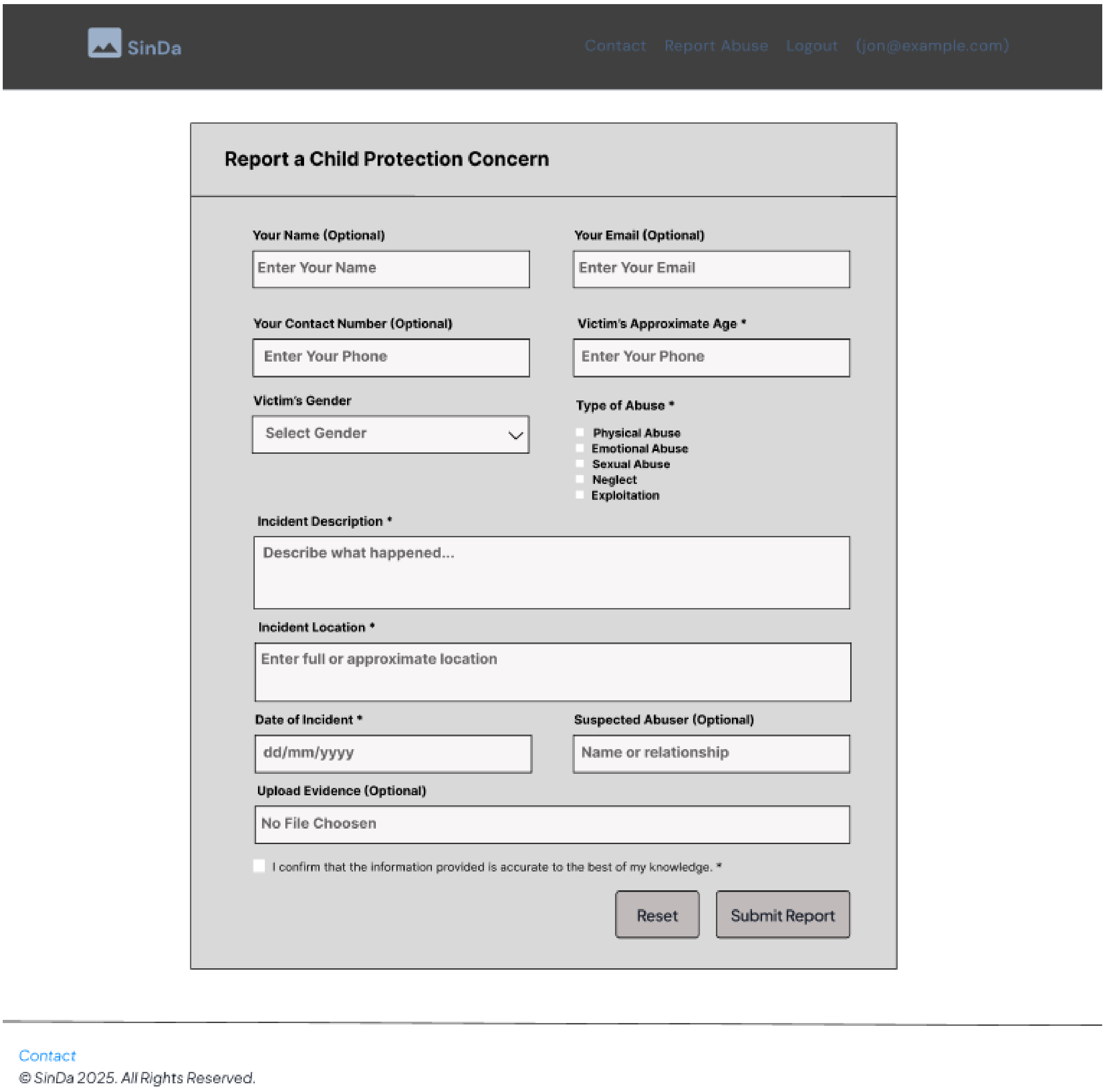


Figure 161: Report Page

This wireframe shows the Report a Child Protection Concern page, which enables public users to securely submit detailed abuse reports. The form balances required and optional fields by collecting essential information such as reporter details, victim information, abuse type, incident description, location, and date, while also allowing optional inputs like suspected abuser details and evidence uploads. At the bottom, users must confirm the accuracy of their submission before proceeding, with options to either reset the form or submit it for processing. This design ensures that critical data is always captured for investigation while offering flexibility for additional details, thereby improving report accuracy and usability.

4.4.8 My Submitted Report Page

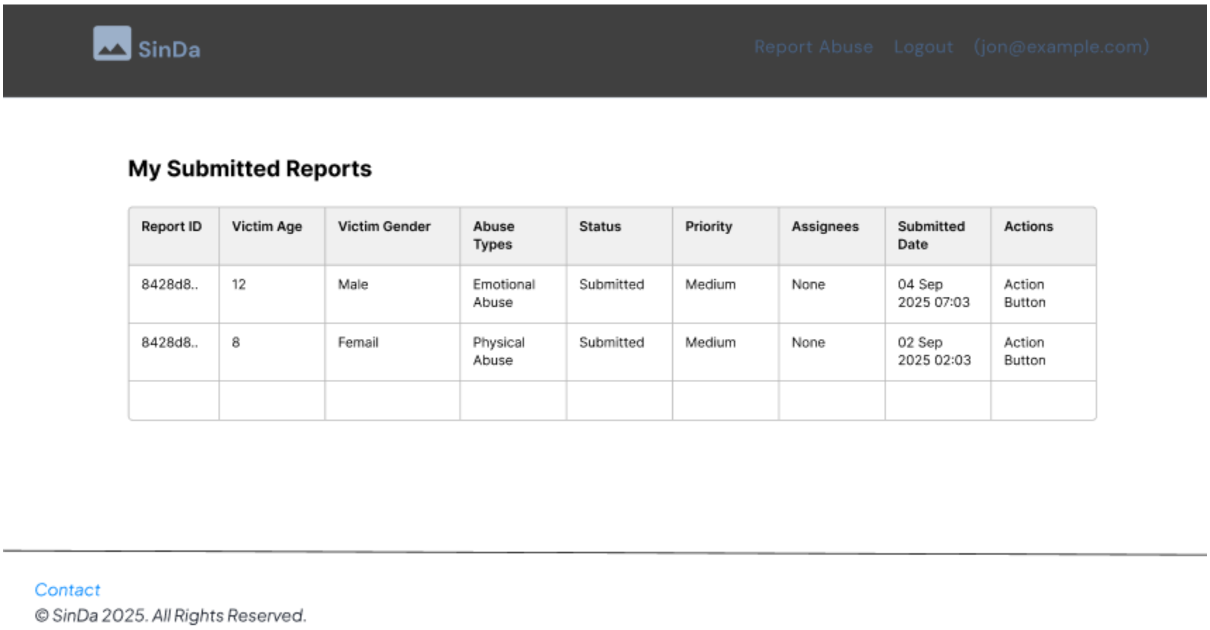


Figure 162: My Submitted Report Page

This wireframe shows the My Submitted Reports page, where public users can view and track the cases they have reported. The table lists key details for each report, including the report ID, victim's age and gender, type of abuse, status, priority, assigned personnel, submission date, and available actions. This design allows users to monitor their reports in one place and check for updates or follow-up actions.

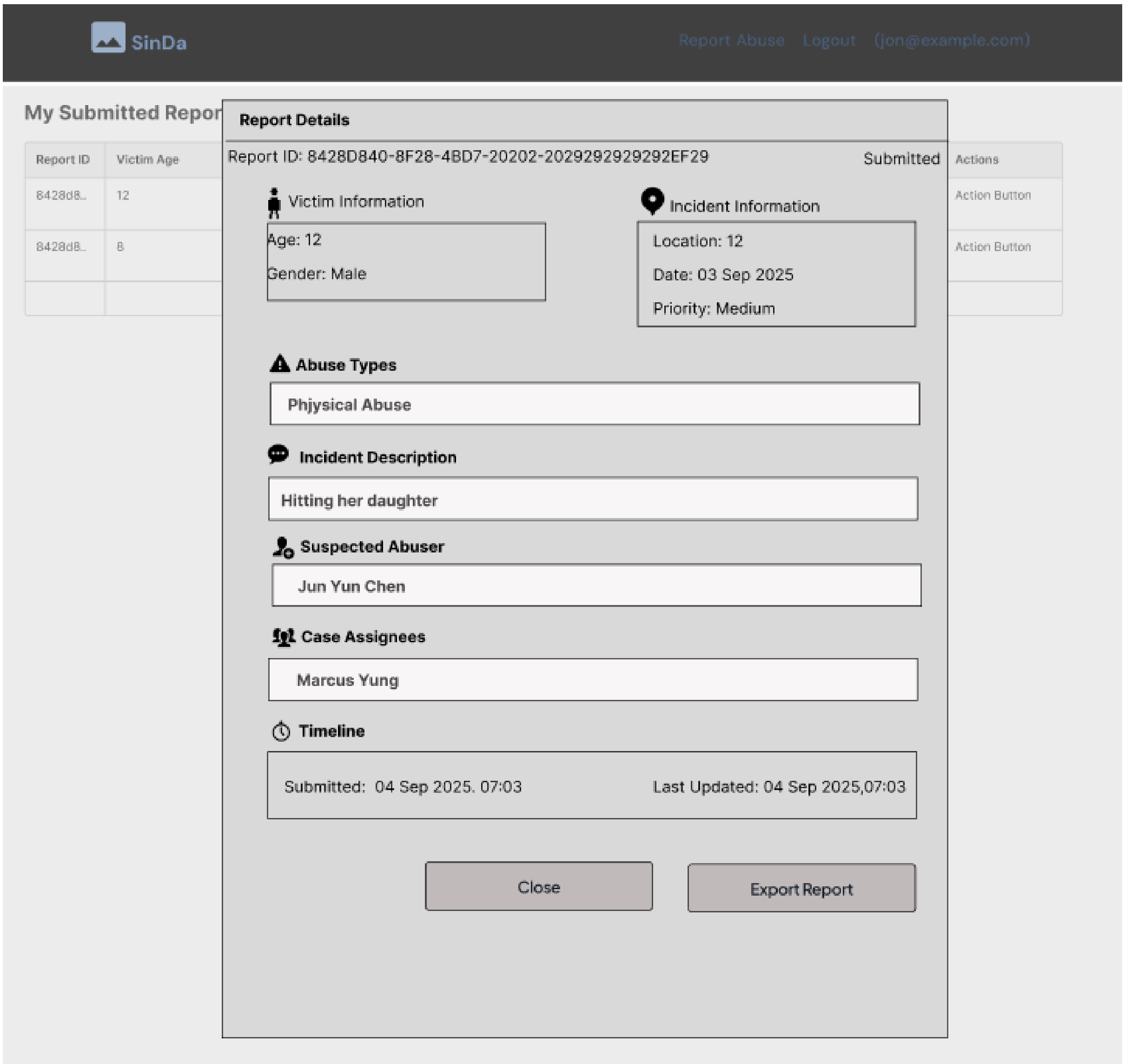


Figure 163: My Submitted Report Page

This wireframe shows the Report Details modal, which appears when a public users clicks the action button from the "My Submitted Reports" page. It provides a detailed view of the report, including victim and incident information, type of abuse, incident description, suspected abuser, assigned case worker, and a timeline showing submission and update history. At the bottom, users can either close the modal or export the report for record-keeping or official use.

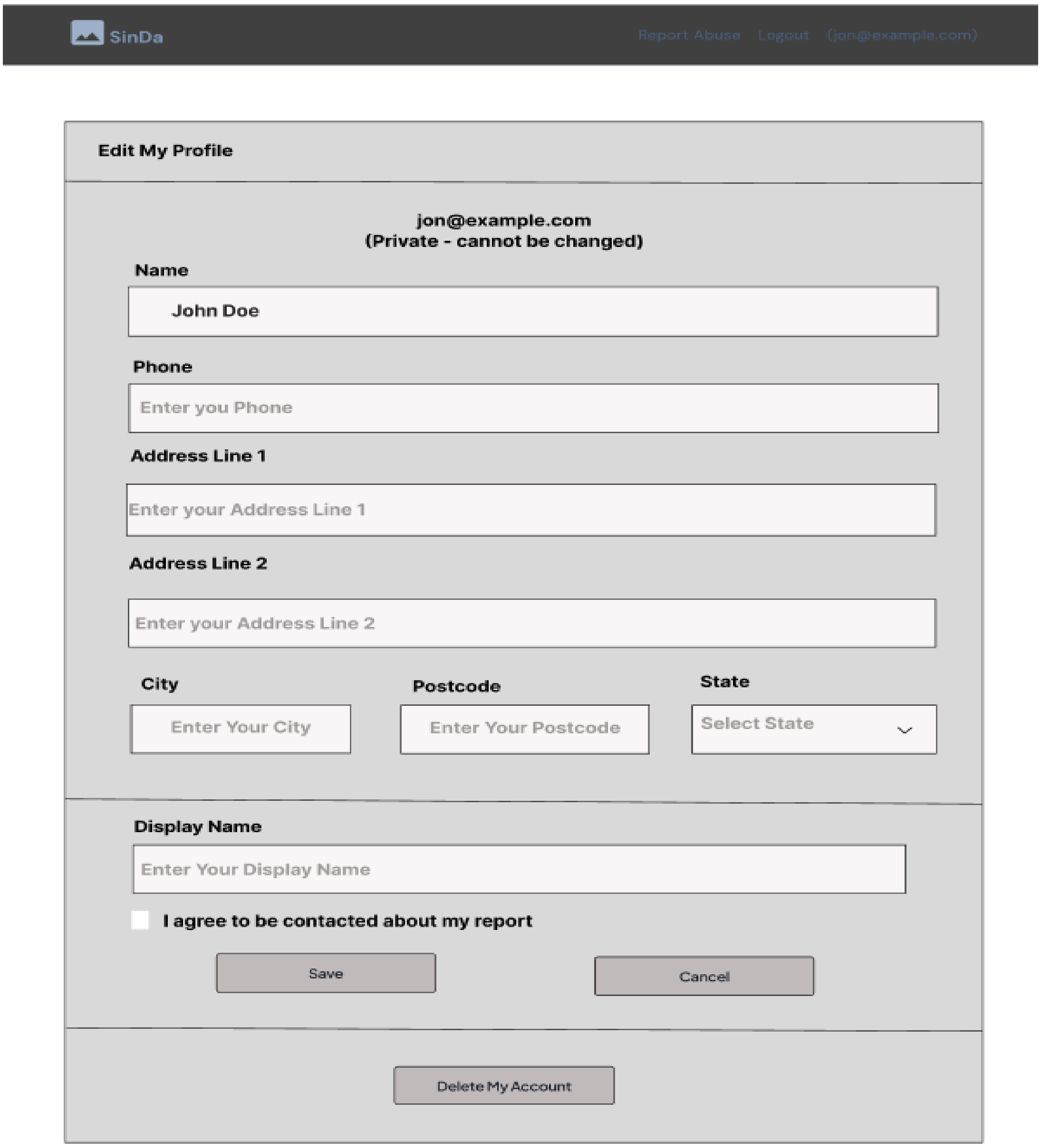


Figure 164: Edit My Profile Page

This wireframe shows the Edit My Profile page, where public users can manage their personal details and preferences. The form includes fields for name, phone number, address, city, postcode, state, and display name, while the email address remains fixed and uneditable for privacy reasons. Users can also choose whether they agree to be contacted about their reports. At the bottom, there are options to save changes, cancel edits, or permanently delete the account.

4.4.9 Dashboard Page (System Administrator)

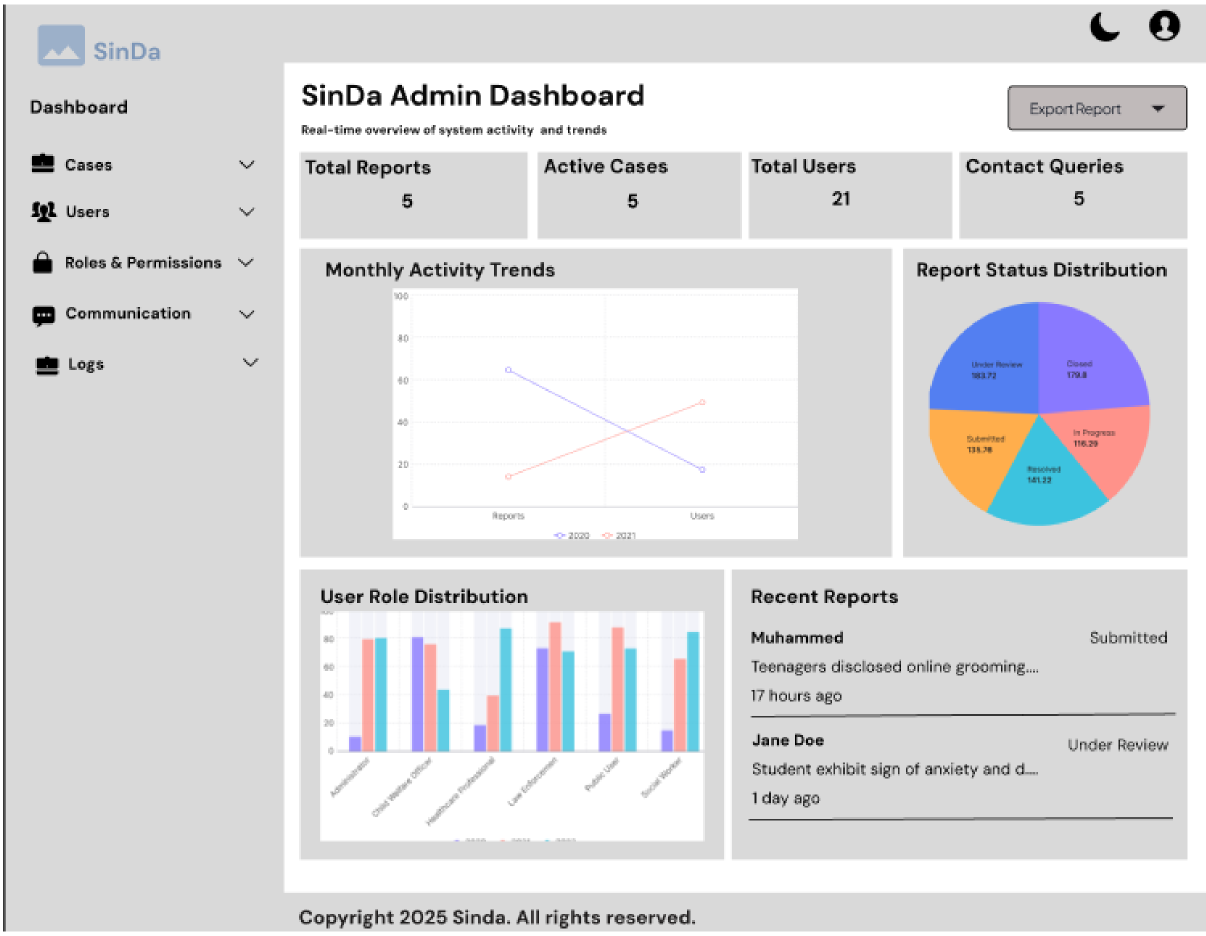


Figure 165: Dashboard (System Administrators)

The SinDa system's Admin Dashboard, which is shown in this wireframe, is a dashboard for most authorized users that displays live data on reports, cases, users, and contact queries. The summary cards on the top part display the total of key metrics, and the visual charts present the monthly trend of activities, the distribution of report status, and the distribution of user roles. The right panel features the latest reports for fast monitoring. The left sidebar menu is for the authorized users to manage cases, users, roles and permissions, communication, and logs while the Export Report button allows the user to download the data for further use. The structure of the layout is incorporated in such a way that the authorized users can promptly evaluate the system's performance and manage cases effectively.

4.4.10 Dashboard Page (Authorized Users)

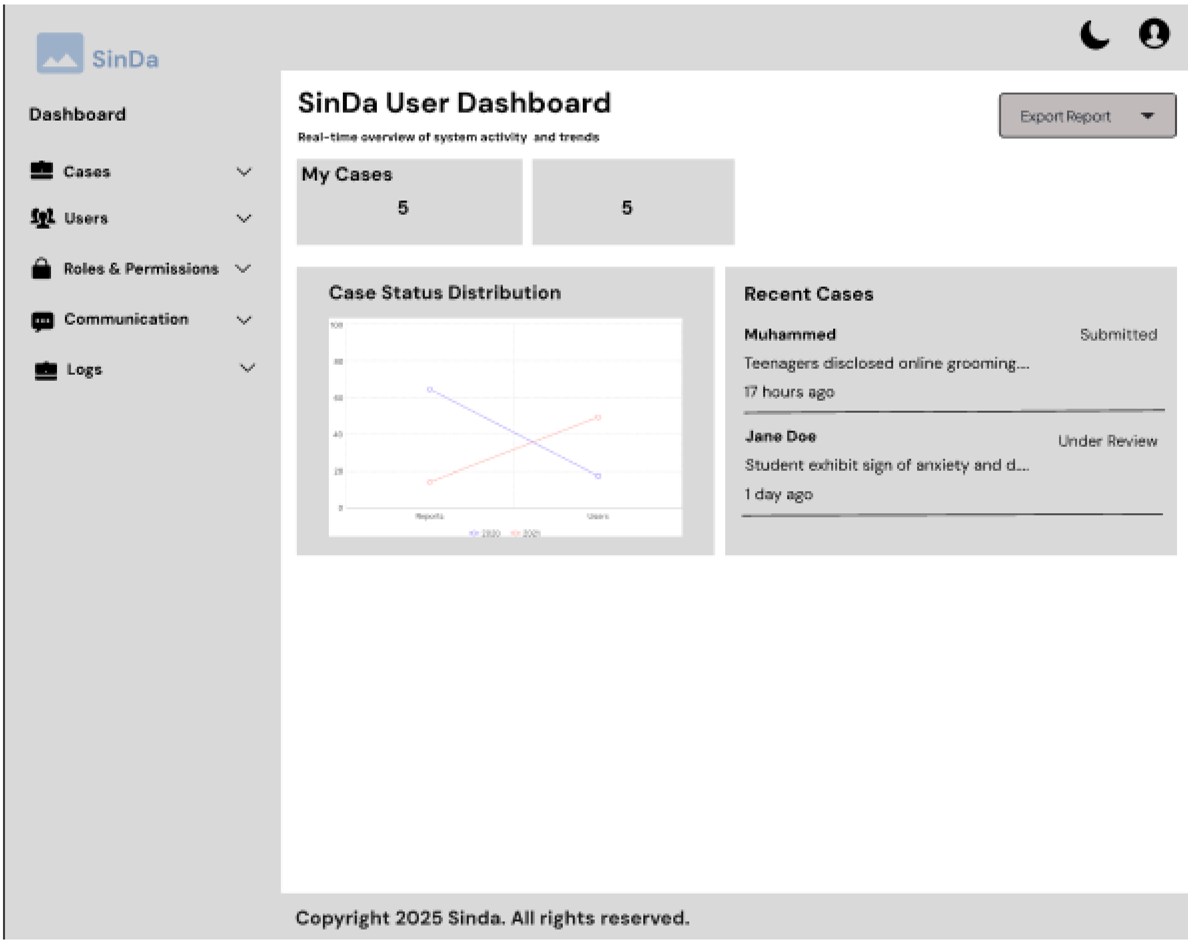


Figure 166: Dashboard (Authorized Users)

This wireframe shows the User Dashboard for authorized users such as social workers, law enforcement, healthcare professionals, and other stakeholders besides the system administrator. It provides an overview of their assigned cases, with key metrics displayed at the top under My Cases. The dashboard also includes a Case Status Distribution chart to visualize case progress and a Recent Cases section for quick access to the latest reports. Like the admin view, the sidebar allows navigation to cases, users, roles and permissions, communication, and logs, while the Export Report button enables data extraction when needed and was given permission too. This design ensures authorized users can efficiently monitor and manage the cases they are responsible for.

4.4.11 Case Management

4.4.11.1 Case Management Page

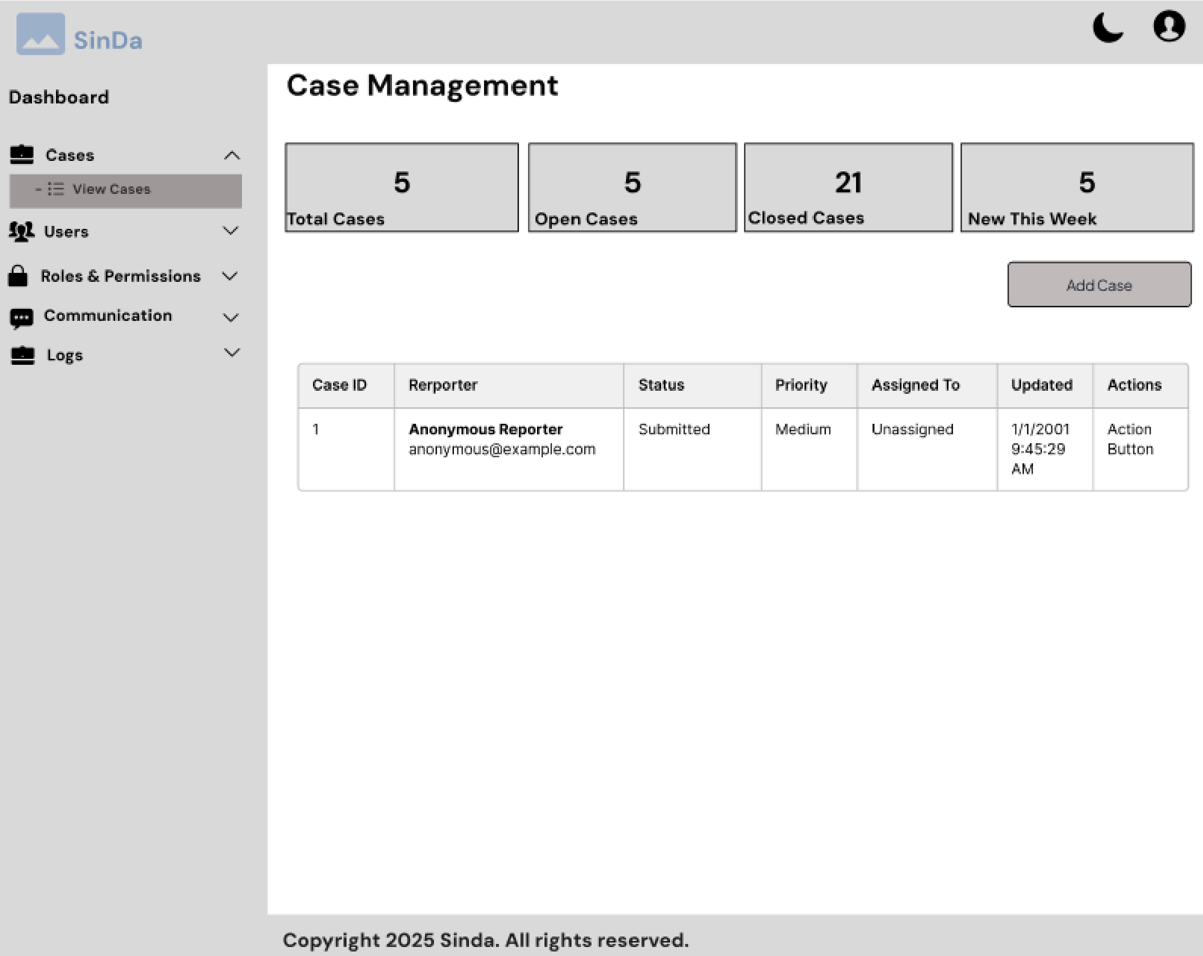


Figure 167: Case Management Page

This wireframe shows the Case Management page, where authorized users can view and manage reported cases. At the top, summary cards display totals for key metrics such as Total Cases, Open Cases, Closed Cases, and New This Week. Below, a table lists individual cases with details like case ID, reporter, status, priority, assigned personnel, last update, and available actions. There is also an Add Case button to allow new cases to be manually entered. This layout ensures users can track case progress efficiently and take appropriate follow-up actions.

4.4.11.2 Add New Case Modal

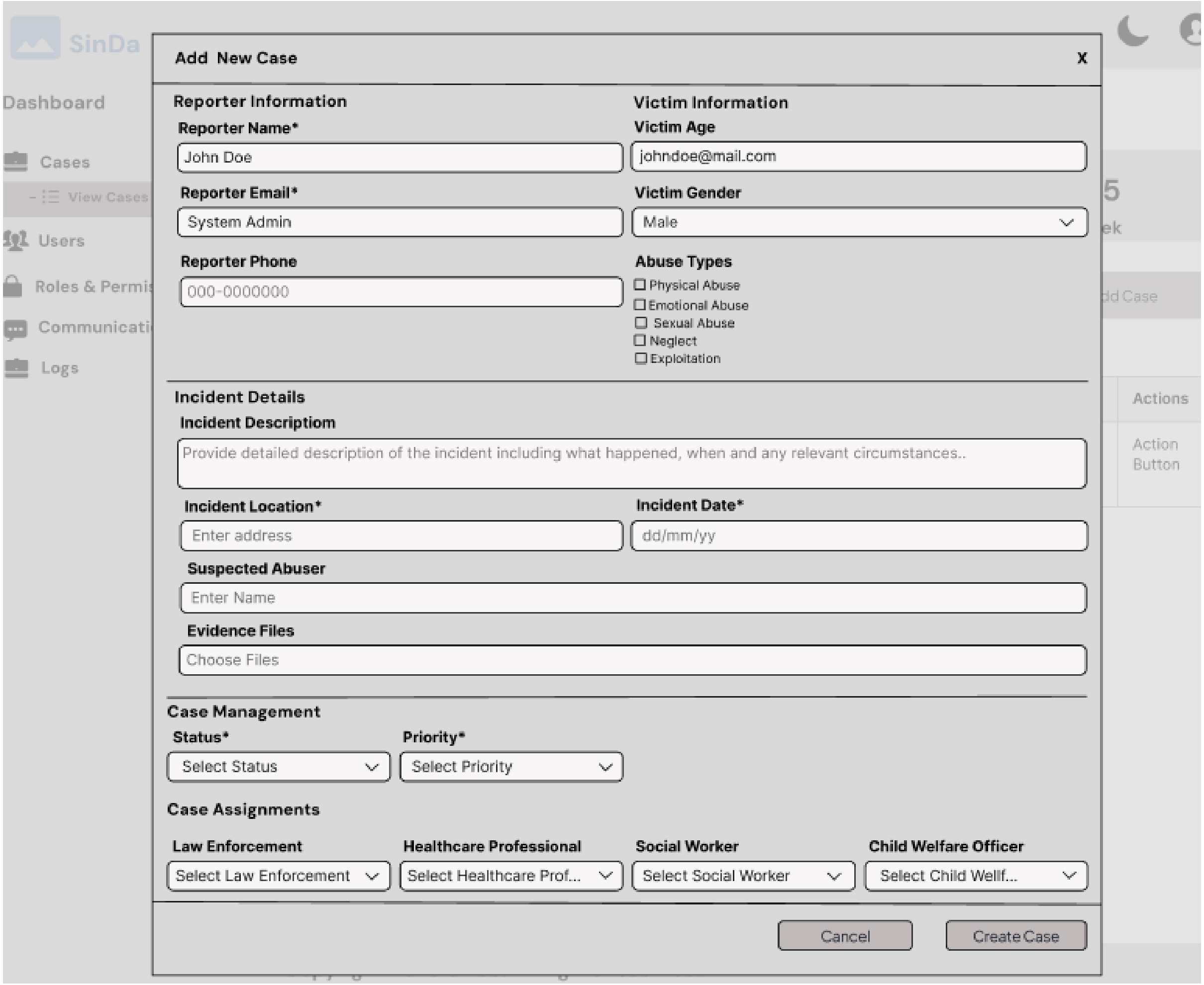


Figure 168: Add New Case Modal

This wireframe shows the Add New Case modal, which allows authorized users to create and register a new child protection case. The form gathers key information including reporter details (name, email, phone), victim details (age, gender, abuse type), and mandatory fields such as incident description, location, and date, while also providing optional inputs for suspected abuser and evidence uploads. Case management controls let users set the case status and priority, and assignments can be directed to relevant personnel such as law enforcement officers, healthcare professionals, social workers, or child welfare officers. At the bottom, options to cancel or create the case are provided. The design ensures that all critical details are captured for proper investigation while maintaining flexibility through optional fields, and structured role-based assignment helps streamline interagency collaboration.

4.4.11.3 Edit Case Modal

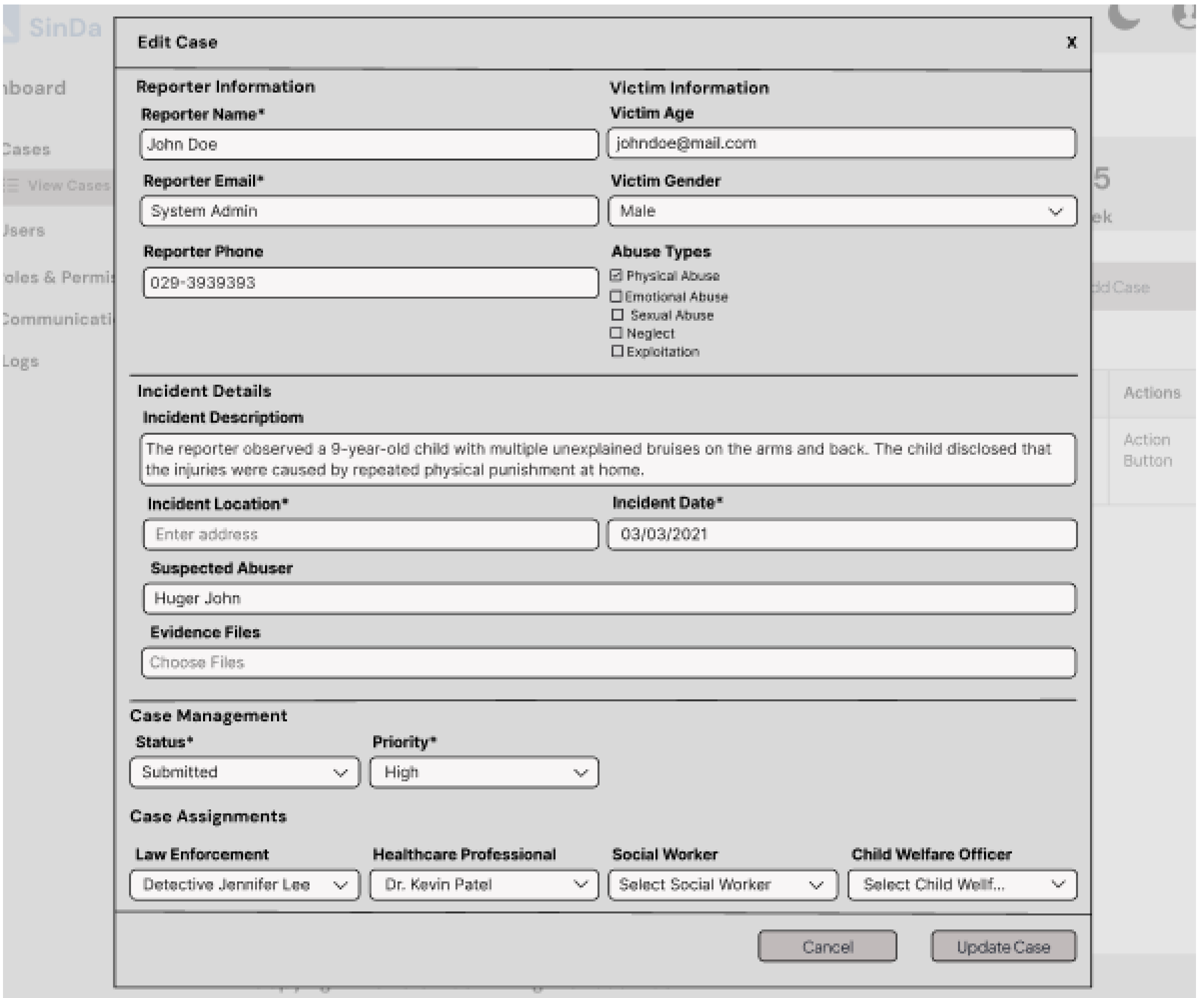


Figure 169: Edit Case Modal

This wireframe shows the Edit Case modal, which allows authorized users to update the details of an existing child protection case. The form presents reporter and victim information, selected abuse types, and a detailed incident description, along with fields for location, date, suspected abuser, and supporting evidence files. Case management controls enable users to adjust the case status and priority, while assignments can be modified to reallocate responsibilities to law enforcement officers, healthcare professionals, social workers, or child welfare officers. At the bottom, users can either cancel or confirm updates through the Update Case button. The design ensures flexibility in managing ongoing cases by keeping essential details visible for quick reference while supporting updates to reflect new information, changes in priority, or reassignment of roles for effective interagency coordination.

4.4.11.4 Delete Case Modal

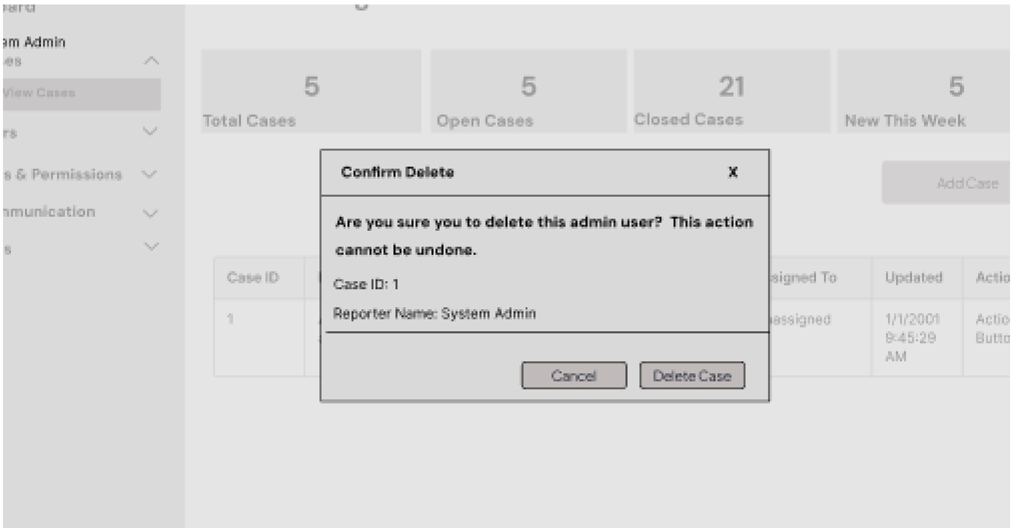


Figure 170: Delete Case Modal

The Confirm Delete popup, which is intended to stop unintentional case deletions, is displayed in this wireframe. The warning message makes sure users understand the gravity of the decision by emphasizing that the action cannot be reversed and is stated clearly. To assist users ensure they are deleting the proper record and lower the possibility of errors, important facts like Case ID and Reporter Name are shown. Cancel and Delete Case are the two separate buttons used by the modal. The delete button is emphasized to demand intentional action, whereas the cancel button is less noticeable to promote thoughtful evaluation. By including a final confirmation step, this design decision strikes a balance between user control and security, enhancing usability while lowering the possibility of data loss.

4.4.11.5 Case History

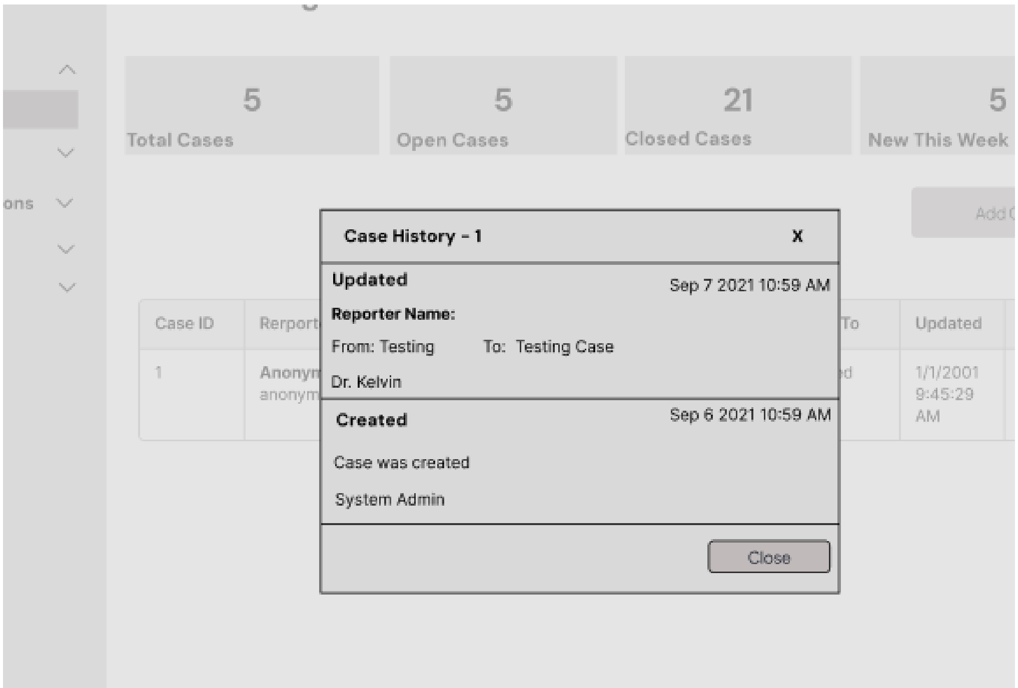


Figure 171: Case History Modal

When a user chooses View Case History from the action menu, the Case History modal is displayed like this wireframe. Who made the modifications, what was changed, and the associated timestamps are all included in the modal's historical record of significant alterations. For accountability, important elements are prominently presented, including the reporter's name and particular update information (such as modifications to case titles or assignments). It is better to read because users can readily discern between the original case creation and subsequent revisions when the "Created" and "Updated" parts are separated. A straightforward departure without affecting data is offered via the Close button at the bottom. By providing all users with clear and easily available information about a case's status, this design guarantees openness, facilitates audit monitoring, and improves interagency collaboration.

4.4.11.6 Case Details Page

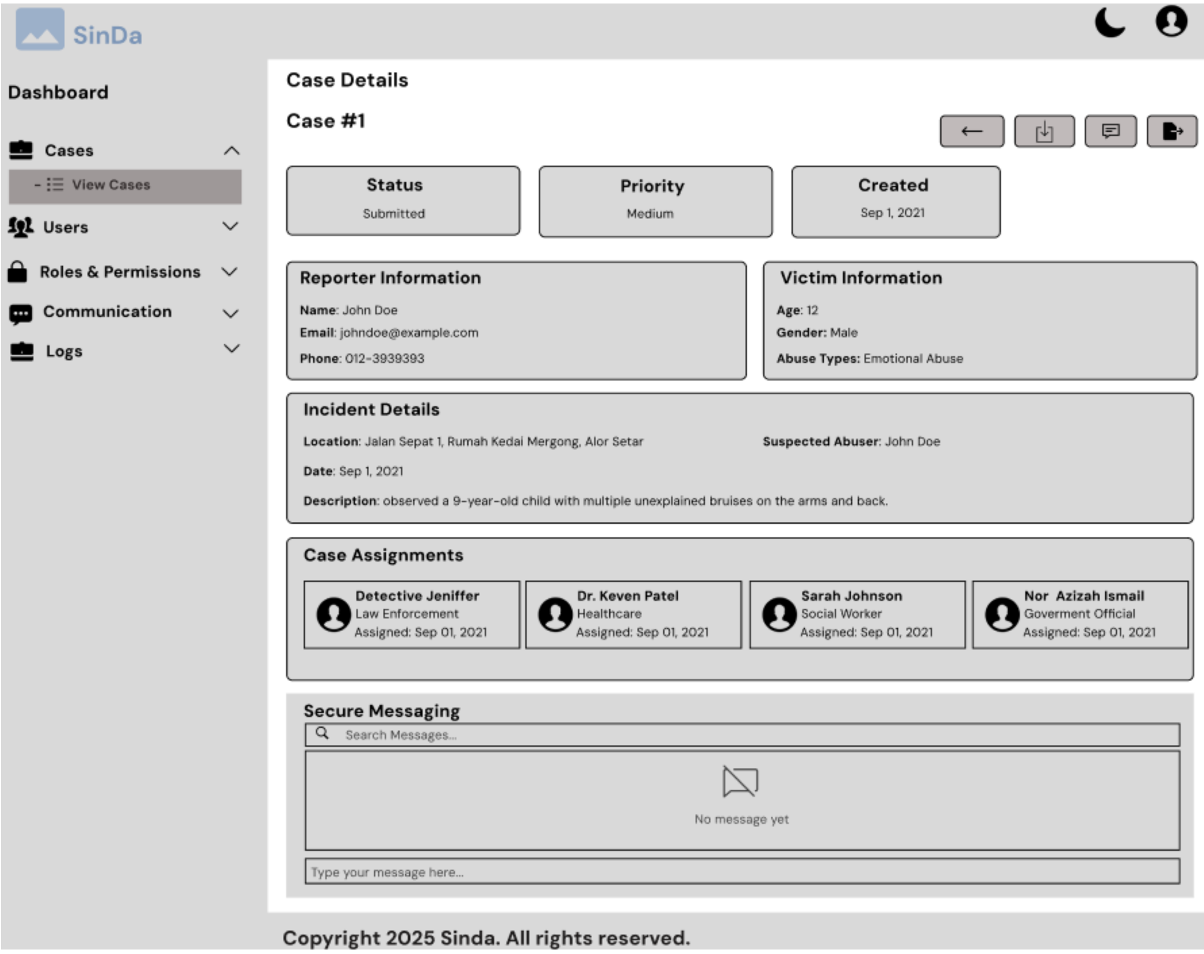


Figure 172: Case Details Page

When a user chooses View Case Details from the action menu, the Case Details page is shown in this wireframe. Case status, priority, and creation date are shown at the top for instant context, followed by reporter and victim details to capture crucial background. This design divides information into easily accessible sections for easy access. The event details section makes sure investigators have all the important information in one place by emphasizing the location, date, alleged abuser, and a detailed narrative. Case assignments are provided using role-based cards (law enforcement, healthcare, social worker, government official), making duties obvious and easy to manage. A secure messaging window at the bottom ensures cooperation without ever leaving the page by enabling interagency discussion within the case. This architecture promotes clarity, controlled information flow, and ease of coordination, lowering cognitive burden while facilitating quick decision-making.

4.4.12 Admin Management

4.4.12.1 Admin Management Page

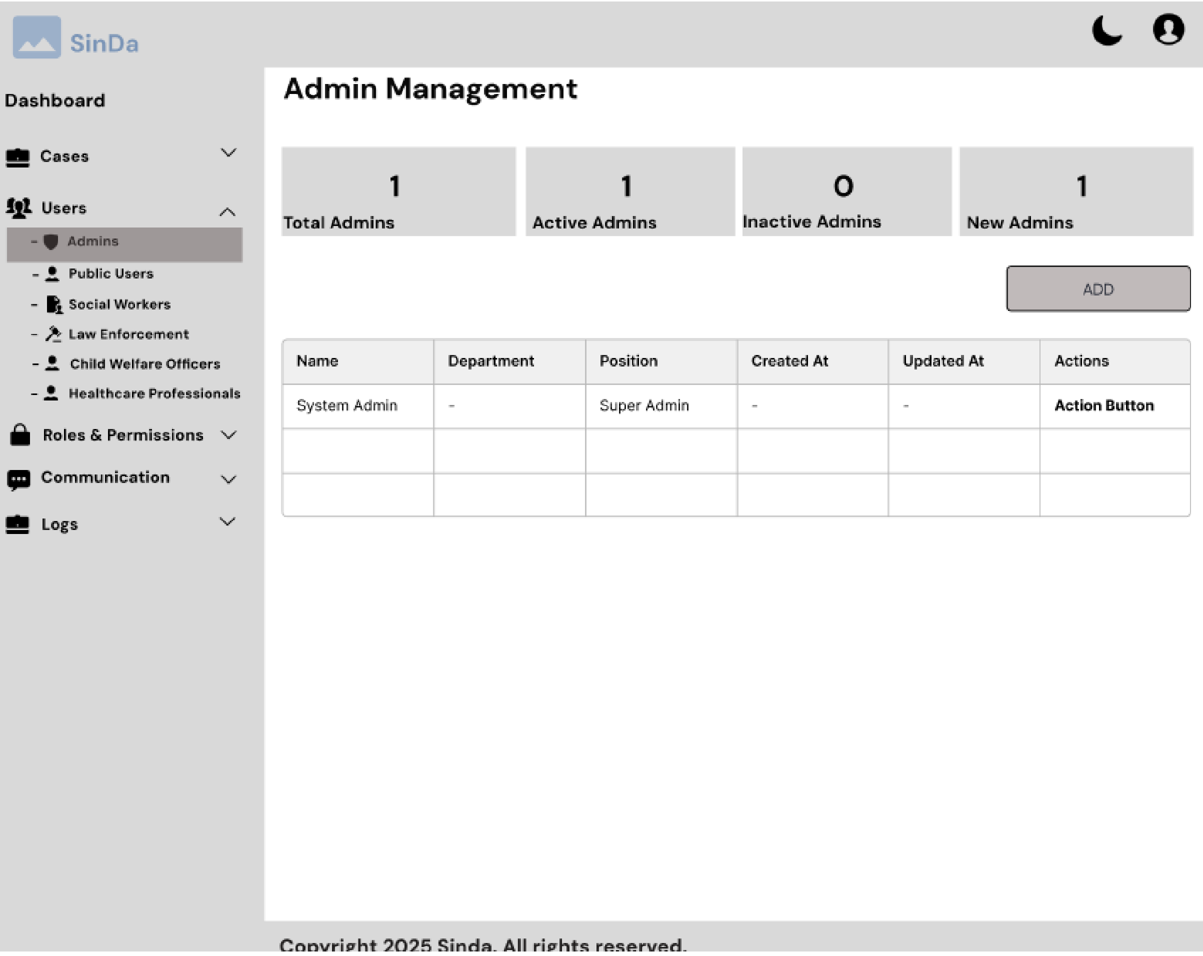


Figure 173: Admin Management Page

The Admin Management page, which lists every administrator in the system, is displayed in this wireframe. Summary cards at the top provide users with a brief overview of the current admin distribution by displaying the total, active, inactive, and recently added admins. To ensure accountability and transparency, each administrator's information is listed in an organised table below, along with their name, department, position, and creation and modification timestamps. Efficient user administration is supported with an Action Button column that enables fast actions like modifying or deleting an administrator. To make adding new admins simple, the Add button is positioned prominently. While the summary cards emphasise important indicators for supervision, the design employs a simple, tabular style for clarity, enabling administrators to swiftly examine and handle information. This method helps guarantee effective administration and job monitoring by striking a balance between concise statistics and thorough documentation.

4.4.12.2 Add New Admin Modal

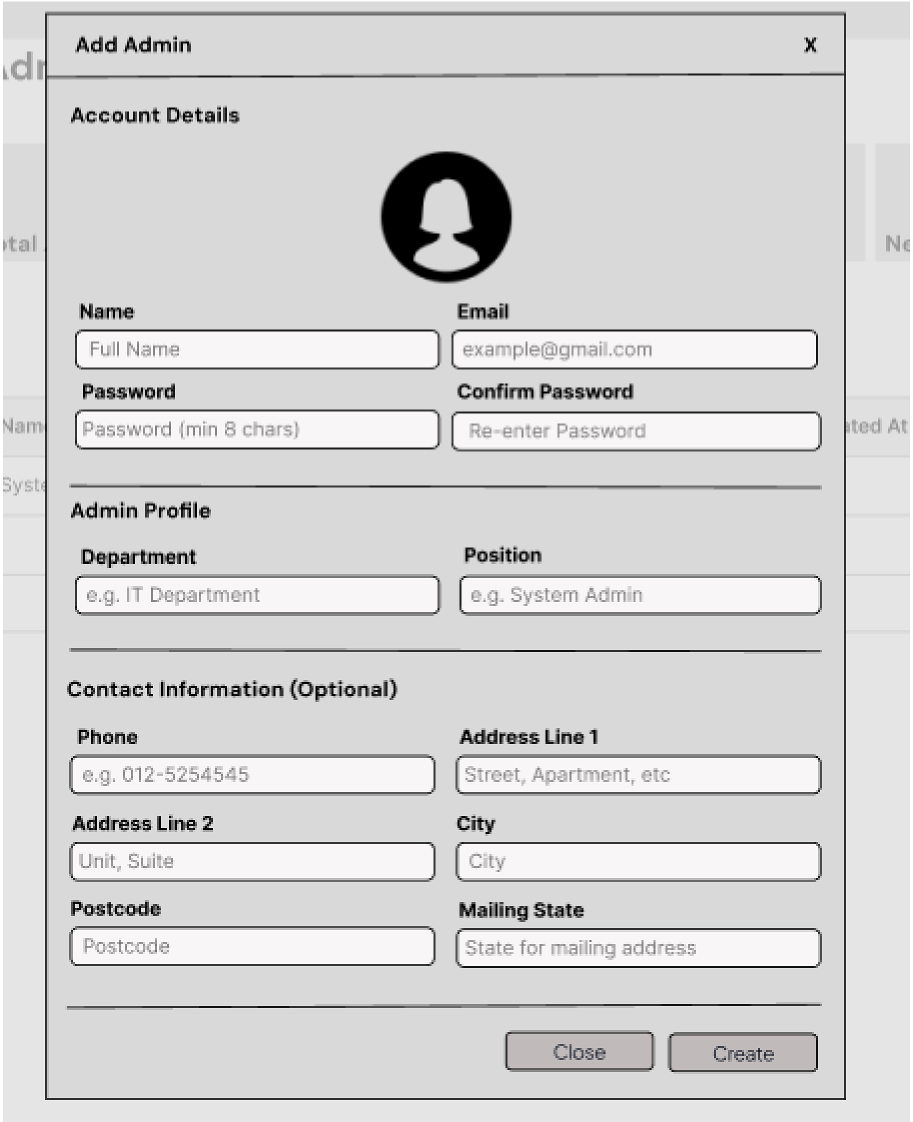


Figure 174: Add New Admin Modal

The Add Admin modal, which enables authorised users to add additional administrators to the system, is depicted in this wireframe. The form is clearly separated into three sections: Contact Information, which contains optional information like an address and phone number, Admin Profile, which contains role-related information like department and position, and Account Details, which contains login credentials. While optional data are separated individually to lessen user stress and allow flexibility, required fields such as name, email, and password are positioned at the top to prioritise the establishment of a vital account. To ensure account security and reduce input mistakes, password and confirm password boxes are added. Clear action options are offered by the Close and Create buttons at the bottom, which provide confirmation or a safe escape. With an emphasis on structured input, error avoidance, and user-friendliness, this architecture guarantees that administrators may be added rapidly while preserving appropriate data security and consistency.

4.4.12.3 Edit Admin Modal

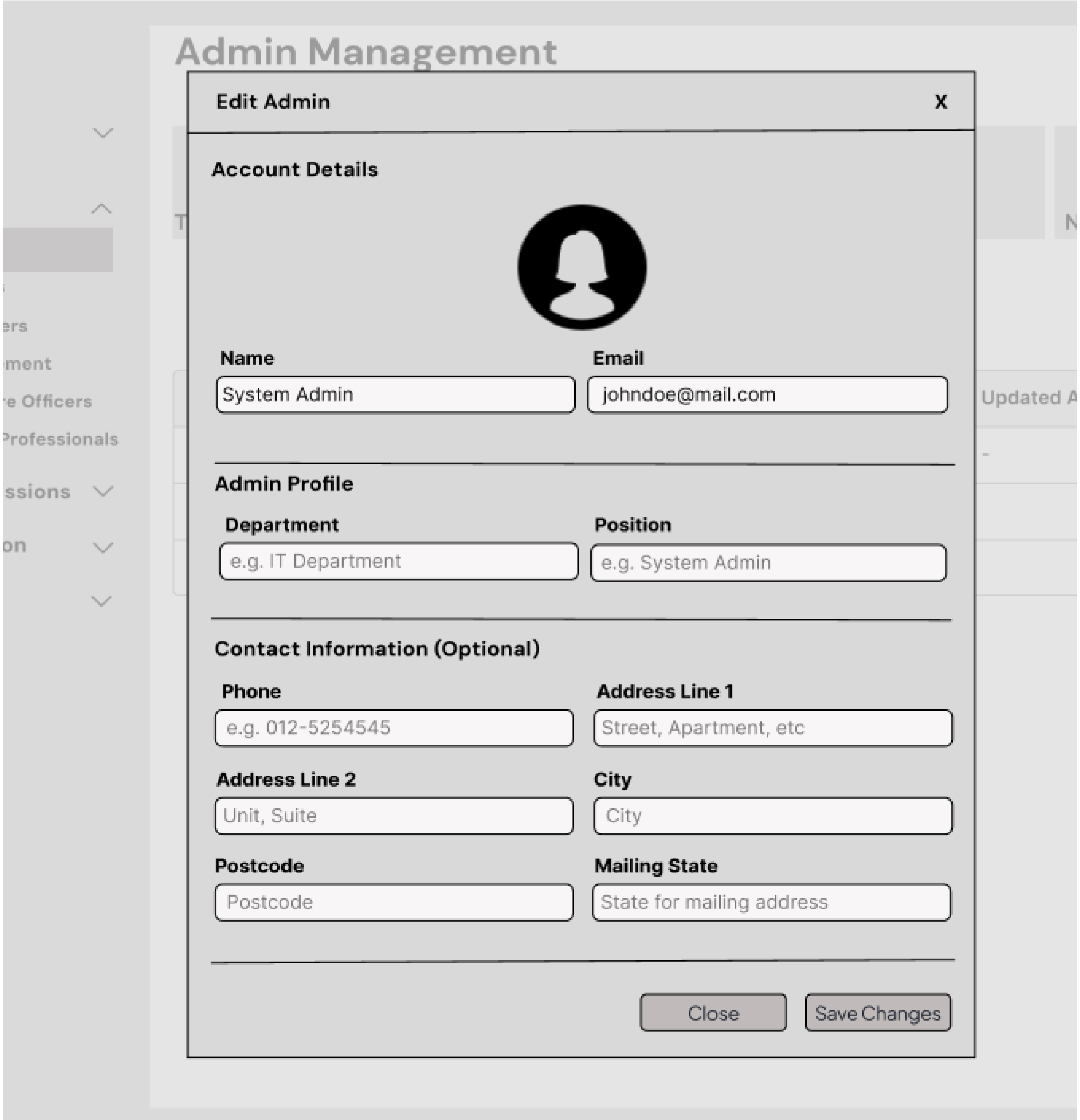


Figure 175: Edit Admin Modal

The Edit Admin modal, which enables authorised users to amend an existing administrator's data, is depicted in this wireframe. The form is divided into three structured sections: Contact Information, which contains optional data like phone number and address; Admin Profile, which contains role-related qualities like department and position and Account Details, which contains essential information like name and email. While optional contact information can still be edited for flexibility, pre-filled fields show the current values to save time and minimise mistakes. The Close and Save Changes action buttons at the bottom offer a clear option for either confirming modifications or deleting edits. By organising relevant fields, facilitating quick updates without sacrificing data integrity, and guaranteeing administrators can easily manage role and contact information, this design places a high priority on usability.

4.4.12.4 Delete Admin Modal

Figure 176: Delete Admin Modal

The admin management Confirm Delete modal, which asks the user a question before permanently deleting an administrator account, is depicted in this wireframe. To prevent deleting errors, the modal explicitly indicates the action being taken and shows the user's name. To provide consumers control over their choice, two buttons are included: Cancel and Delete. The cancel option acts as a safety measure against inadvertent clicks. While the cancel option is still accessible as a safer way out, the delete button is highlighted to highlight the action's irreversible nature. By emphasising accountability, clarity, and error prevention, this design makes sure managers thoroughly consider their options before deleting important user accounts.

4.4.13 Public User Management

4.4.13.1 Public User Management Page

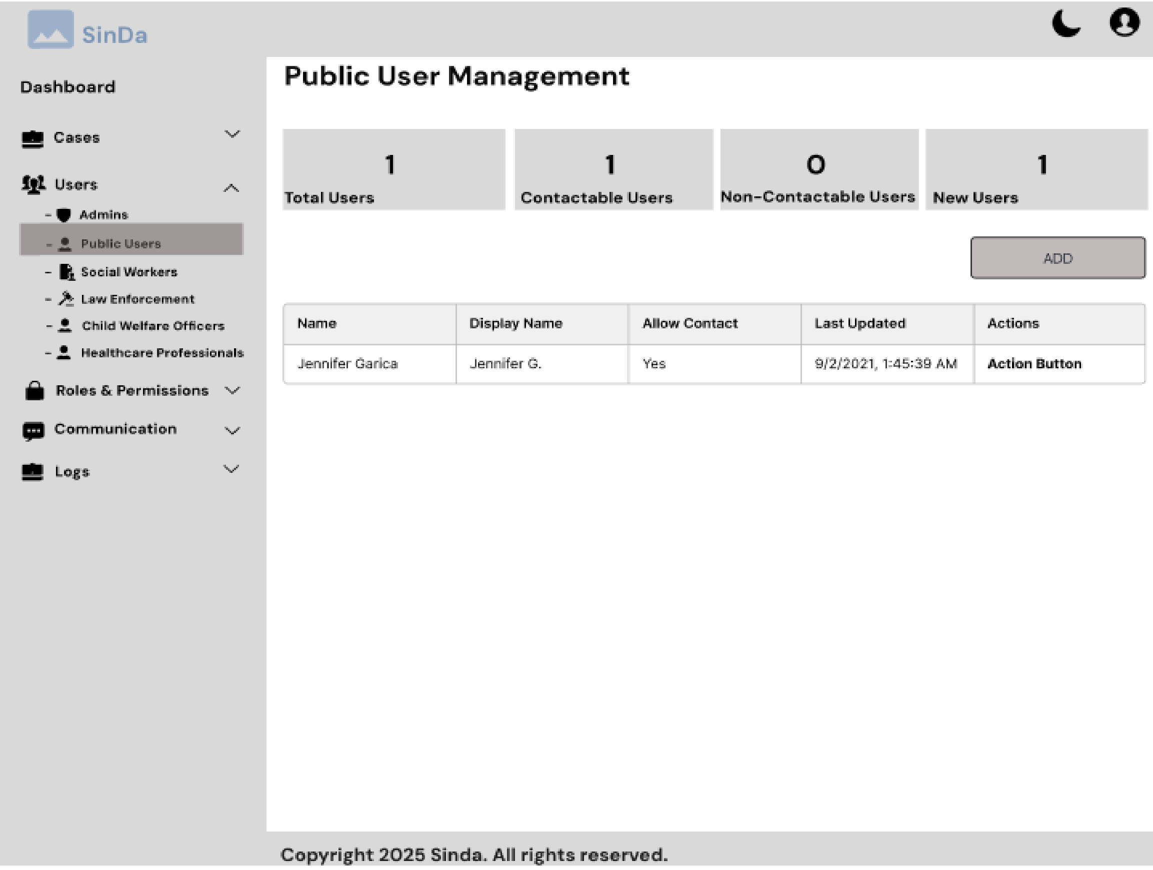


Figure 177: Public User Management Page

The Public User Management page, which enables administrators to supervise system users in general, is displayed in this wireframe. To ensure a rapid overview of user distribution, summary cards at the top display quick information such as the overall number of users, the number of contactable or non-contactable users, and the number of newly added users. To assist administrators in verifying user activity and preferences, the table below provides comprehensive information such as name, display name, contact permission status, and the last modification timestamp. The Add button makes it simple to register new public users, and an Action Button column allows you to perform actions like modifying or disabling a user. Clarity and efficiency are given top priority in this layout, which follows a straightforward and organised format. To enable security and efficient user interaction, this design makes sure administrators can easily control user permissions, preserve communication preferences, and keep an eye on account activity.

4.4.13.2 Add Public User Modal

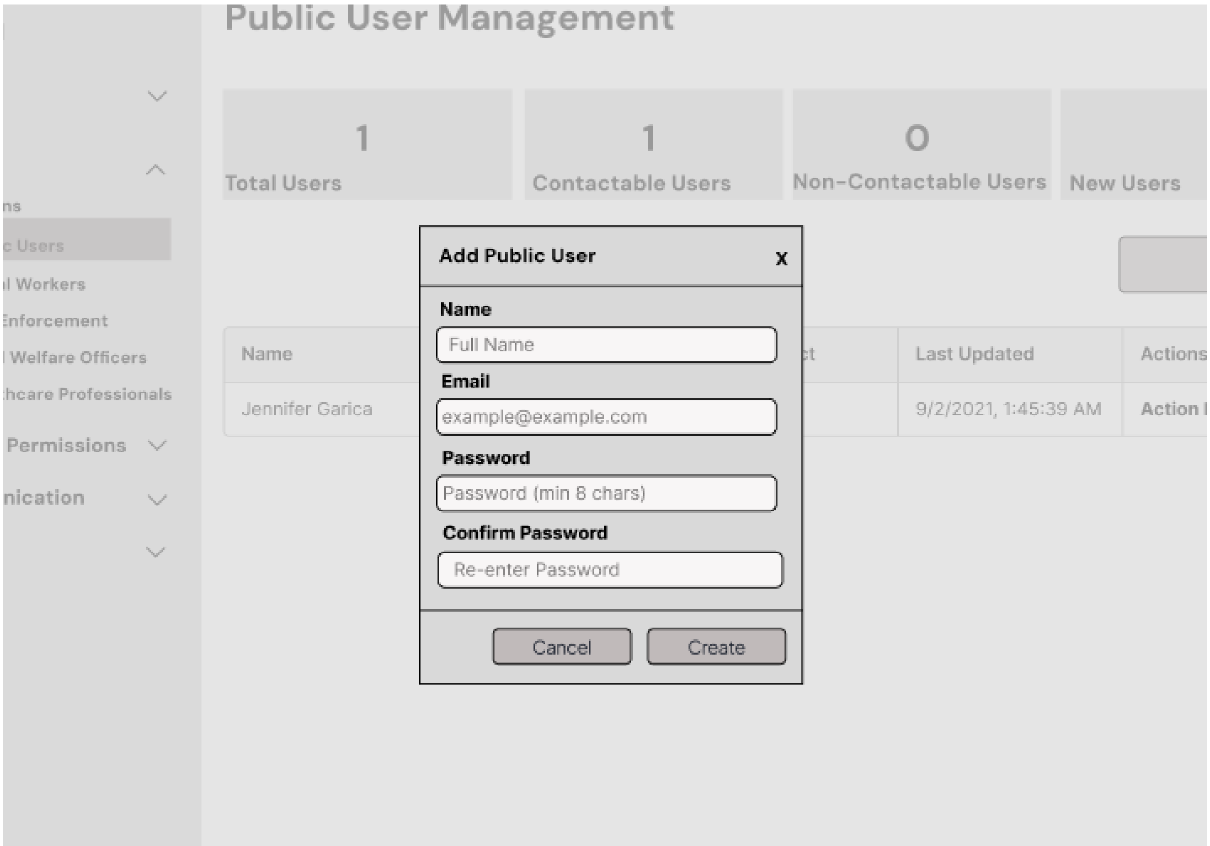


Figure 178: Add Public User Modal

The Add Public User modal, which enables system administrators to add new users, is depicted in this wireframe. With just four needed fields, Name, Email, Password, and Confirm Password the form ensures that just the most important information is gathered for a speedy account creation process. By ensuring consistency, the confirm password box lowers the possibility of mistakes, and account security is supported by the minimum character requirement. The Cancel and Create buttons at the bottom provide users the option to confirm the new account or to quit without making any changes. The small size reduces mental strain, expedites the registration process, and guarantees that administrators may add users quickly and accurately while preserving security.

4.4.13.3 Edit Public User Modal

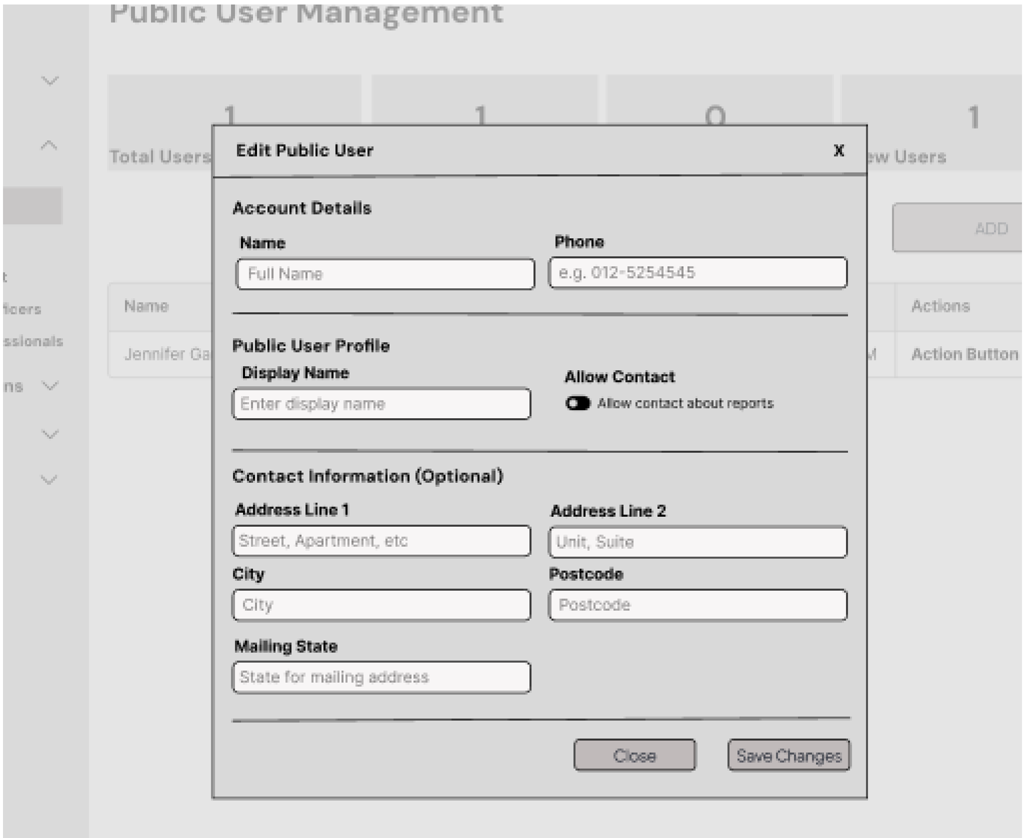


Figure 179: Edit Public User Modal

The Edit Public User modal, seen in this wireframe, enables administrators to amend an existing user's information. The information is structured and easy to understand thanks to the form's division into parts for Contact Information, Account Details, and Public User Profile. Basic information like name and phone number are included in the account data, but the user profile section adds a Display Name and an Allow Contact option that allow users to be contacted regarding reports. More extensive information may be kept without overcrowding the primary form thanks to optional fields like address, city, postcode, and postal state. While pre-filled areas streamline editing and minimise input mistakes, the action buttons Close and Save Changes offer a clear option for completing or rejecting adjustments. Usability, adaptability, and accuracy are prioritised in the design to provide efficient user administration while upholding contact preferences and privacy.

4.4.13.4 Delete Public User Modal

Figure 180: Delete Public User Modal

The purpose of the Delete Public User modal, which is displayed in this wireframe, is to obtain confirmation before deleting a user account permanently. To avoid selecting the incorrect account, the modal displays the user's name and asks the administrator if they are certain they want to delete the user. For decision-making, two buttons are displayed: Cancel and Delete. The cancel option permits a secure escape, while the delete button is highlighted to validate the action. To retain accountability and lower the possibility of unintentional deletions, the design adheres to the principles of clarity and mistake prevention, guaranteeing that administrators make a conscious decision prior to carrying out an irreversible action.

4.4.14 Social Worker Management

4.4.14.1 Social Worker Management Page

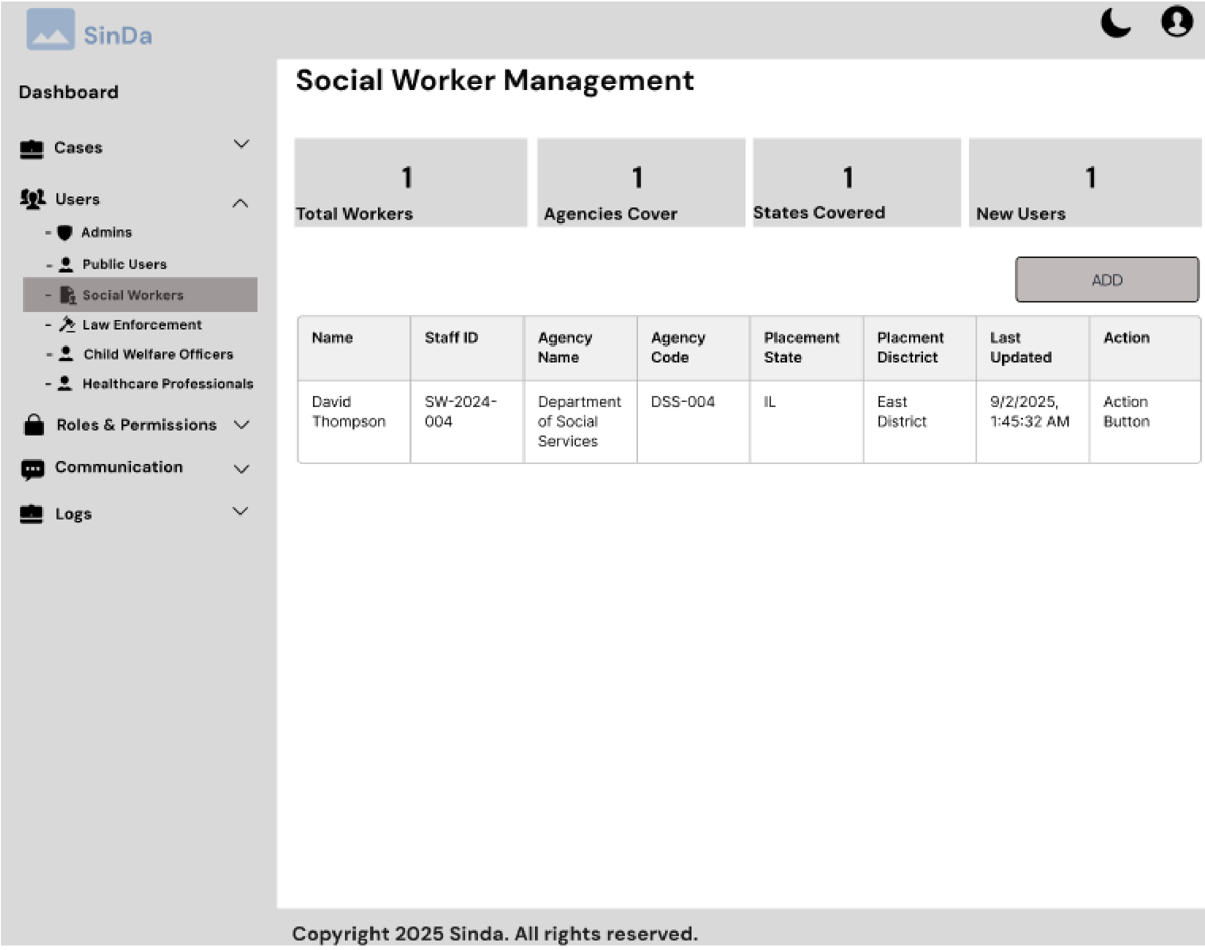


Figure 181: Social Worker Management Page

The Social Worker Management page, seen in this wireframe, gives administrators the ability to supervise and control registered social workers within the system. To give a brief overview of the workforce distribution, summary cards at the top include important facts like the overall number of workers, agencies covered, states covered, and new users. To ensure proper record-keeping and simple verification, the table below shows each social worker's name, staff ID, agency information, placement location, and last updated timestamp. While the Add button enables administrators to register new workers, the Action Button column offers instant access to features like amending, reassigning, or dismissing a social worker. The design supports efficient labour distribution between agencies and regions by using a simple, tabular layout to guarantee data clarity and speedy scanning. Administrators may manage social workers methodically using this architecture, which prioritises efficiency, accountability, and scalability while guaranteeing coverage across jurisdictions.

Add Social Worker Modal

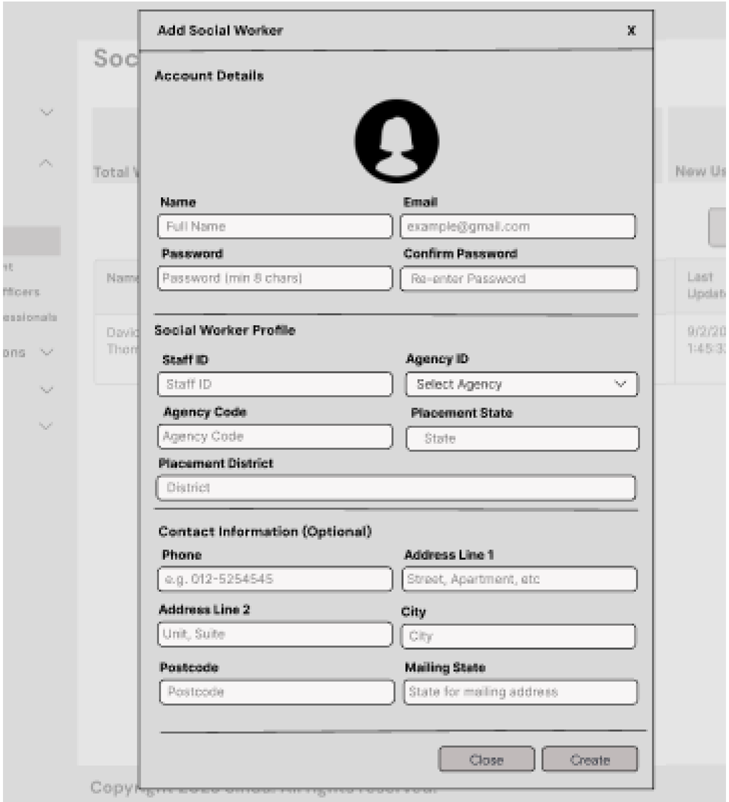


Figure 182: Add Social Worker Modal

This wireframe shows the Add Social Worker modal, which allows administrators to add more social workers to the system. The form is separated into sections for Account Details, Social Worker Profile, and Contact Information to guarantee readability and ease of use. The account area contains crucial login information, such as name, email address, and password with a confirm password to prevent mistakes. The profile area collects essential work-related information, including staff ID, agency ID, agency code, and placement details which contain state and district, to enable proper allocation and jurisdiction tracking. Optional contact fields allow for more detailed entries without overloading the user with information when they initially input it. The Close and Create buttons provide clear action pathways for administrators to safely terminate or approve the addition of a new employee. This design prioritises accuracy, completeness, and flexibility, ensuring that all relevant information is collected for interagency collaboration while facilitating social worker onboarding.

4.4.14.2 Add Social Worker Modal

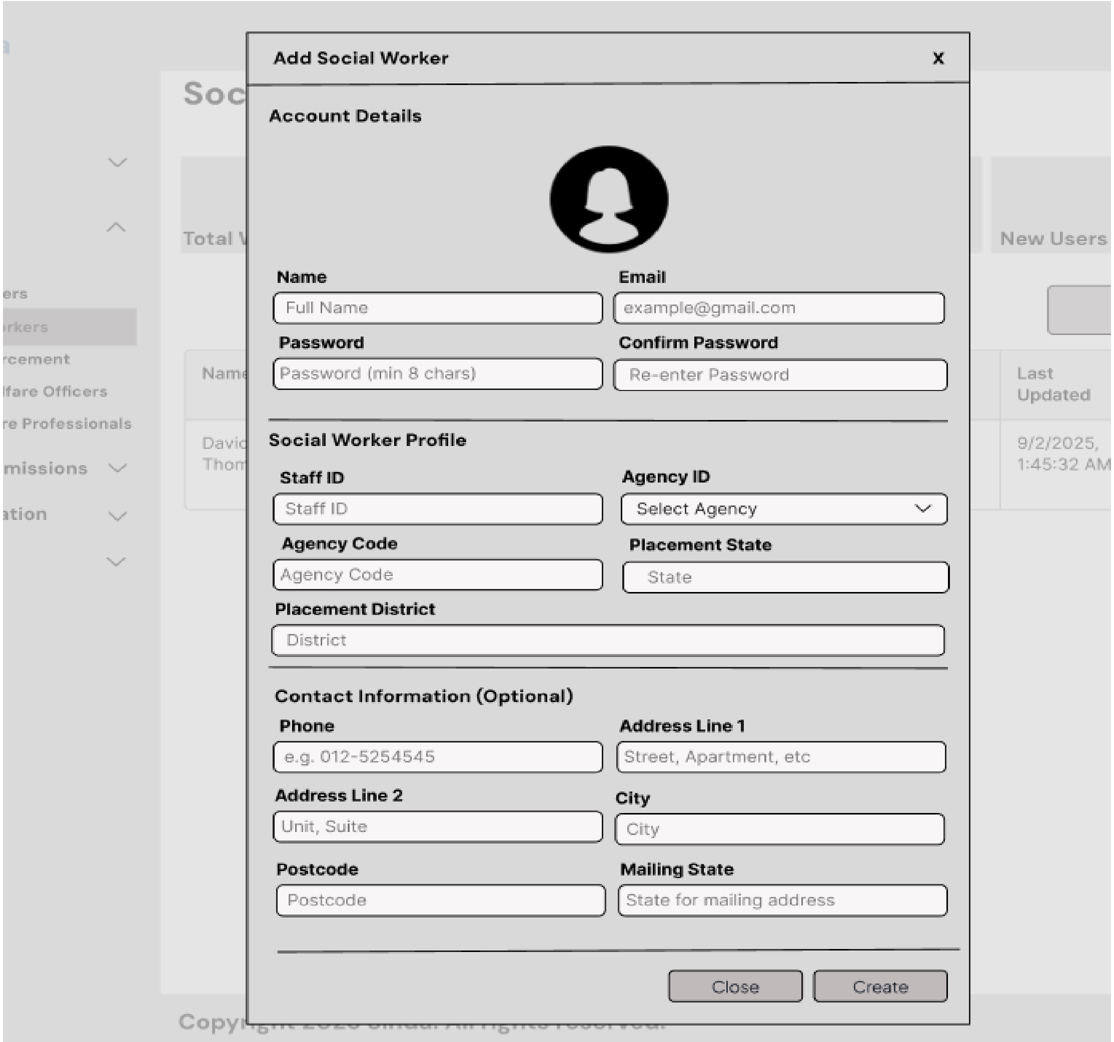


Figure 183: Add Social Worker Modal

This wireframe shows the Add Social Worker modal, designed to let administrators register new social workers into the system. It is divided into three clear sections: account details, social worker profile, and optional contact information. The top section collects essential login credentials like name, email, and password to ensure secure system access. The profile section captures professional identifiers such as staff ID, agency ID, placement state, and district, which are important for role-based assignments and regional case allocation. The optional contact section allows for adding detailed personal information like phone number and address, supporting better communication when needed. The design uses logical grouping, form fields with placeholder text for clarity, and dropdown options for standardized values, making the registration process structured, intuitive, and easy to complete while maintaining data consistency.

4.4.14.3 Edit Social Worker Modal

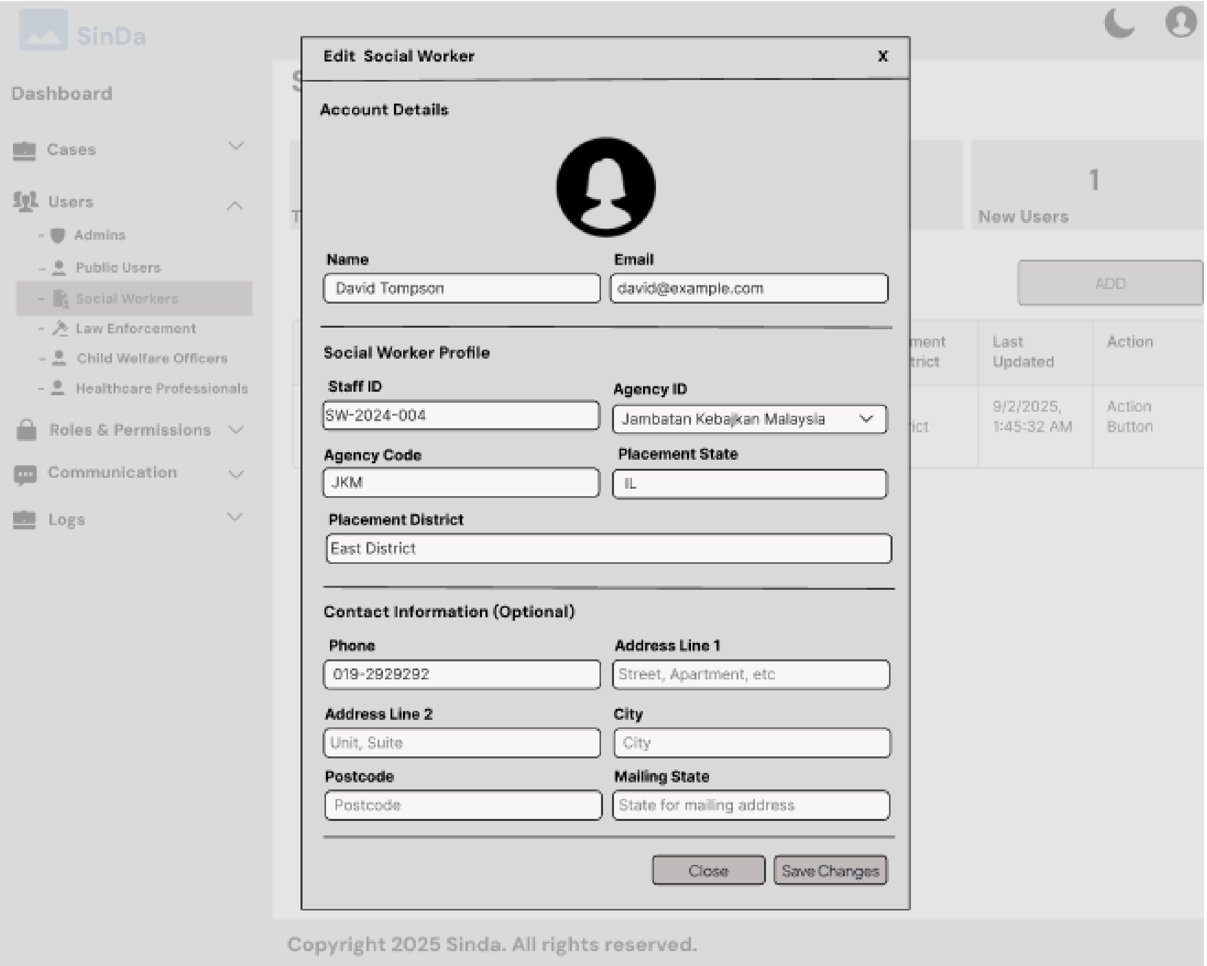


Figure 184: Edit Social Worker Modal

The Add Social Worker modal, which allows administrators to add additional social workers to the system, is depicted in this wireframe. The account data, social worker profile, and optional contact information are the three distinct categories into which it is separated. To provide safe system access, the top part gathers necessary login information, such as name, email address, and password. Professional identifiers including staff ID, agency ID, placement state, and district are recorded in the profile area and are crucial for role-based assignments and regional case distribution. When necessary, the optional contact area facilitates improved communication by enabling the addition of comprehensive personal data, such as address and phone number. The registration process is made organised, simple, and straightforward to complete while preserving data consistency thanks to the design's use of logical grouping, form fields with placeholder text for clarity, and dropdown alternatives for standardised values.

4.4.14.4 Delete Social Worker Modal

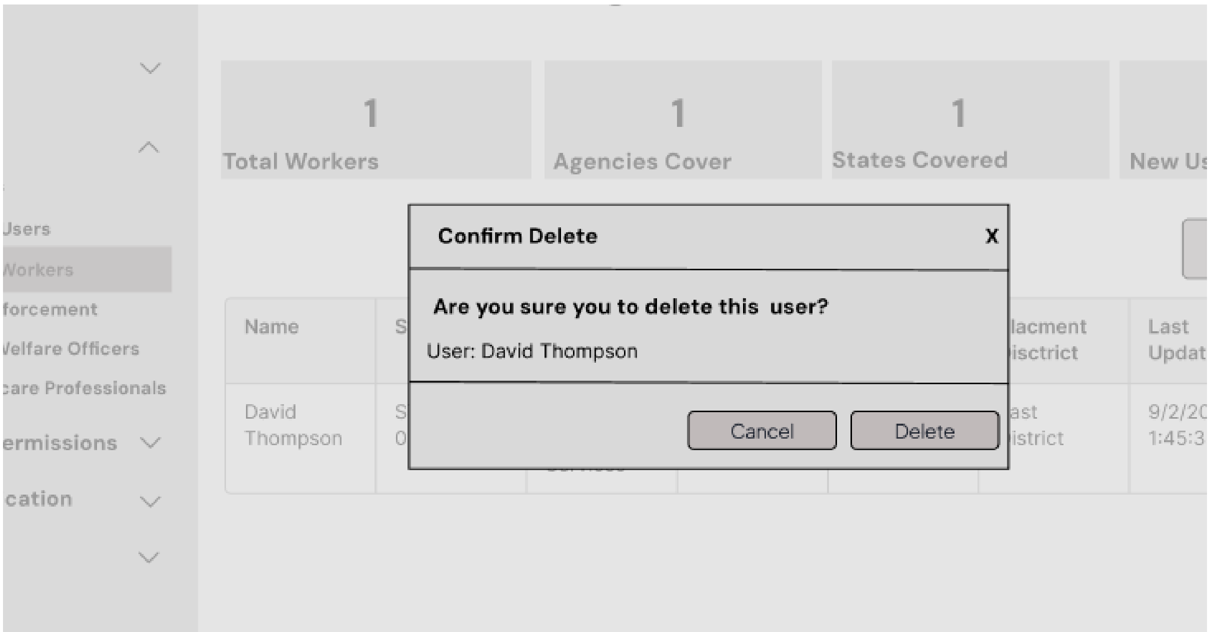


Figure 185: Delete Social Worker Modal

This wireframe shows the Confirm Delete modal, which appears when an administrator attempts to remove a user from the system. The design provides a clear confirmation message with the user's name displayed to prevent accidental deletion. It includes two action buttons: Cancel to safely exit without changes and delete to confirm the action. The simple, centered layout with bold text for the warning ensures that the user pays attention before making a potentially irreversible decision. This design reasoning prioritizes user awareness, reduces mistakes, and enforces accountability in user management.

4.4.15 Law Enforcement Management

4.4.15.1 Law Enforcement Page

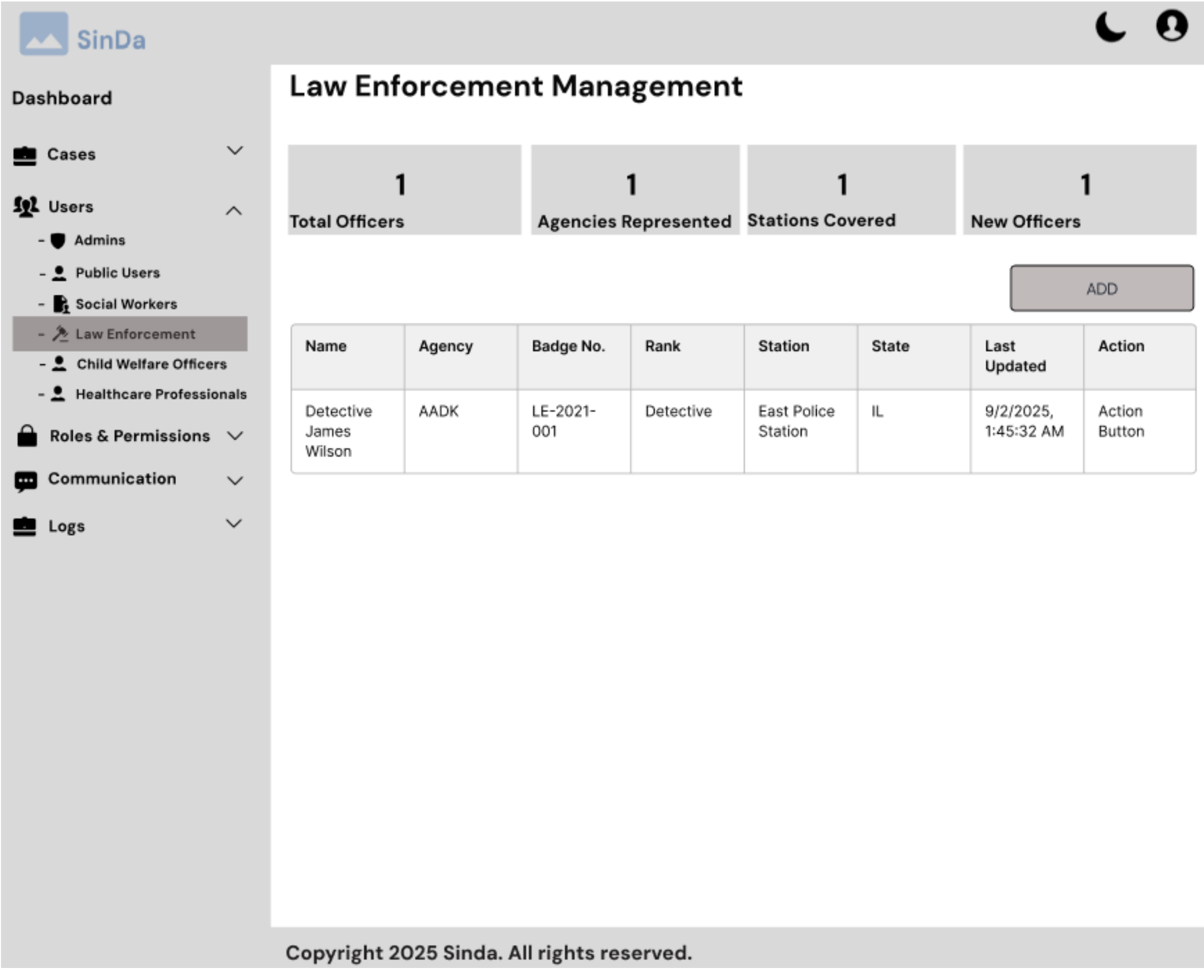


Figure 186: Law Enforcement Page

This wireframe shows the Law Enforcement Management page, which allows administrators to manage law enforcement officers in the system. At the top, four summary cards display quick statistics: total officers, agencies represented, stations covered, and new officers, helping users quickly gauge the scope of representation. Below, a structured table lists detailed officer information including name, agency, badge number, rank, station, state, last updated date, and an action button for further management tasks like editing or deleting. The Add button at the top-right provides a straightforward way to register new officers. The design ensures clarity and efficiency by combining high-level insights with detailed records, making it easy to track, update, and expand law enforcement data systematically.

4.4.15.2 Add Law Enforcement Modal

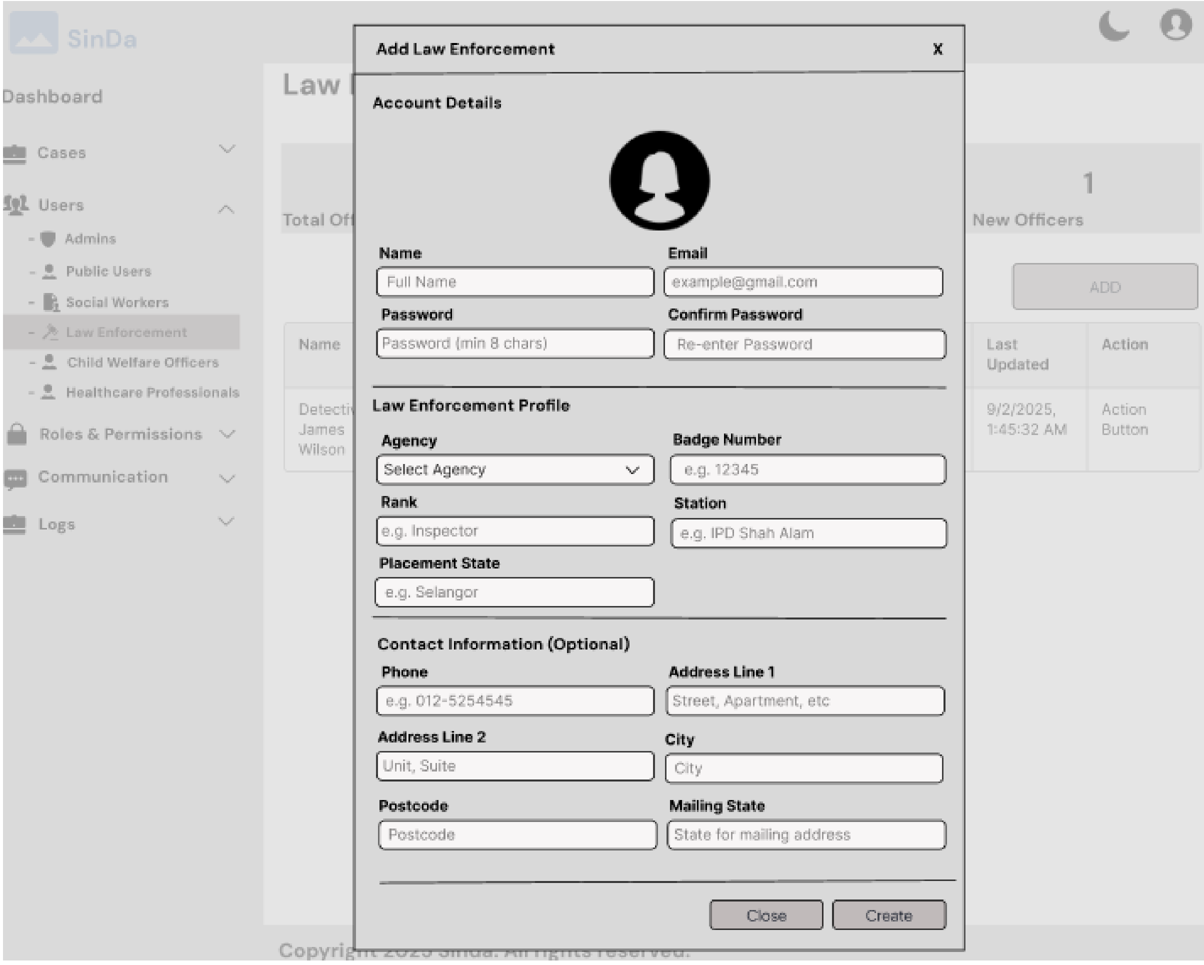


Figure 187: Add Law Enforcement Modal

This wireframe shows the Add Law Enforcement modal, which administrators use to register new law enforcement officers. The design is structured into three sections for clarity. The top Account Details section captures login credentials such as name, email, and password to secure access. The Law Enforcement Profile section records professional details including agency, badge number, rank, station, and placement state, ensuring officers are properly identified and linked to their jurisdiction. The final Contact Information section allows additional details like phone number and address to be stored for communication purposes if needed. Logical grouping, placeholder examples, and dropdown selections help standardize input, reduce errors, and make the process intuitive. The Create and Close buttons at the bottom provide clear next steps, supporting both task completion and easy cancellation.

4.4.15.3 Edit Law Enforcement Modal

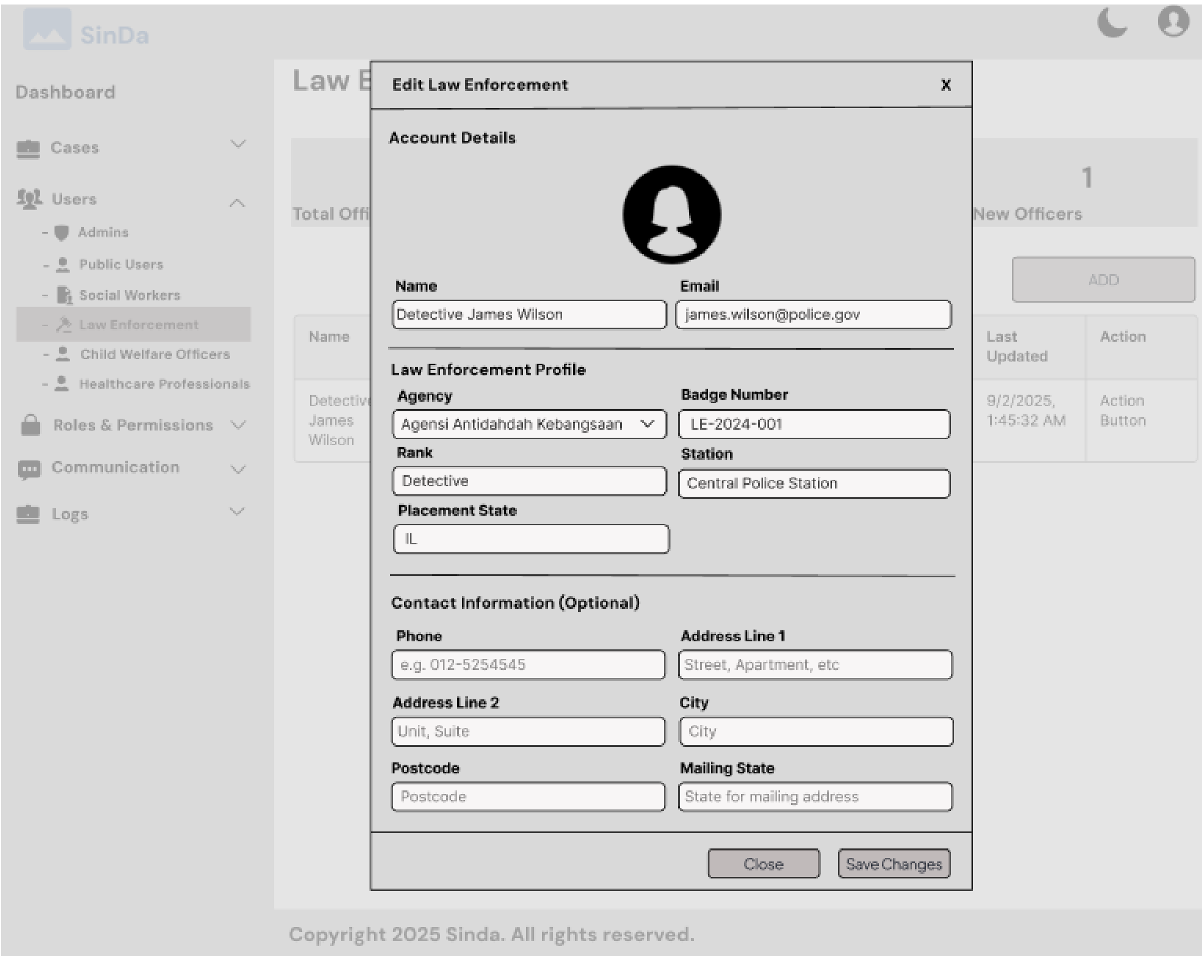


Figure 188: Edit Law Enforcement Modal

This wireframe shows the Edit Law Enforcement modal, which allows administrators to update details of an existing law enforcement officer. The layout is consistent with the add form for familiarity, but pre-filled fields show the officer's current information, such as name, email, agency, badge number, rank, station, and placement state. Optional contact fields remain editable to add or modify communication details. The clear separation into sections for account details, professional profile, and optional contact ensures structured editing without overwhelming the user. The action buttons at the bottom, Close and Save Changes, make it simple to either exit without modification or confirm updates. The design reasoning here focuses on consistency, clarity, and reducing input errors, while providing administrators with an efficient way to maintain accurate and up-to-date officer records.

4.4.15.4 Delete Law Enforcement Modal

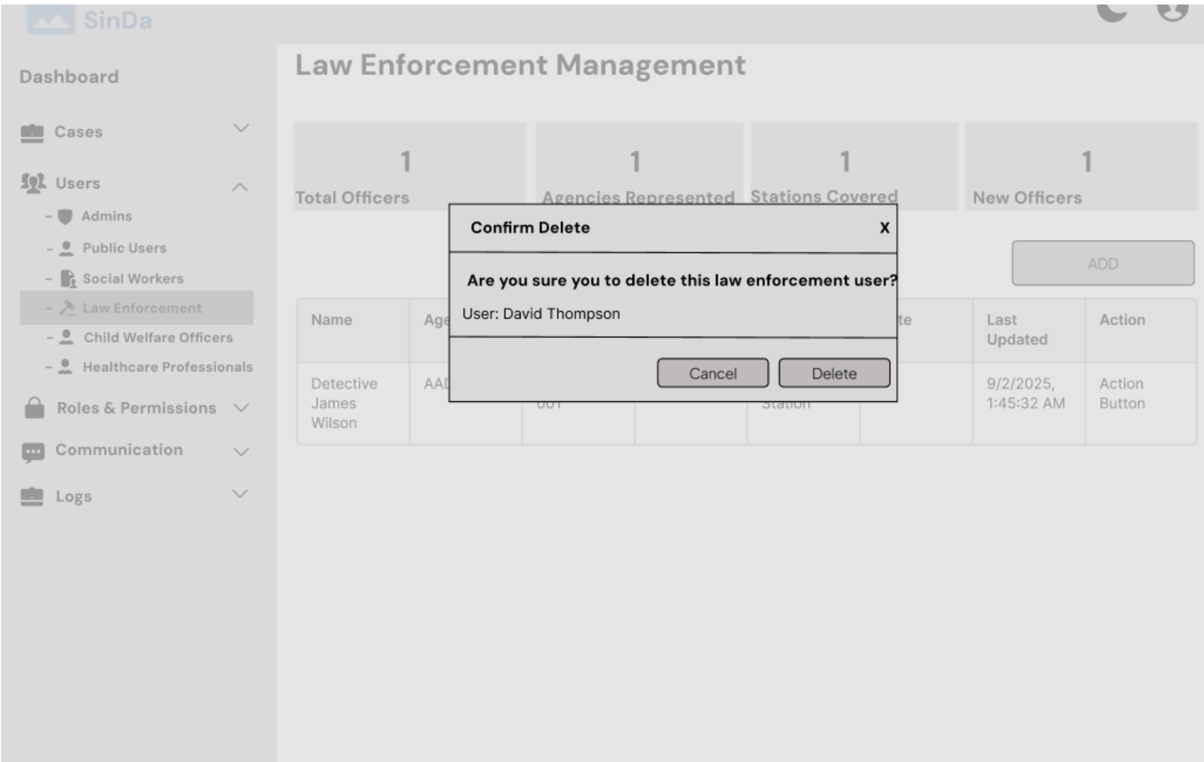


Figure 189: Delete Law Enforcement Modal

This wireframe shows the **Confirm Delete** modal for law enforcement users, which appears when an administrator attempts to remove an officer's account. The confirmation message clearly states the type of user and displays the officer's name to ensure accuracy before deletion. Two action buttons, **Cancel** and **Delete**, give the administrator a safe choice to back out or proceed. The compact, centered layout with bold warning text emphasizes the importance of the decision, helping prevent mistakes while maintaining accountability in user management. The design reasoning prioritizes clarity, user awareness, and safe handling of critical actions.

4.4.16 Government Child Welfare Officer Management

4.4.16.1 Government Child Welfare Officer Management

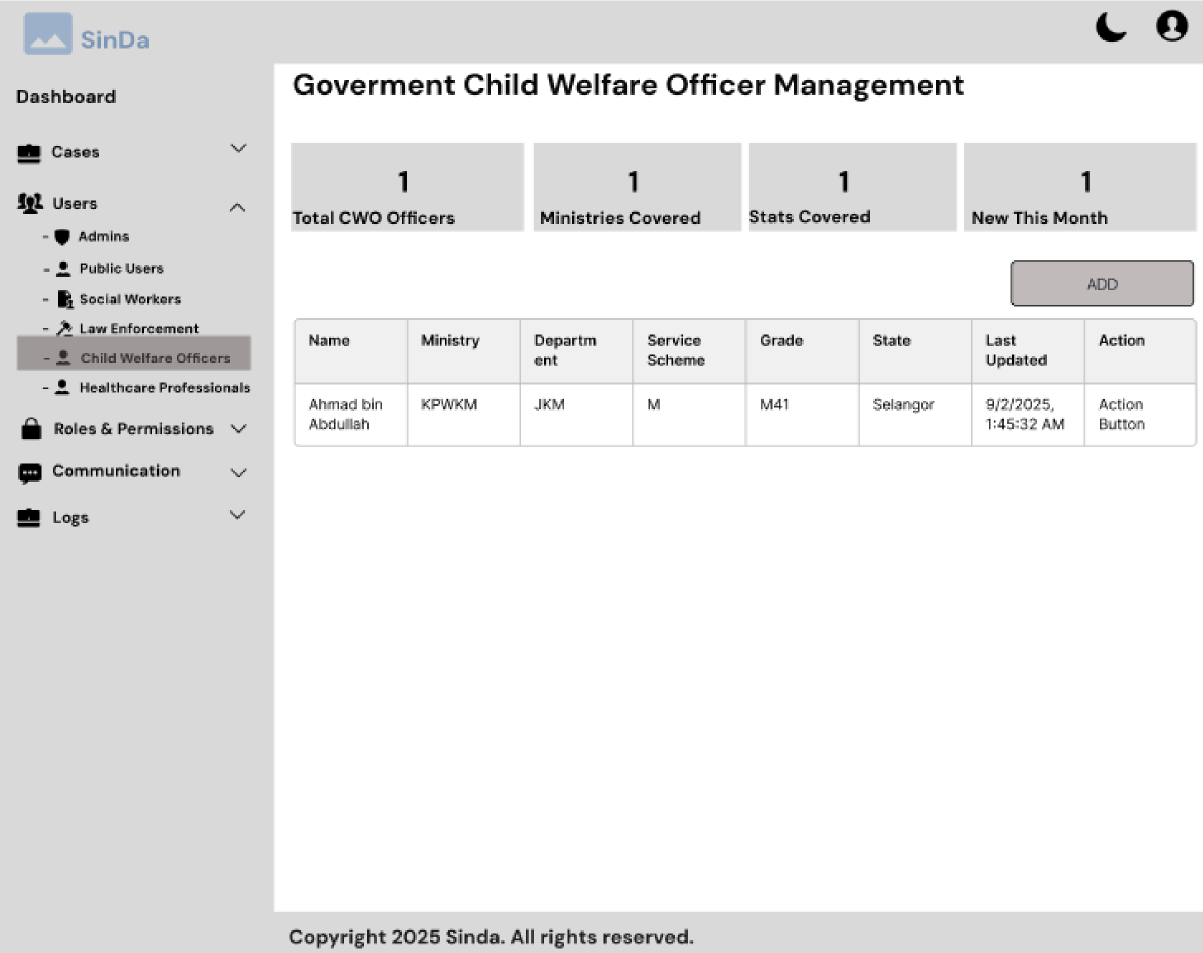


Figure 190: Government Child Welfare Officer Management

This wireframe shows the Government Child Welfare Officer Management page, which provides administrators with tools to oversee CWO officers in the system. At the top, four summary cards display key statistics: total officers, ministries covered, states covered, and new officers added in the current month, giving a quick overview of coverage and growth. Below, a table lists detailed officer records, including name, ministry, department, service scheme, grade, state, last updated timestamp, and an action button for further management such as editing or deleting. The Add button on the top-right enables new officer registrations, ensuring scalability as the network expands. The structured layout, consistent with other management modules, balances high-level summaries with detailed records, promoting ease of navigation and efficient officer management.

4.4.16.2 Add Child Welfare Officer Modal

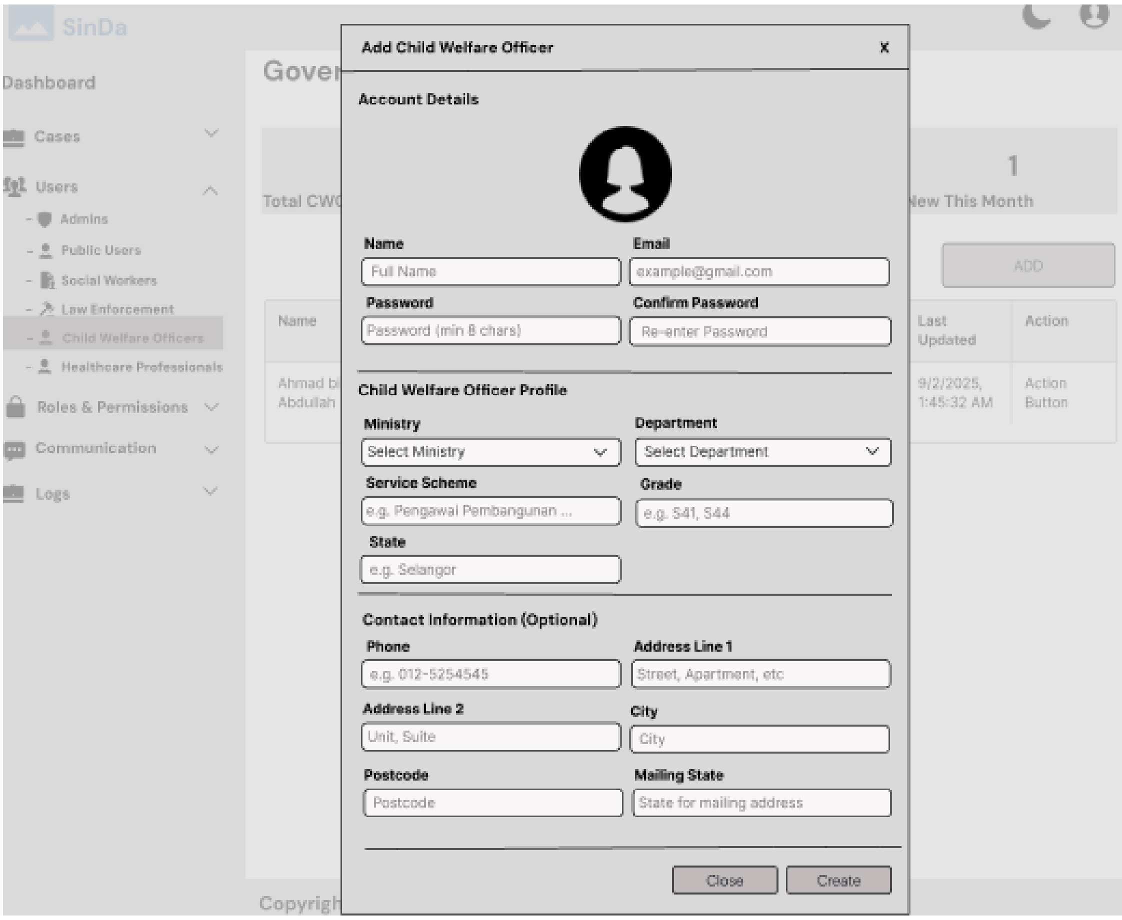


Figure 191: Add Child Welfare Officer

This wireframe shows the Add Child Welfare Officer modal, which administrators use to register new government officers in the system. The design is divided into three structured sections. The Account Details section captures login credentials such as name, email, and password to ensure secure system access. The Child Welfare Officer Profile section collects key professional details including ministry, department, service scheme, grade, and state, ensuring officers are accurately classified under their official roles and jurisdictions. Finally, the Contact Information (Optional) section provides fields for phone and address details, supporting communication if needed. The layout uses dropdowns and placeholder examples to guide users and maintain data consistency. At the bottom, Create and Close buttons allow clear actions for completing or cancelling the registration. The design reasoning emphasizes clarity, standardized input, and efficiency, making officer onboarding streamlined and error-free.

4.4.16.3 Edit Child Welfare Officer Modal

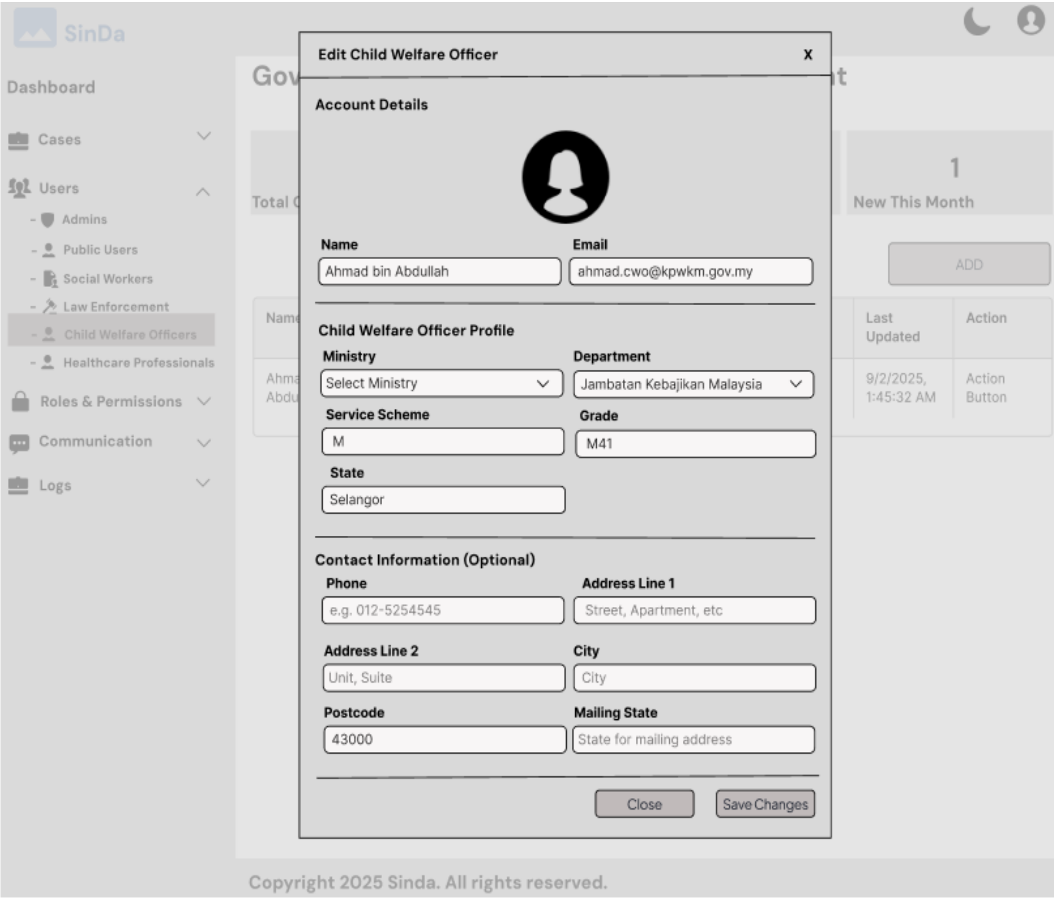


Figure 192: Edit Child Welfare Officer Modal

This wireframe shows the Edit Child Welfare Officer modal, which allows administrators to update details of an existing officer. The form is divided into clear sections for usability. The Account Details section displays editable fields for name and email, pre-filled with the officer's current information. The Child Welfare Officer Profile section captures professional details such as ministry, department, service scheme, grade, and state, helping maintain accurate classification within government structures. The Contact Information section provides fields for phone and address details, allowing updates to communication information when necessary. Pre-filled data ensures accuracy while minimizing re-entry, and the consistent layout with dropdowns and input fields makes editing straightforward. At the bottom, the Close and Save Changes buttons give users clear control over cancelling or confirming edits. The design prioritizes consistency, accuracy, and efficiency, ensuring officer records remain up to date with minimal effort.

4.4.16.4 Delete Child Welfare Officer Modal

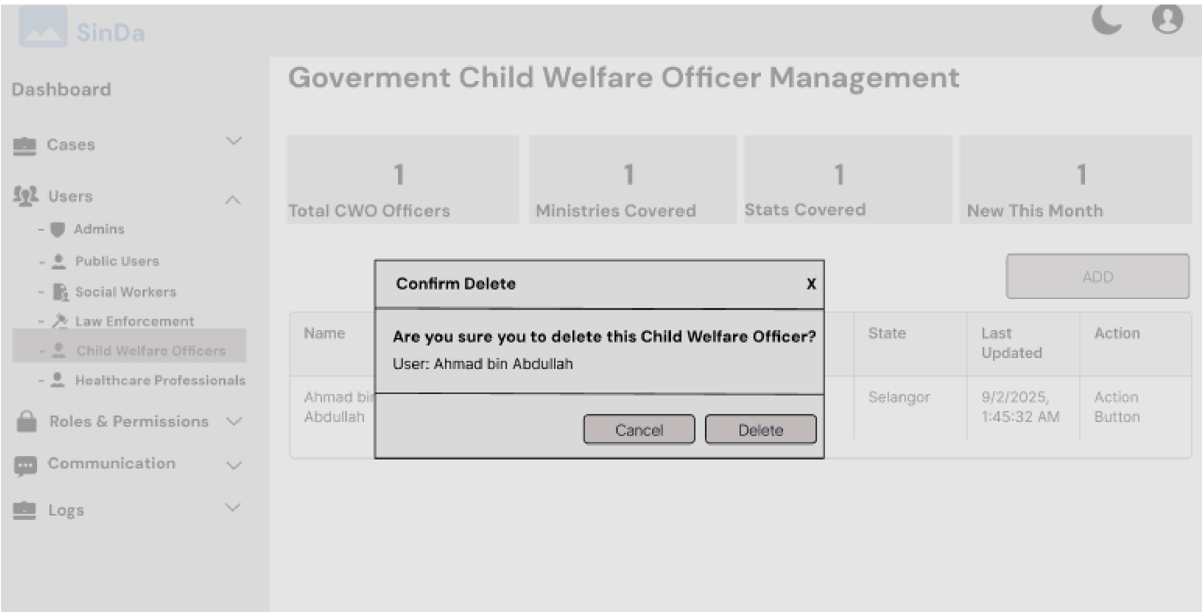


Figure 193: Delete Child Welfare Officer Modal

This wireframe shows the Confirm Delete modal for a Child Welfare Officer, which appears when an administrator initiates a deletion. The confirmation message clearly specifies the officer’s role and displays the officer’s name to avoid mistakes. Two action buttons, Cancel and Delete, give the user a safe choice between aborting or confirming the action. The centered, minimal layout with bold confirmation text highlights the seriousness of the decision, ensuring administrators carefully review before proceeding. The design reasoning emphasizes clarity, accountability, and error prevention in handling sensitive officer records.

4.4.17 Healthcare Professionals Management

4.4.17.1 Healthcare Professional Management Page

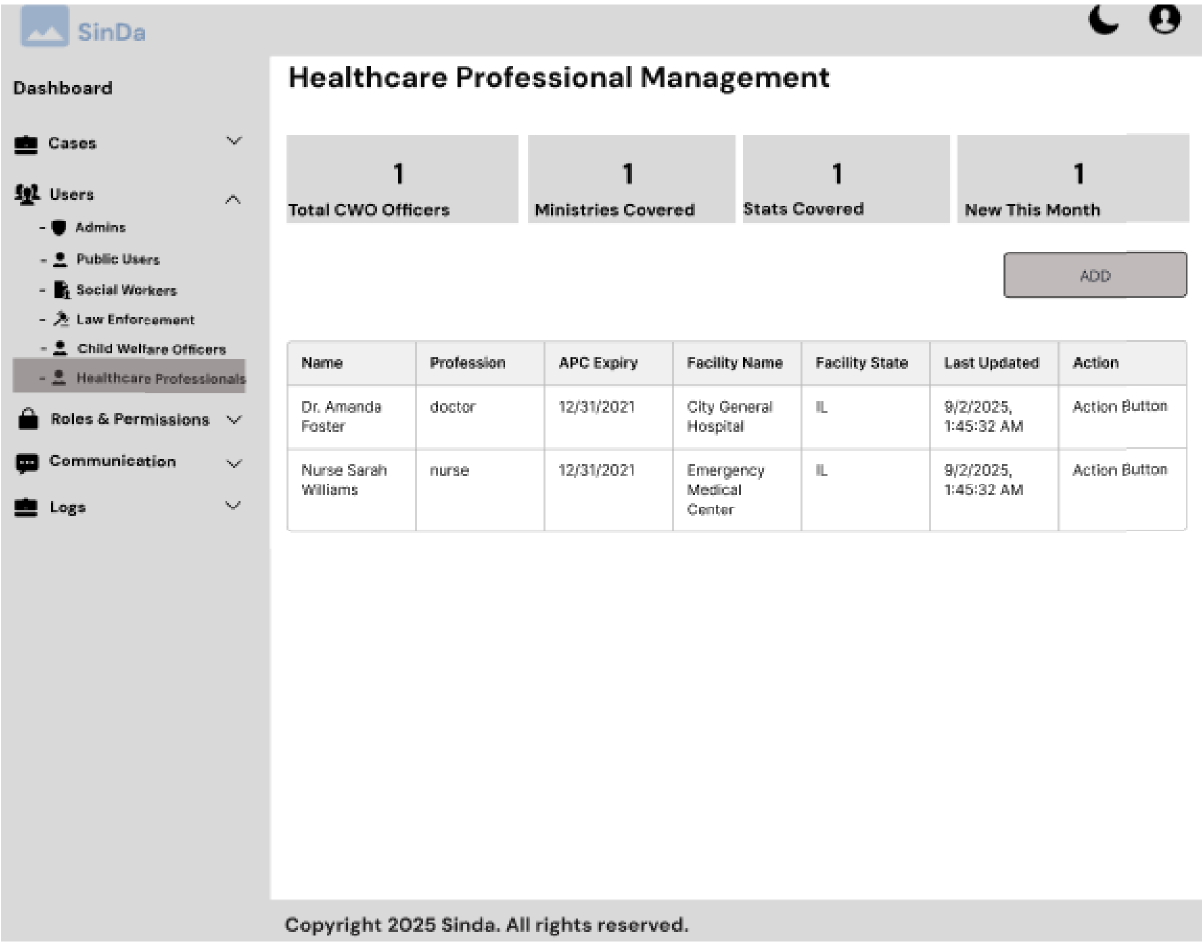


Figure 194: Healthcare Professional Management Page

This wireframe shows the Healthcare Professional Management page, which enables administrators to oversee healthcare users in the system. At the top, four summary cards provide a quick overview with total professionals, ministries covered, states covered, and new professionals added this month, giving instant insights into coverage and growth. Below, a structured table lists detailed records, including name, profession, APC expiry, facility name, facility state, last updated timestamp, and an action button for management tasks like editing or deleting. The Add button on the top-right allows for easy registration of new healthcare professionals. The design ensures consistency with other management modules by combining high-level statistics with detailed listings, promoting clarity, efficiency, and scalability in managing professional users.

4.4.17.2 Add Healthcare Professional Modal

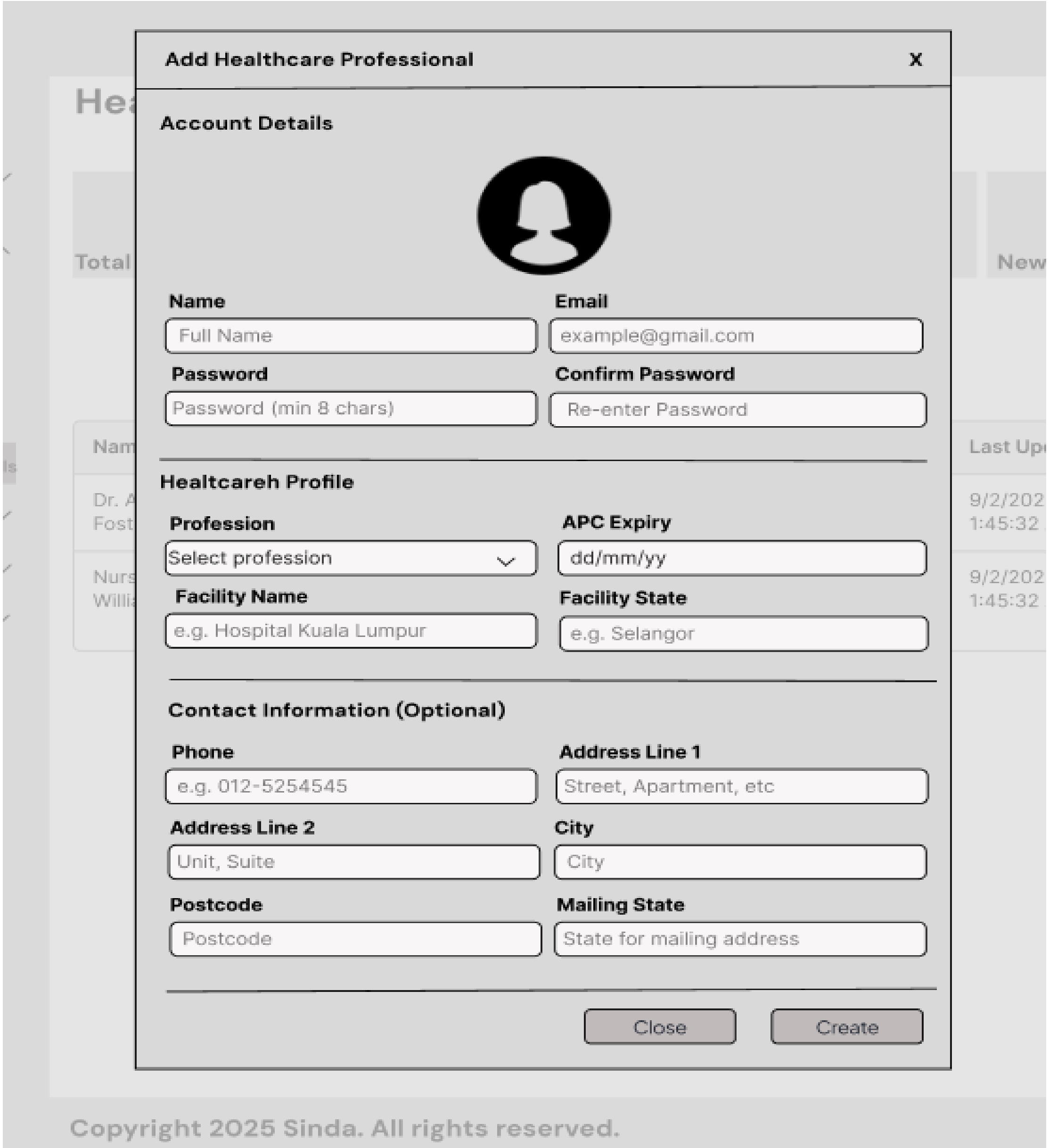


Figure 195: Add Healthcare Professional Modal

This wireframe shows the Add Healthcare Professional modal, which administrators use to register new doctors, nurses, or other healthcare users. The form is divided into three structured sections. The Account Details section collects login credentials including name, email, and password to ensure secure system access. The Healthcare Profile section records professional details such as profession via dropdown, APC expiry date, facility name, and facility state, ensuring proper validation of medical professionals and linking them to their workplace. The Contact Information section allows administrators to input phone numbers and addresses for easier communication if needed. The form uses placeholder text and dropdowns for standardization, reducing input errors and maintaining consistency. The bottom Create and Close buttons provide clear user control for either completing or cancelling the registration. The design reasoning emphasizes structure, accuracy, and simplicity, making the onboarding process both reliable and efficient.

4.4.17.3 Edit Healthcare Professional Modal

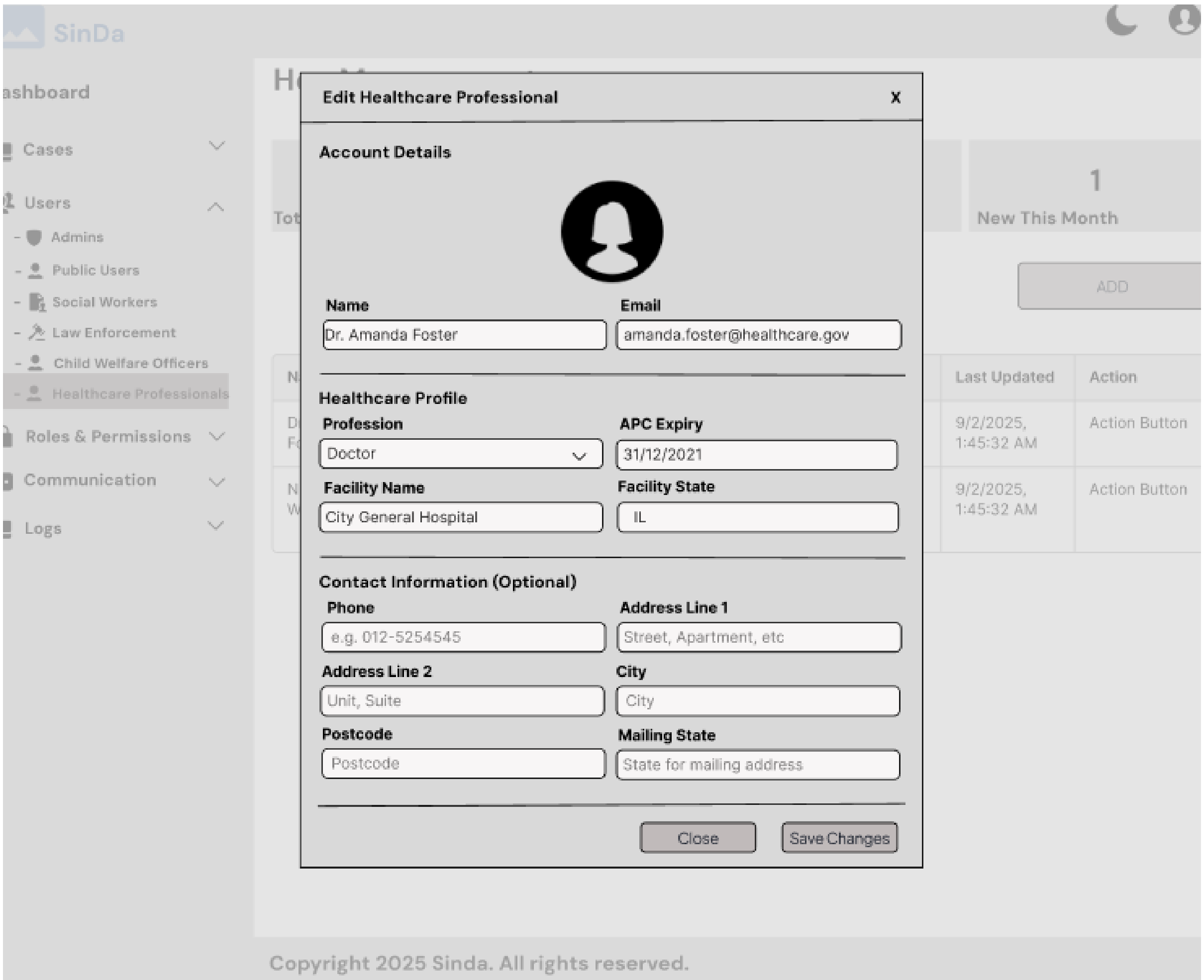


Figure 196: Edit Healthcare Professional Modal

This wireframe shows the Edit Healthcare Professional modal, which allows administrators to update details of an existing healthcare user. The layout follows the same structure as the add form for consistency, but with pre-filled information such as name, email, profession, APC expiry, facility name, and facility state, making it easy to review and adjust. Optional contact information fields are also available for updates if needed. The design supports efficiency by minimizing data re-entry while ensuring information stays accurate and up to date. At the bottom, the Close and Save Changes buttons give clear options to either exit without changes or confirm updates. The reasoning behind this design focuses on consistency, clarity, and streamlined editing to maintain professional records effectively.

4.4.17.4 Delete Healthcare Professional Modal

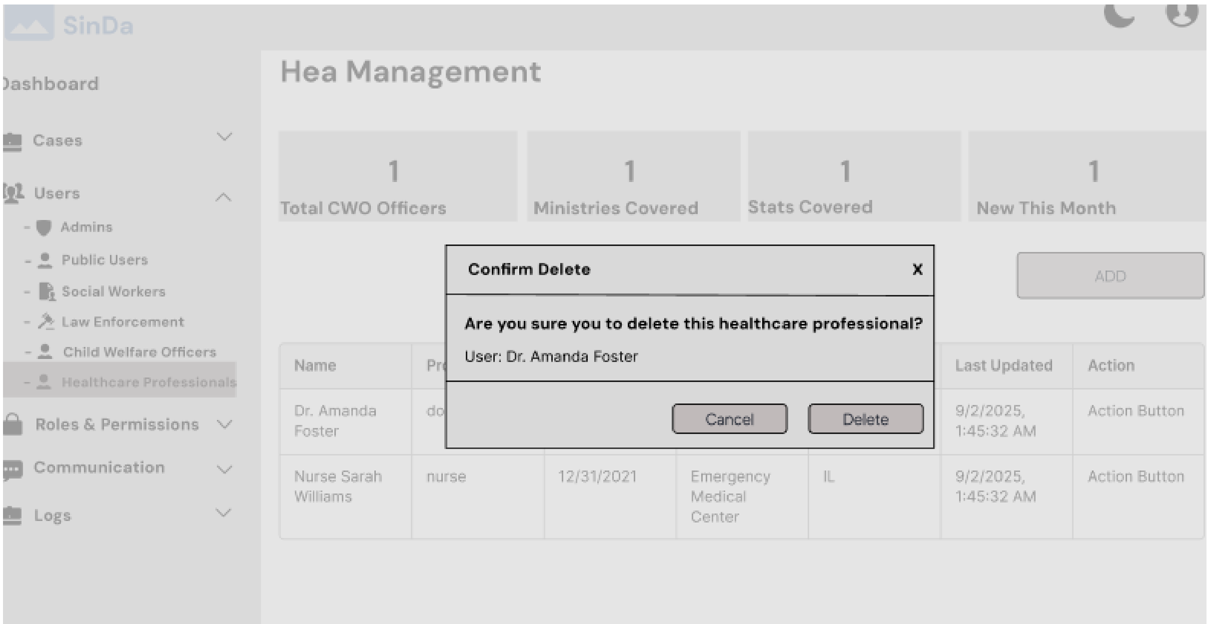


Figure 197: Delete Healthcare Professional Modal

This wireframe shows the Confirm Delete modal for a healthcare professional, which appears when an administrator initiates a deletion. The confirmation message specifies the role and displays the professional’s name to prevent mistakes. The layout is simple and focused, with two clear action buttons: Cancel to safely back out and Delete to confirm the removal. The centered design and bold text draw attention to the seriousness of the action, ensuring that administrators double-check before proceeding. The reasoning behind this design emphasizes clarity, error prevention, and accountability when handling sensitive healthcare user records.

4.4.18 Role and Permission Management

4.4.18.1 Role Management Page

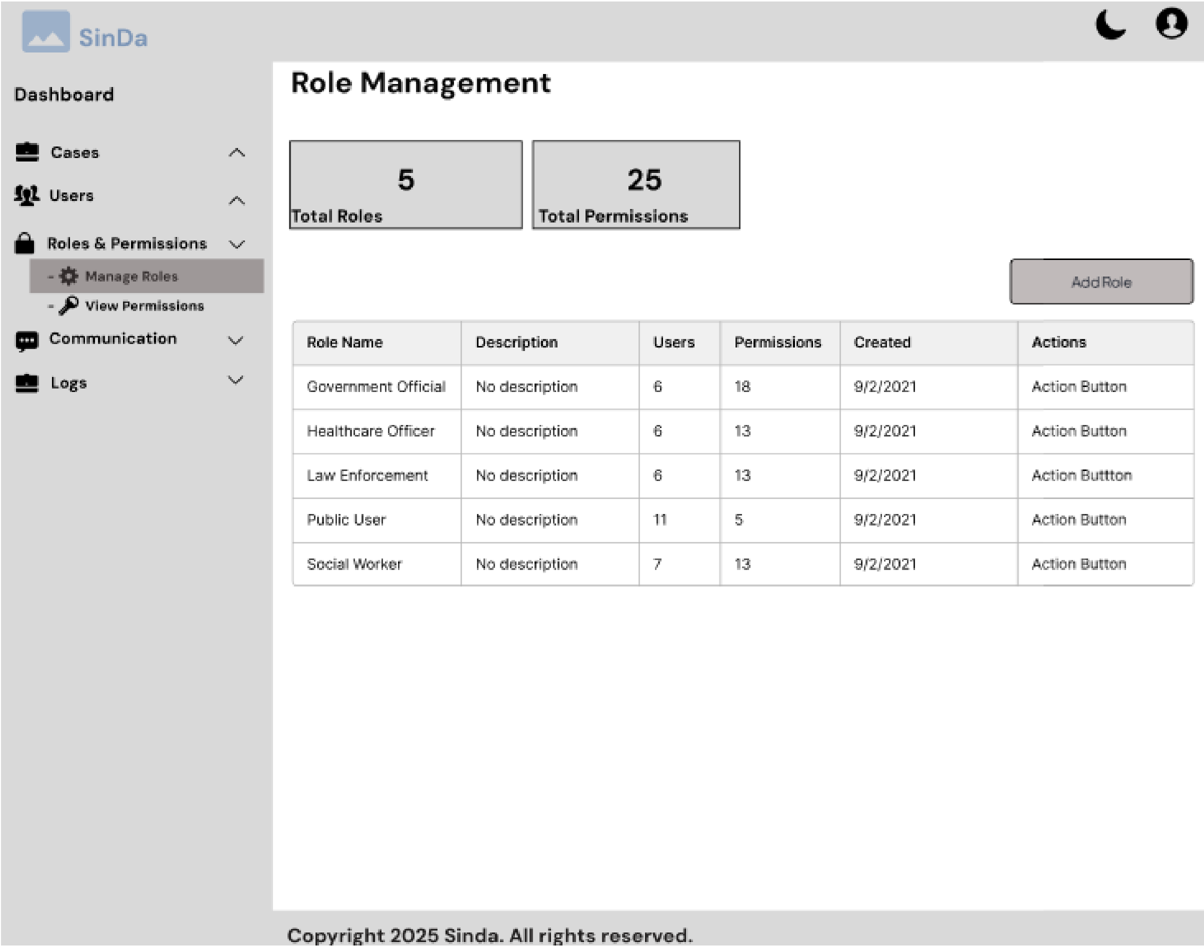


Figure 198: Role Management Page

This wireframe shows the Role Management page, which allows administrators to oversee system roles and their permissions. At the top, two summary cards provide key insights: the total number of roles and the total number of permissions, giving a quick overview of the and an Actions column for managing each role such as editing, updating permissions, or deleting. The Add Role button at the top-right enables administrators to quickly create new roles, ensuring flexibility as the system grows. The clean tabular layout organizes all critical details briefly, while summary cards provide a high-level snapshot for decision-making. The design reasoning emphasizes both clarity and control administrators can easily track how many users are linked to each role, verify permission counts, and adjust efficiently. This balance of overview and detailed management helps maintain a secure and structured role-based access system.

4.4.18.2 Add Role Modal

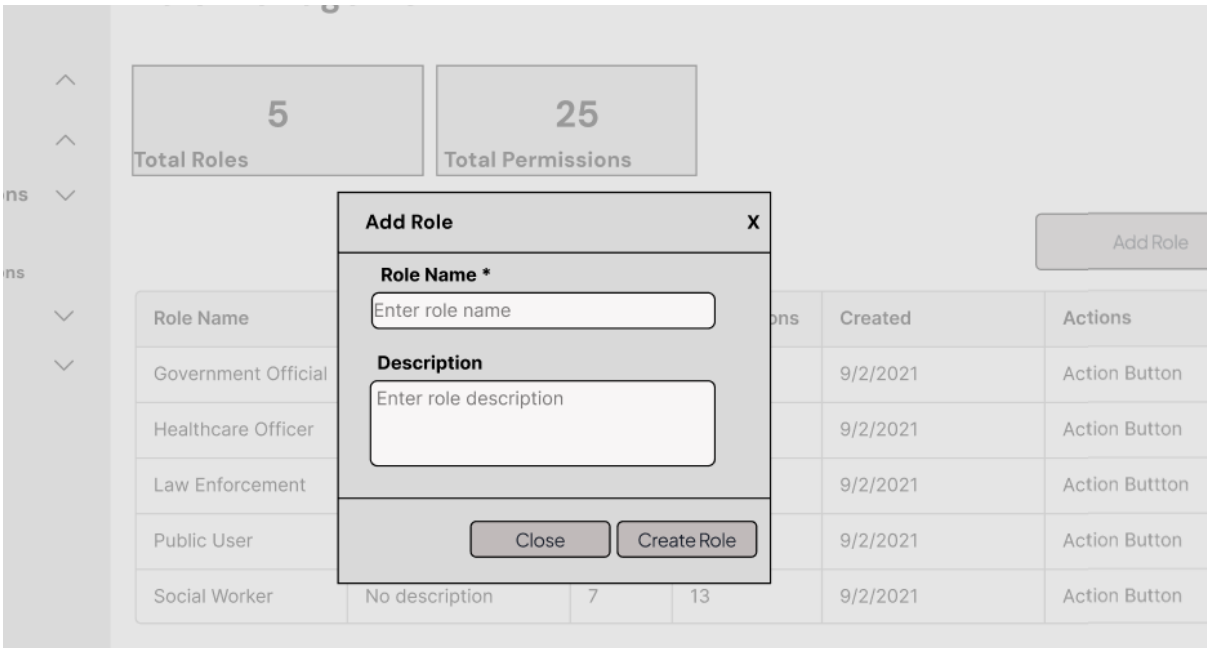


Figure 199: Add Role Modal

The Add Role modal, which enables administrators to establish new system roles, is depicted in this wireframe. The form has two basic fields, Role Name, which is necessary to specify the role's title, and Description, which may optionally offer background information regarding the role's duties or goals. There are two action buttons at the bottom: Add Role to validate creation and Close to quit without making any changes. Administrators may add responsibilities fast and without needless steps because to the layout's emphasis and compactness. Because of the design reasoning's emphasis on efficiency, clarity, and simplicity, expanding the role structure as organisational demands change is simple.

4.4.18.3 Edit Role Modal

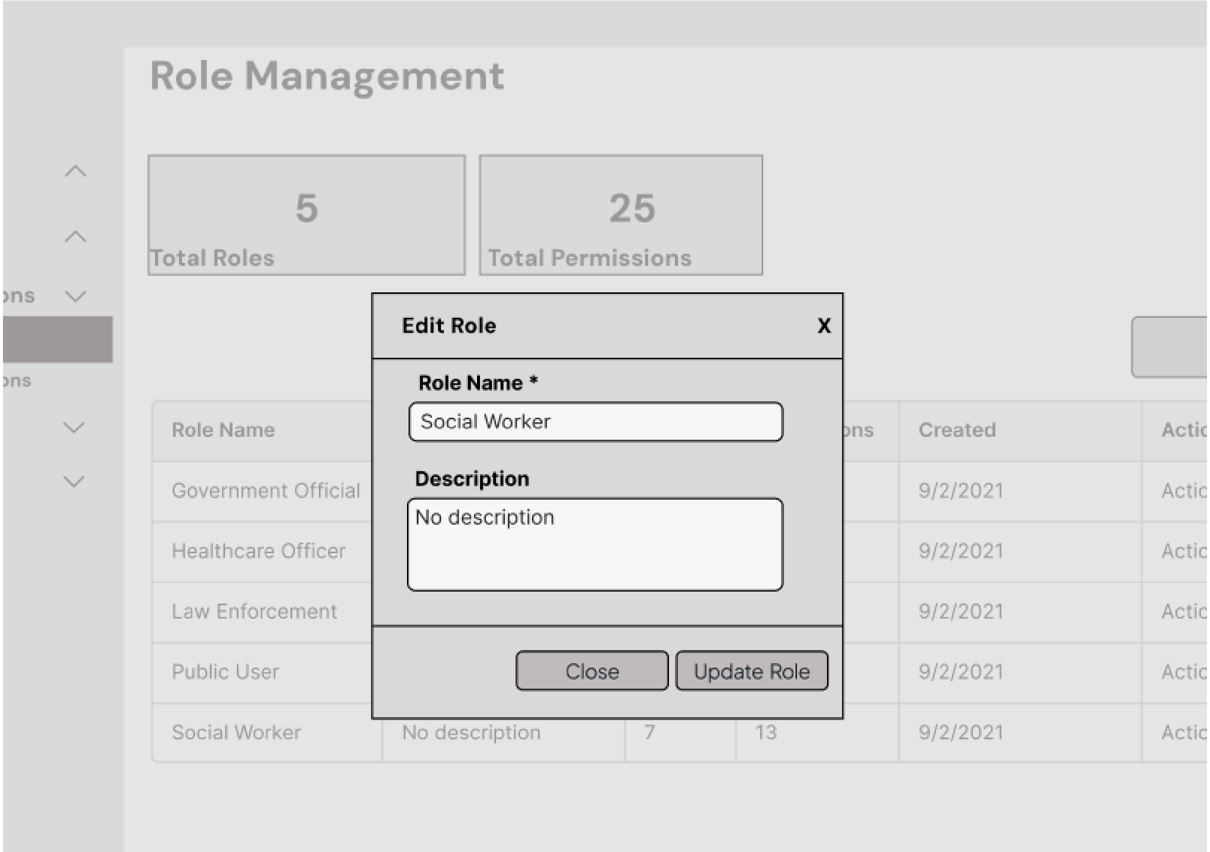


Figure 200: The Edit Role Modal

The Edit Role modal, seen in this wireframe, enables administrators to modify the specifics of an existing system role. The form has two fields: Role Name, which is a mandatory area that is filled up with the current role name, and Description, which is where you may add or change more information about the position's purpose. Two action buttons at the bottom offer choices: Update Role to validate and save revisions, and Close to quit without making any changes. Re-entry is decreased, correctness is guaranteed, and rapid revisions to job descriptions are supported by the straightforward, pre-filled architecture. By emphasising uniformity, effectiveness, and clarity, the design reasoning aids administrators in upholding a flexible and well-organised role structure.

4.4.18.4 Delete Role Modal

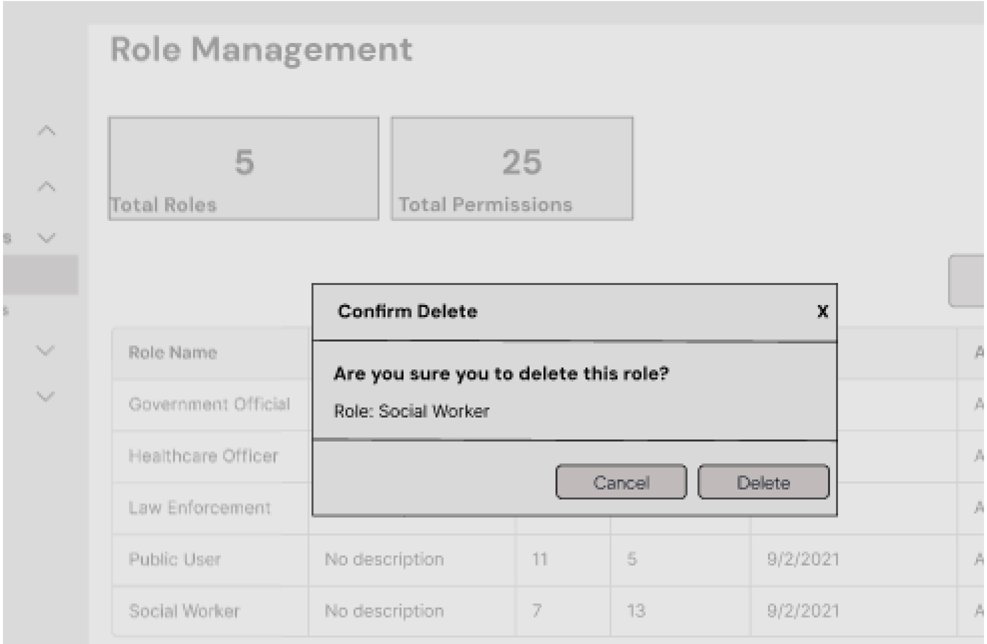


Figure 201: Delete Role Modal

The Confirm Delete modal for a role in the Role Management module is displayed in this wireframe. To avoid unintentional deletions, the confirmation message clearly asks the administrator if they are certain they want to remove the position and shows the role name. There are two options at the bottom: Delete to confirm and continue and cancel to securely end the activity without making any modifications. The simple, centred design emphasises how important the action is, making sure administrators look again before making permanent changes. When handling system-wide responsibilities that affect permissions and user access, the design logic places a strong emphasis on responsibility, clarity, and mistake prevention.

4.4.18.5 Role Details Page

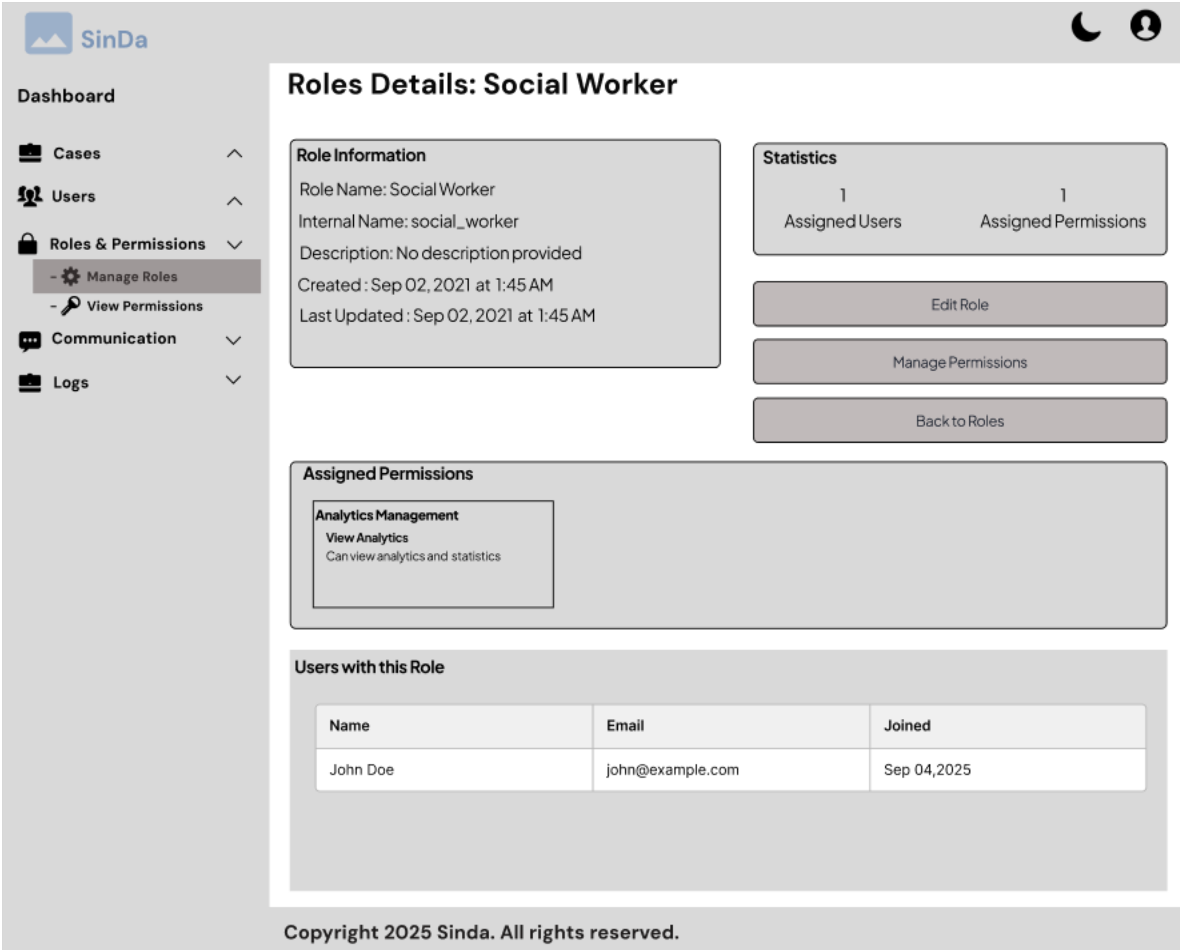


Figure 202: Role Details Page

This wireframe shows the Role Details page which provides administrators with a detailed overview of the role's configuration and assignments. On the left, the Role Information panel displays key details including role name, internal name, description, and timestamps for creation and last update, ensuring traceability. On the right, the Statistics panel summarizes the number of assigned users and permissions. Below, two sections break down specifics. Assigned Permissions lists the permissions linked to the role, for example, Analytics Management to View Analytics, while Users with this Role lists members currently assigned, showing their name, email, and date joined. At the top-right, quick-action buttons, Edit Role, Manage Permissions, and Back to Roles which offer direct management functions without navigating away. The design reasoning emphasizes transparency, quick reference, and ease of management, allowing administrators to see both high-level and detailed role information in one centralized view.

4.4.18.6 Assign Permission Page

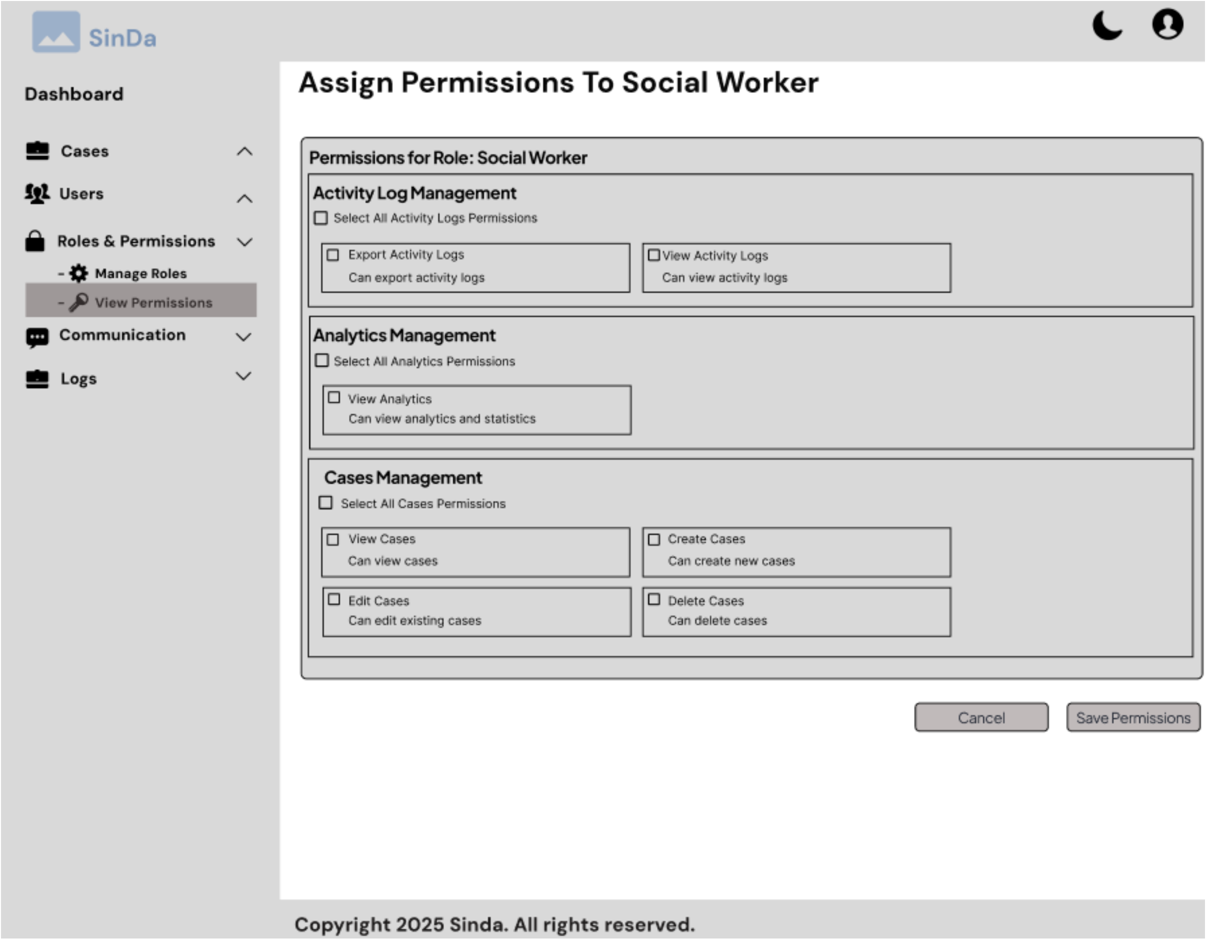


Figure 203: Assign Permission

This wireframe shows the Assign Permissions page, where administrators can configure specific access rights for a selected role. The layout is organized into three main permission groups: Activity Log Management, Analytics Management, and Cases Management. Each group provides checkboxes for individual permissions such as exporting logs, viewing analytics, creating or editing cases, and includes a “Select All” option for quickly granting all permissions within that category. Short descriptions beneath each option clarify the exact capability it grants. At the bottom, two action buttons, Cancel and Save Permissions allow the administrator to either discard or confirm changes. The design focuses on clarity and efficiency, enabling fine-grained control while also supporting bulk assignment for faster configuration.

4.4.18.7 Manage Permission Page

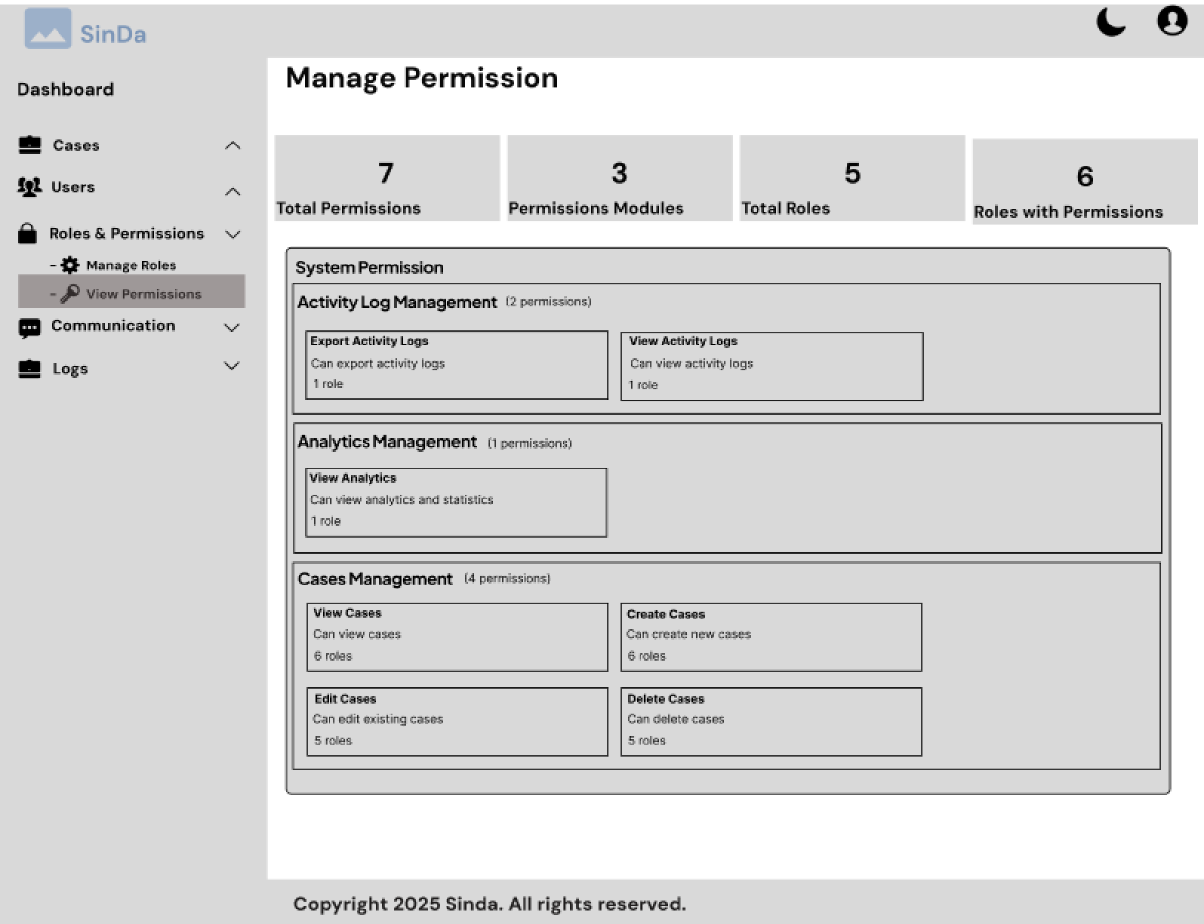


Figure 204: Manage Permission Page

The Manage Permission page, which gives administrators a centralised view of all system permissions, is depicted in this wireframe. Key information including the overall number of permissions, permission modules, roles, and the number of roles that presently have permissions assigned are highlighted at the top of summary cards. The modules for Activity Log Management, Analytics Management, and Cases Management are where permissions are arranged below. A brief explanation of the action and the number of roles that presently possess the permission are listed for each module's specific permissions such as export activity logs, view analytics, create cases, modify cases, etc. Administrators can swiftly examine how rights are distributed across roles, spot overlaps, and preserve system transparency thanks to the organised layout. The design supports effective permission management by emphasising both high-level insights and in-depth breakdowns.

4.4.19 Contact Queries Management

4.4.19.1 Contact Queries Management Page

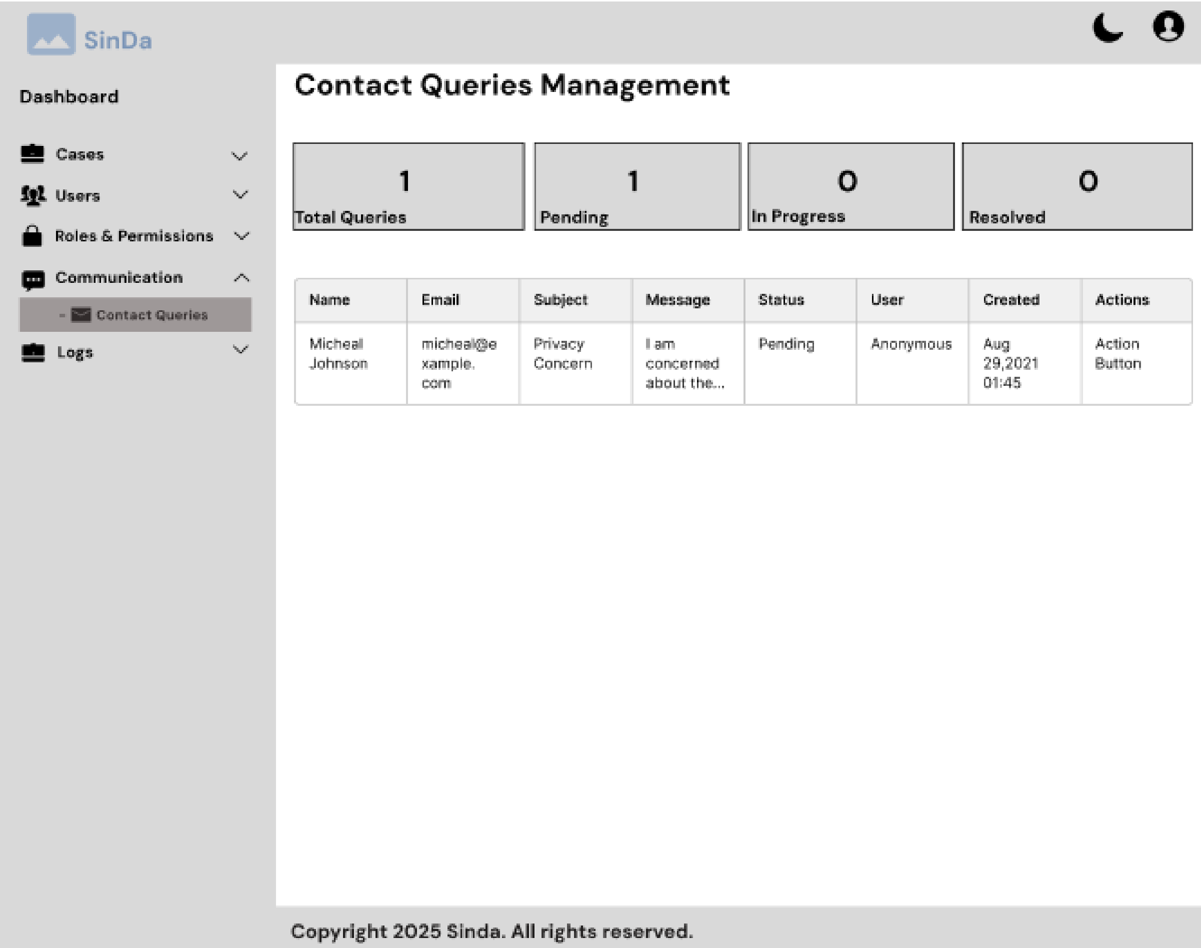


Figure 205: Contact Queries Management Page

The Contact Queries Management page, seen in this wireframe, enables administrators to examine and respond to incoming contact enquiries. Key information, such as the overall number of enquiries, those that are pending, in process, and resolved, are shown at the top of summary cards. With columns for the sender's name, email, topic, message excerpt, status, user type (such as anonymous), the date produced, and an actions column with a query management button, the table below displays individual enquiries. To ensure that administrators can prioritise and respond effectively, the structure offers both a concise summary of the overall query statuses and comprehensive information for each submission. Transparent communication management is supported by the design's emphasis on clarity and organised case handling.

4.4.19.2 Contact Queries Details Page

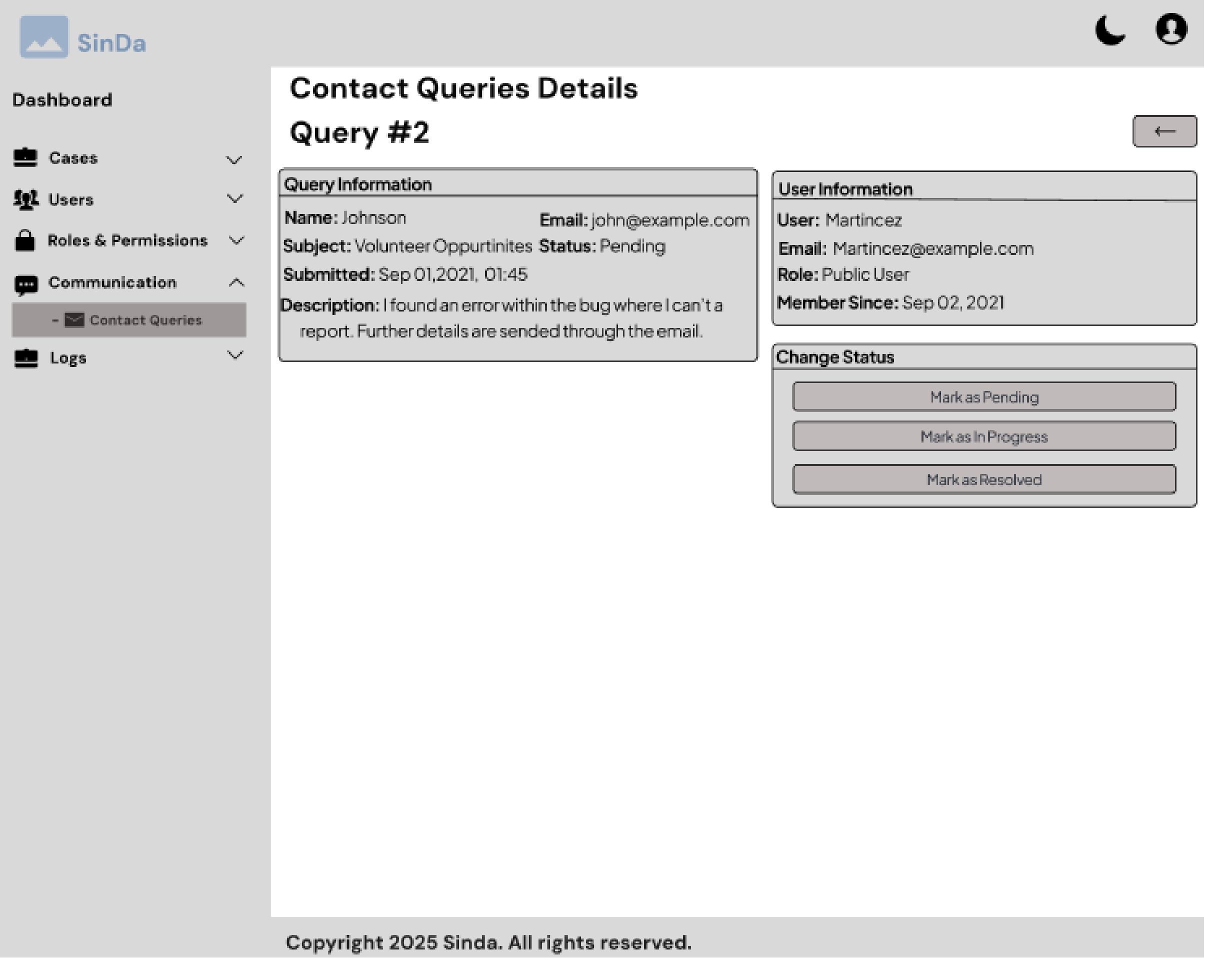


Figure 206: Contact Queries Details Page

The Contact Queries Details page, which gives administrators comprehensive details on a specific inquiry, is displayed in this wireframe. There are three sections to the layout. Details including the sender's name, email address, subject, status, submission date, and a thorough explanation of the question are shown in the question Information panel on the left. The User Information panel on the right displays the username, email address, role, and membership date of the registered user. Quick-action buttons to update the query status (Pending, In Progress, or Resolved) are available beneath that in the Change Status panel. Easy access to the previous list view is made possible by a back button at the top. The design makes query handling simple and effective by guaranteeing administrators may examine context, comprehend the related user, and respond right away.

4.4.19.3 Delete Contact Quary Modal

Figure 207: Contact Quary Modal

When an administrator tries to delete a contact query from the system, the Confirm Delete modal occurs, as seen in this wireframe. To prevent errors, the modal displays a clear warning message such as "Query: Privacy Concern" requesting confirmation before deletion. It also includes the precise question subject. There are two action buttons: Delete to confirm and remove the inquiry and Cancel to exit the modal without making any changes. By avoiding unintentional deletions and promoting responsibility in handling private user conversations, the design guarantees thoughtful decision-making.

4.4.20 Session Logs Page

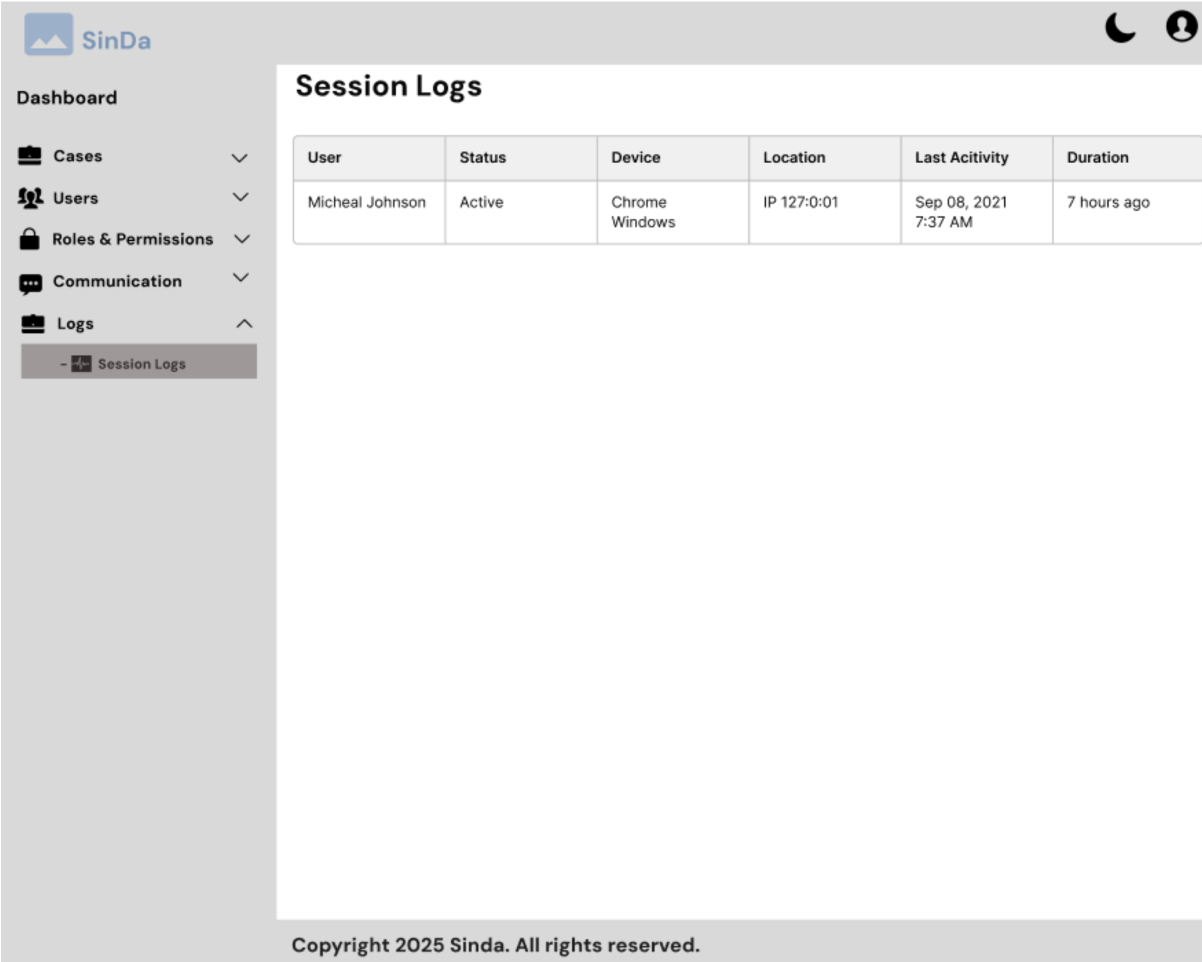


Figure 208: Session Logs Page

The Session Logs page, which monitors user behaviour within the system for security and transparency, is displayed in this wireframe. A table including the user's name, session status, device utilised, location (IP address), last activity timestamp, and session length is displayed on the site. This enables system administrators to keep an eye on ongoing or previous sessions, spot any odd logins, and guarantee system use responsibility. With columns that are clearly labelled to offer instant insights into user activity and system access patterns, the design is simple.

4.4.21 Profile Page

4.4.21.1 Edit My Profile Page (Administrator)

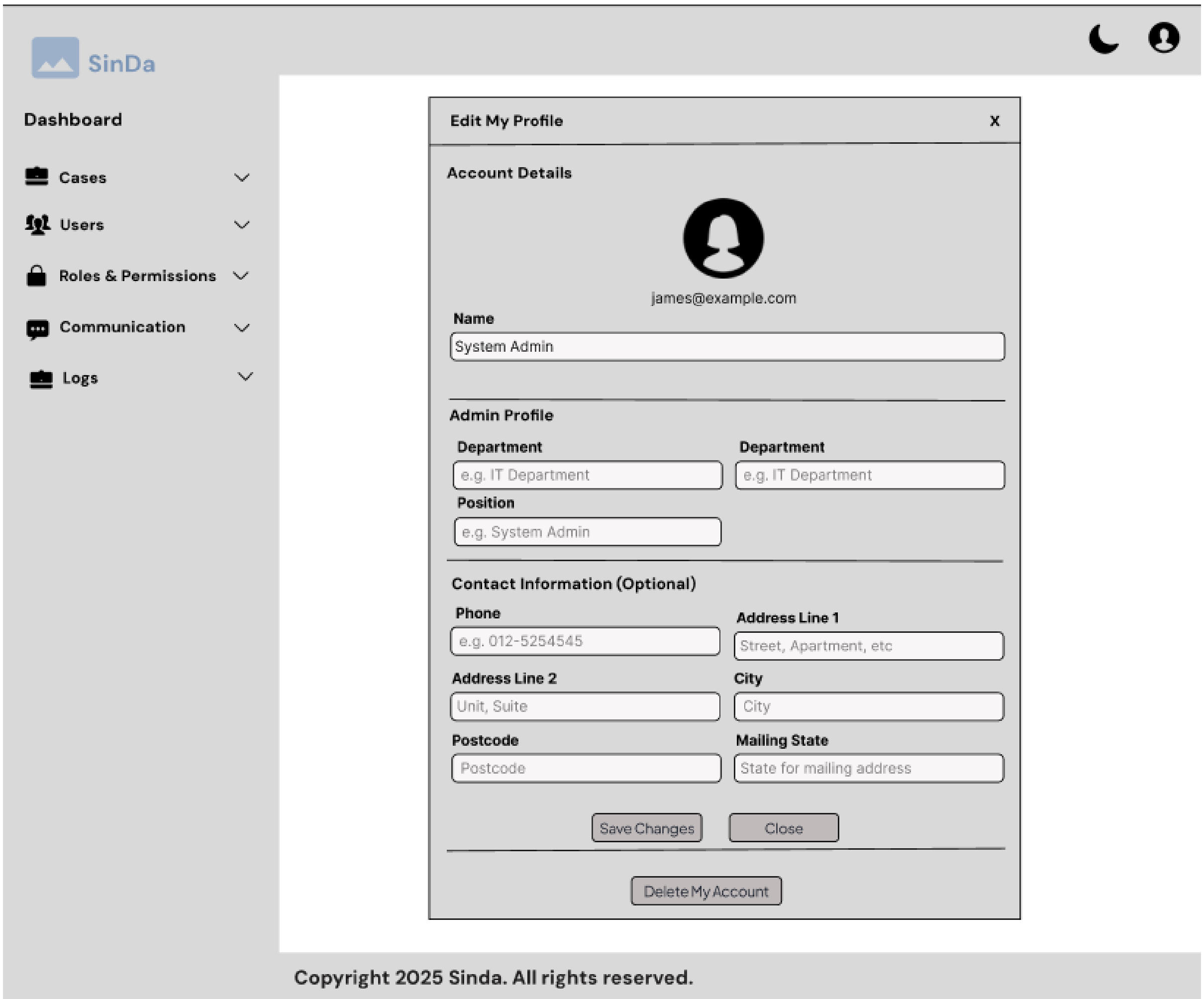


Figure 209: Edit My Profile Page (Administrator)

This wireframe shows the Edit My Profile page for Administrator, which appears when a user clicks the avatar icon and selects "View Profile." At the top, the Account Details section displays the user's email, profile avatar, and a field to update their name. Below, the Admin Profile section allows users to fill in role-related details such as department and position. The Contact Information section provides fields for phone, address, city, postcode, and state, giving flexibility for users to manage personal information. At the bottom, action buttons include Save Changes to confirm updates, Close to exit without saving, and Delete My Account for account removal. The design balances personal customization with administrative details, ensuring users can manage both their role-related profile and optional contact information in one place.

4.4.21.2 Edit My Profile Page (Law Enforcement)

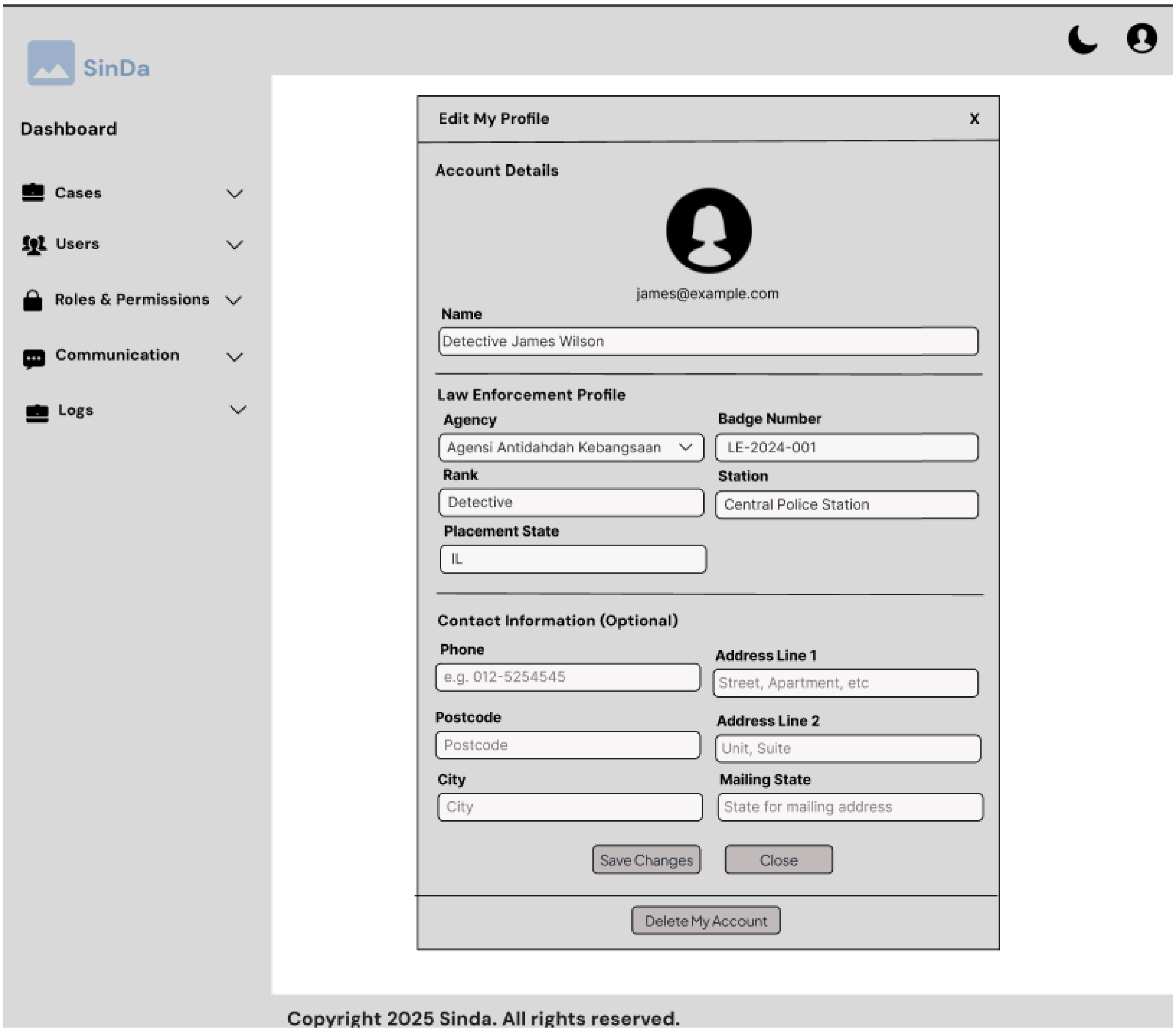


Figure 210: Edit My Profile Page (Law Enforcement)

When a Law Enforcement user clicks the avatar icon and chooses "View Profile," the Edit My Profile page is shown in this wireframe. The Account Details area at the top shows the user's email address, avatar, and a name update field. Professional information including agency, badge number, rank, station, and placement state is recorded in the Law Enforcement Profile section below. This guarantees that role-specific data is appropriately maintained and documented. After that, the user can provide their phone number, address, postcode, city, and state in the Contact Information area for mailing reasons. Action options at the bottom include Delete My Account to remove an account, Close to quit without saving, and Save Changes to amend information. Users may handle both personal and professional data in one location thanks to the design's balance of general account details and specialised job information.

4.4.21.3 Edit My Profile Page (Social Worker)

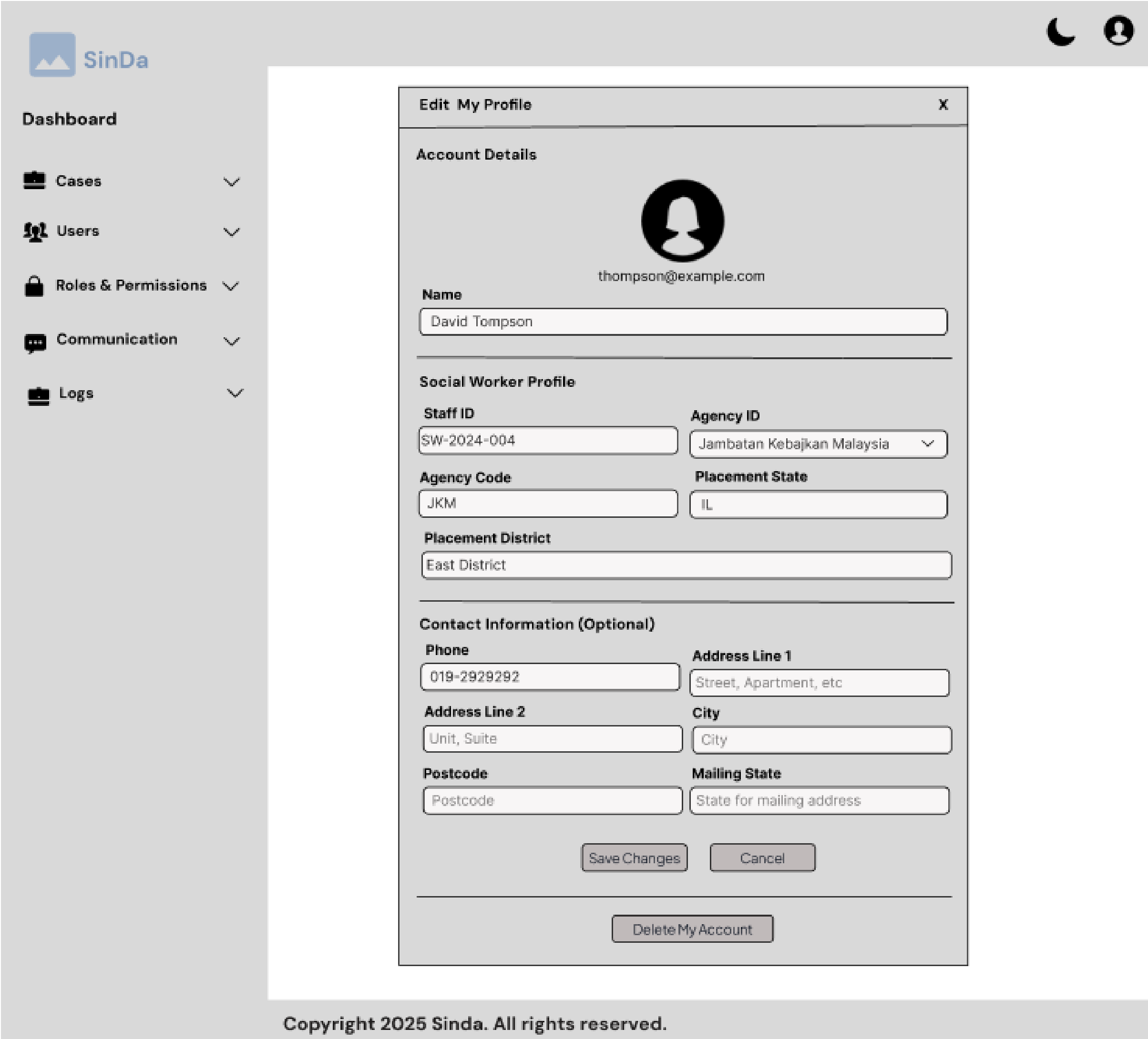


Figure 211: Edit My Profile page (Social Worker)

Above is the wire frame for Social Worker when they view their profile. The top section, Account Details, displays the user's email, avatar, and a field to update their name. Below that, there is a dedicated section for Social Worker Profile, which collects role-specific details such as staff ID, agency ID, agency code, placement state, and placement district. This ensures professional information is properly captured for case management and accountability. Following this, the Contact Information section allows the user to provide personal details like phone number, address, postcode, city, and state. At the bottom, action buttons include Save Changes to confirm updates, Cancel to discard them, and Delete My Account for account removal. The design integrates both general account settings and specialized role data, allowing seamless profile management.

4.4.22 Edit My Profile Page (Government Child Welfare Officer)

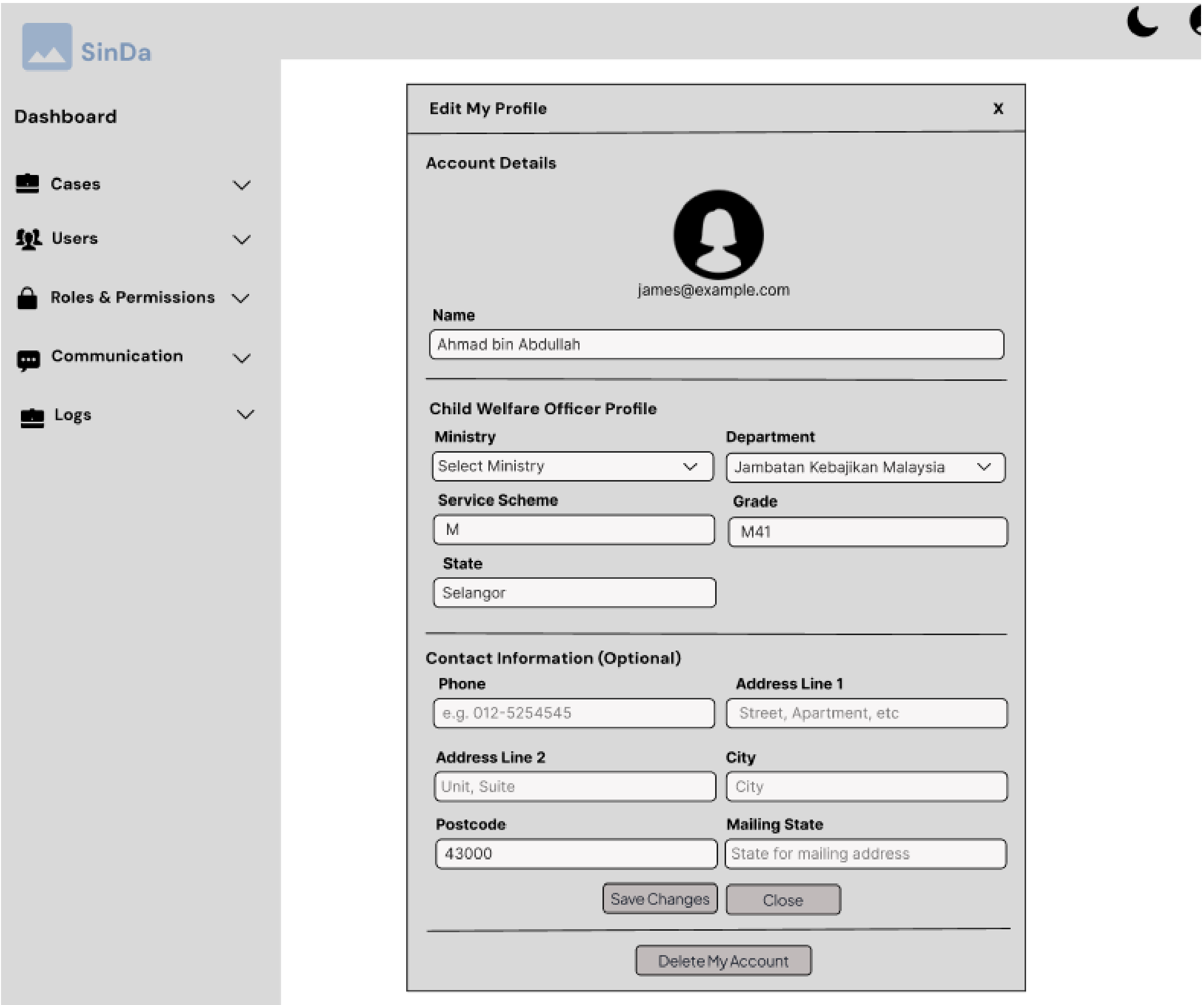


Figure 212: Edit My Profile Page (Government Child Welfare Officer)

This wireframe shows the Edit My Profile page for Child Welfare Officers. At the top, the Account Details section displays the user's email, avatar, and a field for updating their name. Below, there is a dedicated Child Welfare Officer Profile section, which captures role-specific details such as ministry, department, service scheme, grade, and state. This ensures the system stores professional information necessary for structured case management. The Contact Information section follows, allowing the officer to add personal details like phone number, address, postcode, city, and mailing state. At the bottom, action buttons include Save Changes to confirm updates, Close to exit without saving, and Delete My Account for account removal. The design balances general user information with specialized role details, supporting both personalization and accountability.

4.4.23 Edit My Profile Page (Healthcare Professional)

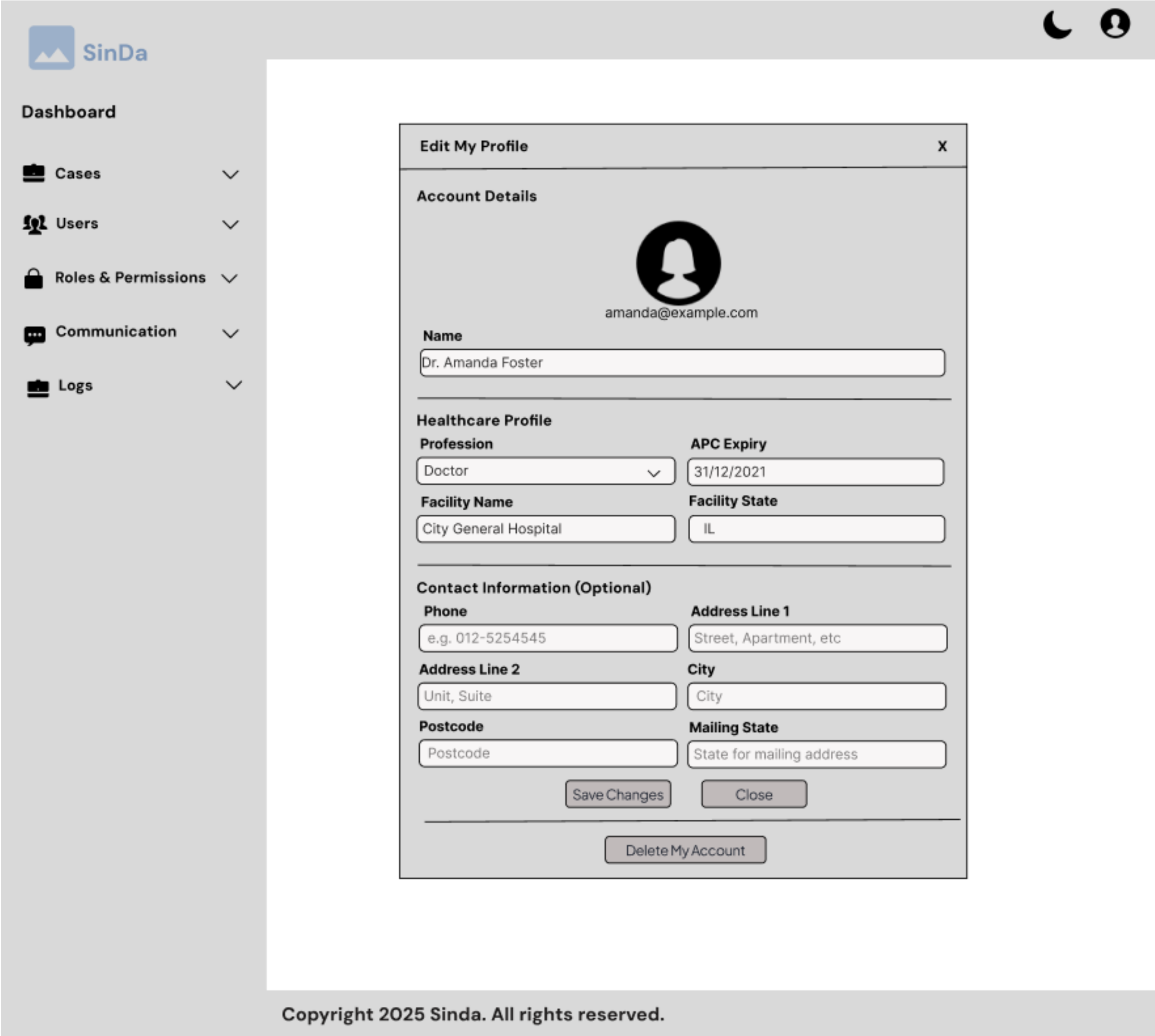


Figure 213: Edit My Profile Page (Healthcare Professional)

This wireframe shows the Edit My Profile page for Healthcare Professionals. Again, the top section, Account Details, displays the user's email, avatar, and a field for updating their name. Below, there is a dedicated Healthcare Profile section where healthcare professionals can enter role-specific details such as profession, APC expiry date, facility name, and facility state. This ensures the system captures important professional and regulatory information. The Contact Information section follows, allowing the user to add personal details such as phone number, address, postcode, city, and mailing state. At the bottom, action buttons include Save Changes to confirm updates, Close to discard them, and Delete My Account for account removal. The design integrates both general account settings and specialized professional data, enabling seamless management of user profiles.

4.4.24 Edit My Profile Page (Public User)

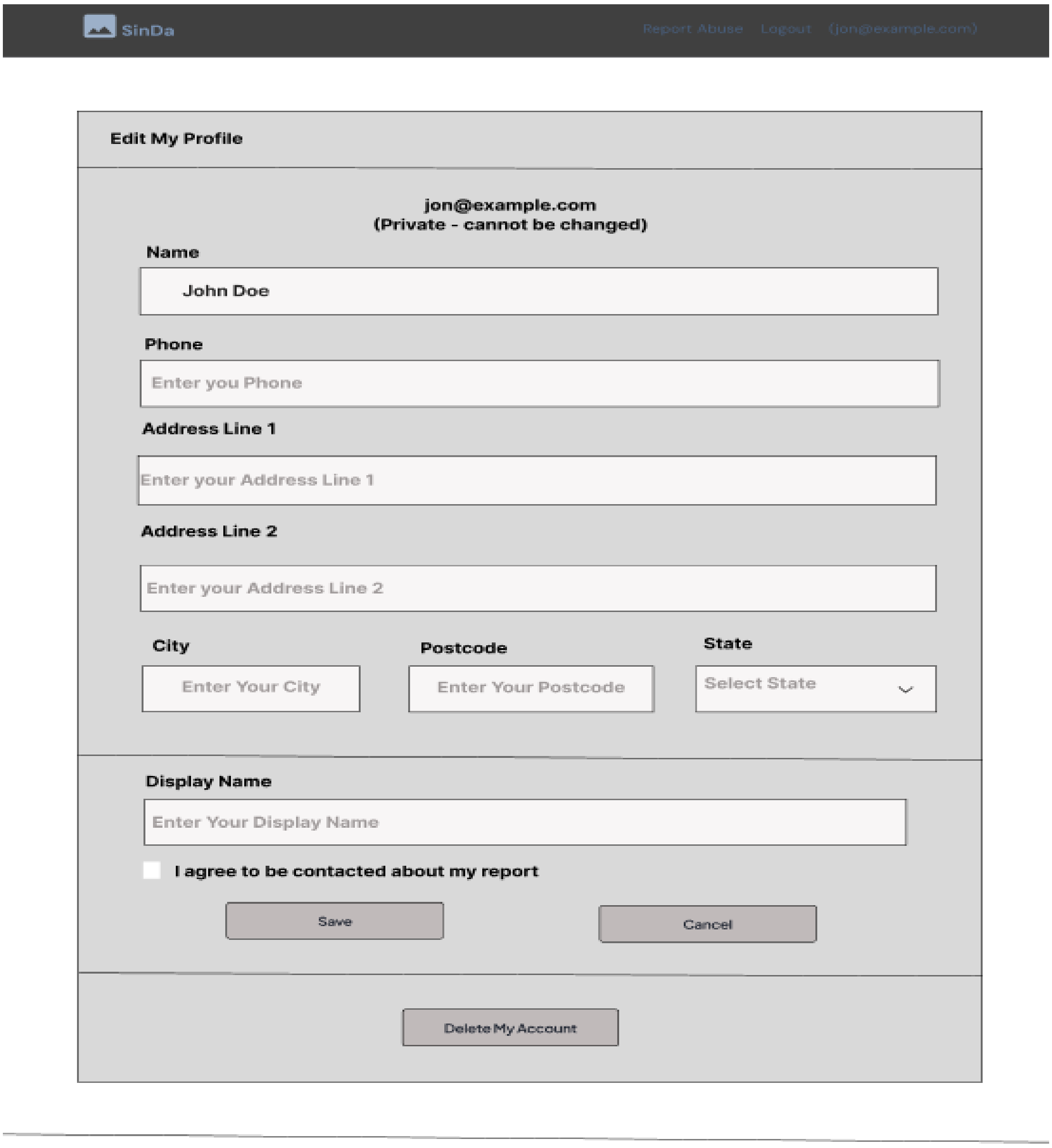


Figure 214: Edit My Profile Page (Public User)

The Edit My Profile page for a public user is displayed in this wireframe. The user's email is shown at the top, but it is tagged as private and cannot be edited. The user can change their name, phone number, address (lines 1 and 2), city, postcode, and state using the editable fields below. A Display Name box in a separate section allows users to customise how their name displays in the system. The user can consent to follow-ups about instances they have reported by checking the "I agree to be contacted about my report" checkbox. Action choices at the bottom include Delete My Account to remove the account, Cancel to delete changes, and Save to confirm alterations. By safeguarding email identity while allowing customisation and communication preferences, the design prioritises user privacy.

4.5 Test Case Preparation

4.5.1 Introduction

To make sure the system is dependable, secure, and operates as intended across all its modules, software testing is an essential step in the development process. Unit testing and user acceptance testing (UAT) are the two primary testing methodologies established for this project. Each has a distinct function in the verification procedure. Individual components like, anonymous reporting, and authentication and others are all subjected to unit testing. Defects can be found early and fixed before integration by separating these modules and executing test cases with both valid and incorrect inputs, which lowers the possibility of more serious system failures. Contrarily, UAT is done at the conclusion of development to assess the system from the viewpoint of the user. To verify that the system fulfils its goals and offers a seamless and safe user experience, testers in various positions engage with real-world scenarios, such as sending an anonymous complaint, gaining access to assigned cases, or posting secure messages. A reliable and trustworthy system that is prepared for deployment in real-world use is the result of unit testing and UAT working together to create a balanced assurance framework. Unit testing guarantees technical accuracy, while UAT verifies practical usability.

4.5.2 Unit Testing

4.5.2.1 Sign In

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid email and password	- Login successful - Redirect to appropriate page		
TC2	User enters invalid email format (e.g., "test@")	- Browser prevents form submission - Display browser validation message		
TC3	User enters wrong password	-Display "Credentials do not match" error		
TC4	User leaves email field empty	- Browser prevents form submission - Display browser validation message		
TC5	User leaves password field empty	- Browser prevents form submission - Display browser validation message		
TC6	User clicks "Remember me" checkbox	-Login successful with remember token set		

TC7	User clicks "Forgot Password" link	-Open password reset modal		
TC8	User clicks "Sign Up" link	-Redirect to registration page		

Table 82: Sign In Unit Testing

4.5.2.2 Reset Password Unit Testing

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User enters valid email in forgot password form	- Send password reset email - Display "Reset link sent" message		
TC2	User enters invalid email format in forgot password	- Display "Enter valid email" error - Keep modal open		
TC3	User enters non-existent email in forgot password	- Display "Credentials do not match" error		
TC4	User clicks reset link from email	- Display password reset form - Show new password and confirm fields		
TC5	User enters matching new passwords (8+ chars)	- Password reset successful - Redirect to login page - Display success message		

TC6	User enters passwords that don't match	- Display "Passwords do not match" error - Stay on reset form		
TC7	User enters password less than 8 characters	- Display "Password must be at least 8 characters" error - Stay on reset form		

Table 83 Reset Password Unit Testing

4.5.2.3 Sign Up

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid username, email, and matching passwords	- Account created successfully - Redirect to login page - Display "Account created successfully" message		
TC2	User enters incomplete email format (e.g., "gtes@"	- Browser prevents form submission - Display browser validation message		
TC3	User enters already registered email	- Display "This email is already registered" error - Keep form data for retry		

TC4	User leaves username field empty	- Browser prevents form submission - Display browser validation message		
TC5	User leaves email field empty	- Browser prevents form submission - Display browser validation message- Display success message		
TC6	User enters passwords that don't match	- Display "Passwords do not match" error - Stay on reset form		
TC7	User enters password less than 8 characters	- Display "Password must be at least 8 characters" error - Stay on reset form		
TC8	User leaves password field empty	- Browser prevents form submission - Display browser validation message		
TC9	User doesn't check Terms & Conditions checkbox	- Browser prevents form submission - Display browser validation message		
TC10	User clicks "Sign in" link	- Redirect to login page -Maintain guest session		

TC11	User clicks "Back to Home" link	- Redirect to landing page - Maintain guest session		

Table 84: Sign Up Unit Test

4.5.2.4 Create Report

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid username, email, and matching passwords	- Report submitted successfully - Display "Your report has been submitted successfully" message		
TC2	User leaves victim age field empty	- Browser prevents form submission - Display browser validation message		
TC3	User enters victim age greater than 17	- Browser prevents form submission - Display "Please enter an age between 0 and 17" message		
TC4	User doesn't select victim gender	- Browser prevents form submission - Display browser validation message		

TC5	User doesn't select any abuse type	- Display "Please select at least one type of abuse" error - Stay on form		
TC6	User leaves incident description empty	- Browser prevents form submission - Display browser validation message		
TC7	User leaves incident location empty	- Browser prevents form submission - Display browser validation message		
TC8	User leaves incident date empty	- Browser prevents form submission - Display browser validation message		
TC9	User doesn't check confirmation checkbox	- Browser prevents form submission - Display browser validation message		
TC10	User uploads more than 5 files	- Display "Maximum 5 files allowed" error - Clear file selection		
TC11	User uploads invalid file type	- Display file type validation error - Clear file selection		

TC12	User uploads file larger than 20MB	- Display file size validation error - Clear file selection		
TC13	User enters incomplete email format (e.g., "gtes@"	- Browser prevents form submission - Display browser validation message		
TC14	User clicks Reset button	- Clear all form fields - Reset to initial state		
TC15	Logged-in user accesses report form	- Pre-fill name, email, phone from profile - Make email field readonly		

Table 85: Create Report

4.5.2.5 Create Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User fills all required fields and submits valid query	- Query submitted successfully - Display "Thank you for your message. We will get back to you soon!" message - Redirect to same page		

TC2	User leaves name field empty	- Browser prevents form submission - Display browser validation message		
TC3	User enters invalid email format (e.g., "test@")	- Browser prevents form submission - Display browser validation message		
TC4	User leaves email field empty	- Browser prevents form submission - Display browser validation message		
TC5	User leaves subject field empty	- Browser prevents form submission - Display browser validation message		
TC6	User leaves message field empty	- Browser prevents form submission - Display browser validation message		
TC7	User enters subject with more than 15 words (e.g., 20 word sentence)	- Browser prevents form submission - Display browser validation message		
TC8	User enters message with more than 150 words (e.g., 200 word message)	- Browser prevents form submission - Display browser validation message		

Table 86: Create Query

4.5.2.6 View Submitted Report

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	Logged-in user accesses "My Reports" page with existing reports	- Display reports table with all user's reports		
TC2	Logged-in user accesses "My Reports" page with no reports	- Display "No Reports Found" message - Show "Create Your First Report" button		
TC3	User clicks "View Details" button on a report	- Open modal with full report details Display victim info, incident details, abuse types - Show assignees and timeline information		
TC4	User clicks "Export Report" button	- Generate PDF download - Display "PDF generated successfully" message - Download file with proper naming convention		

TC5	User searches for specific report in table	- Filter table results based on search term - Update result count		
TC6	User closes report detail modal	- Close modal and return to table view - Maintain table state and position - Preserve current filters/sort		

Table 87: View Submitted Report

4.5.2.7 Authorized users Dashboard

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	Administrator user accesses dashboard	- Display "SinDa Admin Dashboard" heading - Show all 4 statistics cards (Total Reports, Active Cases, Total Users, Contact Queries) - Display "Real-time overview of system activity and trends"		
TC2	Government Child Welfare Officer accesses dashboard	- Display "SinDa Government Official Dashboard" heading		

		- Show all 4 statistics cards - Display "Overview of all cases and system activity"		
TC3	Social Worker accesses dashboard	- Display "SinDa Social Worker Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases		
TC4	Law Enforcement accesses dashboard	- Display "SinDa Law Enforcement Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases		
TC5	Healthcare Professional accesses dashboard	-Display "SinDa Healthcare Professional Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases		
TC6	User clicks "Export Reports"	- Export dropdown menu opens - Shows CSV Export and PDF Export options		

		- Displays available export types (All Cases, All Users, All Queries, Session Logs)		
TC7	Dashboard shows recent reports table	- Display table with latest 5 reports - Show reporter name, description, status - Include timestamps		
TC8	Dashboard handles empty data	- Show "No case data available" message - Display appropriate empty states - Handle zero counts gracefully		

Table 88: Authorized user Dashboard

4.5.2.8 Case Management

4.5.2.8.1 View Cases

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses case management page with different roles	- **Administrator/Child Welfare Officer**: Page loads with All cases and statistics		

		- **Social Worker/Law Enforcement/Healthcare**: Page loads with only assigned cases and filtered statistics - Data table shows cases based on user role		
TC2	User views case data table with existing cases	- Data table displays cases based on user role		
TC3	User views case data table with no accessible cases	- Empty data table displayed - "No data available" message shown		
TC4	User searches case by reporter name in data table (e.g., search "John")	- Filtered results displayed from user's accessible cases - Only matching cases shown		
TC5	User without create permission views case management page	- Page loads successfully - Add Case button not visible		
TC6	User with create permission views case management page	- Page loads successfully - Add Case button visible		
TC7	User with create permission views case	- Add Case modal opens successfully - Modal displays all form fields		

	management page clicks "Add" button			
TC8	User without view history permission views case management page	- Page loads successfully - View History buttons not visible in action dropdown		
TC9	User with view history permission views case management page	- Page loads successfully - View History buttons visible in action dropdown		
TC10	User with view history permission clicks "View History" button	- History modal opens with loading state - Case history data loads and displays		
TC11	User without edit permission views case management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC12	User with edit permission views case management page	- Page loads successfully - Edit button visible in action dropdown		
TC13	User with edit permission clicks "Edit" button	- Edit modal opens with loading state - Case data loads and populates form fields		

		- All form fields are editable and pre-filled		
TC14	User without delete permission views case management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC15	User with delete permission views case management page	- Page loads successfully - Delete buttons visible in action dropdown		
TC16	User with delete permission clicks "Delete" button	- Delete confirmation modal opens - Case ID and reporter name displayed		
TC17	User clicks "Cancel" button in delete modal	- Modal closes without action - Case remains in system - Return to cases list view		
TC18	Admin clicks "Delete Case" button in modal	- Case deleted successfully - Display "Case deleted successfully" message - Remove case from list		
TC19	User without export permission views case management page	- Page loads successfully - Export PDF buttons not visible in action dropdown		

TC20	User with export permission clicks "Export PDF" button	- PDF generation starts with loading state - PDF file downloads automatically		

Table 89: View Cases

4.5.2.8.2 Create Case

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Add Case" button	- Add Case modal opens - Form displays with all fields		
TC2	User fills all required fields and submits	- Case created successfully - Success message displayed - Case appears in table		
TC3	User submits form with missing reporter name	- Browser prevents form submission - Display browser validation message		
TC4	User submits form with missing reporter email	- Browser prevents form submission - Display browser validation message		

TC5	User submits form with invalid email format (e.g., "john@")	- Browser prevents form submission - Display browser validation message		
TC6	User submits form with missing incident description	- Browser prevents form submission - Display browser validation message		
TC7	User submits form with missing incident location	- Browser prevents form submission - Display browser validation message		
TC8	User submits form with missing incident date	- Browser prevents form submission - Display browser validation message		
TC9	User submits form with missing status	- Browser prevents form submission - Display browser validation message		
TC10	User submits form with missing priority	- Browser prevents form submission - Display browser validation message		
TC11	User uploads more than 5 files (e.g., selects 6 images)	"You can only select up to 5 files" error message displayed - Files cleared from input		

Table 90: Create Case

4.5.2.8.3 Edit Case Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User with edit permission clicks "Edit" button	- Edit modal opens with loading state - Case data loads and populates form fields		
TC2	User successfully updates case with valid data	- Case updated successfully - Success message displayed - Changes reflected in data table		
TC3	User submits form with missing reporter name	- Browser prevents form submission - Display browser validation message		
TC4	User submits form with missing reporter email	- Browser prevents form submission - Display browser validation message		
TC5	User submits form with invalid email format (e.g., "john@")	- Browser prevents form submission - Display browser validation message		
TC6	User submits form with missing incident description	- Browser prevents form submission - Display browser validation message		

TC7	User submits form with missing incident location	- Browser prevents form submission - Display browser validation message		
TC8	User submits form with missing incident date	- Browser prevents form submission - Display browser validation message		
TC9	User submits form with missing status	- Browser prevents form submission - Display browser validation message		
TC10	User submits form with missing priority	- Browser prevents form submission - Display browser validation message		
TC11	User uploads more than 5 files (e.g., selects 6 images)	"You can only select up to 5 files" error message displayed - Files cleared from input		
TC12	User creates case and cancels form	- Modal closes without saving - No case created - Form data cleared - No success/error messages		

Table 91: Edit Case Details

4.5.2.8.4 View Case Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "View Case Details" button	- Case details page loads successfully - All case information displays correctly		
TC2	Admin user views case details page	- Full case information displayed - All action buttons visible (Edit, Export, History)		
TC3	Government Child Welfare Officer views any case details	- Full case information displayed - All administrative functions available		
TC4	User views case with evidence files	- Evidence section displays correctly - File viewer modal opens when clicking evidence		
TC5	User clicks "View History" button	- Case history modal opens - Historical actions displayed chronologically		
TC6	User clicks "Export Case" button	- PDF download starts automatically		

TC7	User clicks "Edit Case" button	- Edit Case modal opens with loading state - Case data loads and populates form fields		
TC8	User clicks “Message”	- Page smoothly scrolls to messages section		
TC9	User views case with no assignees and checks messages section	- "No Assignees Found" displayed - Message composition form disabled		
TC10	User views case with assignees and no messages	- "No messages yet" displayed - "Start the conversation by sending the first message" shown		
TC11	User views case with existing messages	- All messages displayed chronologically - Sender names and timestamps shown		
TC12	User searches for messages using search bar	- Search results filtered correctly - Clear search button appears		

Table 92: View Case Details

4.5.2.9 Admin Management

4.5.2.9.1 View Admins

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses admin management page	- Page loads successfully with admin dashboard - Statistics cards display (Total Admins, Active Admins, Inactive Admins, New Admins) - Data table shows admin users list - Add button is visible		
TC2	User views admin data table with no admin users	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches admin by name in data table (e.g., search "John")	- Filtered results displayed - Only matching admins shown		
TC4	User without create permission views admin management page	- Page loads successfully - Add button not visible		

TC5	User with create permission views case management page	- Page loads successfully - Add button visible		
TC6	User without edit permission views admin management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC7	User with edit permission views admin management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views admin management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC9	User with delete permission views admin management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 93: View Admins

4.5.2.9.2 Create Admin

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User submits admin creation form with all required fields filled correctly	- Admin user successfully created - Success message displayed - Modal closes automatically - New admin appears in admin list		
TC2	User submits admin creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits admin creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits admin creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits admin creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		
TC6	User submits admin creation form with	- Browser prevents form submission		

	password less than 8 characters	- Display browser validation message: "Please lengthen this text to 8 characters or more"		
TC7	User submits public user creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC8	User submits admin creation form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC9	User submits admin creation form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User creates admin and cancels form by clicking 'Close' button	- Modal closes without saving - No admin user created - Form data cleared		

Table 94: Create Admin

4.5.2.9.3 Edit Admin Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User submits edit admin form with all required fields updated correctly required fields filled correctly	- Admin user successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in admin list		
TC2	User submits edit admin form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit admin form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits edit admin form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User edits admin and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to admin data - Form data cleared		
TC6	User uploads new avatar	- Avatar previews review the selected avatar		

Table 95: Edit Admin Details

4.5.2.9.4 Delete Admin

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms admin deletion by clicking "Delete" button in modal	- Admin user successfully deleted from database - Success message displayed - Admin removed from admin list - Modal closes automatically		
TC2	User cancels admin deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to admin data - Admin remains in admin list		
TC3	User cancels admin deletion by clicking "Close" (X) button in modal	Modal closes without deleting - No changes made to admin data - Admin remains in admin list		

Table 96: Delete Admin

4.5.2.10 Public User Management

4.5.2.10.1 View Public Users

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User accesses public user management page	- Page loads successfully with public user dashboard - Statistics cards display (Total Users, Contactable Users, Non-Contactable Users, New Users) - Data table shows public users list		
TC2	User views public user data table with no public users	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches public user by name in data table (e.g., search "John")	- Filtered results displayed - Only matching public users shown		
TC4	User without create permission views public user management page	- Page loads successfully - Add button not visible		
TC5	User with create permission views public user management page	- Page loads successfully - Add button visible		

TC6	User without edit permission views public user management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC7	User with edit permission views public user management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views public user management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC7	User with delete permission views public user management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 97: View Public Users

4.5.2.10.2 Create Public User

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits public user creation form with all required fields filled correctly	- Public user successfully created - Success message displayed - Modal closes automatically		

		- New public user appears in public user list		
TC2	User submits public user creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits public user creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits public user creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits public user creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		
TC6	User submits public user creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"		

TC7	User submits public user creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC8	User submits public user creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		
TC9	User creates public user and cancels form by clicking 'Close' button	- Modal closes without saving - No public user created - Form data cleared		

Table 98: Create Public User

4.5.2.10.3 Edit Public User Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit public user form with all required fields updated correctly	-Public user successfully updated - Success message displayed - Modal closes automatically		

		- Updated data reflected in public user list		
TC2	User submits edit public user form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit public user form with missing display name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User edits public user and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to public user data - Form data cleared		
TC5	User edits admin and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to admin data - Form data cleared		

Table 99: Edit Public User Details

4.5.2.10.4 Delete Public User

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User confirms public user deletion by clicking "Delete" button in modal	- Public user successfully deleted from database - Success message displayed - Public user removed from public user list - Modal closes automatically		
TC2	User cancels public user deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to public user data - Public user remains in public user list		
TC3	User cancels public user deletion by clicking "Close" (X) button in modal	Modal closes without deleting - No changes made to public user data - Public user remains in public user list		

Table 100: Delete Public User

4.5.2.11 Social Worker Management

4.5.2.11.1 View Social Workers

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User accesses social worker management page	- Page loads successfully with social worker dashboard - Statistics cards display (Total Workers, Agencies Covered, States Covered, New Users) - Data table shows social worker users list		
TC2	User views social worker data table with no social workers	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches social worker by name in data table (e.g., search "John")	- Filtered results displayed - Only matching social workers shown		
TC4	User without create permission views social worker management page	- Page loads successfully - Add button not visible		

TC5	User with create permission views social worker management page	- Page loads successfully - Add button visible		
TC6	User without edit permission views social worker management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC7	User with edit permission views social worker management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views social worker management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC9	User with delete permission views social worker management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 101: View Social Workers

4.5.2.11.2 Create Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User submits social worker creation form with all required fields filled correctly	- Social worker successfully created - Success message displayed - Modal closes automatically - New social worker appears in social worker list		
TC2	User submits social worker creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits social worker creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits social worker creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits social worker creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		
TC6	User submits social worker creation form with	- Browser prevents form submission		

	password less than 8 characters	- Display browser validation message: "Please lengthen this text to 8 characters or more"		
TC6	User submits public social worker form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC7	User submits social worker creation form with missing staff ID field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC8	User submits social worker creation form with missing agency name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC9	User submits social worker creation form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User submits social worker creation	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

	form with missing placement district field			
TC11	User selects "Other" for agency name but doesn't provide agency name other	- Form submission proceeds but server validation catches this - Display server validation error: "Please enter the agency name when 'Other' is chosen"		
TC12	User creates social worker and cancels form by clicking 'Close' button	- Modal closes without saving - No social worker created - Form data cleared		

Table 102: Create Social Worker

4.5.2.11.3 Edit Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit social worker form with all required fields updated correctly	- Social worker successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in social worker list		

TC2	User submits edit social worker form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit social worker form with missing staff ID field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits edit social worker form with missing agency name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits edit social worker form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC6	User submits edit social worker form with missing placement district field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC7	User edits social worker and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to social worker data - Form data cleared		
TC8	User uploads new avatar	- Avatar previews the selected avatar		

TC9	User updates agency information (agency name, agency code)	- Agency information successfully updated - Success message displayed - Updated agency information reflected in social worker profile		
TC10	User updates placement information (state, district)	Placement information successfully updated - Success message displayed - Updated placement information reflected in social worker profile		

Table 103: Edit Social Worker

4.5.2.11.4 Delete Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms social worker deletion by clicking "Delete" button in modal	- Social worker user successfully deleted from database - Success message displayed - Social worker removed from social worker list - Modal closes automatically		

TC2	User cancels social worker deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list		
TC3	User cancels social worker deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list		

Table 104: Delete Social Worker

4.5.2.12 Law Enforcement Management

4.5.2.12.1 View Law Enforcements

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses law enforcement management page	- Page loads successfully with law enforcement dashboard - Statistics cards display (Total Officers, Agencies Represented, Stations Covered, New Users)		

		- Data table shows law enforcement users list		
TC2	User views law enforcement data table with no law enforcement officers	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches law enforcement by name in data table (e.g., search "John")	- Filtered results displayed - Only matching law enforcement officers shown		
TC4	User without create permission views law enforcement management page	- Page loads successfully - Add button not visible		
TC5	User with create permission views law enforcement management page	- Page loads successfully - Add button visible		
TC6	User without edit permission views law	- Page loads successfully - Edit buttons not visible in action dropdown		

	enforcement management page			
TC7	User with edit permission views law enforcement management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views law enforcement management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC9	User with delete permission views law enforcement management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 105: View Law Enforcements

4.5.2.12.2 Create Law Enforcement

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits law enforcement creation form	- Law enforcement officer successfully created - Success message displayed		

	with all required fields filled correctly	- Modal closes automatically - New law enforcement officer appears in law enforcement list		
TC2	User submits law enforcement creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits law enforcement creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits law enforcement creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits law enforcement creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		

TC6	User submits law enforcement creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"		
TC7	User submits law enforcement creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC8	User submits law enforcement creation form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC9	User submits law enforcement creation form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User submits law enforcement creation form with missing law enforcement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC11	User submits law enforcement creation form with no agency selected (empty selection)	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC12	User submits law enforcement creation form with duplicate badge number for same agency	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"		
TC13	User creates law enforcement and cancels form by clicking 'Close' button	- Modal closes without saving - No law enforcement officer created - Form data cleared		

Table 106: Create Law Enforcement

4.5.2.12.3 Edit Law Enforcement

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit law enforcement form with all required fields updated correctly	- Law enforcement officer successfully updated - Success message displayed - Modal closes automatically		

		- Updated data reflected in law enforcement list - Updated data reflected in social worker list		
TC2	User submits edit law enforcement form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit law enforcement form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits edit law enforcement form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits edit law enforcement form with missing law enforcement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC6	User submits edit law enforcement form	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

	with no agency selected (empty selection)			
TC7	User submits edit law enforcement form with duplicate badge number for same agency (different user)	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"		
TC8	User edits law enforcement and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to law enforcement data - Form data cleared		
TC9	User uploads new avatar	- Avatar previews the selected avatar		
TC10	User updates agency information (agency, badge number, rank, station)	- Agency information successfully updated - Success message displayed - Updated agency information reflected in law enforcement profile		
TC11	User updates law enforcement state information	- Law enforcement state successfully updated - Success message displayed		

		- Updated state information reflected in law enforcement profile		
TC12	User updates contact information (phone, address, city, state, postcode)	- Contact information successfully updated - Success message displayed - Updated contact information reflected in user profile		

Table 107: Edit Law Enforcement

4.5.2.12.4 Delete Law Enforcement

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms law enforcement deletion by clicking "Delete" button in modal	- Law enforcement user successfully deleted from database - Success message displayed - Law enforcement removed from law enforcement list - Modal closes automatically		
TC2	User cancels law enforcement deletion by	- Modal closes without deleting - No changes made to law enforcement data		

	clicking "Cancel" button in modal	- Law enforcement remains in law enforcement list		
TC3	User cancels law enforcement deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to law enforcement data - Law enforcement remains in law enforcement list		

Table 108: Delete Law Enforcement

4.5.2.13 Child Welfare Officer Management

4.5.2.13.1 View Child Welfare Officers

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses child welfare officer management page	- Page loads successfully with CWO dashboard - Statistics cards display (Total CWO Officers, Ministries Covered, States Covered, New This Month) - Data table shows CWO users list		

TC2	User views CWO data table with no child welfare officers	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches CWO by name in data table (e.g., search "John")	- Filtered results displayed - Only matching law enforcement officers shown		
TC4	User without create permission views CWO management page	- Page loads successfully - Add button not visible		
TC5	User with create permission views CWO management page	- Page loads successfully - Add button visible		
TC6	User without edit permission views CWO management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC7	User with edit permission views CWO management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views CWO management	- Page loads successfully - Delete buttons not visible in action dropdown		

TC9	User with delete permission views CWO management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 109: View CWOs

4.5.2.13.2 Create Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits CWO creation form with all required fields filled correctly	- Child welfare officer successfully created - Success message displayed - Modal closes automatically - New CWO appears in CWO list		
TC2	User submits CWO creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits CWO creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC4	User submits CWO creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits CWO creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"		
TC6	User submits CWO creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"		
TC7	User submits CWO creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC8	User submits CWO creation form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC9	User submits CWO creation form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User submits CWO creation form with missing CWO state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC11	User creates CWO and cancels form by clicking 'Close' button	- Modal closes without saving - No child welfare officer created - Form data cleared		

Table 110: Create CWO

4.5.2.13.3 Edit Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit CWO form with all required fields updated correctly	- Child welfare officer successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in CWO list		

TC2	User submits edit CWO form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit CWO form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits edit CWO form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits edit CWO form with missing CWO state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC6	User edits CWO and cancels form by clicking 'Close' button	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC8	User uploads new avatar	- Avatar previews the selected avatar		
TC9	User updates service scheme and grade information	- Service scheme and grade information successfully updated - Success message displayed - Updated information reflected in CWO profile		

TC10	User updates CWO state information	- CWO state successfully updated - Success message displayed - Updated state information reflected in CWO profile		

Table 111: Edit CWO

4.5.2.13.4 Delete Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms CWO deletion by clicking "Delete" button in modal	- Child welfare officer successfully deleted from database - Success message displayed - CWO removed from CWO list - Modal closes automatically		
TC2	User cancels CWO deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to CWO data - Child welfare officer remains in CWO list		
TC3	User cancels CWO deletion by	- Modal closes without deleting - No changes made to CWO data		

	clicking "Close" (X) button in modal	- Child welfare officer remains in CWO list		

Table 112: Delete CWO

4.5.2.14 Healthcare Professional Management

4.5.2.14.1 View Healthcare Professionals

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses healthcare professional management page	- Page loads successfully with healthcare dashboard - Statistics cards display (Total Professionals, Doctors, Nurses, Recently Added) - Data table shows healthcare users list		
TC2	User views healthcare data table with no healthcare professionals	- Empty data table displayed - "No data available" message shown - Statistics show zero values		
TC3	User searches healthcare professional by name in	- Filtered results displayed		

	data table (e.g., search "John")	- Only matching law enforcement officers shown		
TC4	User without create permission views healthcare management page	- Page loads successfully - Add button not visible		
TC5	User with create permission views healthcare management page	- Page loads successfully - Add button visible		
TC6	User without edit permission views healthcare management page	- Page loads successfully - Edit buttons not visible in action dropdown		
TC7	User with edit permission views healthcare management page	- Page loads successfully - Edit buttons visible in action dropdown		
TC8	User without delete permission views healthcare management page	- Page loads successfully - Delete buttons not visible in action dropdown		
TC9	User with delete permission views healthcare management page	- Page loads successfully - Delete buttons visible in action dropdown		

Table 113: View Healthcare Professionals

4.5.2.14.2 Create Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits healthcare creation form with all required fields filled correctly	- Healthcare professional successfully created - Success message displayed - Modal closes automatically - New healthcare professional appears in healthcare list		
TC2	User submits healthcare creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits healthcare creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits healthcare creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"		
TC5	User submits healthcare creation form with email	- Form submission proceeds but server validation catches this		

	that already exists in database	- Display server validation error: "The email has already been taken"		
TC6	User submits healthcare creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"		
TC7	User submits healthcare creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"		
TC8	User submits healthcare creation form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please select an item in the list"		
TC9	User submits healthcare creation form with missing facility name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User submits healthcare creation form with missing state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC11	User creates CWO and cancels form by clicking 'Close' button	- Modal closes without saving - No child welfare officer created - Form data cleared		

Table 114: Create Healthcare Professional

4.5.2.14.3 Edit Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit healthcare form with all required fields updated correctly	- Healthcare professional successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in healthcare list		
TC2	User submits edit healthcare form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits edit healthcare form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC4	User submits edit healthcare form with missing facility name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits edit healthcare form with missing state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC6	User edits healthcare professional and cancels form by clicking 'Close' button	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC8	User uploads new avatar	- Avatar previews the selected avatar		
TC9	User updates facility information (facility name, state)	- Facility information successfully updated - Success message displayed - Updated facility information reflected in healthcare profile		
TC10	User updates APC expiry date	- APC expiry date successfully updated - Success message displayed - Updated APC expiry reflected in healthcare profile		

Table 115: Edit Healthcare Professional

4.5.2.14.4 Delete Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms healthcare deletion by clicking "Delete" button in modal	- Healthcare professional successfully deleted from database - Success message displayed - Healthcare professional removed from healthcare list - Modal closes automatically		
TC2	User cancels healthcare deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list		
TC3	User cancels healthcare deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list		

Table 116: Delete Healthcare Professional

4.5.2.15 Role and Permission Management

4.5.2.15.1 View Role Management

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User access' role management page	- Page loads successfully with role dashboard - Statistics cards display (Total Roles, Total Permissions) - Data table shows roles list (excluding admin role)		
TC2	User clicks on a role name to view details	- Role details page opens successfully - Shows role information, assigned users, and permissions		
TC3	User clicks on actions dropdown for a role	- Dropdown menu appears with options: View, Edit, Permissions, Delete		
TC4	User clicks "View" option from actions dropdown	- Role details page opens successfully - Shows role information, assigned users, and permissions		

TC5	User without create permission views role management page	- Page loads successfully - Add Role button not visible		
TC6	User with create permission views role management page	- Page loads successfully - Add Role button visible		
TC7	User without edit permission views role management page	- Page loads successfully - Edit option not visible in actions dropdown		
TC8	User with edit permission views role management page	- Page loads successfully - Edit option visible in actions dropdown		

Table 117: View Role Management

4.5.2.15.2 Create Role

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits role creation form with all required fields filled correctly	- Role successfully created - Success message displayed - Modal closes automatically - New role appears in roles list		

TC2	User submits role creation form with missing role name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC3	User submits role creation form with role name less than 2 characters	- Form submission proceeds but server validation catches this - Display server validation error: "Role name must be at least 2 characters long"		
TC4	User submits role creation form with role name containing invalid characters (e.g., "Role@123")	- Form submission proceeds but server validation catches this - Display server validation error: "Role name can only contain lowercase letters, numbers, and underscores"		
TC5	User submits role creation form with role name that already exists	- Form submission proceeds but server validation catches this - Display server validation error: "This role name already exists"		
TC6	User creates role and cancels form by clicking 'Close' button	- Modal closes without saving - No role created - Form data cleared		

Table 118: Create Role

4.5.2.15.3 Delete Role

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Delete" button for a role with no assigned users	- Delete confirmation modal opens successfully - Role name displayed in confirmation message - "Are you sure you want to delete this role?" message shown		
TC2	User confirms role deletion by clicking "Delete" button in modal	- Role successfully deleted from database - Success message displayed - Role removed from roles list - Modal closes automatically		
TC3	User cancels role deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to role data - Role remains in roles list		

TC4	User cancels role deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to role data - Role remains in roles list		

4.5.2.15.4 Assign Permission

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Assign Permissions" button for a role	- Permission assignment page opens successfully - Shows all available permissions grouped by module - Current role permissions are pre-selected		
TC2	User assigns permissions to a role and submits form	- Permissions successfully assigned to role - Success message displayed - Redirected back to role management page		
TC3	User cancels permission assignment by clicking "Cancel" button	- Redirected back to roles index - No changes made to role permissions		

		- Original permissions remain unchanged		
TC4	User selects/deselects individual permissions	- Selected permissions count updates dynamically - Individual permission checkboxes reflect selection state		
TC5	User selects "Select All" for a module	- All permissions in that module are selected - Module's "Select All" checkbox shows checked state		
TC6	User deselects "Select All" for a module	- All permissions in that module are deselected - Module's "Select All" checkbox shows unchecked state		

Table 119: Assign Permission

4.5.2.16 Contact Queries Management

4.5.2.16.1 View Contact Queries

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**

TC1	User accesses contact queries management page	- Page loads successfully with contact queries dashboard - Statistics cards display (Total Queries, Pending, In Progress, Resolved) - Data table shows contact queries list - Filters section visible with status, date filters		
TC2	User views contact queries data table with no queries	- Empty data table displayed - "No contact queries found" message shown - Statistics show zero values		
TC3	User searches contact query by name/email/subject in data table	- Filtered results displayed - Only matching contact queries shown		
TC4	User applies status filter (Pending, In Progress, Resolved)	- Data table filtered by selected status - Only queries with matching status displayed		

TC5	User applies date range filter	- Data table filtered by date range - Only queries within date range displayed		
TC6	User clicks "Clear" button to reset filters	- All filters reset to default values - Data table shows all contact queries		
TC7	User clicks on actions dropdown for a contact query	- Dropdown menu appears with options: View, Edit, Delete		
TC8	User clicks "View" option from actions dropdown	- Contact query details page opens successfully - Shows full query information including message content		
TC9	User without view permission accesses contact queries page	- Access denied or limited view based on permissions		
TC10	User with view permission views contact queries page	- Page loads successfully with all viewable elements		

Table 120: View Contact Queries

4.5.2.16.2 View Contact Query Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User views contact query details page	- Contact query details displayed - Shows name, email, subject, message, status, creation date		
TC2	User without edit permission views contact query details	- Contact query details displayed - Status change buttons not visible		
TC3	User with edit permission views contact query details	- Contact query details displayed - Status change buttons visible (Mark as Pending, In Progress, Resolved)		
TC4	User views contact query from anonymous user	- Contact query details displayed - User information section shows "Anonymous user"		
TC5	User views contact query from registered user	- Contact query details displayed - User information shows username, email, role, member since date		

Table 121: View Contact Query Details

4.5.2.16.3 Edit Contact Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User clicks "Mark as Pending" status button	- Status successfully updated to "Pending" - Success message displayed - Page refreshes with updated status		
TC2	User clicks "Mark as In Progress" status button	- Status successfully updated to "In Progress" - Success message displayed - Page refreshes with updated status		
TC3	User clicks "Mark as Resolved" status button	- Status successfully updated to "Resolved" - Success message displayed - Page refreshes with updated status		
TC4	User without edit permission tries to change status	- Status change buttons not visible - No status change functionality available		
TC5	User with edit permission views contact query details	- Status change buttons visible - Can change status using the buttons		

Table 122: Edit Contact Query

4.5.2.16.4 Delete Contact Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Delete" button for a contact query with valid permissions	- Delete confirmation modal opens successfully - Contact query subject displayed in confirmation message - "Are you sure you want to delete this contact query?" message shown		
TC2	User confirms contact query deletion by clicking "Delete" button in modal	- Contact query successfully deleted from database - Success message displayed - Contact query removed from contact queries list - Modal closes automatically		
TC3	User cancels contact query deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list		

TC4	User cancels contact query deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list		

Table 123: Delete Contact Query

4.5.2.17 View Session Logs

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses session logs page	- Page loads successfully with "Session Logs" title - Data table with columns: User, Status, Device Info, Location, Last Activity, Duration - Filters button visible		
TC2	User views session logs data table with no logs	- Empty data table displayed - "No activity logs found" message shown		
TC3	User clicks "Filters" button	- Filter section expands		

		- Shows user filter dropdown, date from/to inputs - Apply Filters and Clear Filters buttons visible		
TC4	User applies user filter	- Data table filtered by selected user - Only logs for selected user displayed		
TC5	User applies date range filter	- Data table filtered by date range - Only logs within date range displayed		
TC6	User clicks "Apply Filters" button	- Filters applied to data table - Table reloads with filtered results		
TC7	User clicks "Clear Filters" button	- All filters reset to default values - Data table shows all session logs		
TC8	User without view permission accesses session logs page	- Access denied or limited view based on permissions		
TC9	User with view permission views session logs page	- Page loads successfully with all viewable elements		

Table 124: View Session Log

4.5.2.18 Profiles

4.5.2.18.1 Edit Profile (System Administrator)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses admin profile edit page	- Page loads successfully with "Edit Admin Profile" title - Form pre-populated with current admin data - Avatar upload section visible		
TC2	User submits admin profile form with all required fields updated correctly	- Admin profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile		
TC3	User submits admin profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits admin profile form with missing display name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC5	User submits admin profile form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC6	User submits admin profile form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC7	User uploads new avatar image	- Avatar preview shows selected image		
TC8	User cancels profile edit by clicking "Cancel" button	- Redirected back to dashboard - No changes saved to profile		
TC9	User clicks "Delete My Account" button	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible		

Table 125: Edit Profile (System Administrator)

4.5.2.18.2 Edit Profile (Law Enforcement)

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**

TC1	User accesses law enforcement profile edit page	- Page loads successfully with "Edit Law Enforcement Profile" title - Form pre-populated with current law enforcement data - Avatar upload section visible		
TC2	User submits law enforcement profile form with all required fields updated correctly	- Law enforcement profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile		
TC3	User submits law enforcement profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits law enforcement profile form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits law enforcement profile form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC6	User submits law enforcement profile form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC7	User uploads new avatar image	- Avatar preview shows selected image		
TC8	User cancels profile edit by clicking "Cancel" button	- Redirected back to dashboard - No changes saved to profile		
TC9	User clicks "Delete My Account" button	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible		

Table 126: Edit Profile (Law Enforcement)

4.5.2.18.3 Edit Profile (Social Worker)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses social worker profile edit page	- Page loads successfully with “Edit Social Worker Profile” title		

		- Form pre-populated with current social worker data - Avatar upload section visible		
TC2	User submits social worker profile form with all required fields updated correctly	- Social worker profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile		
TC3	User submits social worker profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC4	User submits social worker profile form with missing staff ID field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC5	User submits social worker profile form with missing agency name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC6	User selects “Other” from agency dropdown	- Additional text input field appears - “Other” input field becomes required		

TC7	User selects predefined agency from dropdown	- Agency code field auto-populates - Agency code field becomes readonly		
TC8	User submits social worker profile form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC9	User submits social worker profile form with missing placement district field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC10	User uploads new avatar image	- Avatar preview shows selected image		
TC11	User cancels profile edit by clicking "Cancel" button	- Redirected back to dashboard - No changes saved to profile		
TC12	User clicks "Delete My Account" button	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible		

Table 127: Edit Profile (Social Worker)

4.5.2.18.4 Edit Profile (Child Welfare Officer)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses CWO profile edit page	- Page loads successfully with "Edit CWO Profile" title - Form pre-populated with current CWO data - Avatar upload section visible		
TC2	User submits CWO profile form with all required fields updated correctly	- CWO profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile		
TC3	User submits CWO profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC4	User submits CWO profile form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits CWO profile form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		

TC6	User submits CWO profile form with missing state field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC7	User uploads new avatar image	- Avatar preview shows selected image		
TC8	User cancels profile edit by clicking “Cancel” button	- Redirected back to dashboard - No changes saved to profile		
TC9	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible		

Table 128: Edit Profile (Child Welfare Officer)

4.5.2.18.5 Edit Profile (Healthcare Professional)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses healthcare profile edit page	- Page loads successfully with “Edit Healthcare Profile” title - Form pre-populated with current healthcare data		

		- Avatar upload section visible		
TC2	User submits healthcare profile form with all required fields updated correctly	- Healthcare profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile		
TC3	User submits healthcare profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC4	User submits healthcare profile form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"		
TC5	User submits healthcare profile form with missing facility name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC6	User submits healthcare profile form with missing facility state field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC7	User uploads new avatar image	- Avatar preview shows selected image		

TC8	User cancels profile edit by clicking “Cancel” button	- Redirected back to dashboard - No changes saved to profile		
TC9	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible		

Table 129: Edit Profile (Healthcare Professional)

4.5.2.18.6 Edit Profile (Social Worker)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses public user profile edit page	- Page loads successfully with "Edit Profile" title - Form pre-populated with current public user data - Email field shows as readonly with note "Private — cannot be changed"		
TC2	User submits public user profile form with all	- Public user profile successfully updated - Success message displayed		

	required fields updated correctly	- Redirected back to landing page - Updated data reflected in profile		
TC3	User submits public user profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”		
TC4	User cancels profile edit by clicking "Cancel" button	- Redirected back to landing page - No changes saved to profile		
TC5	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible		

Table 130: Edit Profile (Social Worker)

4.5.2.19 Delete My Account – All User Type

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User clicks "Delete My Account" button (any user type)	- Delete confirmation modal opens - Warning message: "Are you absolutely sure you want to delete your account?		

		This action cannot be undone and all your data will be permanently removed." - "Yes, Delete My Account" and "Cancel" buttons visible		
TC2	User confirms account deletion by clicking "Yes, Delete My Account" button	- Account successfully deleted from database - Success message displayed - User logged out and redirected to appropriate page - All user data permanently removed		
TC3	User cancels account deletion by clicking "Cancel" button	- Modal closes without deleting - No changes made to account - User remains logged in		
TC4	User cancels account deletion by clicking "Close" (X) button	- Modal closes without deleting - No changes made to account - User remains logged in		

Table 131: Delete My Account - All User Type

4.5.3 User Acceptance Testing (UAT) Preparation

Name of the Tester: **Role:** **Date:** **Time Start:** **Time End:**				
Name of parameter/Testing Category	**Excellent (5) points**	**Very Good (4) points**	**Fair (2-3) points**	**Poor (1-0) points**
User Registration & Authentication (Public user registration, role-based login, password reset, session management)				
Anonymous Child Abuse Reporting (Submit reports without revealing identity, incident details, evidence upload, truth confirmation)				
Case Management & Assignment (Case creation, assignment to social workers/law enforcement/healthcare, status updates, priority levels)				
Role-Based Dashboard Access (Admin, Government Official, Social Worker, Law Enforcement, Healthcare, Public User dashboards)				

Interagency Case Messaging (Secure communication between assigned case workers, real-time messaging, message history)				
Case History & Audit Trail (Comprehensive case tracking, status changes, activity logs, accountability)				
Contact Query Management (Public inquiries, response tracking, query resolution, admin management)				
Reports & Analytics (Case statistics, downloadable reports, monthly trends, system analytics, CSV/PDF exports)				
Activity Logs & Security (Login/logout tracking, failed login monitoring, IP address logging, device information)				
User Profile Management (Role-specific profile editing, avatar support, account management, account deletion)				
System Performance & Responsiveness (Page load speed, real-time updates, error handling, database optimization)				

User Interface & Navigation (Ease of use, responsive design, accessibility, clear layout, modern UI)				

Feedback from the Tester:	**Response from the Developer:**

Table 132: User Acceptance Testing Template (UAT)

4.6 Execution

4.6.1 Screenshots

4.6.1.1 Landing Page (Not Signed In)

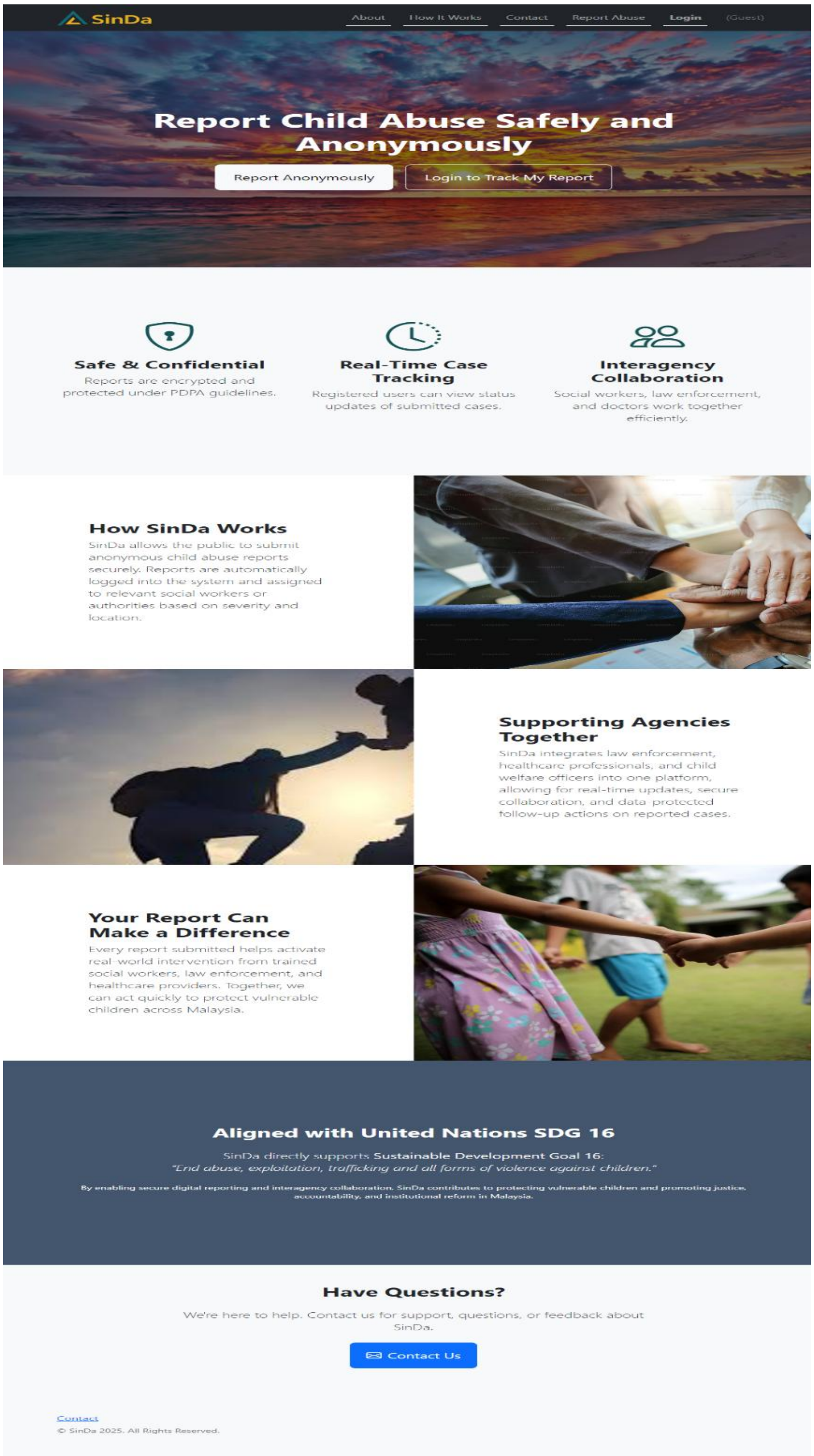


Figure 215: Landing Page (Not Signed In)

The SinDa landing page offers the public a safe way to anonymously report child abuse. It draws attention to important aspects including interagency cooperation, confidentiality, and real-time case tracking. The page highlights the importance of reporting, describes the process for handling reports, and links the system to SDG 16 to protect children from violence. There are three buttons where if the public user clicks on Report Anonymously button it will direct user to the report page. The second button "Login to Track My Report", if the user clicked on it, it would direct user to the sign in page to let the signed in user to see their submitted report. Lastly, theres also "Contact Us" button which will direct user to contact page to let user submit any query to the system administrator.

4.6.1.2 Landing page (Signed in)

Figure 216: Landing page (Signed in)

The signed-in home page gives users who have been verified access that is tailored to their needs. The user's email address is shown along with My Reports and a Logout button in the updated menu bar. The main area still shows off the most important features, letting users either send reports anonymously or check on the progress of reports they have already sent. The button "Login to Track My Report" has been changed to "Track My Report." This makes the change from public access to safe, role-based interaction smooth.

4.6.1.3 Sign in

4.6.1.3.1 Sign in Page

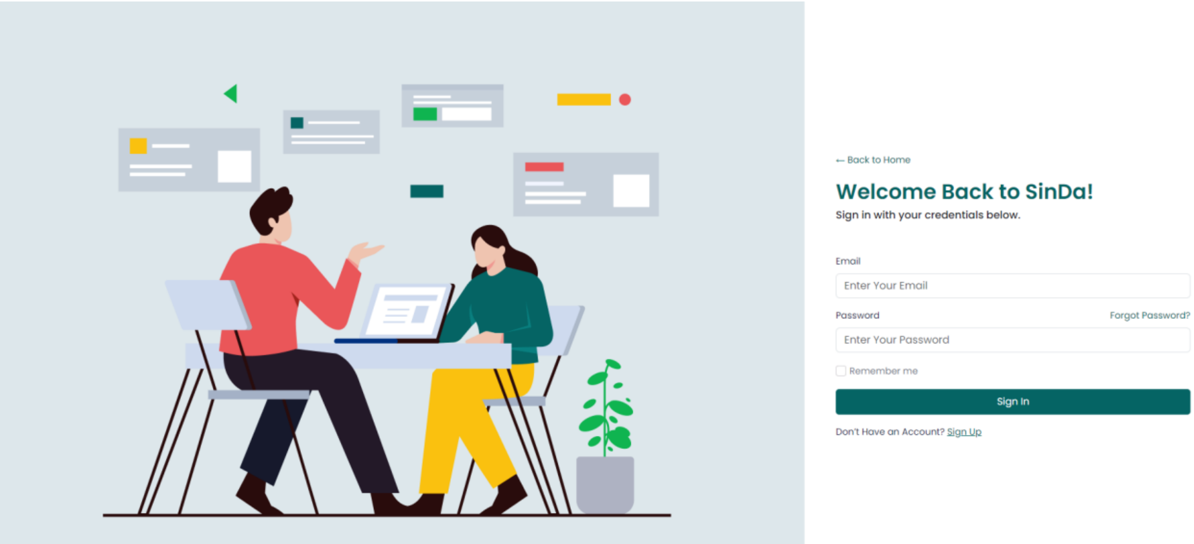
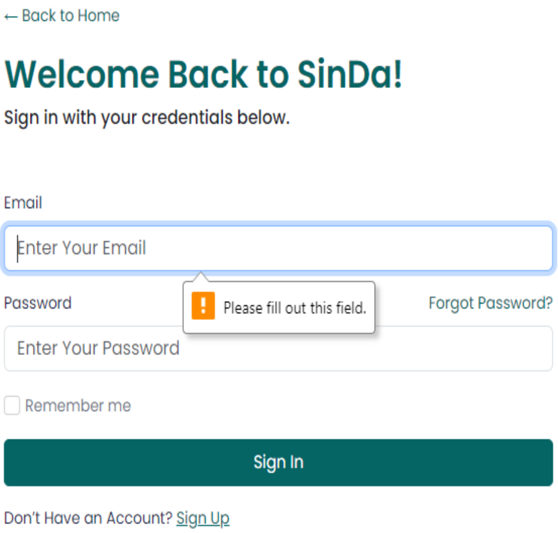
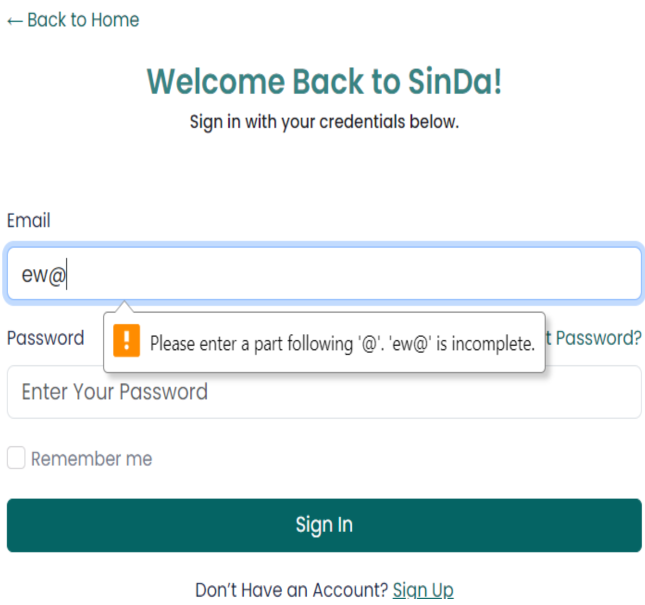


Figure 217: Sign In Page

This is the sign in page which provides a simple and secure interface for registered users to access their accounts. It includes fields for email and password, with options for "Forgot Password" and "Remember me." New users are guided to sign up, ensuring easy access and account management.

4.6.1.3.2 Sign in Page Validations





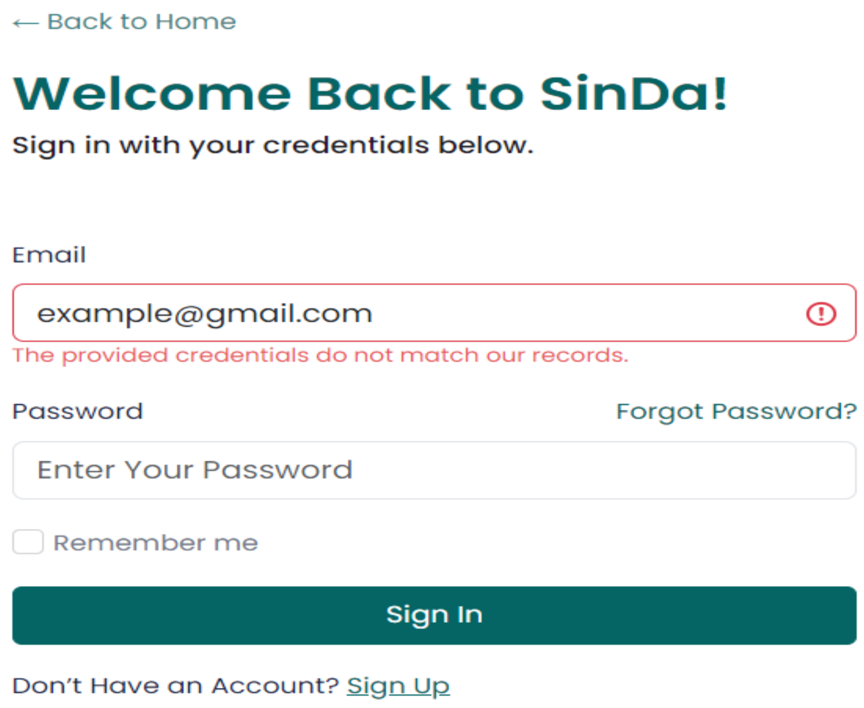


Figure 218: Sign In Page (Validations)

The login page includes input validation to ensure proper credentials are entered. When the user tries to sign in without filling in the email field, a warning message "Please fill out this field" appears. This prevents incomplete submissions and enforces secure login practices. When the user enters an invalid email format, such as leaving it incomplete after the "@" symbol, another validation message appears stating, "Please enter a part following '@'." This ensures the system only accepts properly structured email addresses. Such validations guide users to correct their input before submission, reducing login errors and maintaining secure access to the platform. Then if the user enters a validly formatted email, but the credentials do not exist or are incorrect, the system displays an error message based on the figure 217. This server-side validation ensures that only registered users with correct login details can access the system, thereby maintaining both account security and data protection.

4.6.1.4 Forgot Password

4.6.1.4.1 Reset Your Password modal

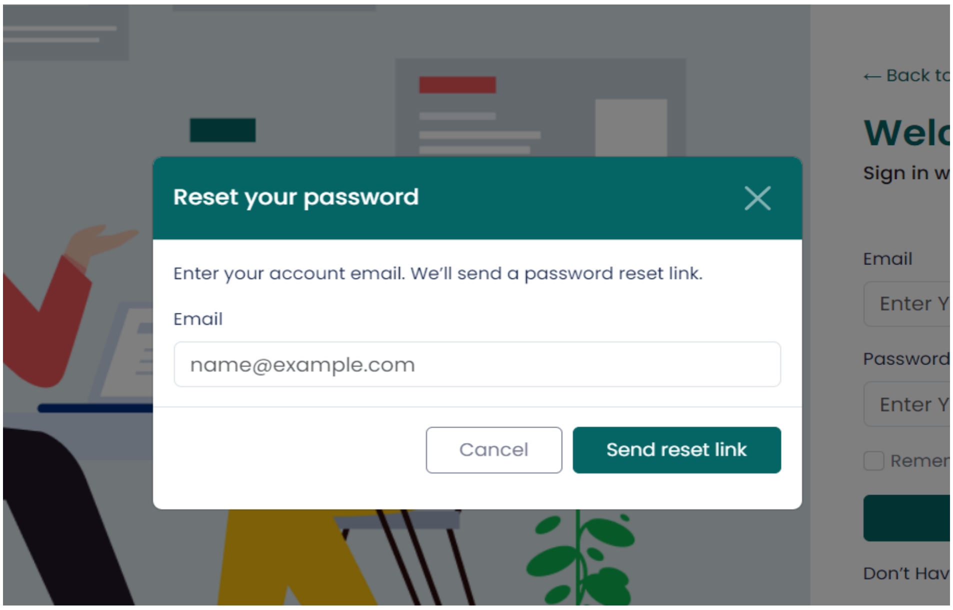
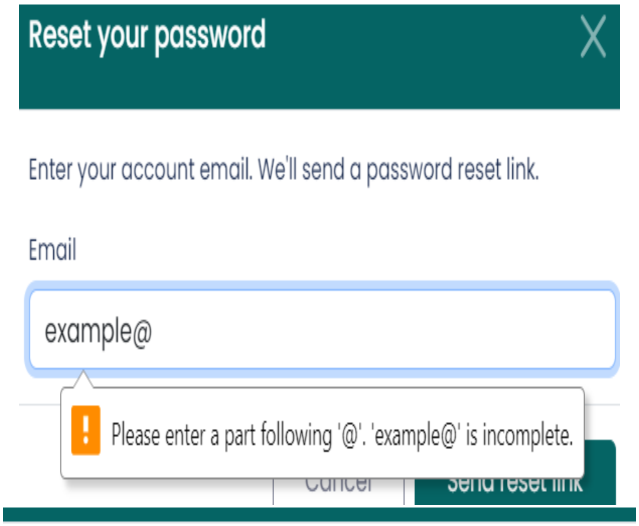
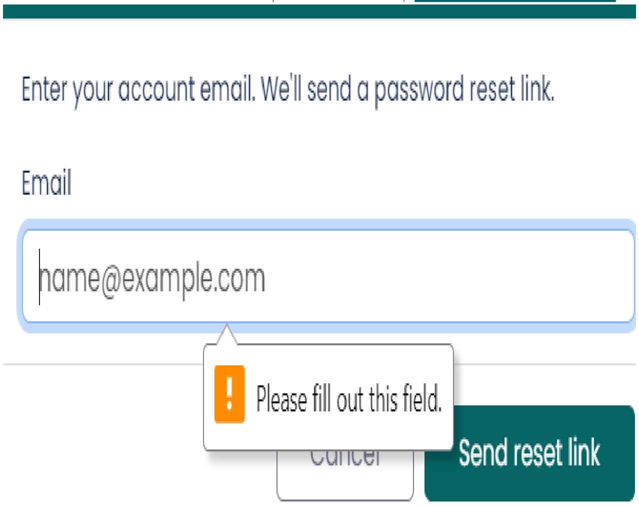


Figure 219: Reset your password modal

When a user clicks "Forgot Password?", a reset password modal appears. It prompts the user to enter their registered email address, and upon submission, a password reset link is sent to their inbox. The modal also includes options to Cancel or Send reset link, ensuring a straightforward and secure recovery process.

4.6.1.4.2 Reset Your Password modal validation





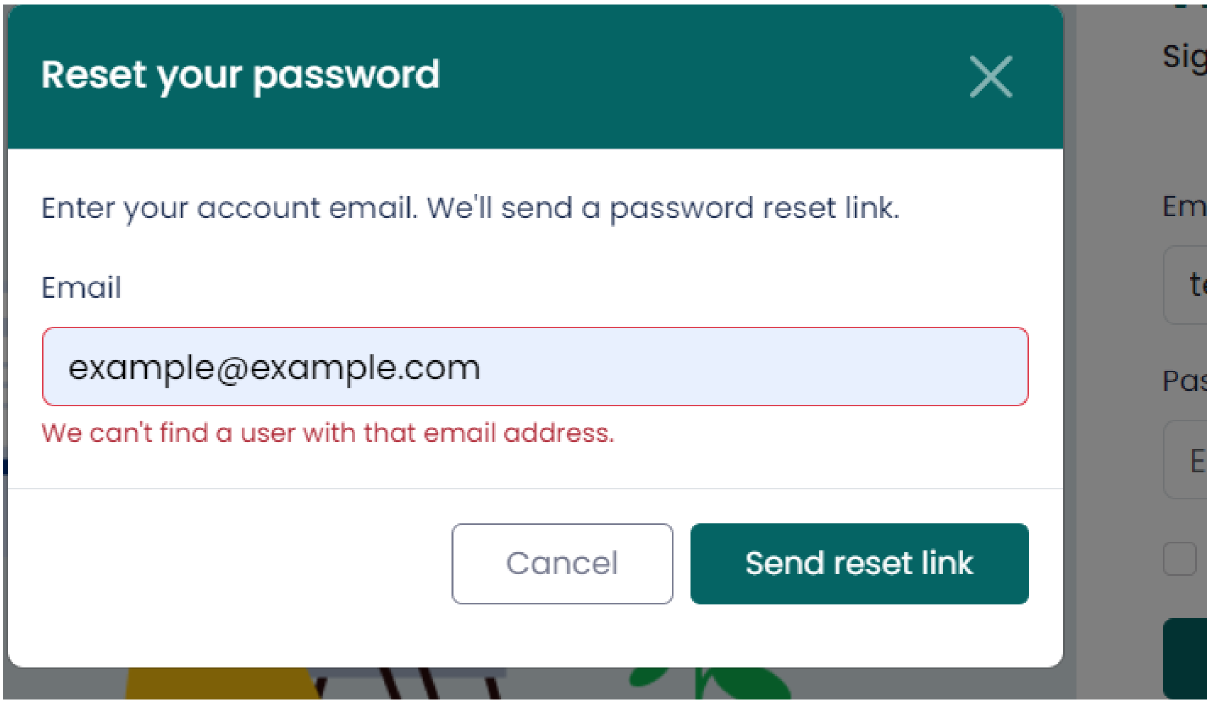


Figure 220: Reset your Password validation

If the user leaves the email fill blank or entered the email in invalid form for example “example@”, there will be browser validations that help to cover that situation. Now, if the user enters an email address that is not registered in the system, the reset password modal displays an error message based on the figure above. This validation ensures that only existing users can request a reset link, prevent misuse and maintain account security.

4.6.1.4.3 Password reset link send successfully

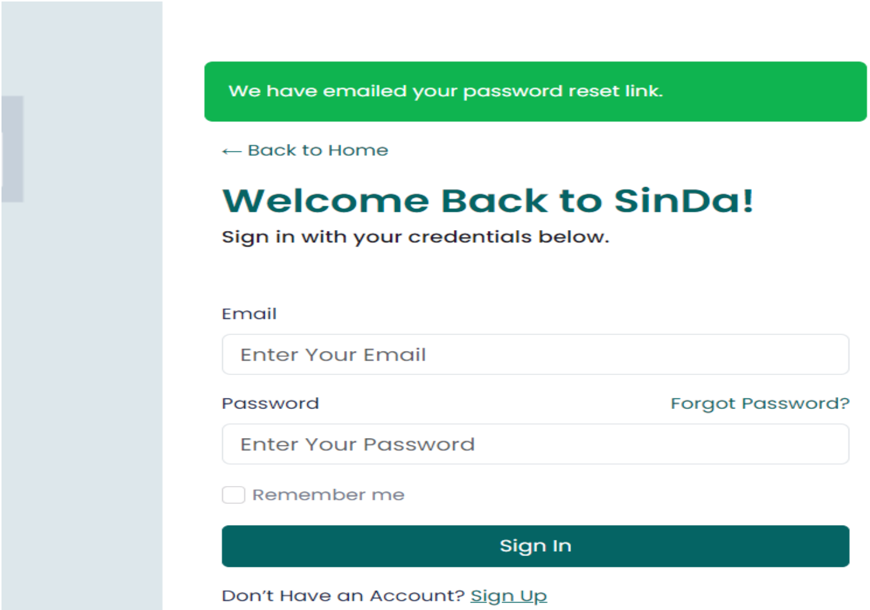


Figure 221: Success Message

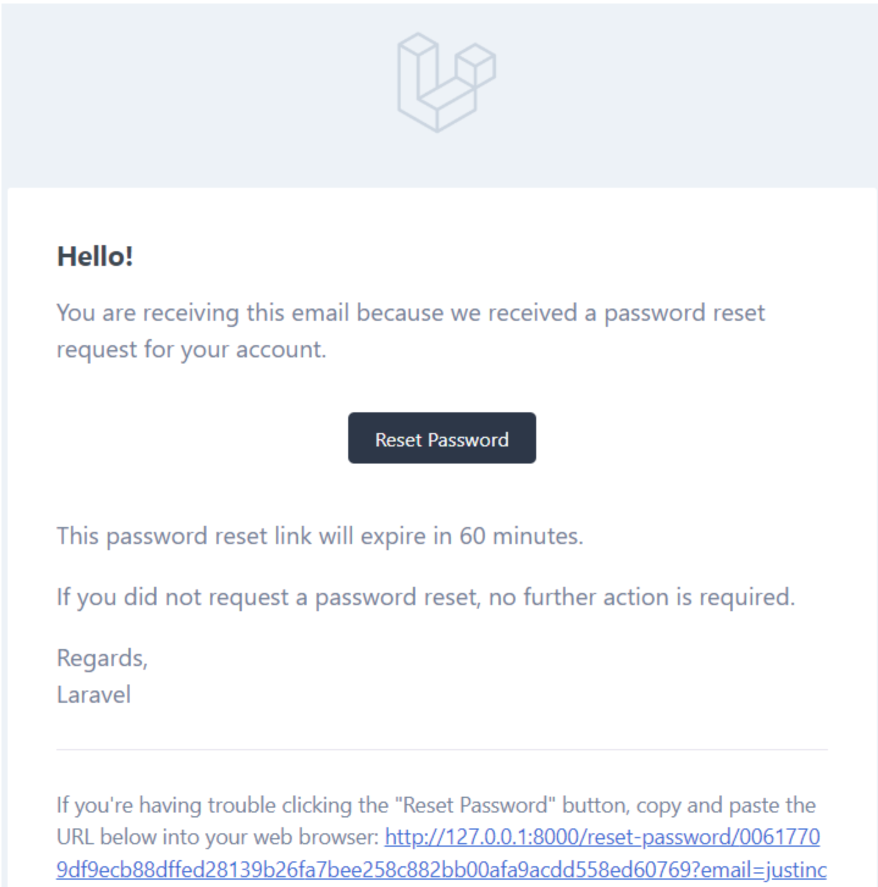


Figure 222: Reset Password email

If the user clicks on "Send reset link" with a valid email address, the system immediately displays a success message indicating that the reset link has been emailed. At the same time, the user is instructed to check their email inbox, where they will receive a secure message containing a reset link. This ensures that the reset process is both acknowledged on-screen and actionable via email, guiding the user through the next step to securely update their password.

4.6.1.4.4 Reset Your Password Page

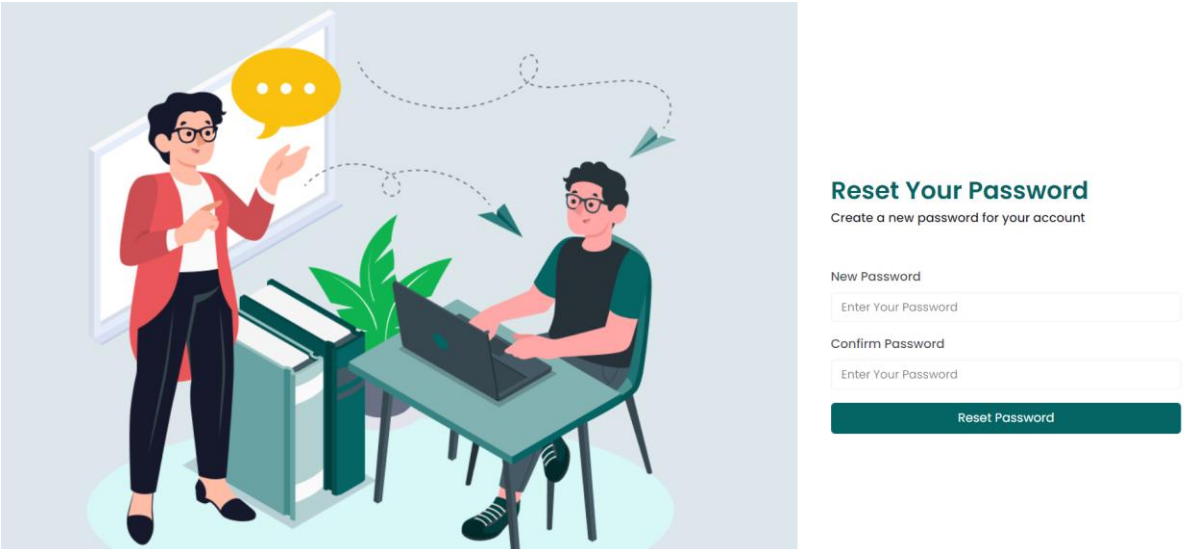
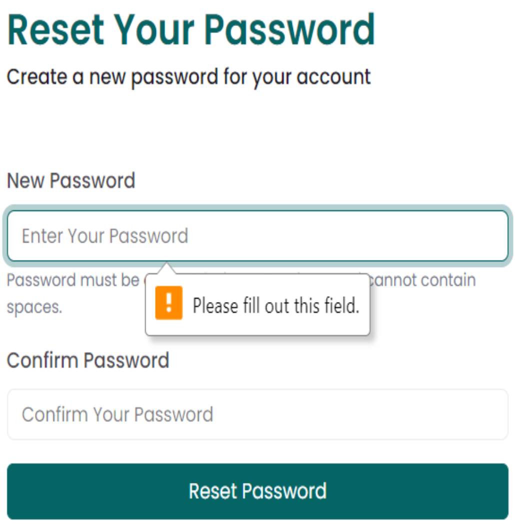
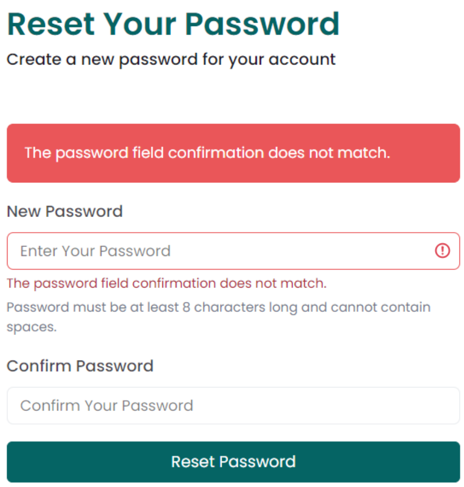


Figure 223: Reset Your Password Page

This is the Reset Your password page, where if the user has clicked on the Reset Password button the mail that they have requested to reset their password. Here, they are prompted to create a new password by entering it twice by inputting the password in New Password and Confirm Password field. If they successfully reset their password they will get redirected to sign in page where they may enter their new passwords.


Reset Your Password
Create a new password for your account
New Password
Enter Your Password
Password must be
annot contain
spaces.
Please fill out this field.
Confirm Password
Confirm Your Password
Reset Password


Reset Your Password
Create a new password for your account
The password field confirmation does not match.
New Password
Enter Your Password
The password field confirmation does not match.
Password must be at least 8 characters long and cannot contain spaces.
Confirm Password
Confirm Your Password
Reset Password

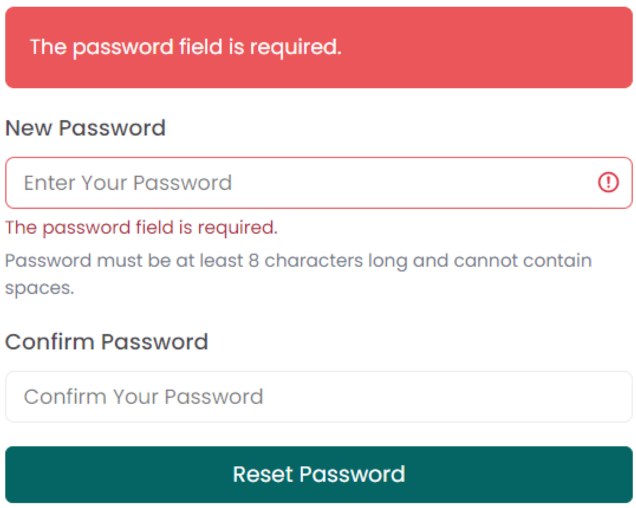

The password field is required.
New Password
Enter Your Password
The password field is required.
Password must be at least 8 characters long and cannot contain spaces.
Confirm Password
Confirm Your Password
Reset Password

Figure 224: Reset password validations

The Reset Password page includes multiple validation checks to ensure strong and accurate credential updates. If the New Password field is left empty, a warning prompts the user to fill it in. If the Confirm Password does not match the new password, an error message appears stating *"The password field confirmation does not match."* Additionally, if the password does not meet security rules minimum 8 characters, no spaces, the system highlights the error. These validations prevent weak or mismatched passwords, ensuring account security and user guidance.

4.6.1.5 Sign up

4.6.1.5.1 Sign up page

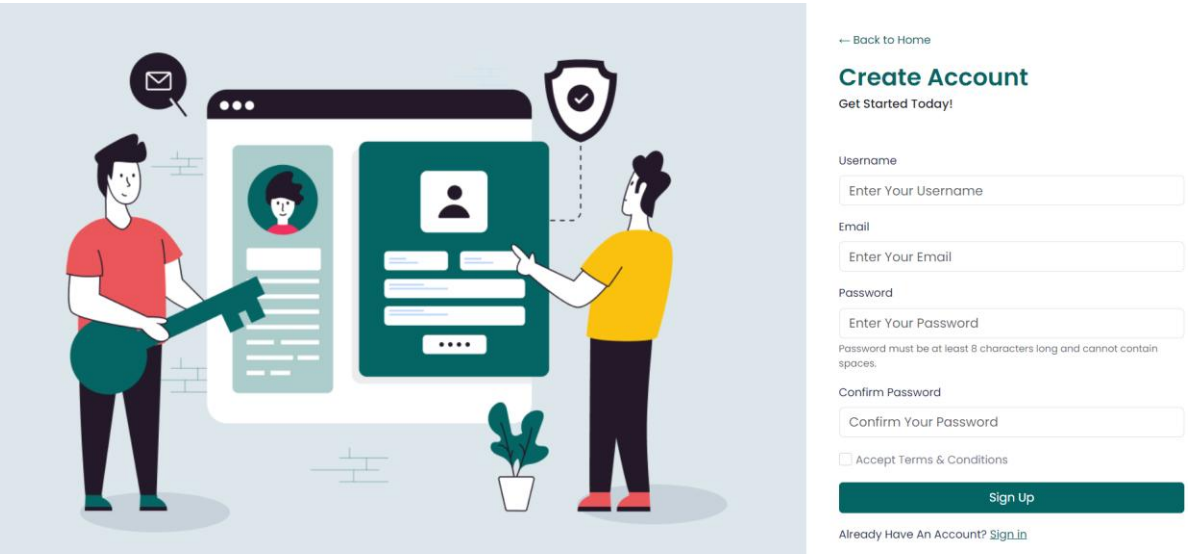
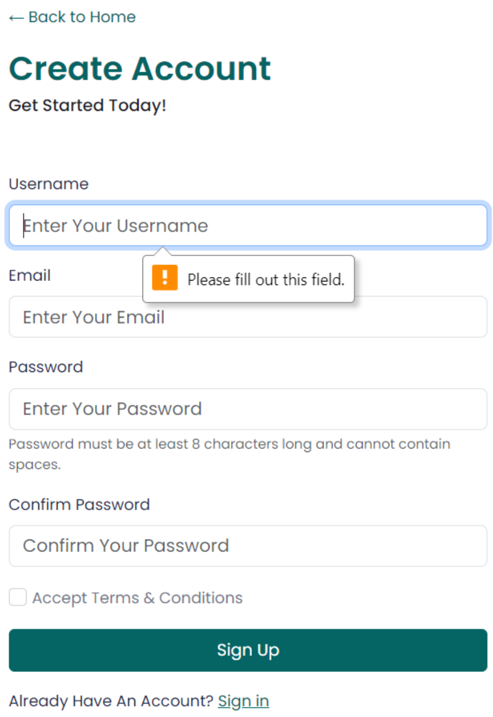
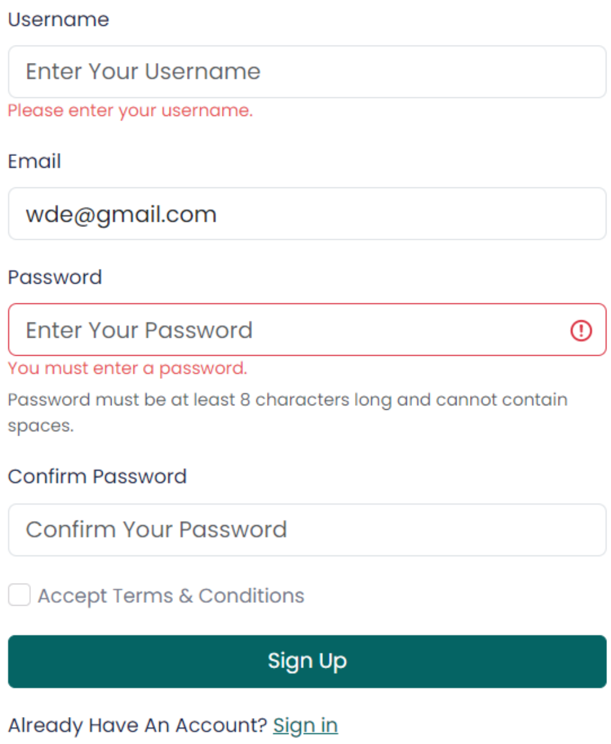


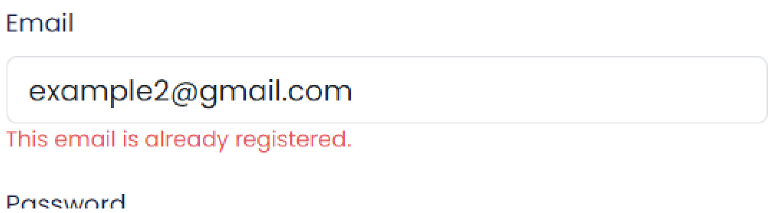
Figure 225: Sign up page

Above is the Sign up page for the system to allows new public users to register for the system. It includes input fields for Username, Email, Password, and Confirm Password, with password rules requiring at least 8 characters and no spaces. Users must also check the Accept Terms & Conditions box before signing up. A link at the bottom provides an option to switch to the login page if they already have an account or if the public user wanted to go back to landing back, they may press the "Back to Home" button to go back to the Landing page.

4.6.1.5.2 Sign up page validations




Username
Enter Your Username
Please enter your username.
Email
wde@gmail.com
Password
Enter Your Password
You must enter a password.
Password must be at least 8 characters long and cannot contain spaces.
Confirm Password
Confirm Your Password
Accept Terms & Conditions
Sign Up
Already Have An Account? Sign in


Email
example2@gmail.com
This email is already registered.

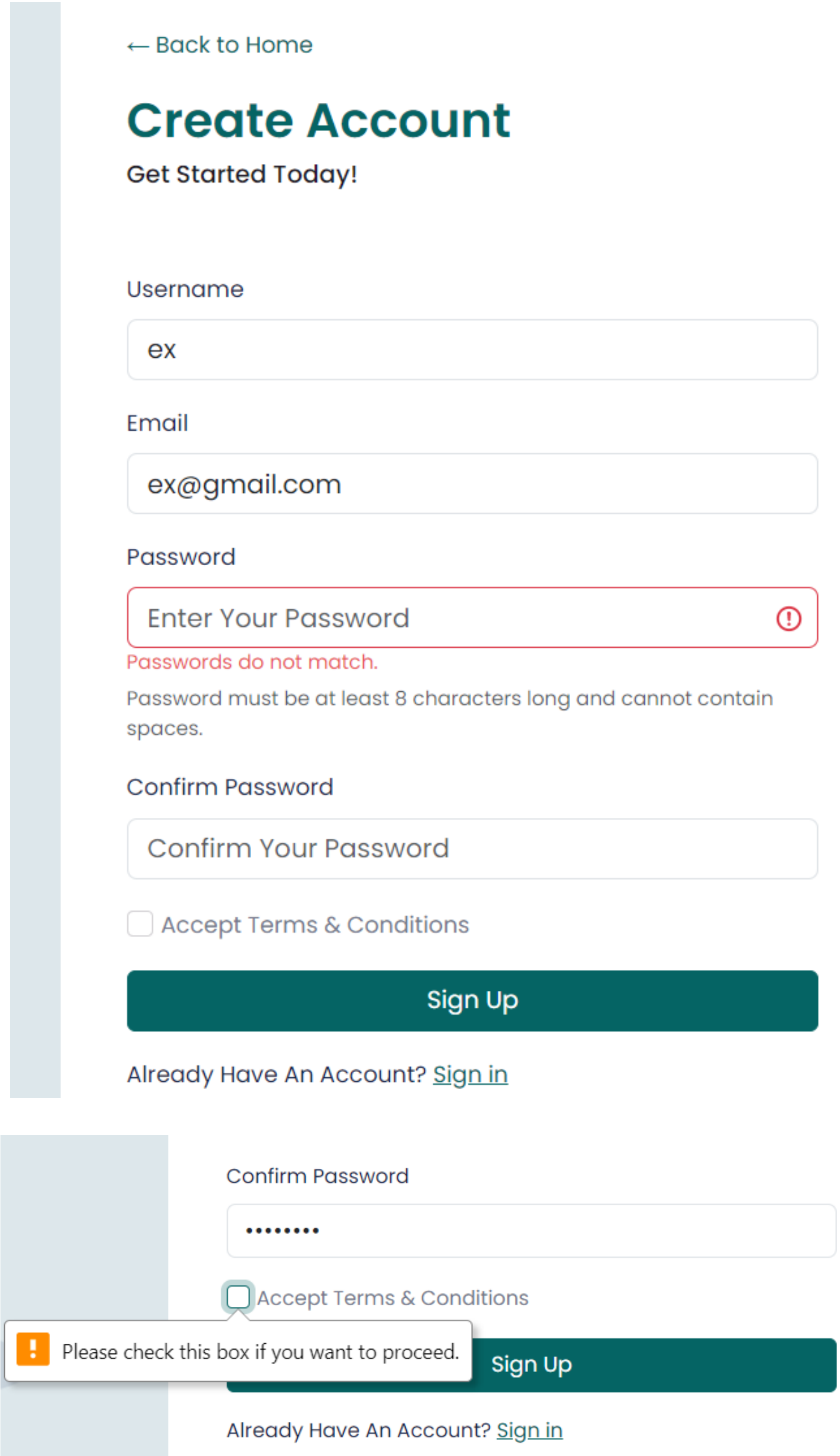


Figure 226: Sign up page validations

Besides this, the page includes multiple validation rules to ensure secure and accurate registration. If a user clicks the "Sign Up" button without completing required fields such as Username or Password, the system prompts them with error messages. It also enforces password policies, requiring a minimum of 8 characters with no spaces, and ensures the Confirm Password field matches the new password. If the entered email is already registered, the system displays an error message to prevent duplicate accounts. Furthermore, users must agree to the Terms & Conditions checkbox before proceeding. These validations work together to guide users in providing complete and correct details, enhancing both data integrity and account security.

4.6.1.5.3 Successful sign up

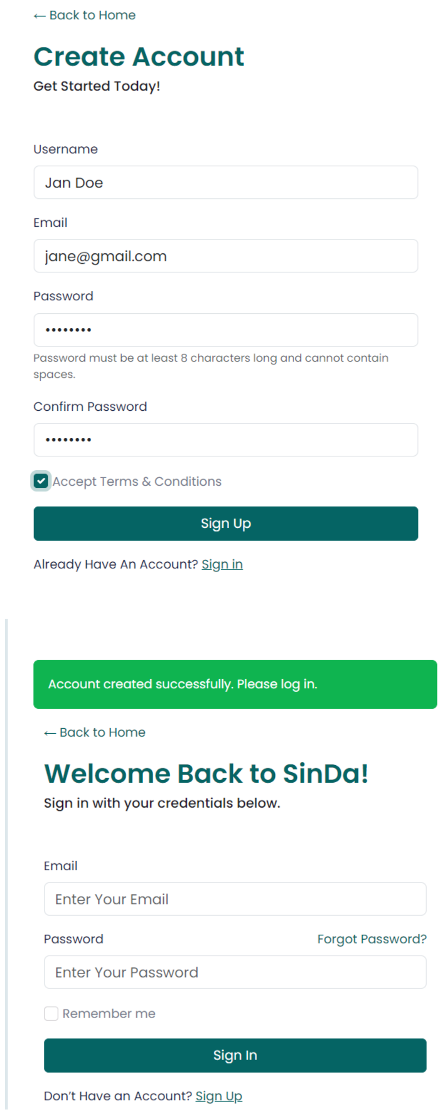


Figure 227: Successful account creation with redirect to Sign In page

When the user submits valid input in the registration form, the system successfully creates the account and redirects them to the Sign In page. A green success message is displayed, confirming the registration and guiding the user to log in with their new credentials.

4.6.1.6 Report Abuse page

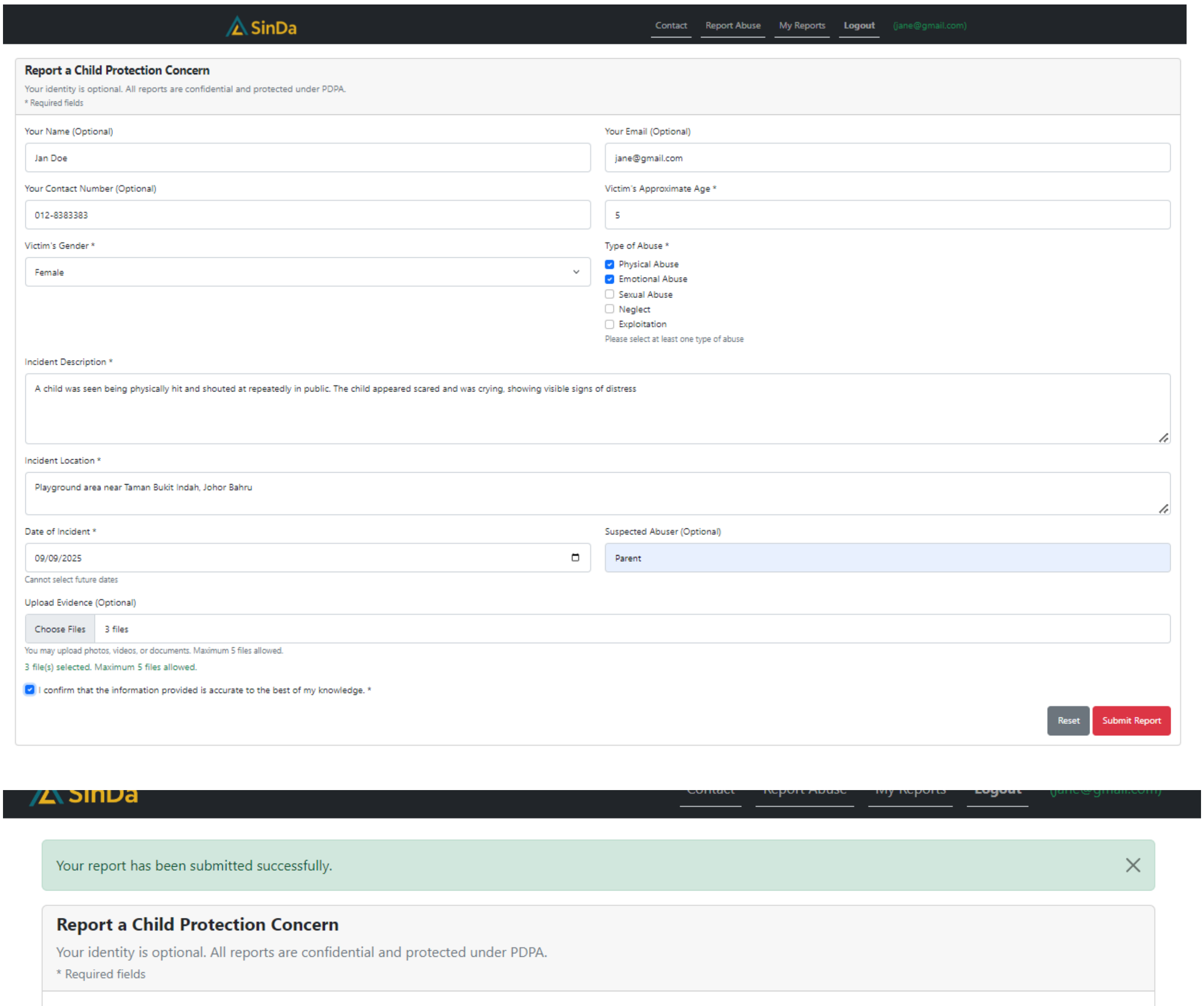


Figure 228: Report Abuse page

Above is the report abuse page which provides a structured form for submitting abuse cases. Public users can optionally provide their name, contact number, and email if they are not signed in. However, for signed-in users, these fields are pre-filled automatically, with the email set to read-only for consistency and security. Mandatory fields include the victim's gender, approximate age, type of abuse, incident description, location, and date of incident. Additional fields allow input of the suspected abuser and uploading of supporting evidence. A confirmation checkbox ensures users acknowledge the accuracy of their report before submission. Once the form is submitted, the system displays a success alert confirming that the report has been sent successfully. The form also supports Reset and Submit Report buttons, ensuring accurate, confidential, and secure case reporting.

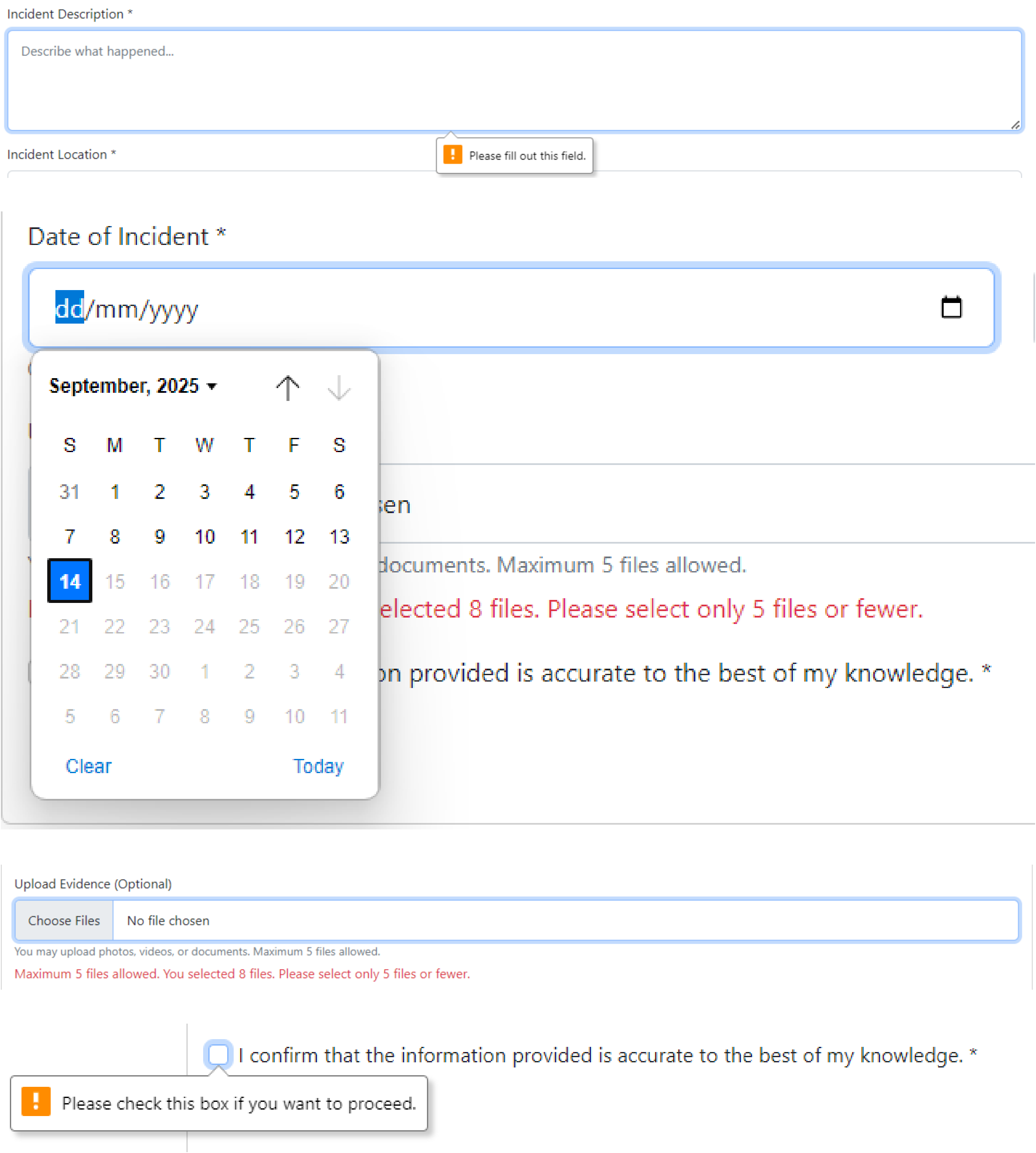


Figure 229: Report Abuse page validations

The Report Abuse page enforces strict validations with browser-based checks applied to all fields based on figures above. Mandatory inputs such as Incident Description, Incident

Location, and Date of Incident must be completed, with the date restricted to prevent future selections. The Upload Evidence field is limited to a maximum of 5 files and exceeding this shows an error. Additionally, the confirmation checkbox must be selected before submission, ensuring user acknowledgment. These browser validations ensure that all required data is provided accurately, preventing incomplete or invalid report submissions.

4.6.1.7 My Submitted Report page

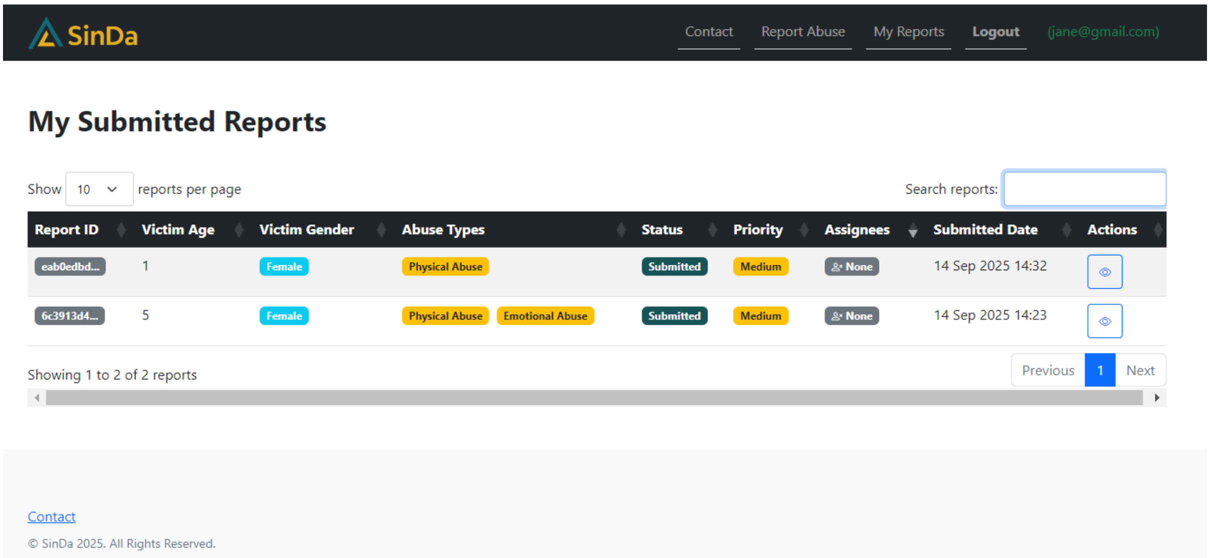


Figure 230: My submitted report page

The My Submitted Reports page allows signed in public users to view and track all reports they have submitted. It displays key details such as Report ID, Victim Age, Victim Gender, Abuse Types, Status, Priority, Assignees, and Submitted Date in a structured table. Users can also search reports, navigate between pages, and use the Actions column for additional interactions. This page ensures transparency and easy monitoring of submitted cases.

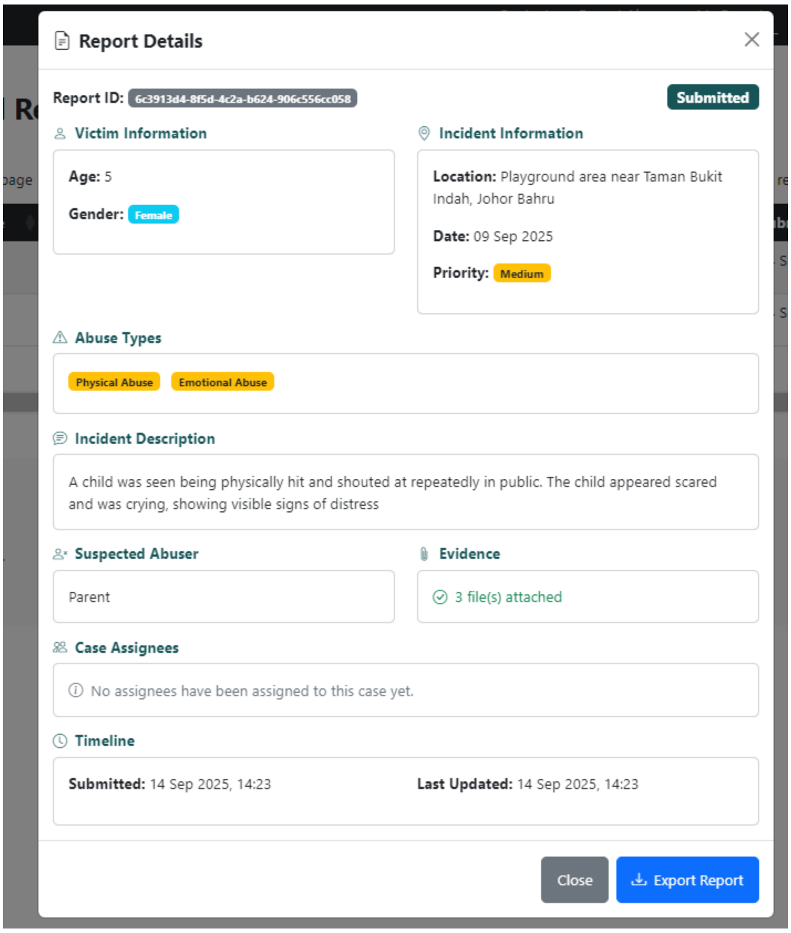


Figure 231: My Submitted Report Page

Public users who is logged in clicks the eye icon in the table of "My Submitted Reports" to see more information about the report. It gives a full picture of the case, including information about the victim, the place and date of the incident, the types of abuse that happened, a description of the incident, a possible abuser, and proof that was attached. The modal also displays case assignees, a timeline with due dates and update dates, and an Export Report choice for keeping track of things. This makes sure that users can easily look over and handle the reports they've sent in.

4.6.1.8 Contact Page

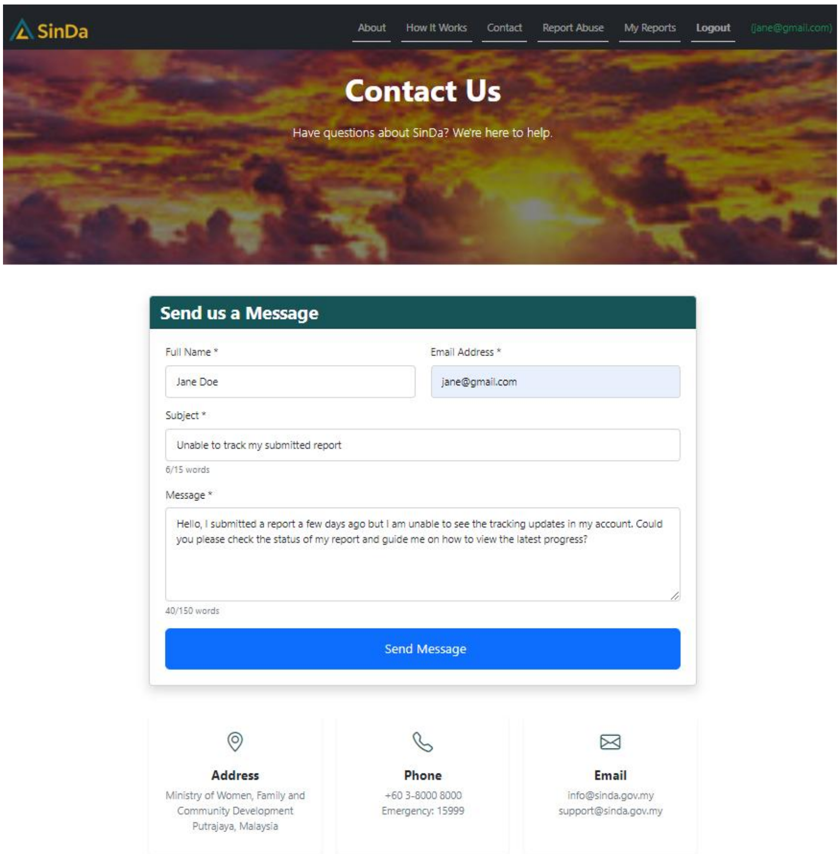


Figure 232: Contact Query page

The Send us a Message page allows users to directly contact the support team for assistance, inquiries, or feedback. The form requires input for Full Name, Email Address, Subject, and Message, with word limits applied to ensure concise communication. The Send Message button submits the query securely, enabling users to reach out for help regarding reports, technical issues, or general information. This feature enhances user engagement by providing a direct communication channel within the platform.

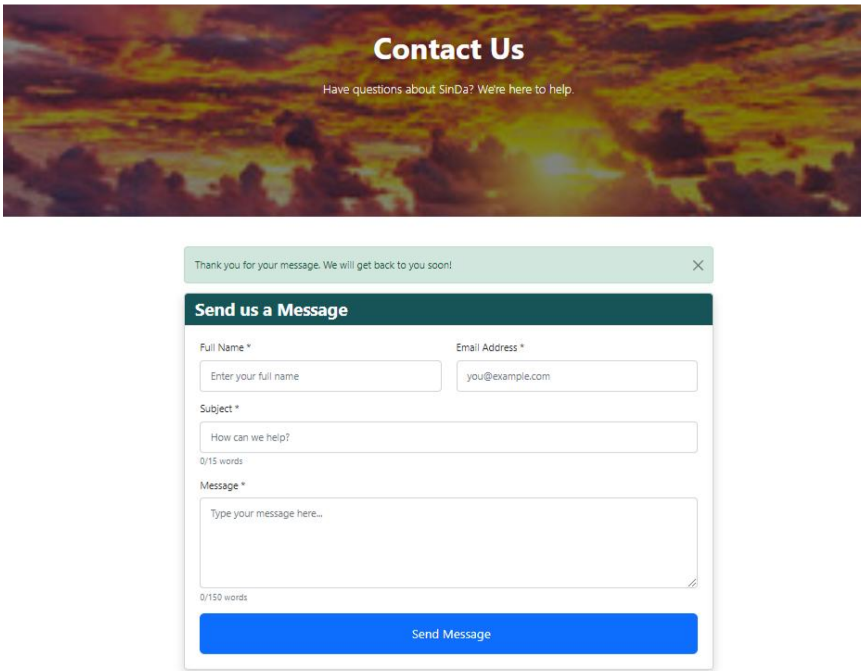
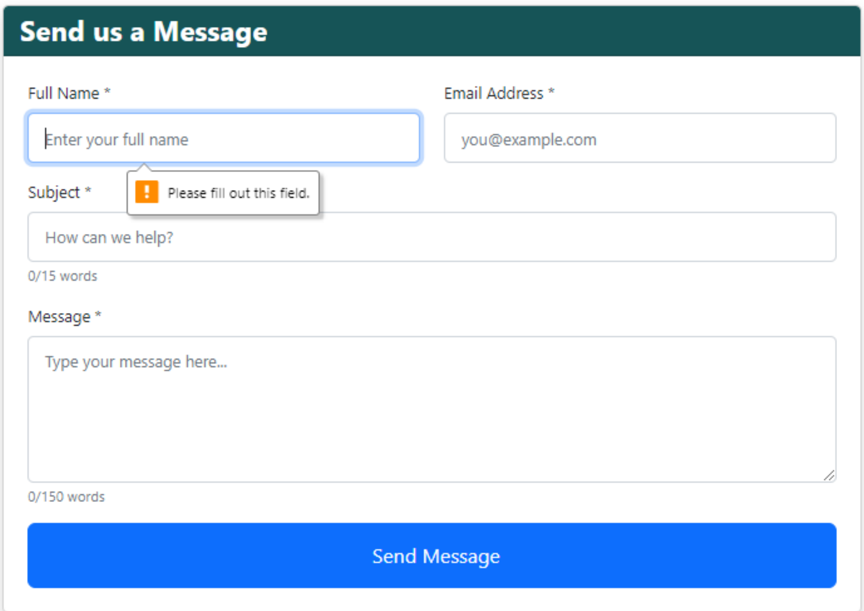


Figure 233: Contact query successful message

Once user fill up the form like the figure 230 and hit the Send Message button, there will be successful message which assures users that their message has been received and will be responded to promptly, providing a reliable communication channel within the platform.



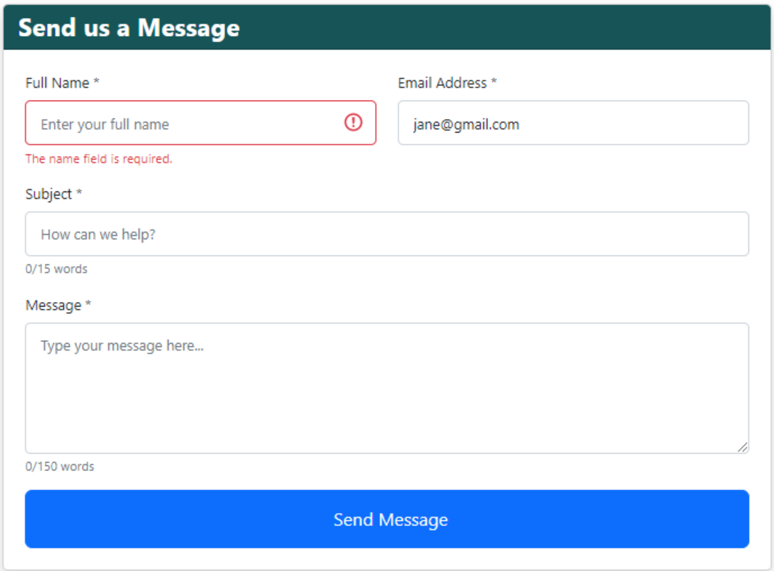


Figure 234: Contact page validations

On the Send us a Message form and other pages, validations run at two levels. First, browser validations require all fields; if the user clicks Send Message with any field empty, inline prompts appear asking to fill them in. Second, server-side validations reject inputs that contain only spaces or otherwise invalid content, returning clear error messages. for example, "The name field is required". With this, these checks ensure meaningful submissions and prevent blank or malformed messages.

4.6.1.9 Public User Profile

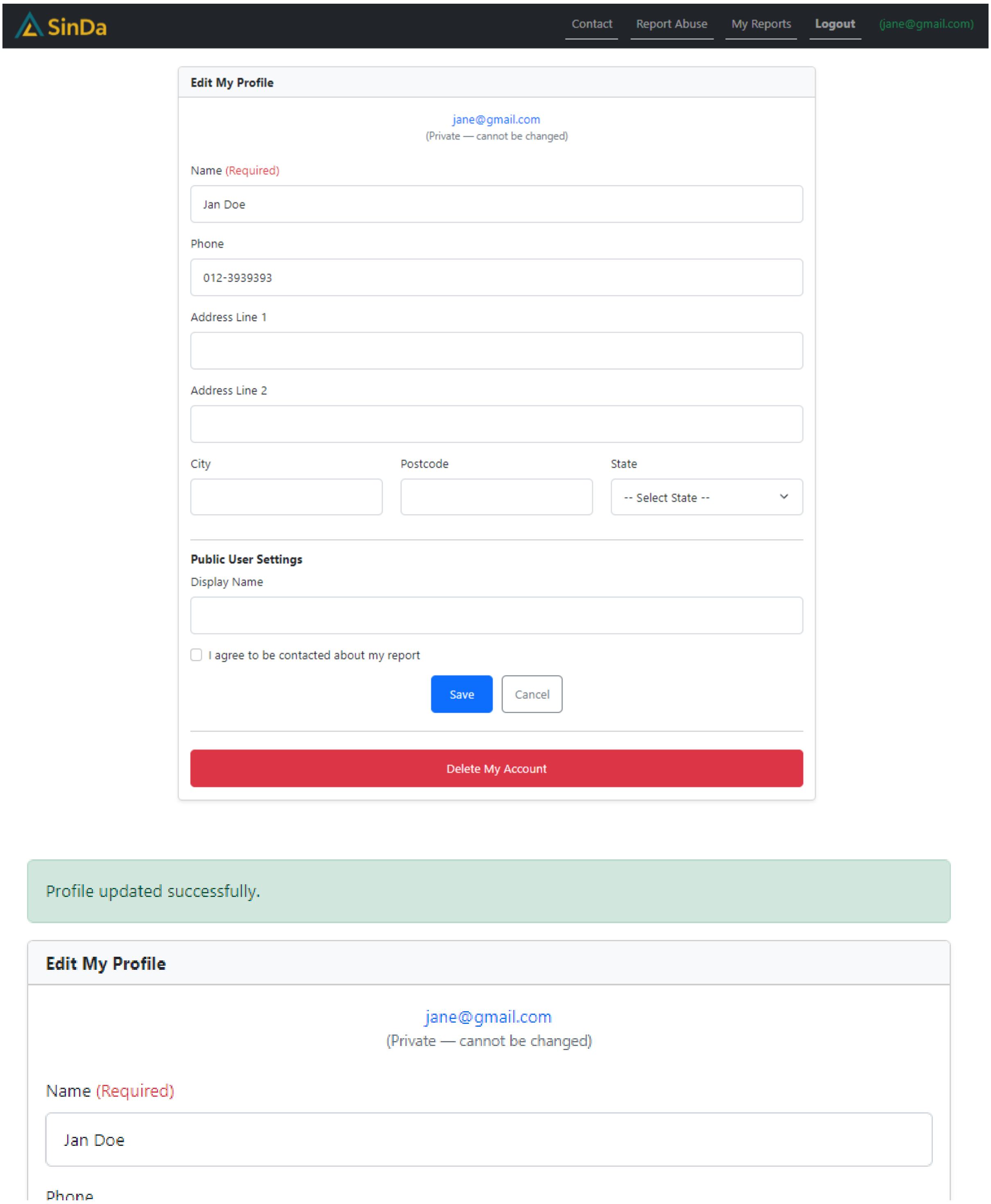


Figure 235: Edit my profile (Public User)

The user can manage their personal information on the Edit My Profile page after clicking on their email address in the navigation bar. Their registered email is shown at the top of the page (read-only), but you can change other data like Name, Phone, Address, City, Postcode, and

State. Under public user options, users can also select a Display Name and decide if they want to be notified regarding their reports. Actions include Delete My Account to remove the account permanently, Cancel to discard edits, and Save to update changes. If the user changes any field and click on the Save button, there will be successful message to confirm the changes. Hence, this page guarantees that users preserve control over their profile information while protecting private data, such as email addresses.

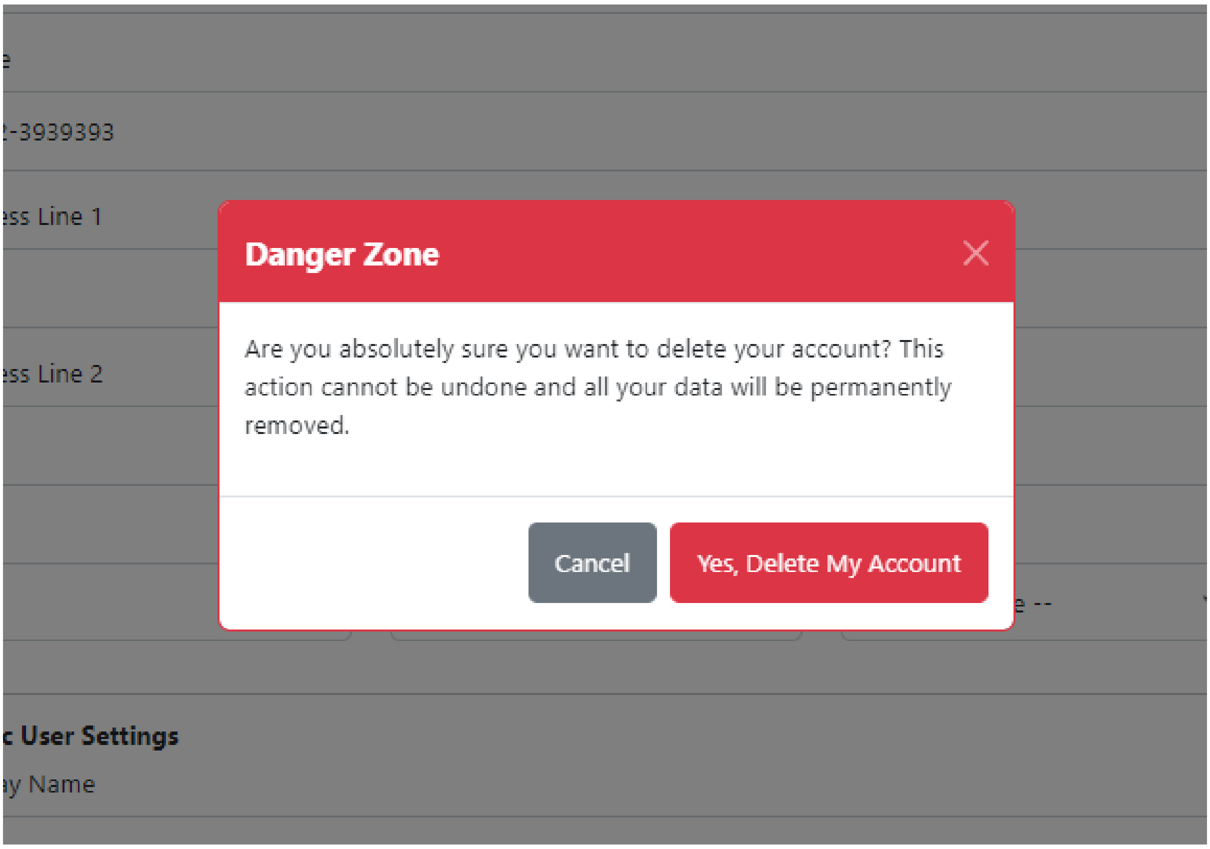


Figure 236: User delete confirmation modal

If the user clicked on Delete my account, the above modal will pop up to confirm the deletion. If the user clicks on the Yes button, they will then sign out from the system and will be redirected back to the Landing page with their account permanently deleted.

4.6.1.10 Authorized Users Dashboard

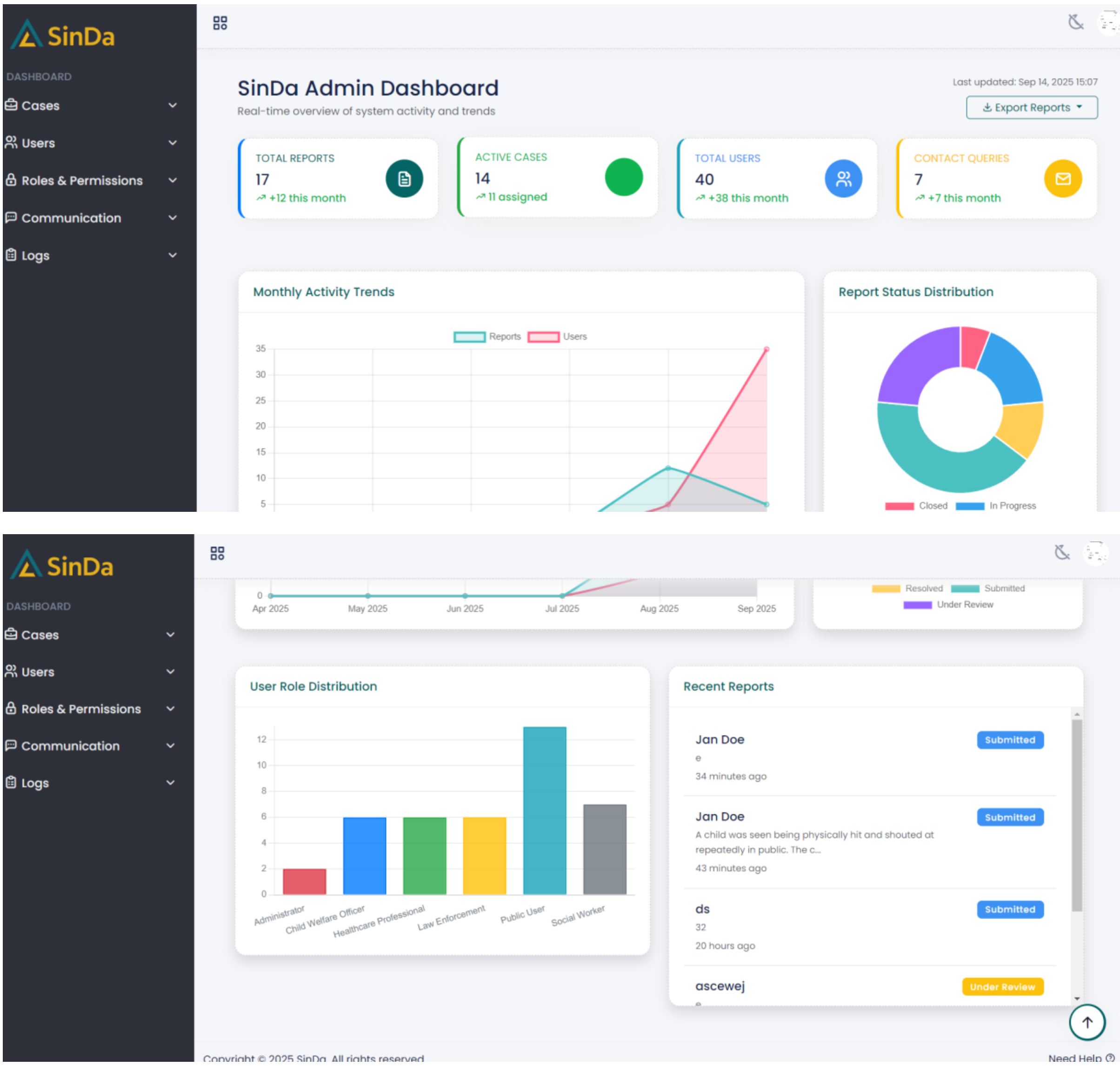


Figure 237: Authorized Users Dashboard (System Administrator)

Based on Figure 235 is the proposed system dashboard for authorized users when they signed in. For system administrators, at the top, summary cards display Total Reports, Active Cases, Total Users, and Contact Queries, with growth indicators for the current month. Visual charts track Monthly Activity Trends and Report Status Distribution, offering insights into how reports progress through different stages. Additional panels include User Role Distribution, showing the breakdown of user types in the system, and Recent Reports, which lists the latest submissions with their status. The sidebar menu provides quick navigation to Cases, Users, Roles & Permissions, Communication, and Logs, ensuring efficient system monitoring and management.

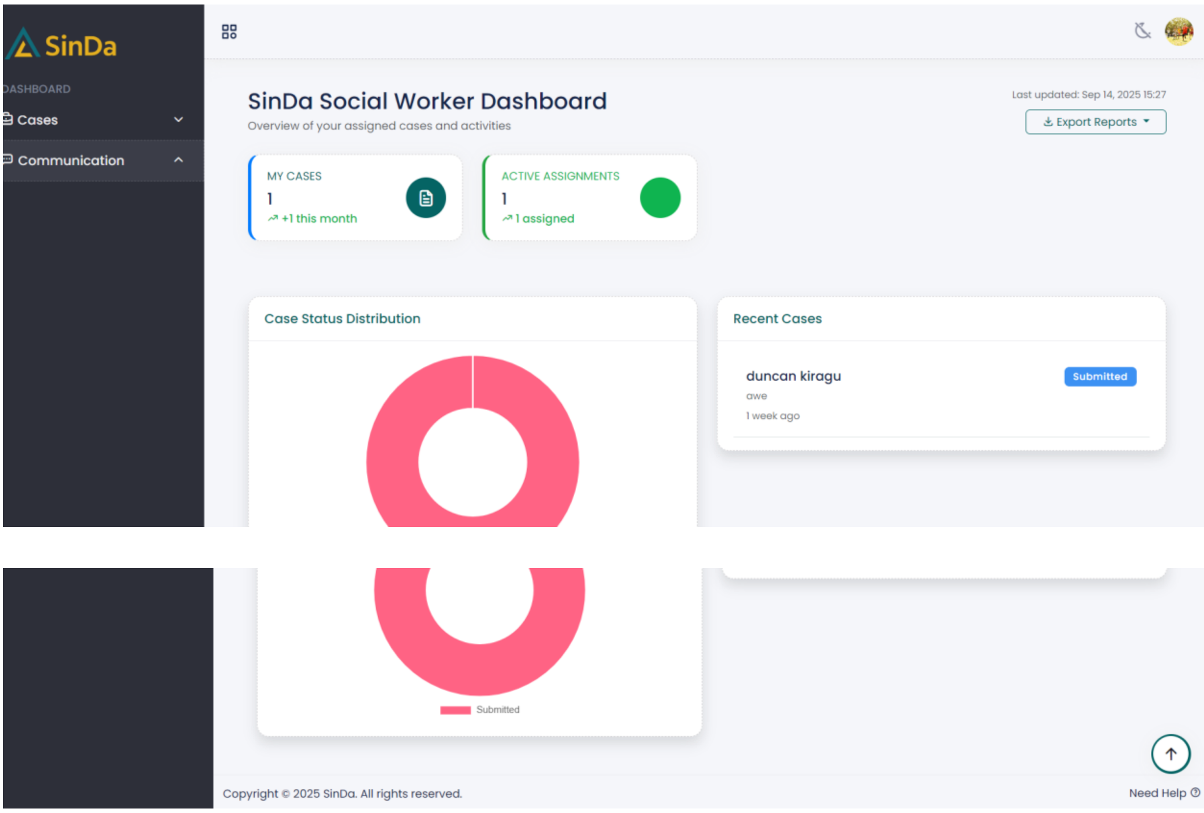


Figure 238: Authorized User Dashboard (Social Worker, Healthcare Professional, CWO)

If the logged-in user is not a system administrator, they are redirected to their respective Authorized User Dashboard (e.g., Social Worker, Healthcare Professional, CWO, or Law Enforcement). This dashboard provides a tailored view with key metrics such as My Cases and Active Assignments, ensuring users can monitor only the cases directly assigned to them. It also includes a Case Status Distribution chart and a Recent Cases panel, allowing authorized users to quickly track progress and stay updated on their responsibilities while maintaining role-based access control.

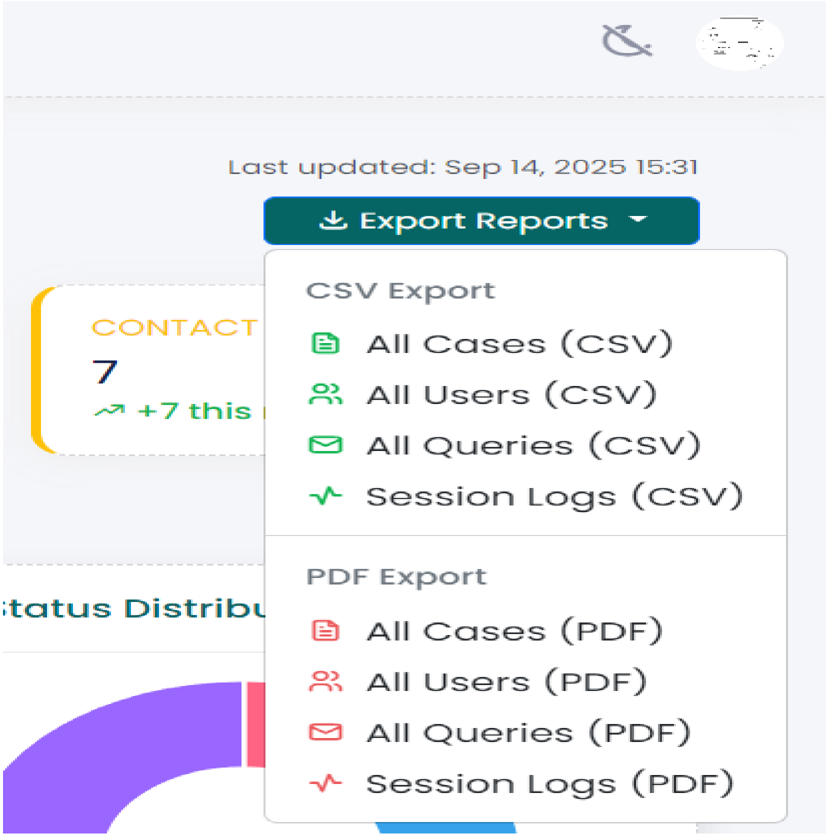
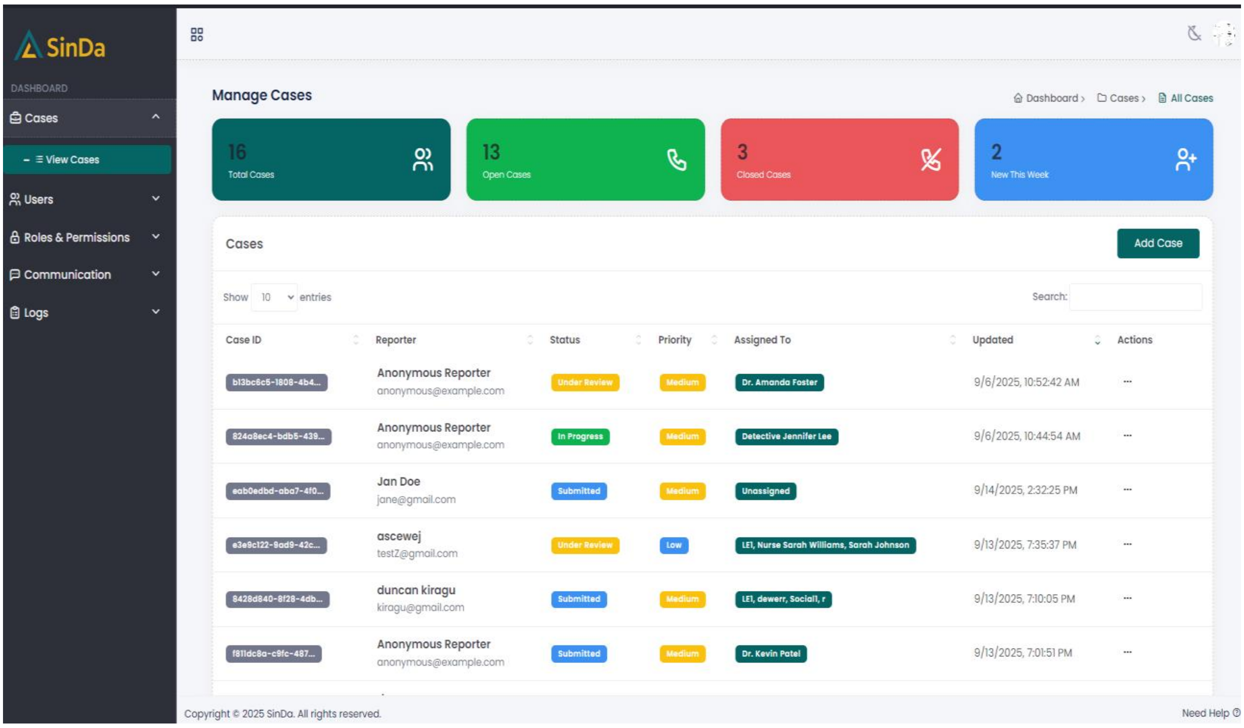


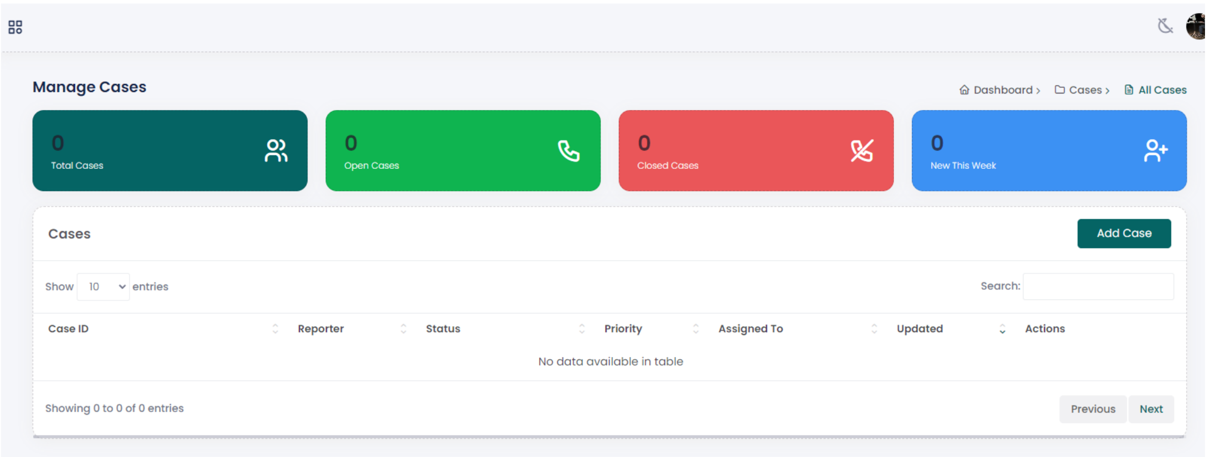
Figure 239: Export Reports

If the signed in user have permission to export reports, there will be export report button at their dashboard based on figure 235. This feature allows users to generate and download system data in both CSV and PDF formats. The export options include All Cases, All Users, All Queries, and Session Logs, providing flexibility for analysis, record-keeping, or external reporting. By supporting multiple formats, the system ensures that exported data can be used for both quick reviews (CSV for spreadsheets) and formal documentation (PDF for reports).

4.6.1.11 Case Management

4.6.1.11.1 Manage Cases Page





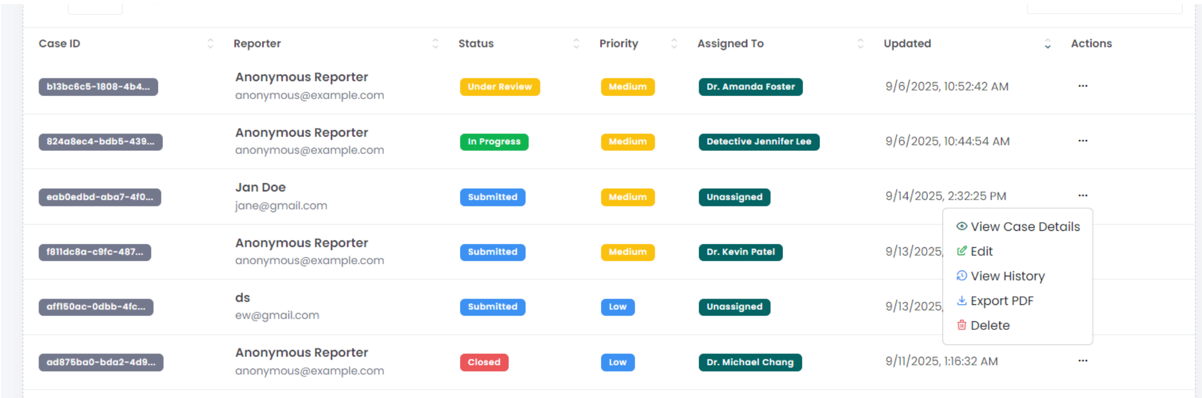


Figure 240: List Cases

Above is the List Cases page when the signed-in user clicks on View Cases from the sidebar. If no cases are assigned to the user, the table will display a message stating “No data available in table.” However, if cases exist, they will be shown in a structured list that includes details such as Case ID, Reporter, Status, Priority, Assigned To, Updated Date, and Actions where if the user click on it here will be dropdown. The top summary cards provide a quick snapshot of Total Cases, Open Cases, Closed Cases, and New Cases This Week, giving users both an overview and detailed access to case management. Additionally, the page includes an Add Case button on the top right of the table if the user have permission to, which allows authorized users to create and submit new cases directly from this page.

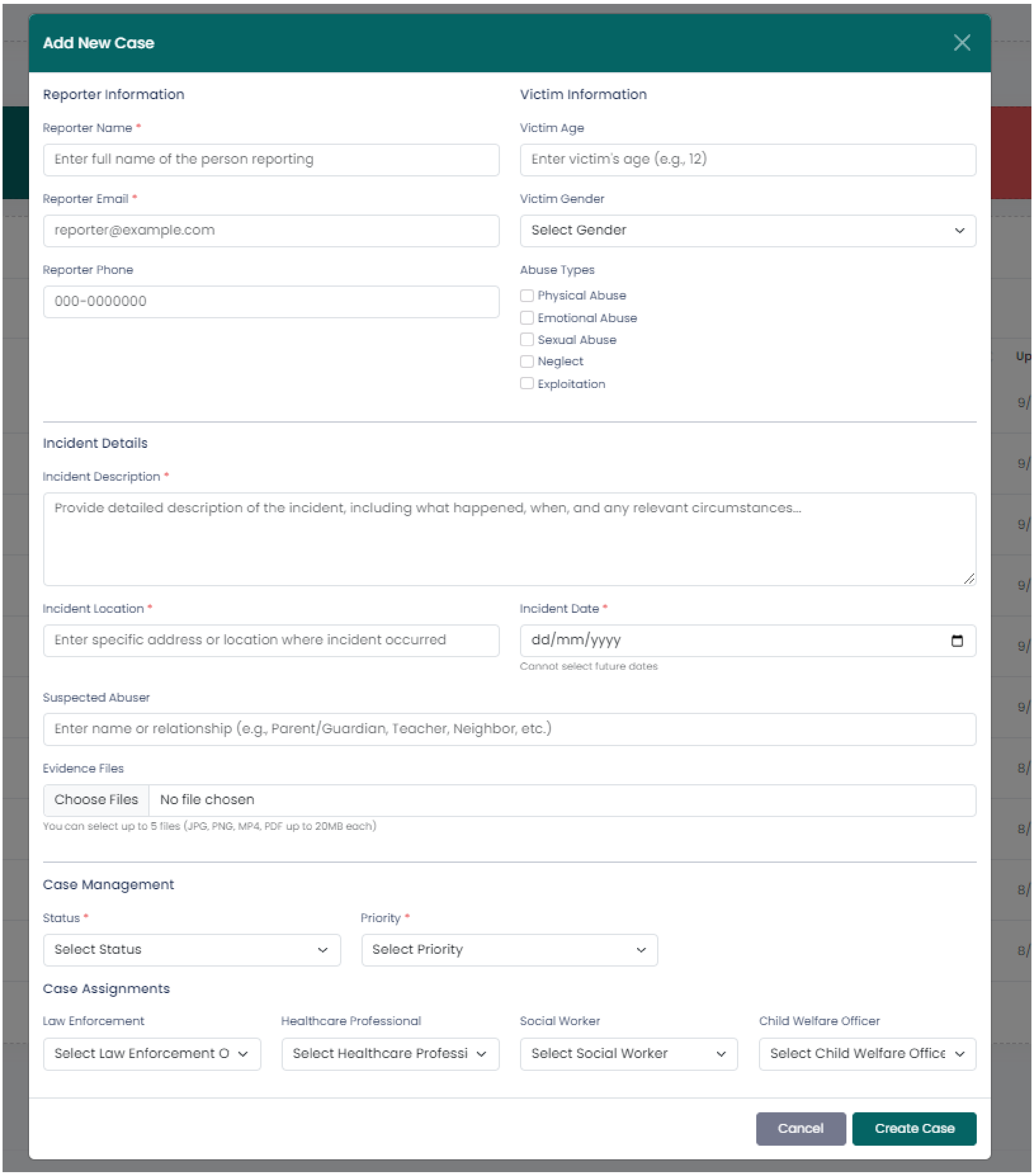


Figure 241: Add case modal

The Add New Case modal provides a structured form for registering child protection cases, ensuring all critical details are captured before submission. It begins with reporter information, followed by victim details such as age, gender, and type of abuse, with multiple categories available including physical, emotional, sexual, neglect, or exploitation. Users must also provide an incident description, location, and date, while optional fields allow adding a suspected abuser and uploading up to five evidence files. To support case tracking, the modal includes fields for selecting case status and priority, along with direct assignment options to law enforcement, healthcare professionals, social workers, or child welfare officers. Once completed, the case can be submitted through the Create Case button.

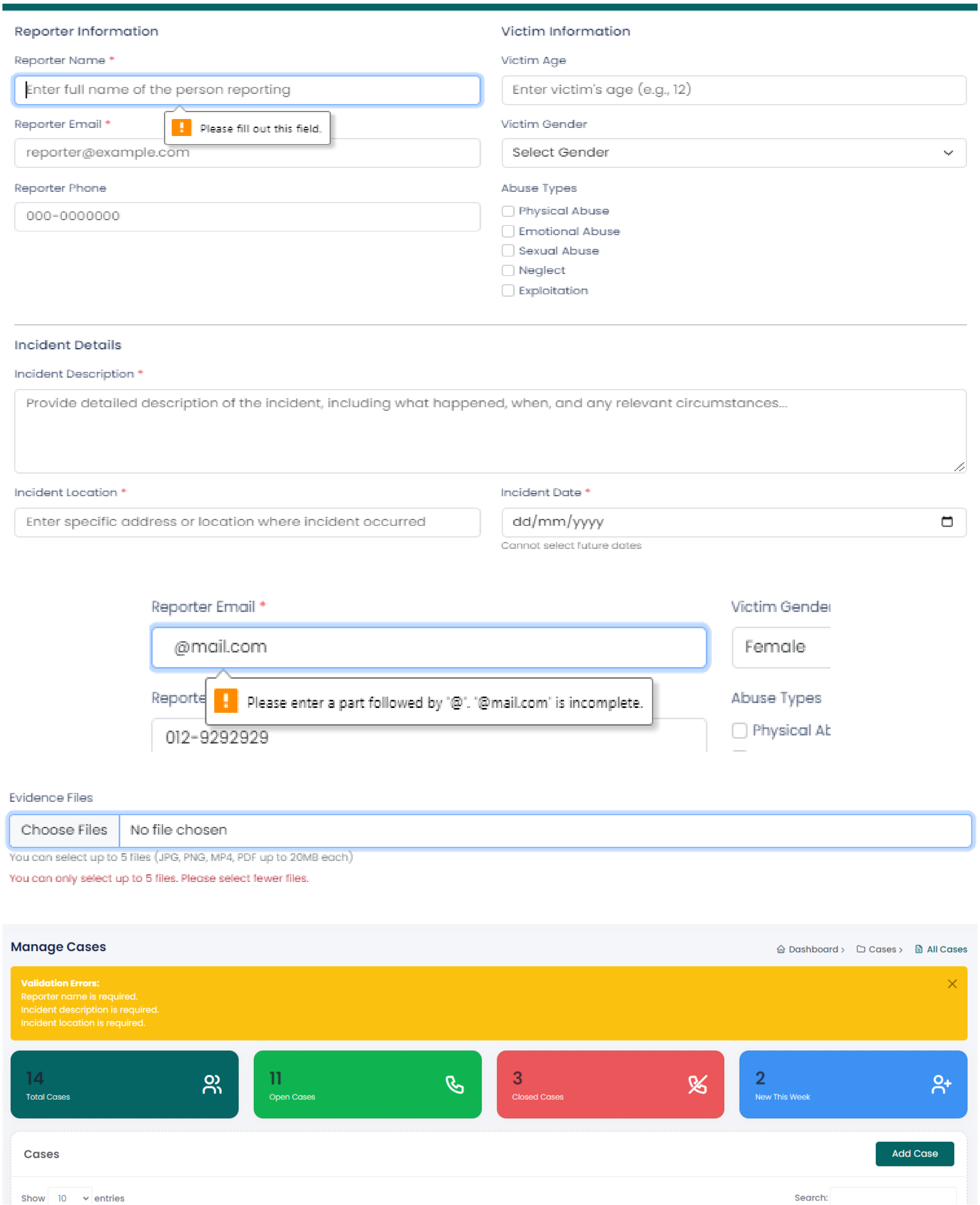


Figure 242: Add Case modal validations

The Add New Case modal uses two-tier validation. First, browser validations require all mandatory fields such as Reporter Name, Reporter Email, Incident Description, Location, Incident Date based on figure above, showing inline prompts like “Please fill out this field” or email-format errors if inputs are incomplete. Second, server-side validations catch malformed entries such as fields containing only spaces or non-meaningful text, returning clear error messages and preventing submission. Together, these checks ensure complete, correctly formatted, and meaningful case data before a case can be created.

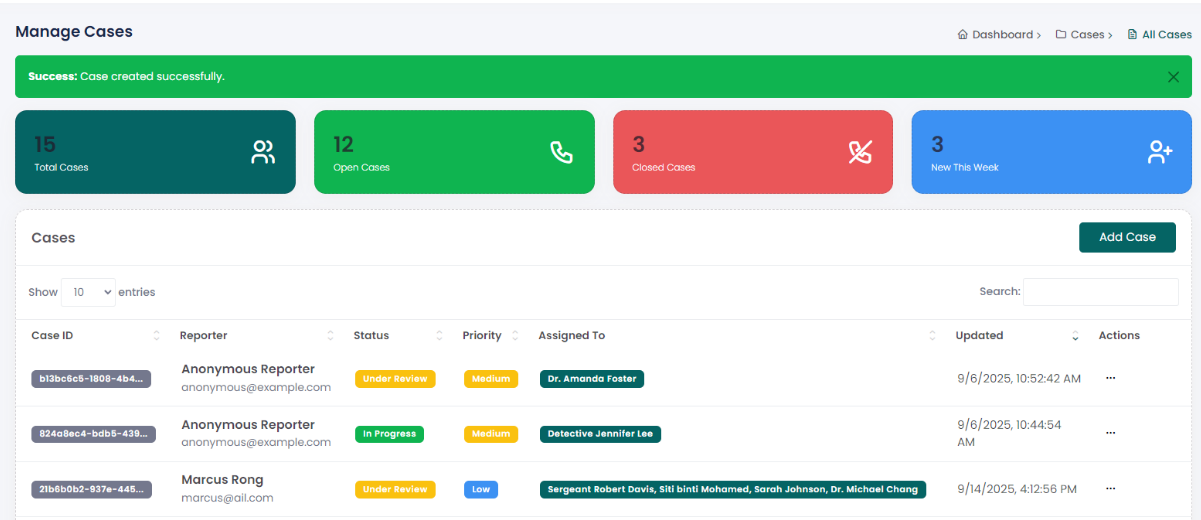


Figure 243: Success case creation message

If the user successfully fills the required field with the correct format and hit the Create Case button, the success alert will pop up like above figure to show the case creation is successful to user.

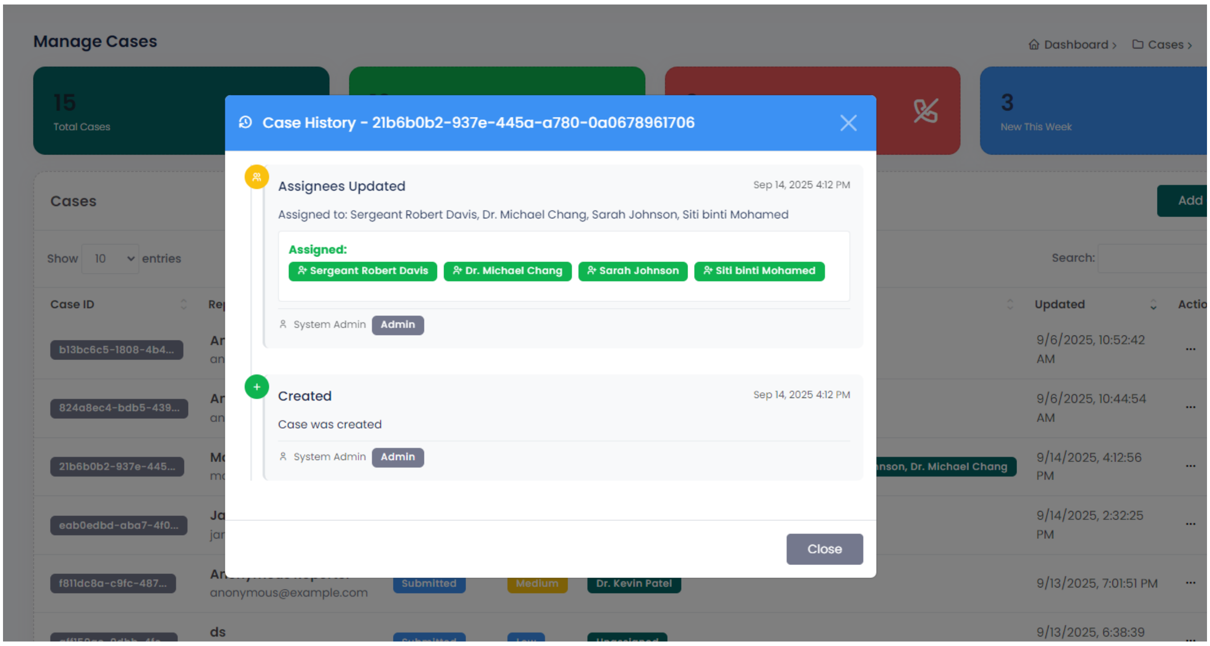


Figure 244: Case History

When the signed-in user clicks View History from the *Actions* dropdown in the case list, this Case History modal is displayed. It provides a chronological record of all activities related to the selected case. For example, it shows when the case was first created, including the date, time, and the role of the user. It also logs updates for instance assignee changes based on figure above, listing the individuals or roles assigned to handle the case, along with timestamps. This transparent history tracking ensures accountability, allowing stakeholders to trace every modification or assignment made to the case. Theres also Close button so user can just close the modal.

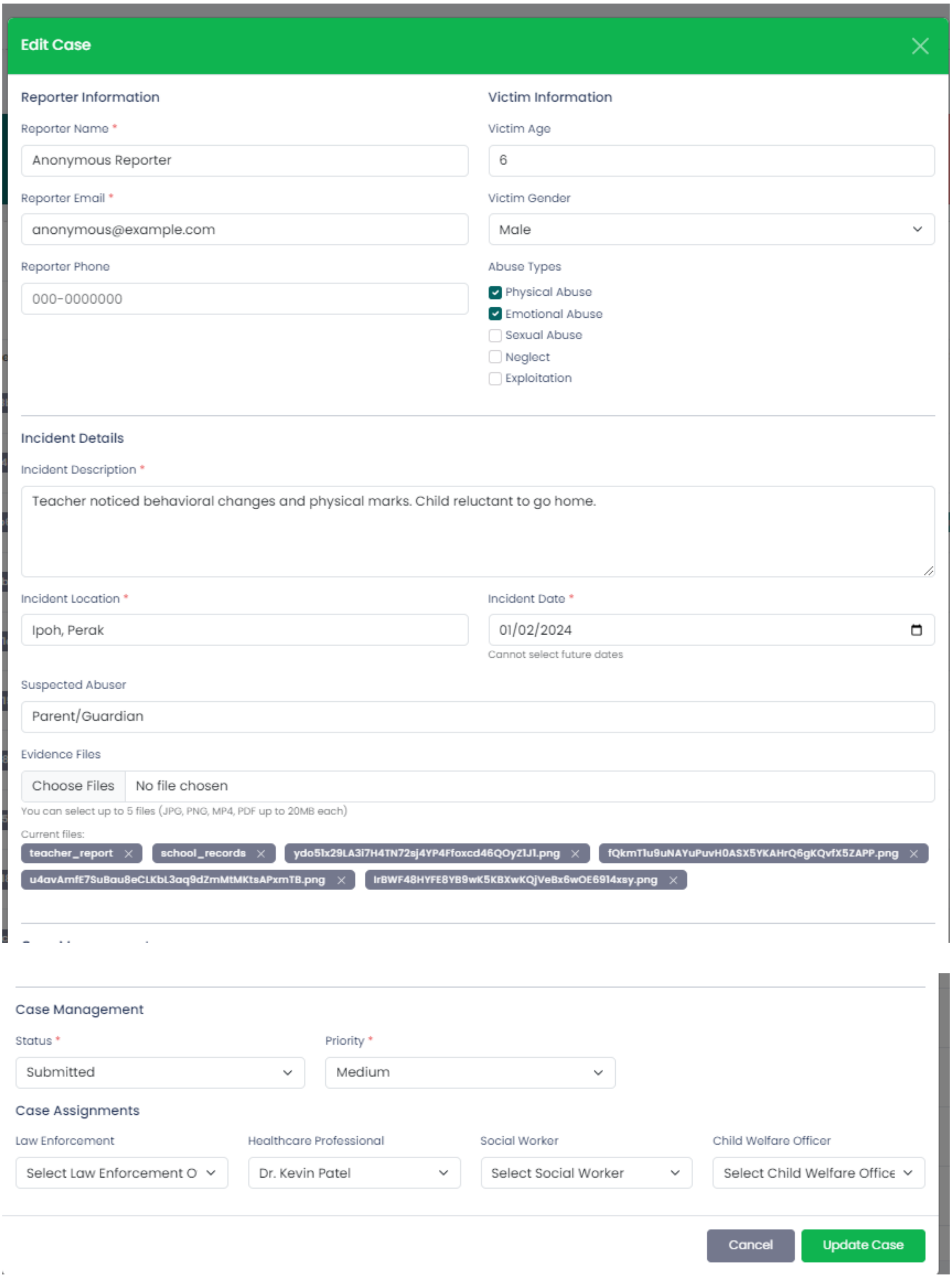


Figure 245: Edit case modal

The Edit Case modal provides functionality like the Add Case modal but is designed for modifying existing case records. Its pre-fills all previously entered information, such as reporter details, victim information, abuse types, incident description, location, date, suspected abuser, and any uploaded evidence. Users can update fields where necessary, including changing case status, adjusting priority, or reassigning the case to different professionals like law enforcement, healthcare, social workers, or child welfare officers. Existing evidence files are displayed with options to remove or add new ones. At the bottom, the Update Case button saves modifications, while the Cancel button exits without applying changes. This ensures that case information can be maintained accurately and kept up to date as investigations progress. The validation rules are consistent with the Add Case modal, requiring all mandatory fields to be properly filled and ensuring correct data formats through both browser and server-side validation, which helps maintain the accuracy and reliability of updates.

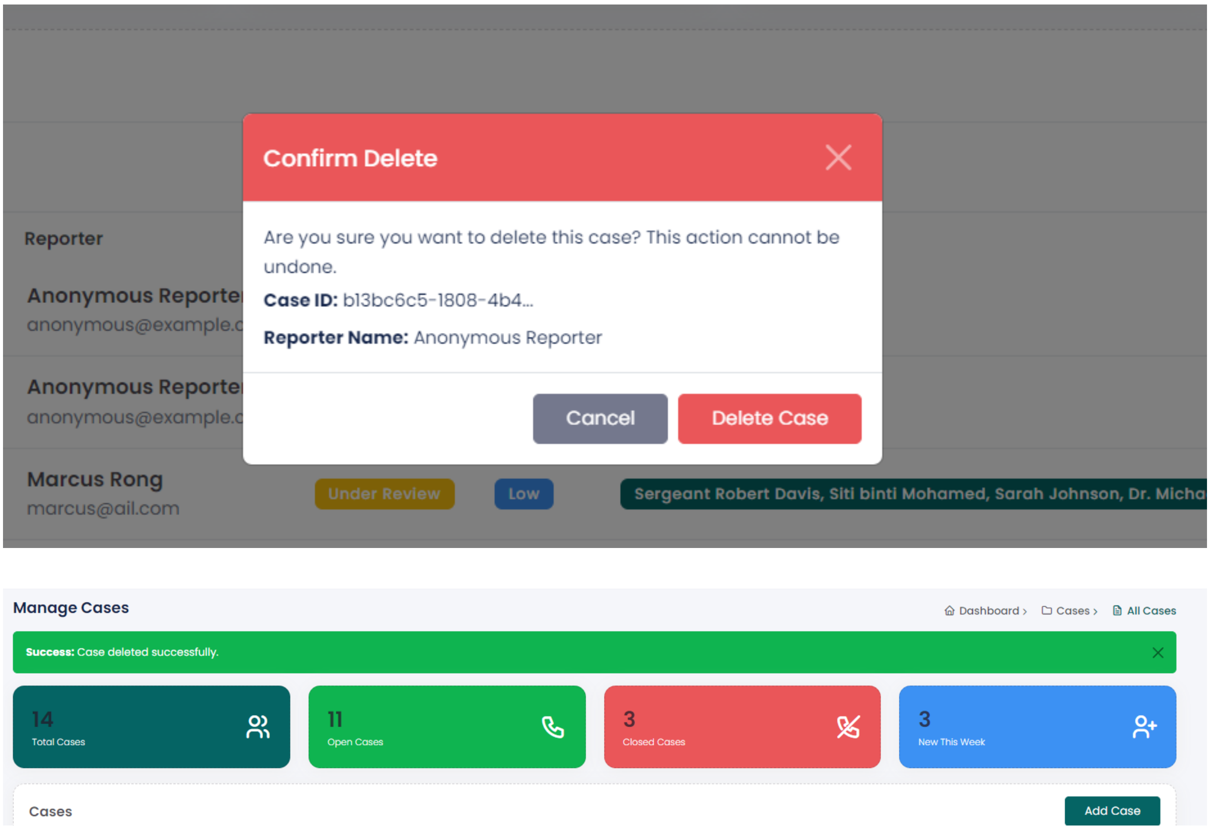


Figure 246: Delete case modal

Above is the delete confirmation modal if a user selected delete from the action dropdown from one of the cases. If the user clicks on Delete Case button, the success alert will pop up similar to figure above.

4.6.1.11.2 View Case Details

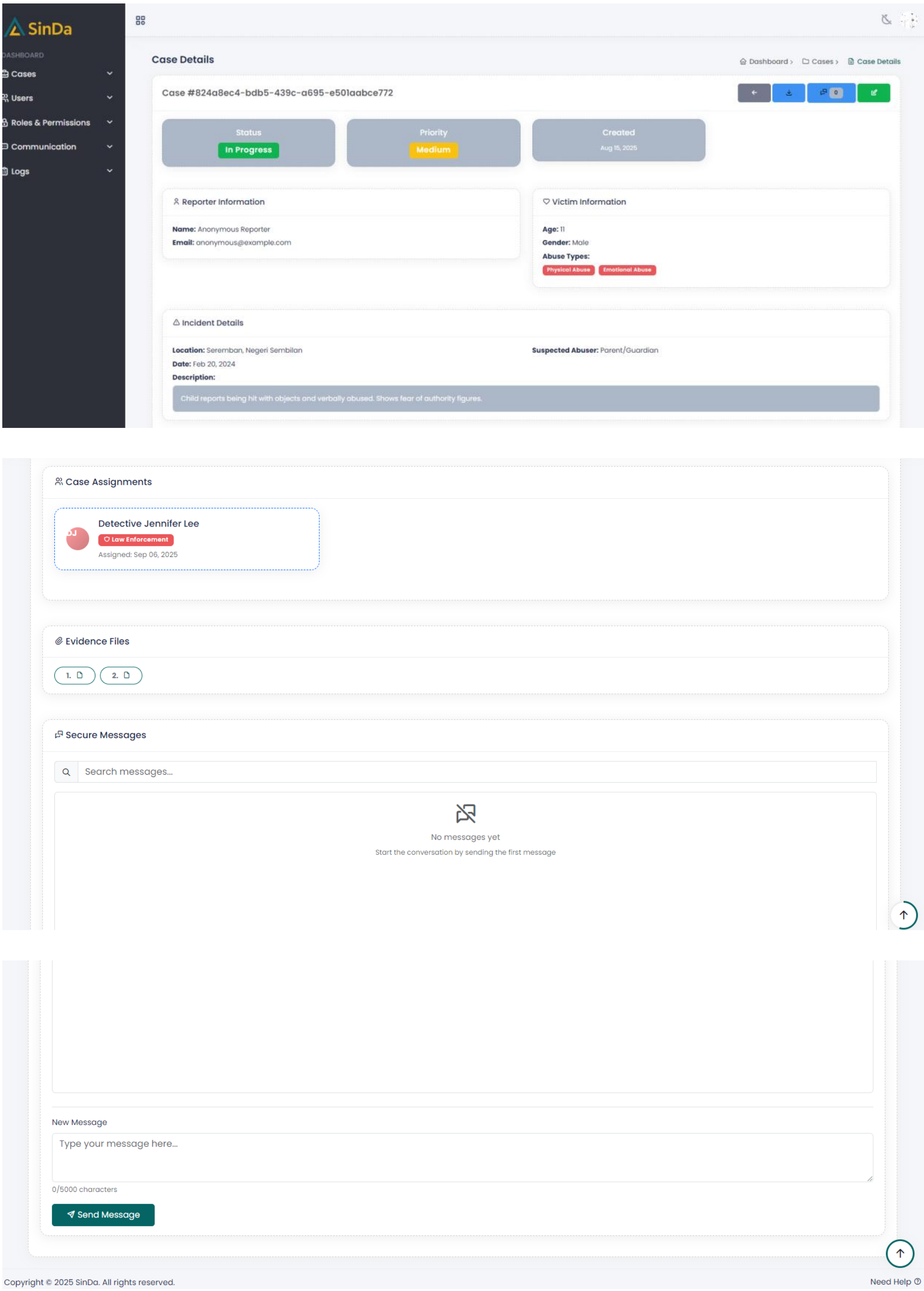
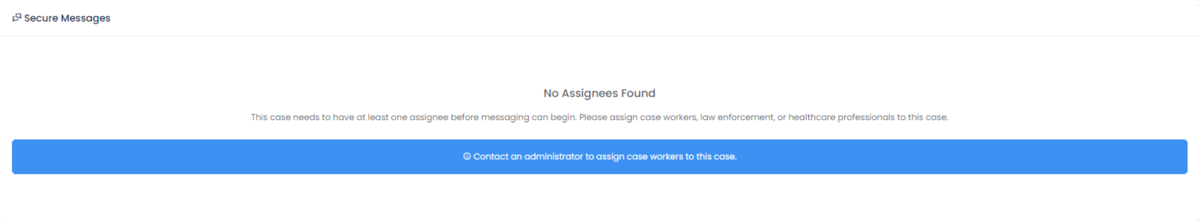


Figure 247: View Case Details

Above are the view case details page where users click on view case details button from the action dropdown. This page function as a centralized view of a specific report, displaying essential information such as case status, priority, and creation date at the top for quick reference. Reporter and victim details are clearly outlined, including contact information, age, gender, and abuse types, while the incident section highlights the location, date, description, and suspected abuser. Case assignments show which professional has been allocated to handle the case, along with their role and assignment date, ensuring accountability. Evidence files are organized for easy access, allowing investigators to review supporting documents or media directly. By clicking on one of the evidence, it will pop up a modal showing the selected evidence as well. At the bottom, a secure messaging feature enables authorized users to communicate within the case record, promoting collaboration and real-time coordination. This page also has quick action button, which is back button which by pressing it, the system will bring the user back to the list case page. Then there also export case in pdf button which export the selected case into pdf format, the message button which will help user to scroll to the secure messaging section, and lastly the edit case button which functions like the list case edit case modal.



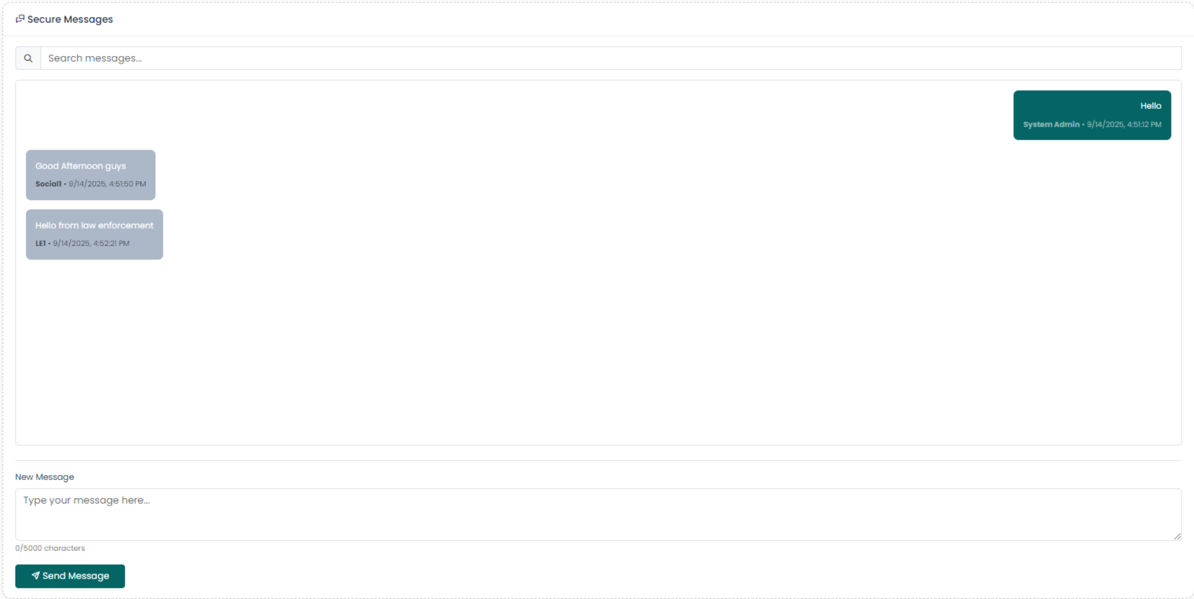


Figure 248: Secure messaging section

The Secure Messages section functions as a built-in communication tool within a case, allowing authorized users such as administrators, law enforcement, and welfare professionals to exchange messages directly. It features a searchable conversation history, displaying each message with the sender's role, name, and timestamp for clear tracking. Messages are displayed in a chat-style interface, where different roles are visually distinguished for easy readability. At the bottom, a text input box enables users to compose new messages with a character limit, and the Send Message button submits the entry instantly. If no assignee has been added to the case, the section instead displays a notice indicating that messaging cannot begin until at least one case worker or professional is assigned, with a prompt to contact the administrator for assignment. This ensures that communication is only initiated when the relevant parties are formally linked to the case, maintaining proper accountability and role-based access.

4.6.1.12 Admin Management

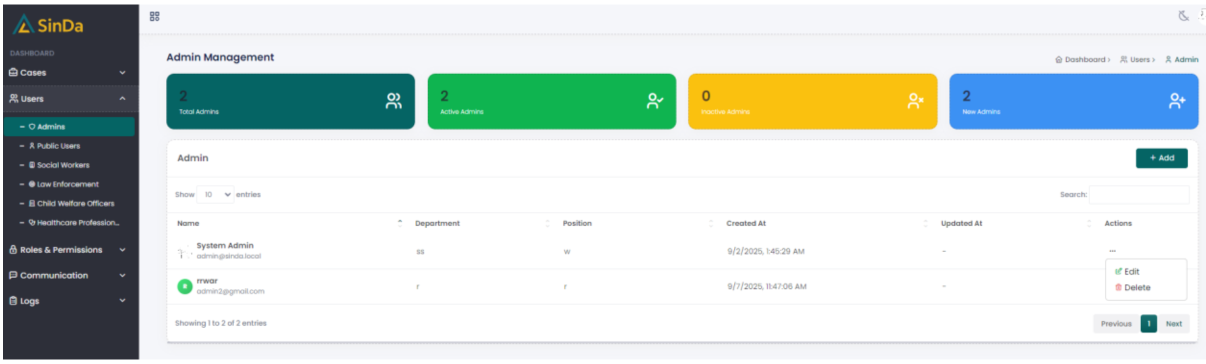


Figure 249: Admin management page

The above figure shows the Admin Management page, which provides an overview and control panel for managing system administrators. At the top, summary cards display the total number of admins, along with counts for active, inactive, and newly added accounts, offering a quick snapshot of the distribution. Below, a detailed table lists each admin's name, department, position, account creation date, last updated date, and includes an action dropdown with options to edit or delete the record. The table also supports search, pagination, and adjustable entry views for easier navigation. Additionally, a dedicated Add button is available to register new administrators directly, streamlining the process of account management. If there's no data available in this table, there will be message-like Figure 238 saying that there's no data available for this table.

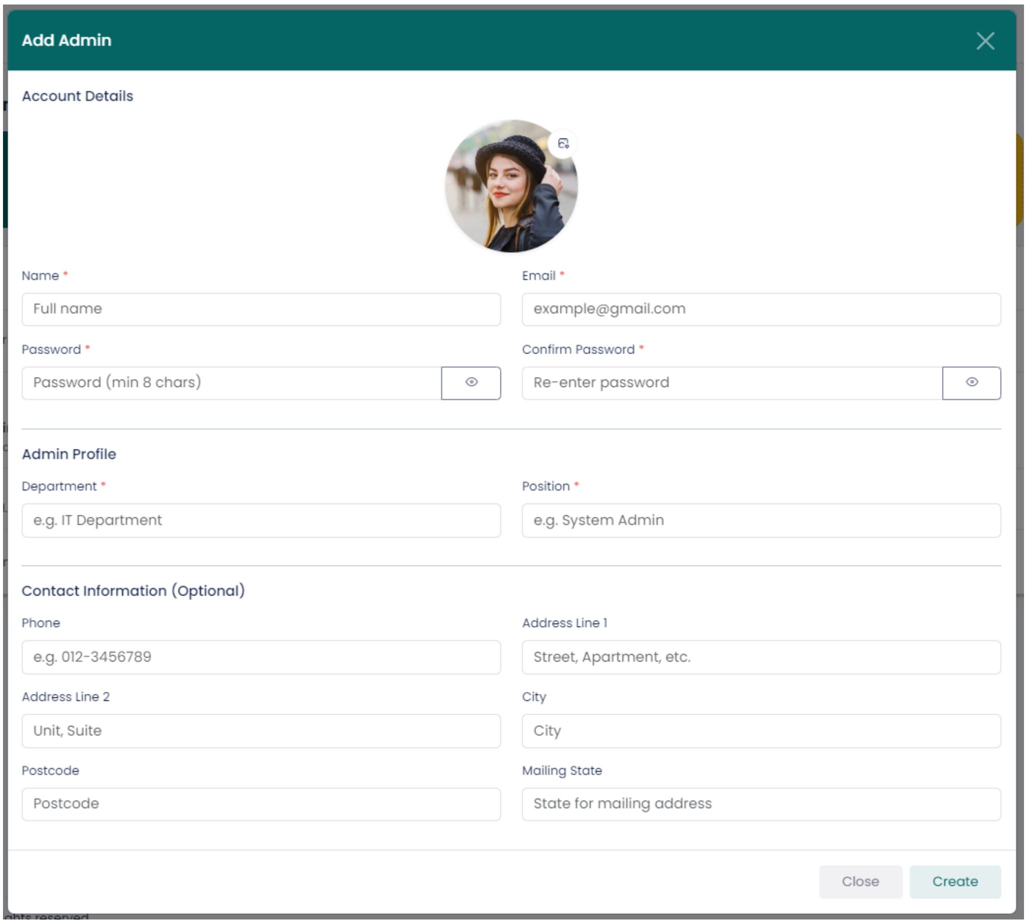
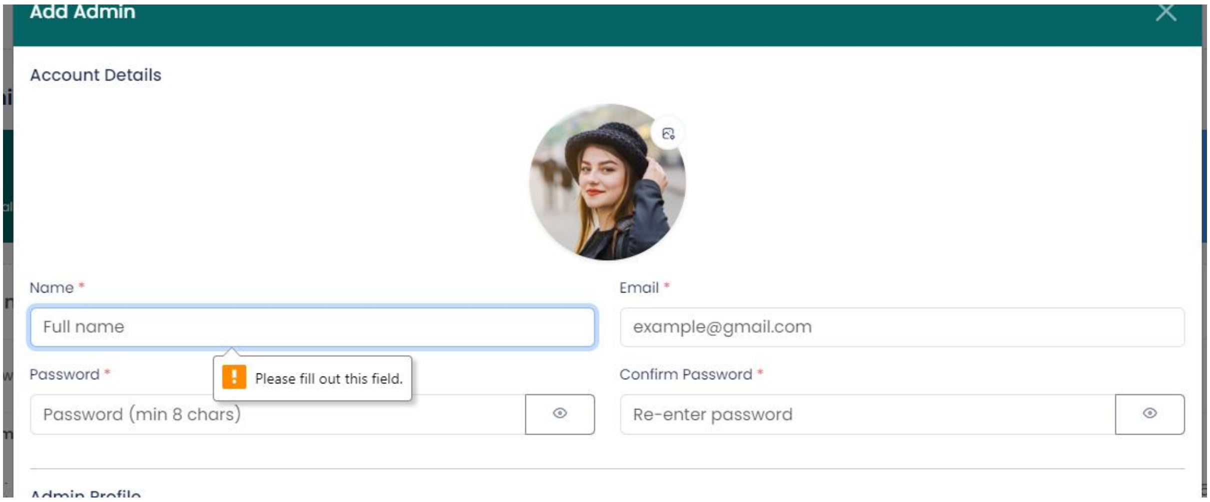
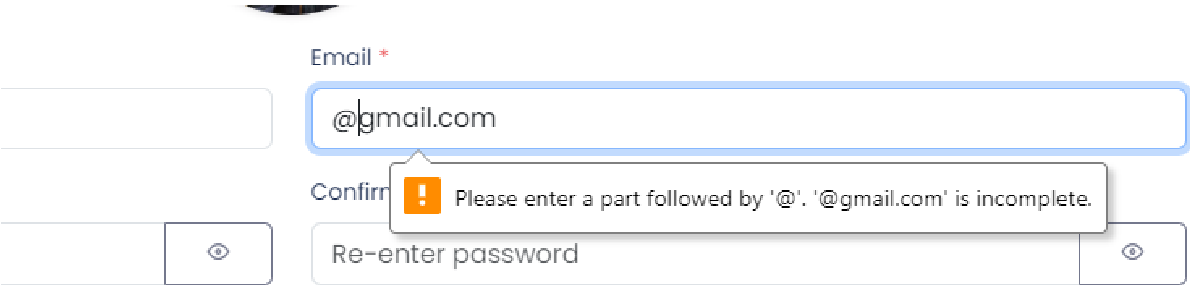
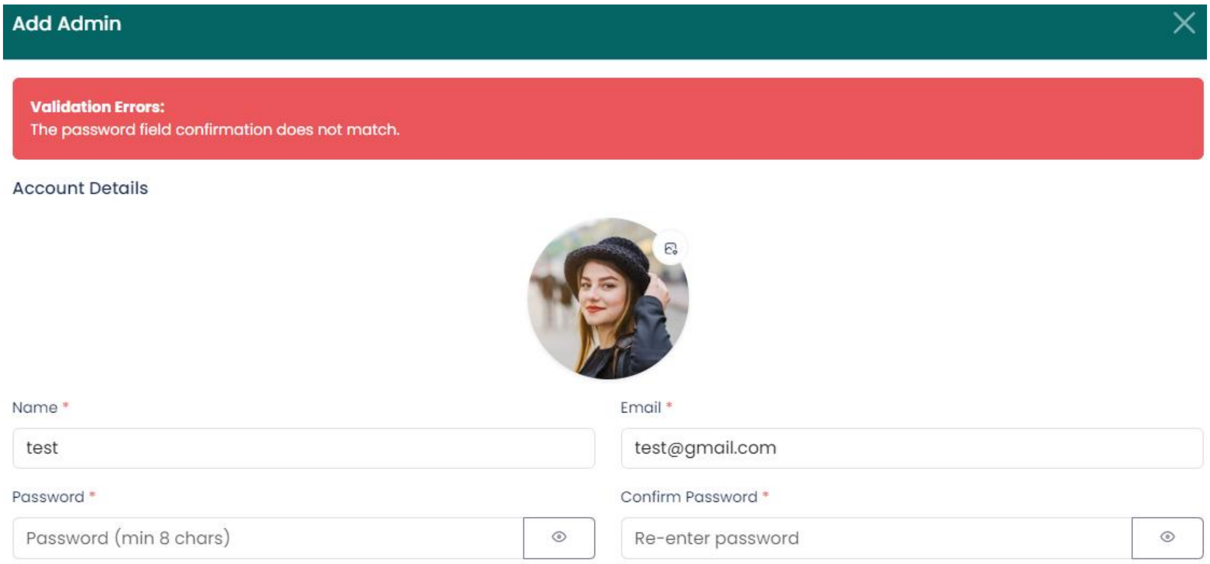


Figure 250: Add Admin modal

Above is the Add Admin modal where it would show up if the user clicked Add button. At the top user can select a profile picture and follow by mandatory field such as full name, email, password and confirm password. Besides that, the admin profile section is also required where user need to fill the department and position to help categorize the administrator role within the organization. Then, there’s an optional section where user can enter phone number which helps to auto format entered phone number into a proper format, address, city, postcode and state for additional record keeping if needed.


Add Admin
Account Details
Name *
Full name
Please fill out this field.
Email *
example@gmail.com
Password *
Password (min 8 chars)
Confirm Password *
Re-enter password


Email *
@gmail.com
Please enter a part followed by '@'. '@gmail.com' is incomplete.
Re-enter password


Add Admin
Validation Errors:
The password field confirmation does not match.
Account Details
Name *
test
Email *
test@gmail.com
Password *
Password (min 8 chars)
Confirm Password *
Re-enter password

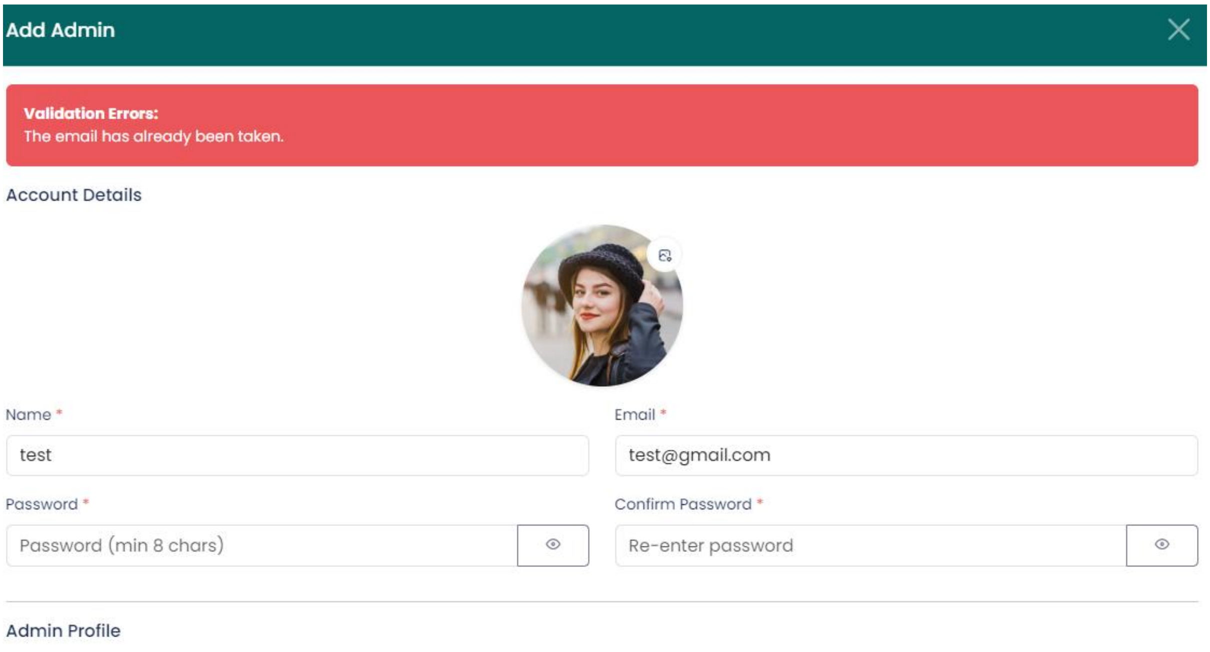


Figure 251: Add admin validations

Likewise, this modal also has applied a two-tier validation process that combines browser-side and server-side for required field based on above figure. If a user clicks the Create button without filling in the required fields, the browser validation immediately triggers, prompting messages such as above figure to guide proper input formatting. Once the form passes browser validation and is submitted, server-side validation takes place to enforce business rules, such as verifying that the password and confirmation match or ensuring the email has not already been taken

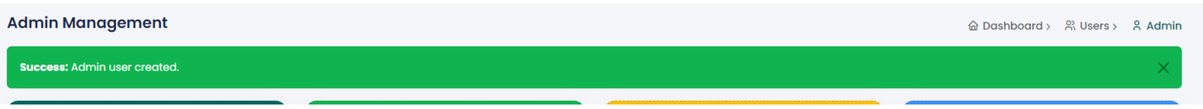


Figure 252: Add admin success message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above.

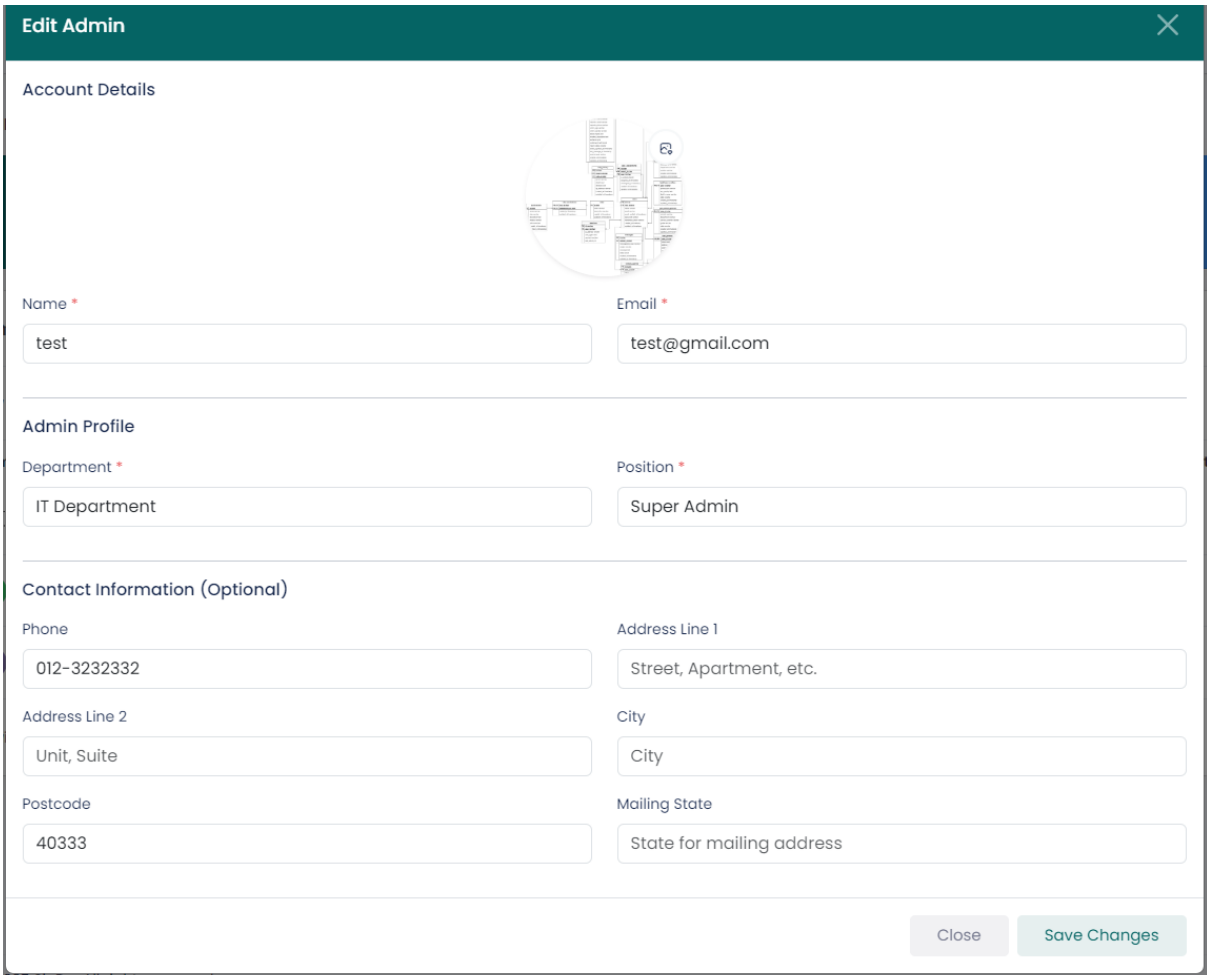


Figure 253:Edit admin modal

Above is the edit admin modal where it is prefilled with details based on above figures. This modal appears if the user has selected the Edit button from the action dropdown. In this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button.

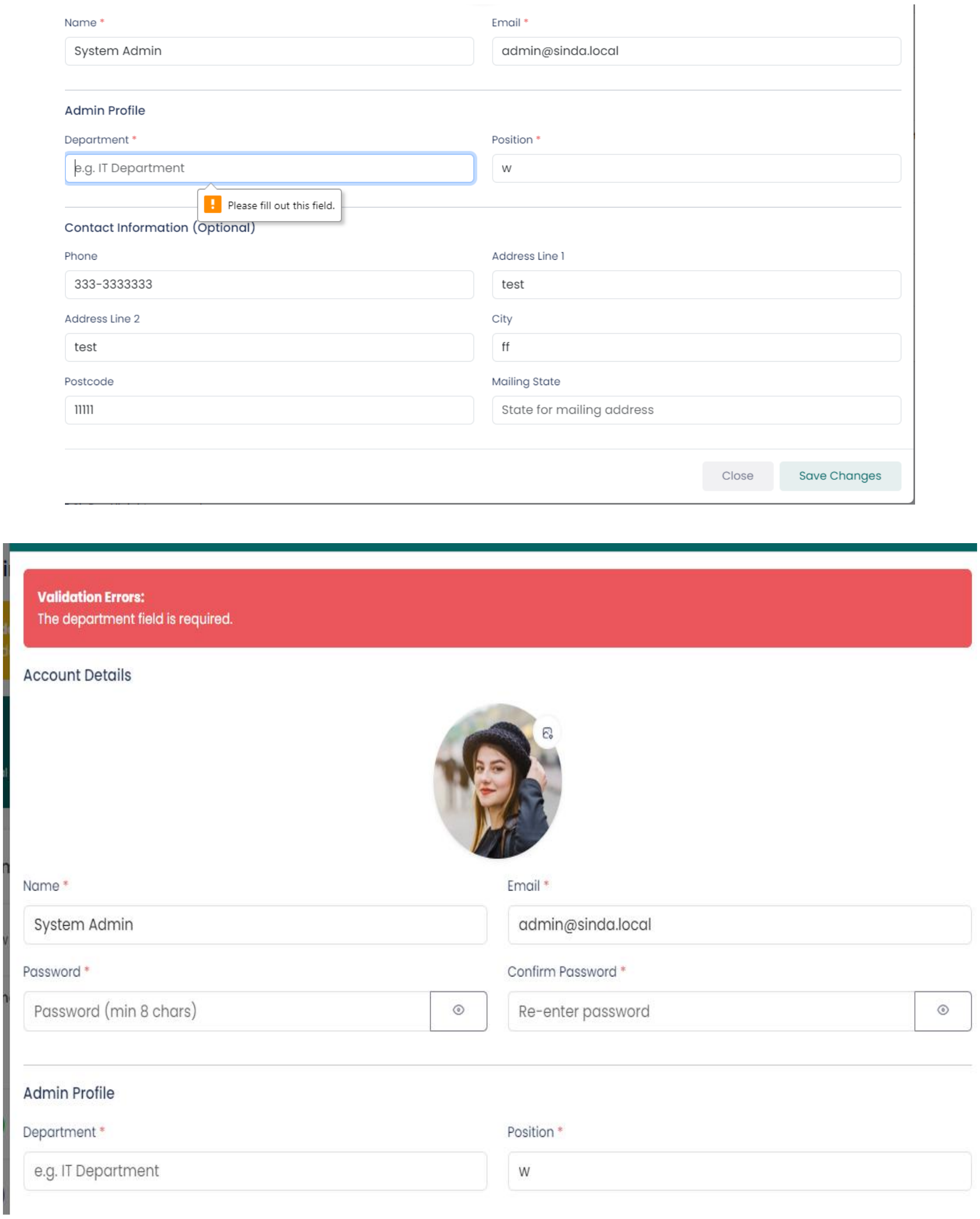


Figure 254: Edit admin modal validation

For the edit admin modal validation, it also a two-tier validation process consisting of browser-side and server-side checks. For example, based on figure above, if the user leaves the Department field which is one of the required field empty and clicks Save Changes, the browser validation immediately highlights the issue with a tooltip message and prevent the submission. Once browser checks are passed, the form undergoes server-side validation, where additional rules are enforced. In this case, the system displays an error message if the field remains empty or the user enter spaces as inputs, ensuring that updates cannot bypass critical requirements.

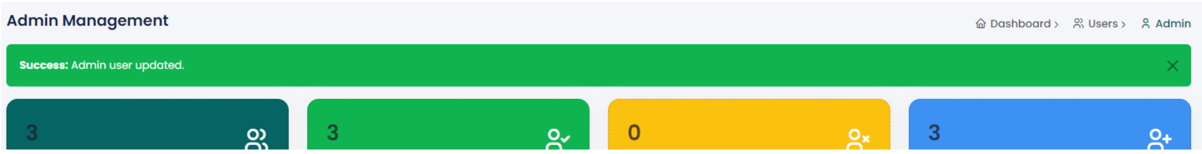


Figure 255: Admin changes success message

If the user appropriately changes the field according to the requirements and click on Save changes button, then the above message will pop up to confirm the changes to user.

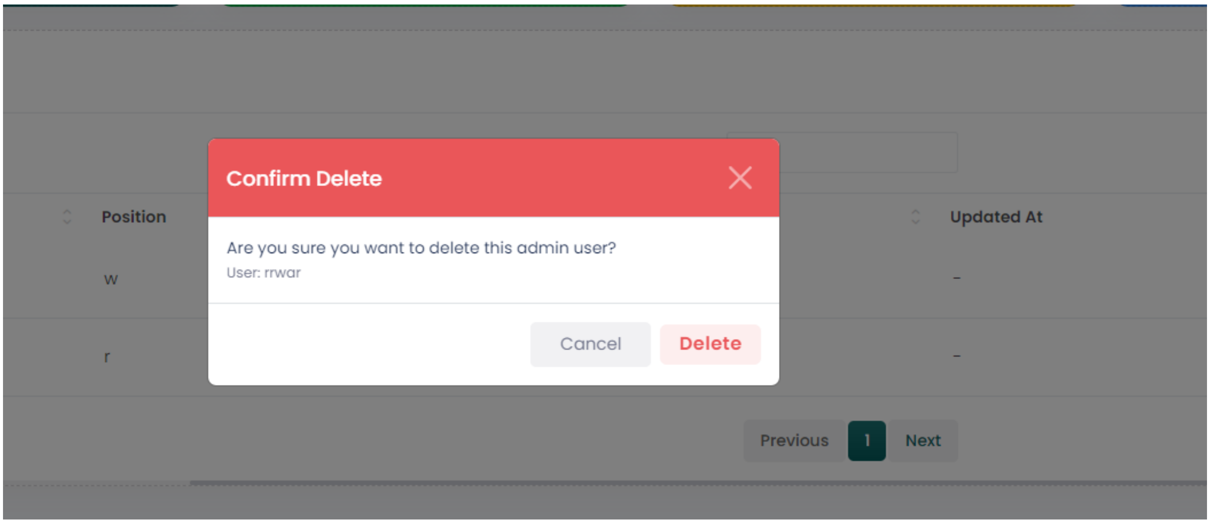


Figure 256: Delete admin confirmation modal

Above is a confirmation dialog that appears when the user clicked on delete button on the action dropdown. If the user wants to remove the selected administrator, the user may click on Delete button which delete the selected administrator from the system else if the user wants to close without deleting that administrator, the user can just click on Cancel to close the modal without deleting the selected administrator.

4.6.1.13 Public User Management

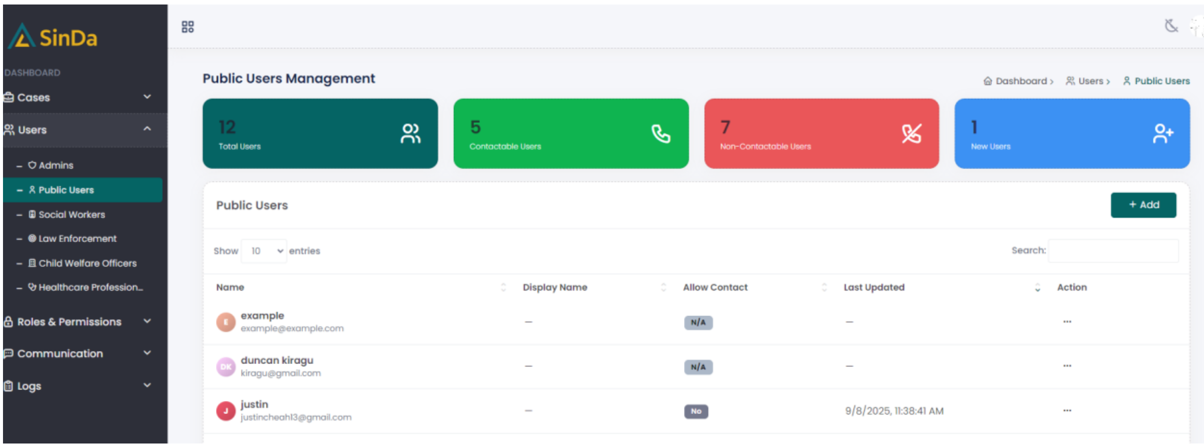


Figure 257: Public User Management page

The Public Users Management page provides administrators with an overview and control panel for managing general users of the system. At the top, summary cards display the total number of registered users, along with counts of contactable users, non-contactable users, and newly added users, giving a clear snapshot of user distribution. Below, a detailed table lists each public user’s name, email, display name, contact preference, last updated date, and an action dropdown for further management options. Additionally, a dedicated Add button is provided to register new public users directly, ensuring administrators can efficiently expand and manage the user base.

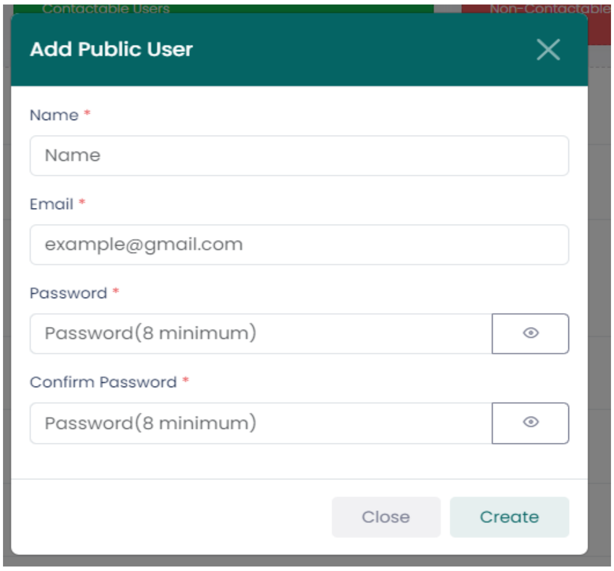
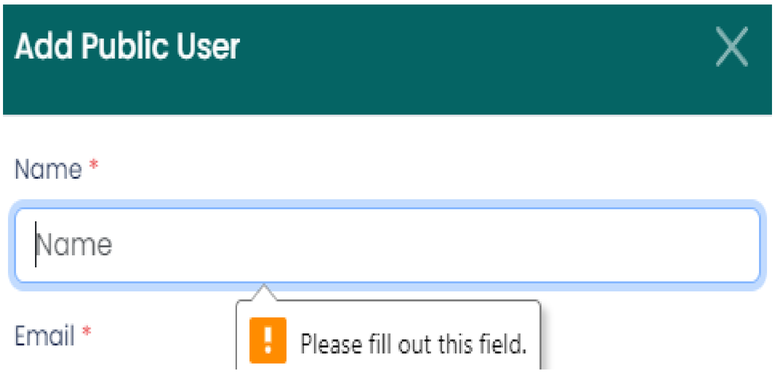
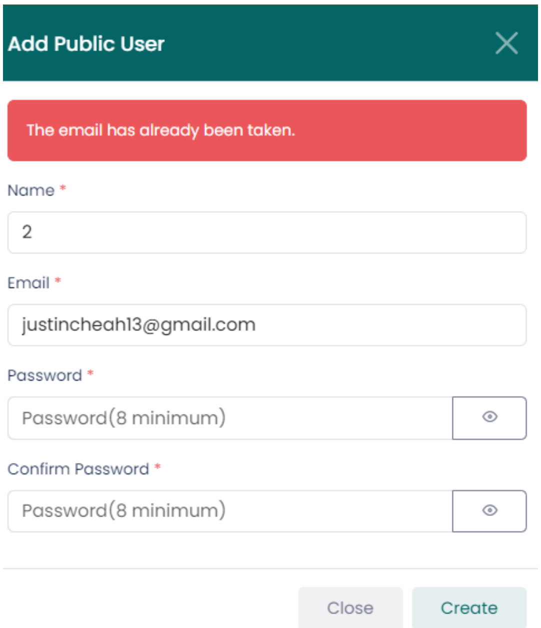


Figure 258: Add public user modal

The Add Public User modal allows administrators to register new users into the system by capturing essential account details. It includes mandatory fields such as Name, Email, Password, and Confirm Password, ensuring that each user has a unique and secure login. The password field requires a minimum of eight characters, and the confirmation field must match to prevent entry errors. At the bottom, the Create button finalizes the registration, while the Close button exits without saving.





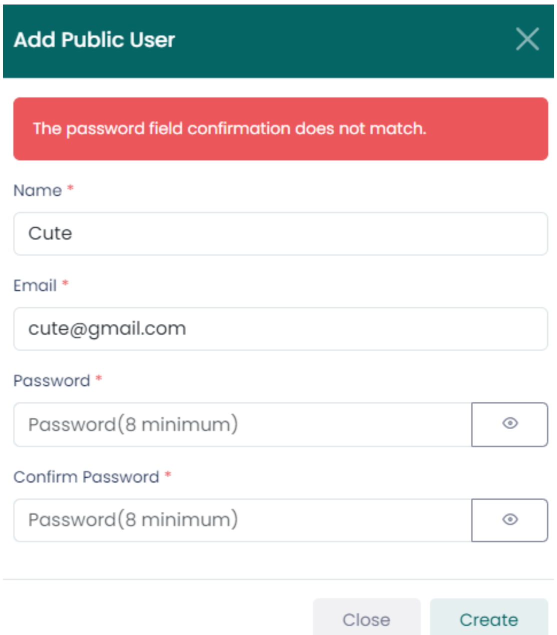


Figure 259: Add public user modal validations

The Add Public User validation ensures that only complete and correct information is accepted when creating a new account. It applies a two-tier validation process consisting of browser-side checks and server-side rules. For example, if a required field such as Name is left empty and the user clicks Create, browser validation immediately triggers with a tooltip base on figure above, preventing submission of incomplete forms. On the server side, deeper checks are enforced, such as verifying whether the Email is unique; if it already exists in the system, an error message is displayed. Similarly, when the Password and Confirm Password fields do not match, the system shows error message ensuring account credentials remain valid and secure.

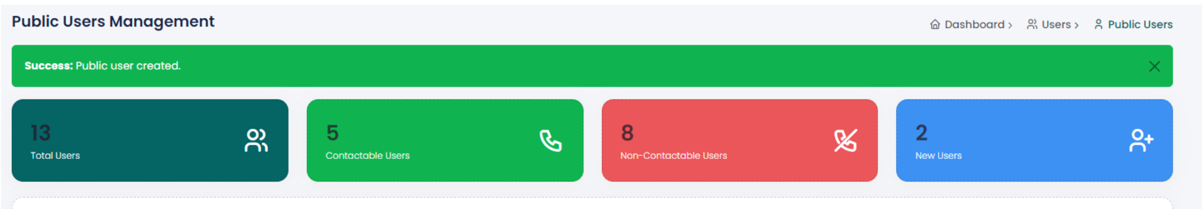


Figure 260: Add public user success message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above.

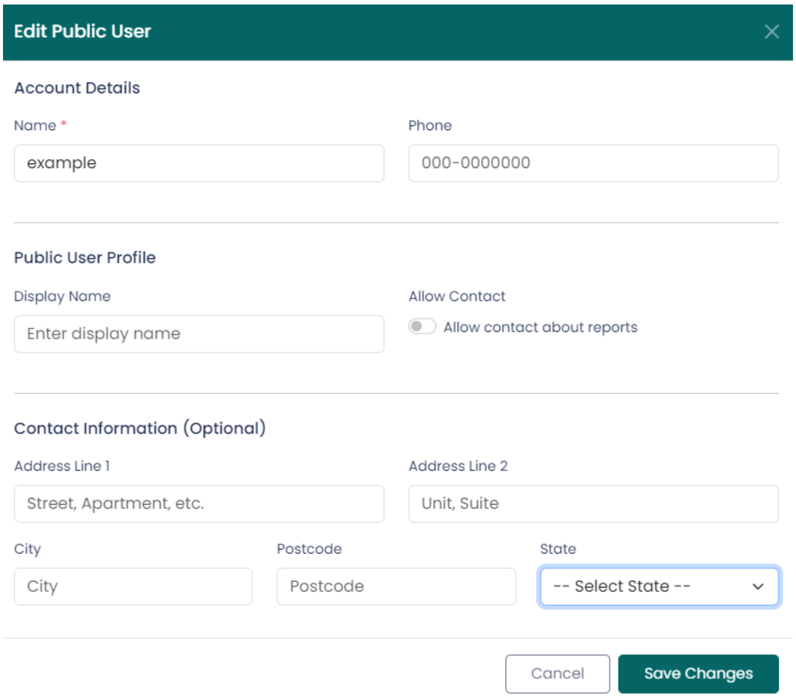


Figure 261: Edit public user modal

Above is the edit public user modal where it is prefilled with details based on above figures. This modal appears if the user has selected the Edit button from the action dropdown. In this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button. Like other forms, this modal follows a two-tier validation process, where browser validation ensures required fields such as the name are filled before submission, while server-side validation enforces consistency and accuracy of the data across the system. This ensures that public user records remain accurate, secure, and up to date.

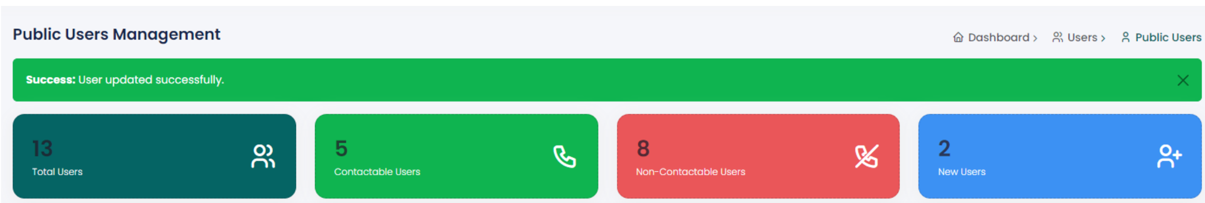


Figure 262: Public user changes success message

If the user appropriately changes the field according to the requirements and click on Save changes button, then the above message will pop up to confirm the changes to user.

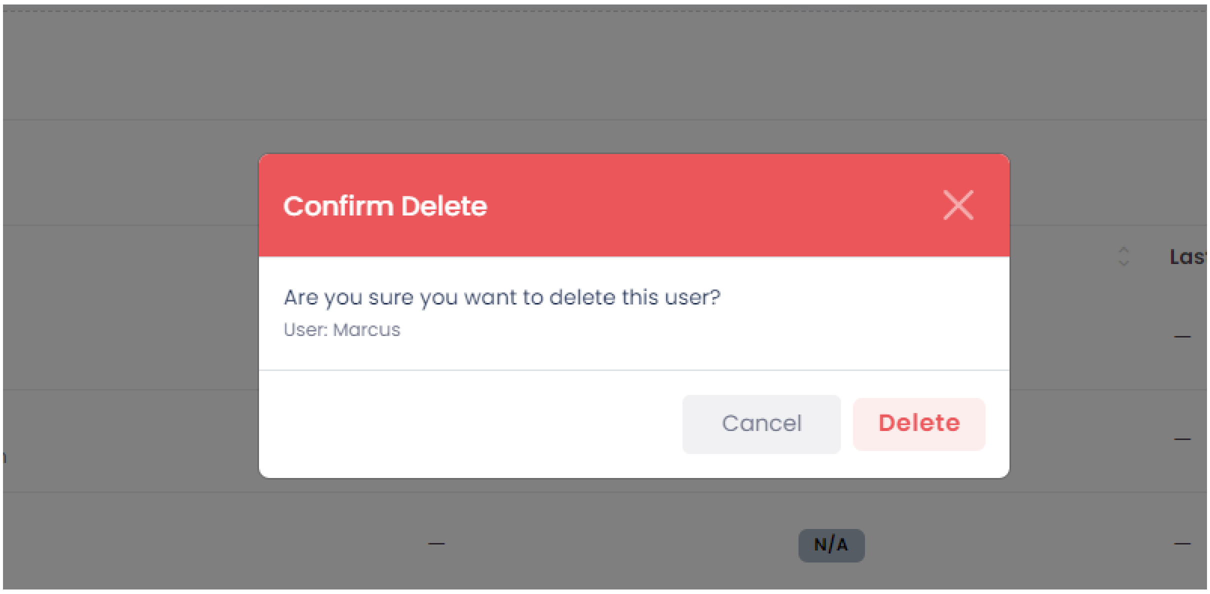


Figure 263: Public user confirm deletion modal

Above is a confirmation dialog that appears when the user clicked on delete button on the action dropdown. If the user wants to remove the selected public user, the user may click on Delete button which delete the selected public user from the system else if the user wants to close without deleting that administrator, the user can just click on Cancel to close the modal without deleting the selected public user.

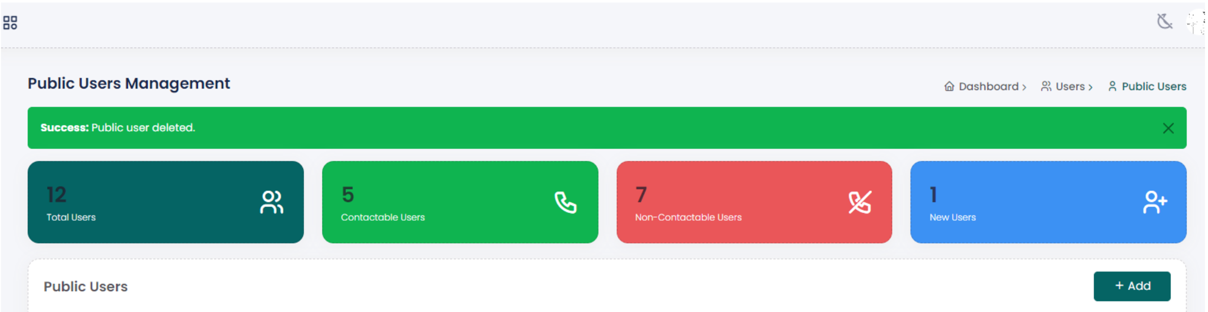


Figure 264: Delete public user successful message

If the user removes the selected public user, a green successful message will be displayed based on above figure.

4.6.1.14 Social Worker Management

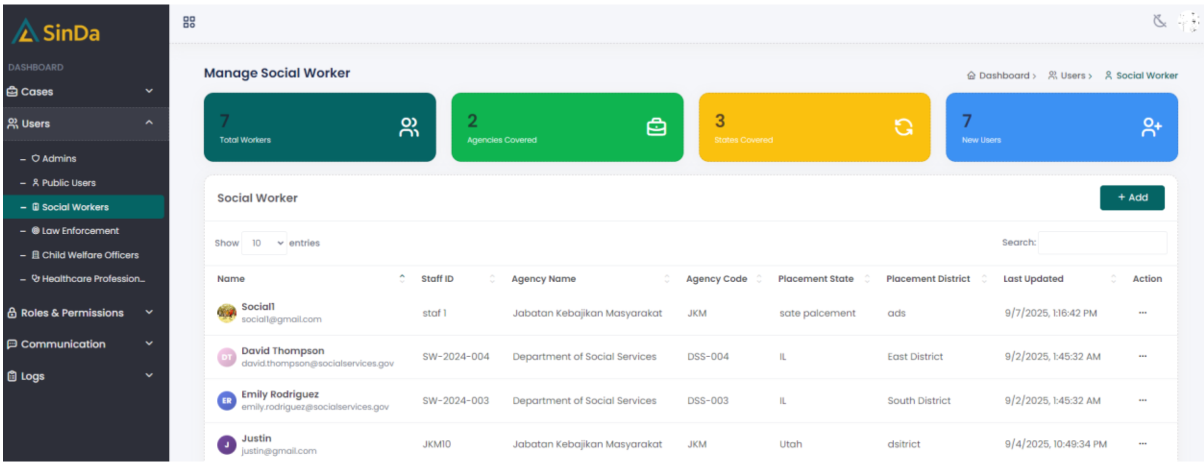


Figure 265: Manage social worker page

The above figure shows the Manage Social Worker page with tools to oversee and manage registered social workers in the system. At the top, summary cards highlight key statistics, including the total number of workers, the number of agencies covered, the number of states covered, and the count of newly added users. Below this, a detailed table lists each social worker along with information such as name, email, staff ID, agency name, agency code, placement state, placement district, and the date of the last update. Additionally, a dedicated Add button is available to register new a social worker directly, streamlining the process of account management. If there’s no data available in this table, there will be message-like Figure 238 saying that there’s no data available for this table.

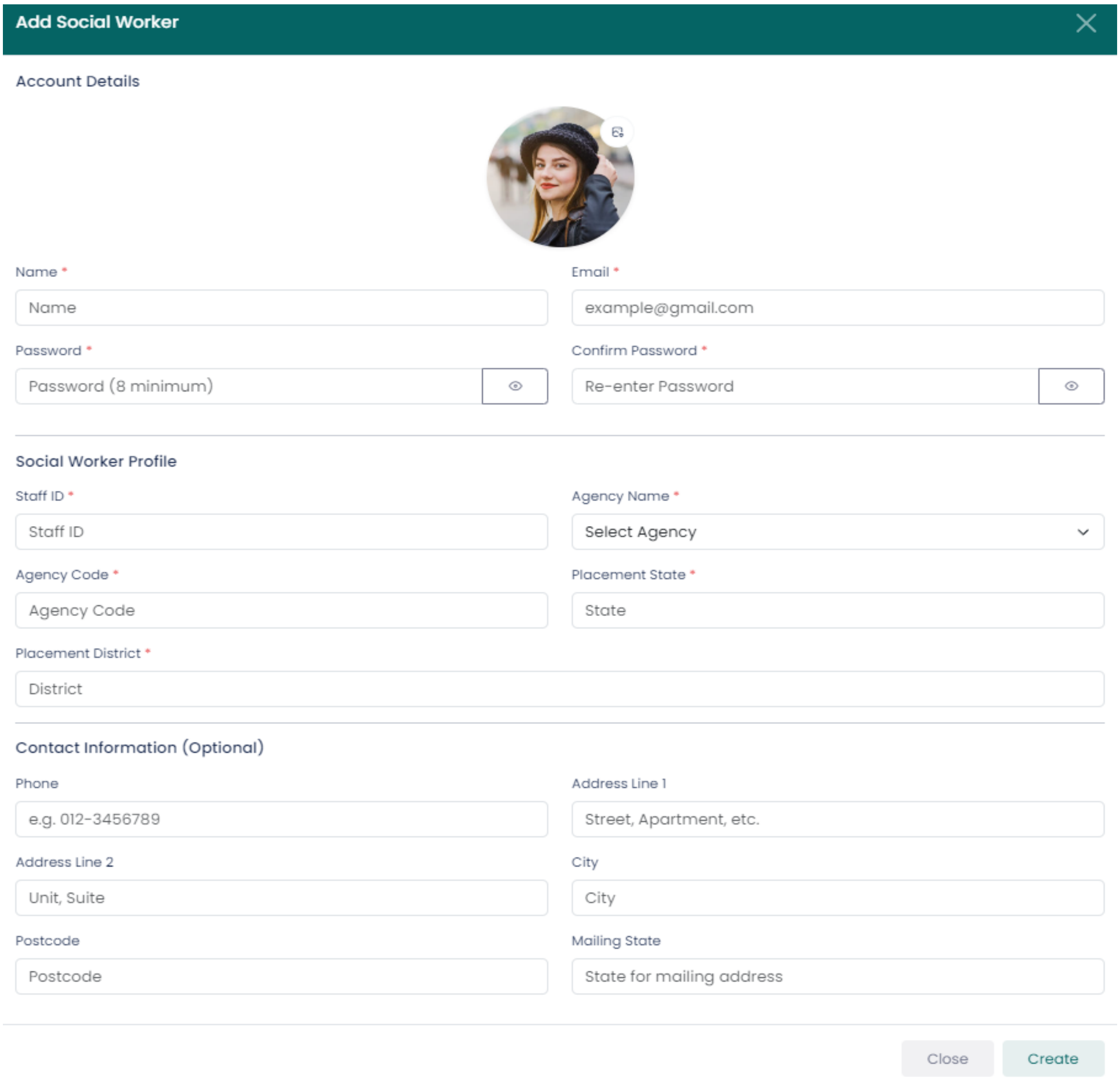


Figure 266: Add social worker modal

Like the rest of the add user modal, these modal pops up when the user clicked Add button. At the top, it requires account details such as name, email, password, and confirm password, ensuring secure login credentials. The Social Worker Profile section captures essential professional information, including staff ID, agency name which is selected from the dropdown, agency code, placement state, and placement district, which are critical for assigning responsibilities and tracking coverage. An additional Contact Information section is provided as optional, allowing entry of phone number, address, city, postcode, and mailing state for more comprehensive records. At the bottom, the Create button finalizes the registration, while the Close button exits without saving changes. Like other registration forms, this modal applies a two-tier validation process, where browser validation ensures all mandatory fields are completed with correct formatting, and server-side validation enforces business rules such as email uniqueness and password confirmation. This ensures accurate, secure, and reliable onboarding of social workers into the system.

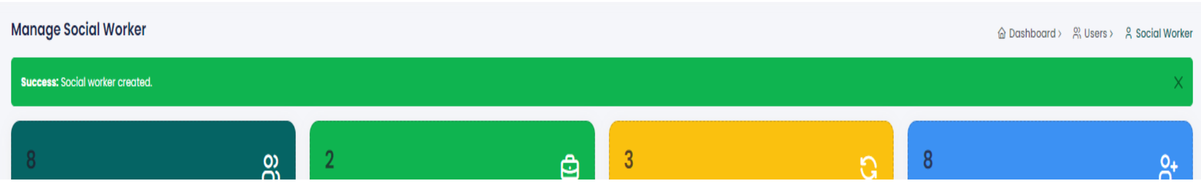


Figure 267: Social worker addition successful message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above.

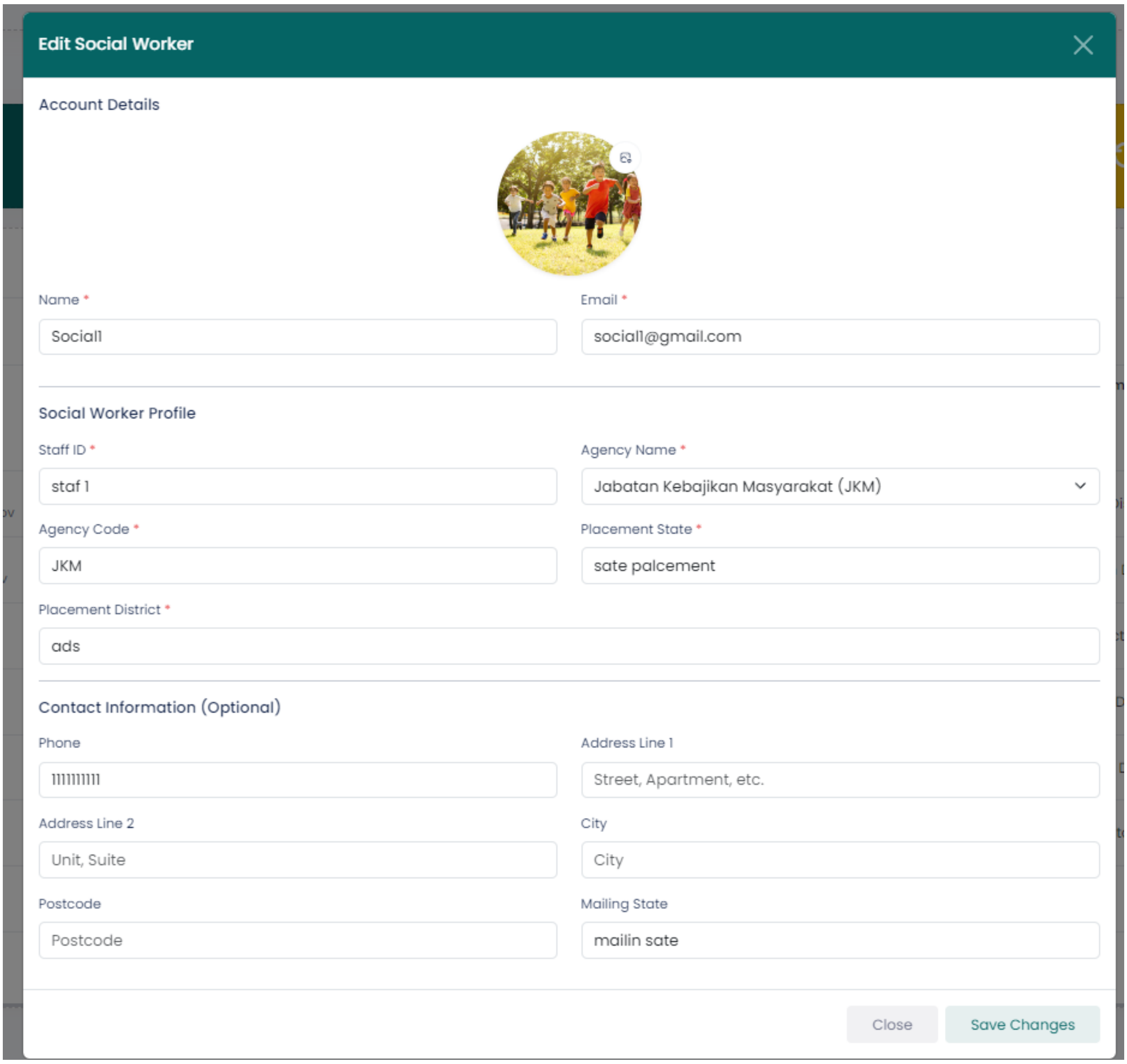


Figure 268: Edit social worker modal

Above is the edit social worker modal where it is prefilled with details based on above figures. This modal appears if the user has selected the Edit button from the action dropdown. In this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button. Like other forms, this modal follows a two-tier validation process, where browser validation ensures required fields such as the name, Staff ID or the rest are filled correctly before submission, while server-side validation enforces consistency and accuracy of the data across the system. This ensures that social worker records remain accurate, secure, and up to date.

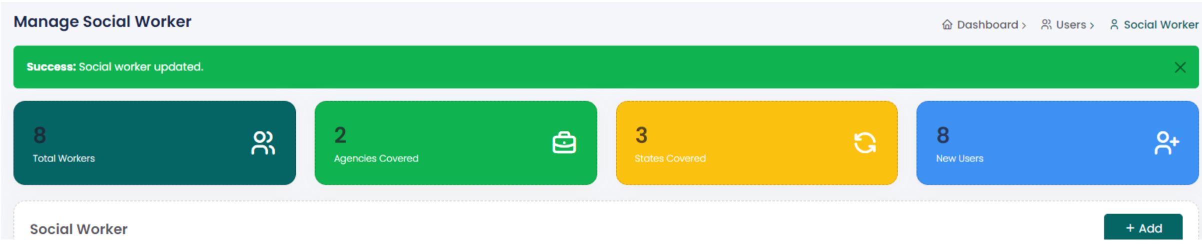


Figure 269: Social worker changes successful message

Once any changes are made correctly by user and use clicked the Save Changes button, there will be green success message that will be displayed based on the above figure.

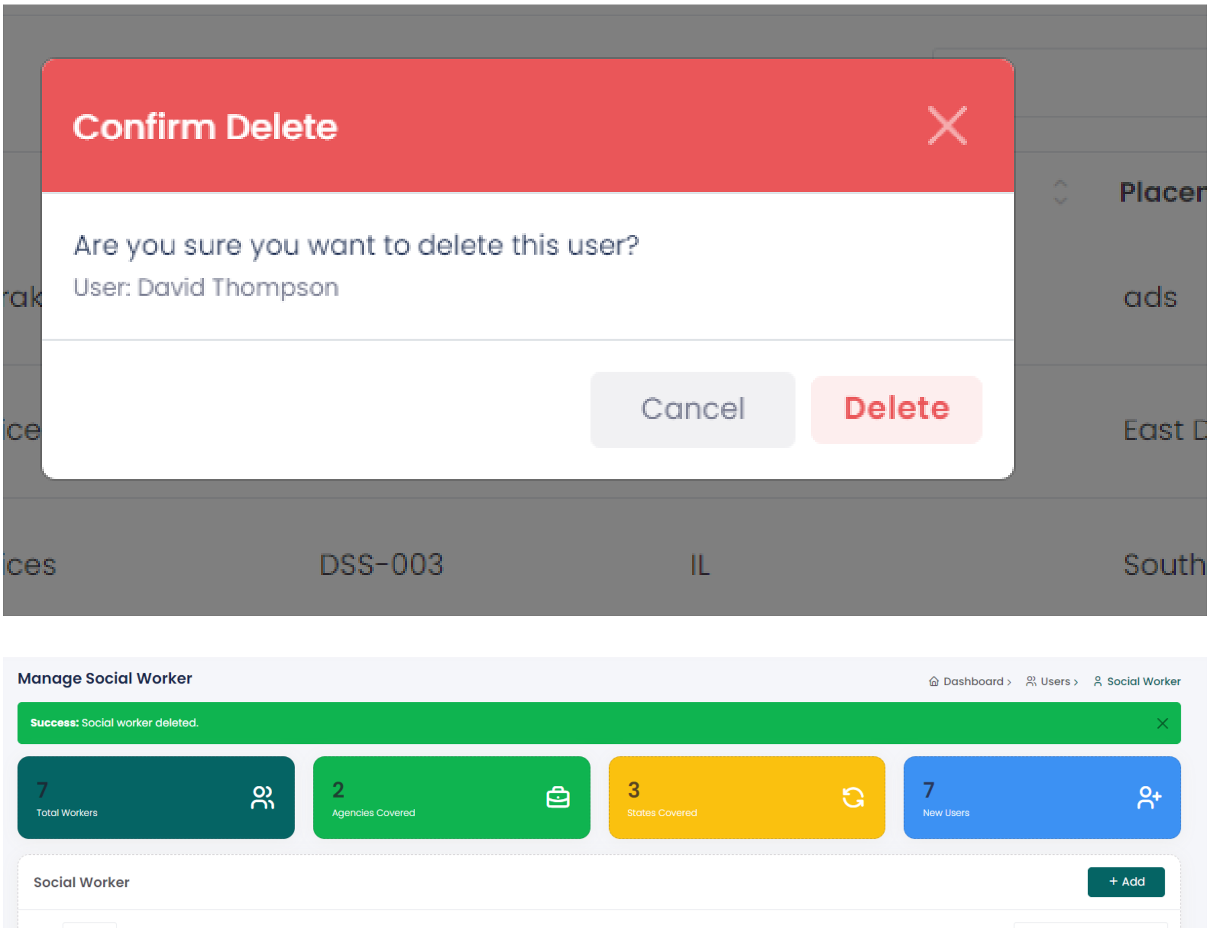


Figure 270: Delete social worker modal and successful message

Like other user management functionalities, if user selected the Delete button form the action dropdown, the above figure modal will pop up to let user confirm the deletion. If user clicked on the Delete button, the green successful message will be displayed, and the user will get deleted from the system. However, if user clicked on Cancel, the modal would close and nothing will happen to that selected social worker.

4.6.1.15 Law Enforcement Management

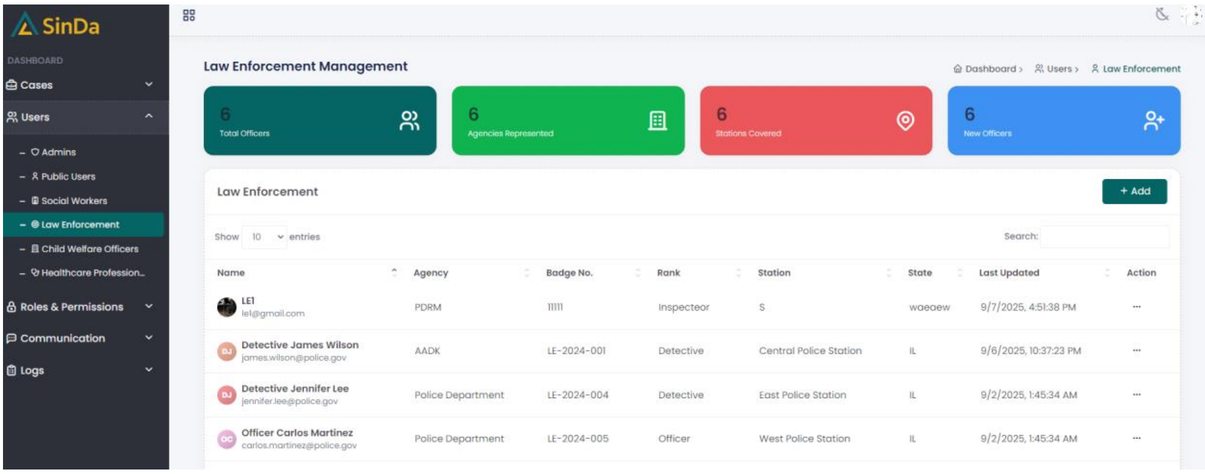


Figure 271: Law enforcement management page

The above page provides users with an organized interface to manage all related law enforcement personnel. At the top, summary cards display key statistics including the total number of officers, agencies represented, stations covered, and newly added officers, giving a quick overview of system records. Below this, a detailed table lists each officer with essential details such as name, email, agency, badge number, rank, assigned station, state, the date of the last update and the actions. On the right, the Add button allows users to register new law enforcement users into the system.

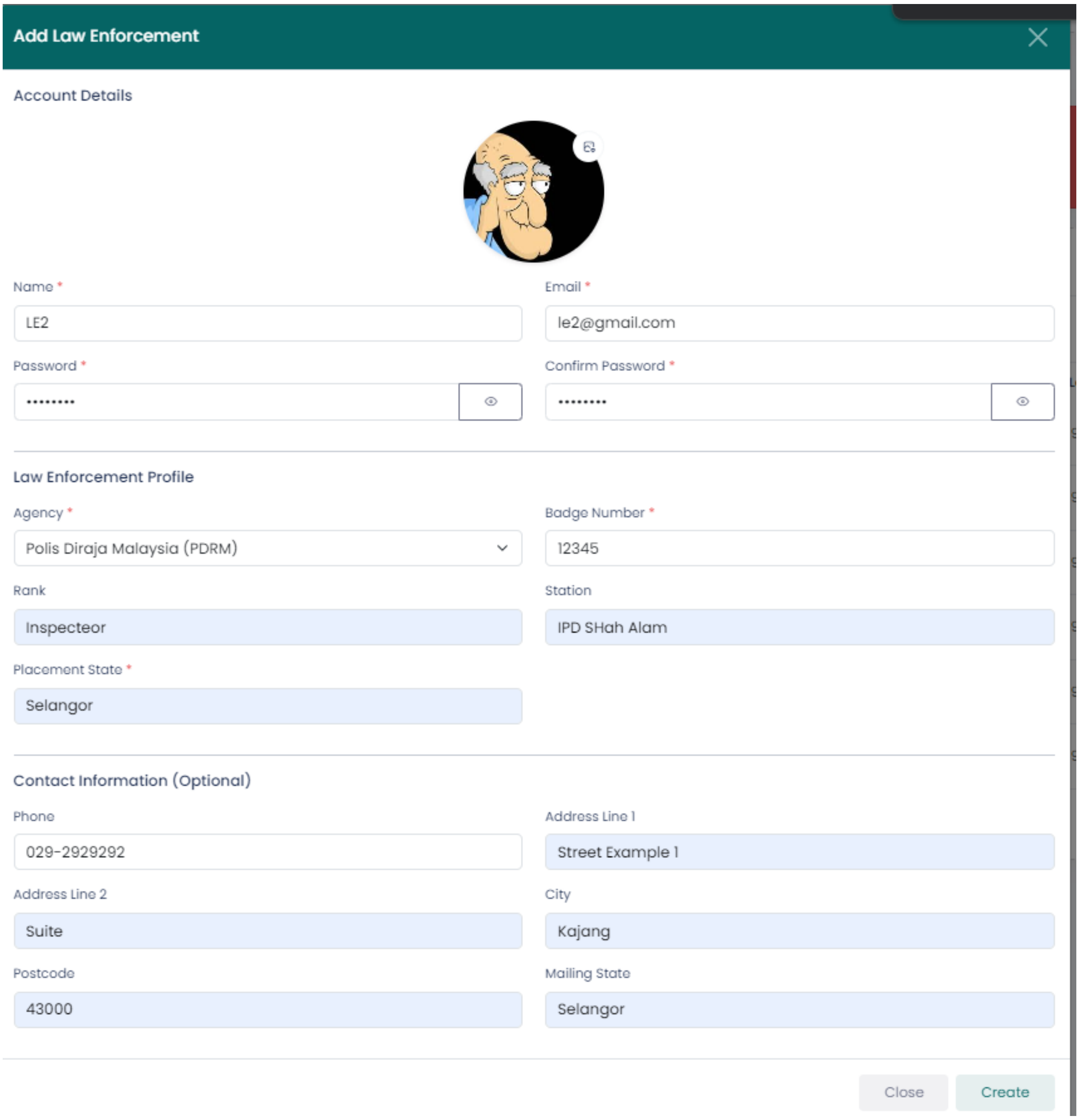


Figure 272: Add law enforcement modal

Based on above figure, it is the form to add new law enforcement into the system. At the top, the Account Details section requires essential inputs such as name, email, password, and confirm password to establish secure login credentials. The Law Enforcement Profile section captures officer-specific information, including agency, badge number, rank, station, and placement state, ensuring that each officer is properly identified and linked to their respective unit. Additionally, the Contact Information section is optional, allowing for the entry of phone numbers, addresses, city, postcode, and mailing state to maintain complete records if necessary. If the user filled up the necessary field and to continue the creation, the user could finalize the addition by clicking the Create button, else if the user doesn't want too, users can click on the close button to close the form. As for the validation, like the rest this modal follow the two-tier validation process, where browser validation ensures the mandatory fields such as name, email address, password, confirm passwords and the others with asterisks are filled properly, while server-side check for unique credentials, valid badge numbers, and secure password confirmation.

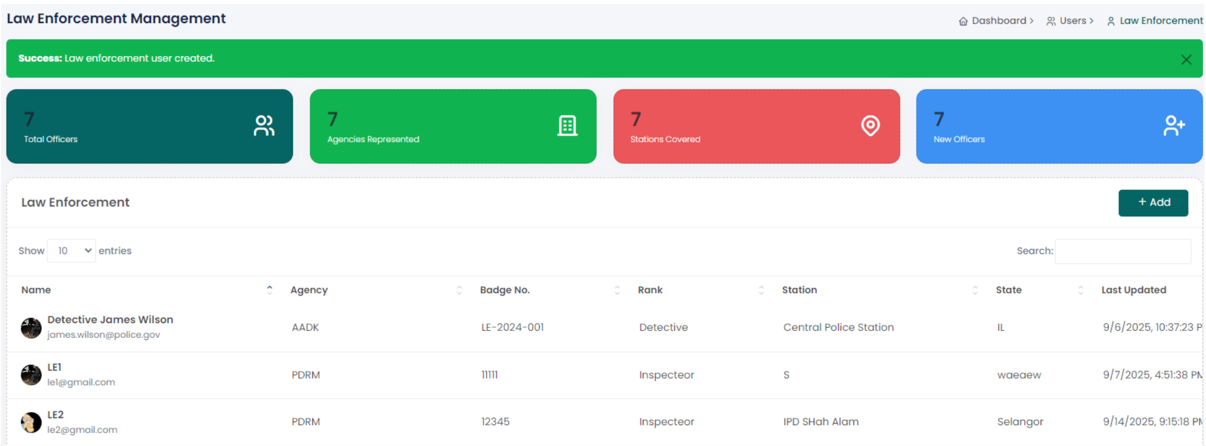


Figure 273: Add law enforcement success message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above and the created user will be in the table.

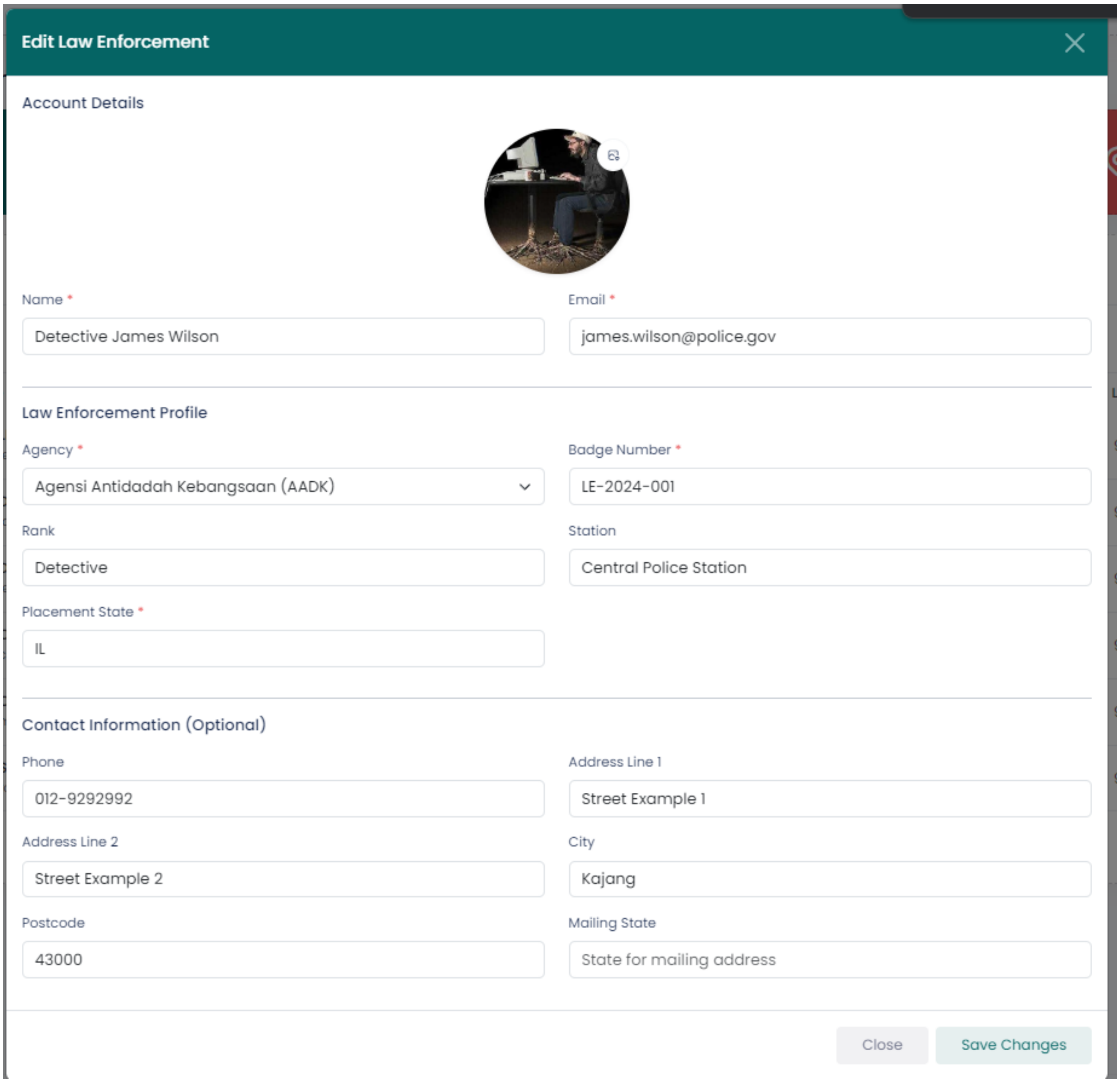


Figure 274: Edit law enforcement modal

Above is the edit law enforcement modal which are prefilled with fields where it appears if users click on Edit button on any of the law enforcement action dropdown. Like others, for this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button. Like other forms, this modal follows a two-tier validation process, where browser validation ensures required fields such as the name, email address, placement state and the rest that have asterisk are filled correctly before submission, while server-side validation enforces consistency and accuracy of the data across the system. This ensures that law enforcement records remain accurate, secure, and up to date.

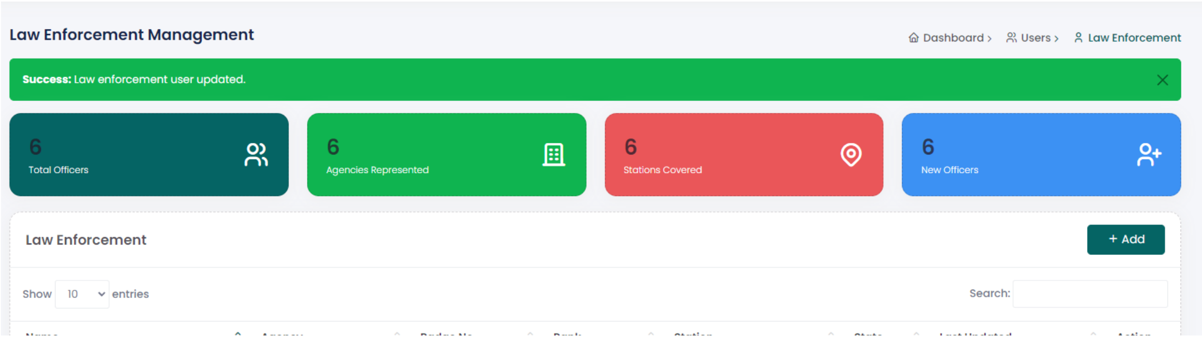


Figure 275: Law enforcement changes successful message

Now if the user saves the changes correctly in the edit law enforcement modal and click on Save Changes button, then there will be green successful message that will be displayed based on the figure above.

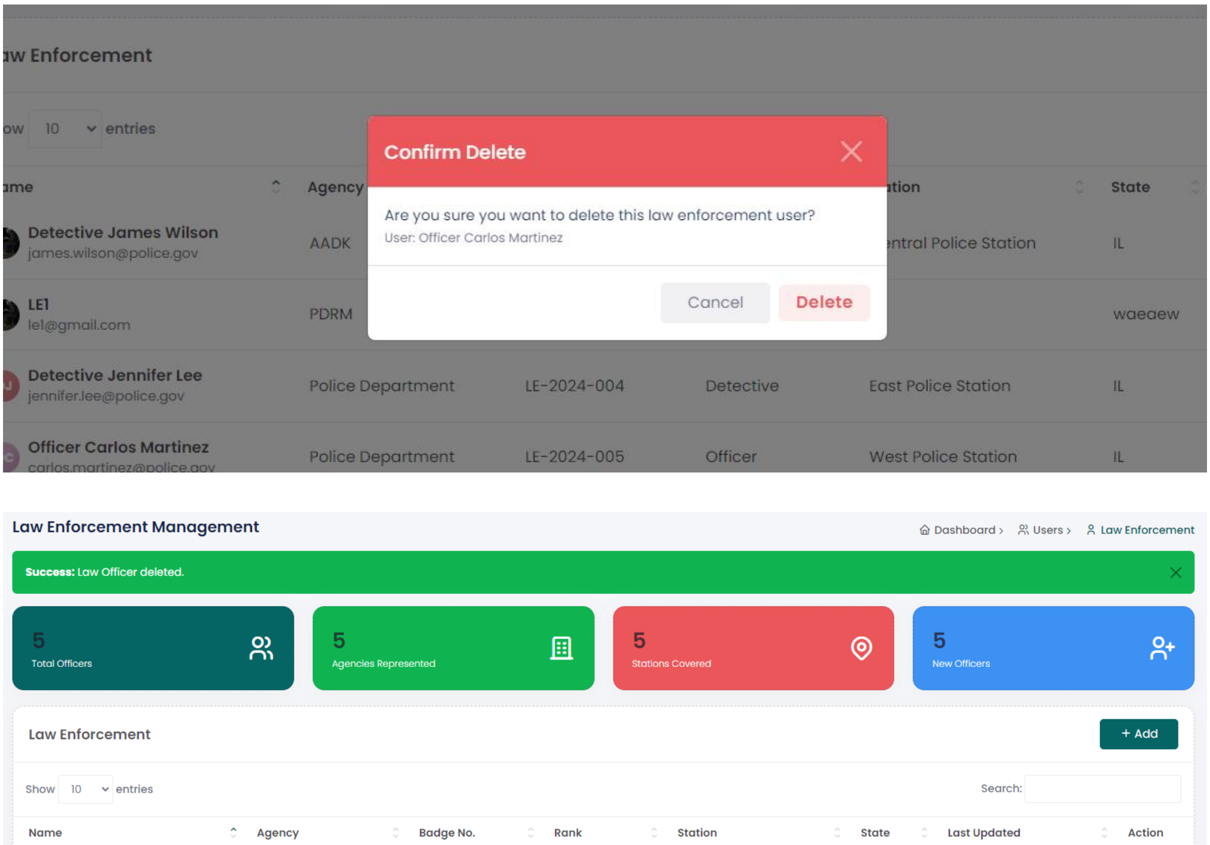


Figure 276: Delete law enforcement modal and successful message

Like other user management functionalities, if user selected the Delete button form the action dropdown, the above figure modal will pop up to let user confirm the deletion. If user clicked on the Delete button, the green successful message will be displayed, and the user will get deleted from the system. However, if user clicked on Cancel, the modal would close, and nothing will happen to that selected law enforcement.

4.6.1.16 Child Welfare Officer Management

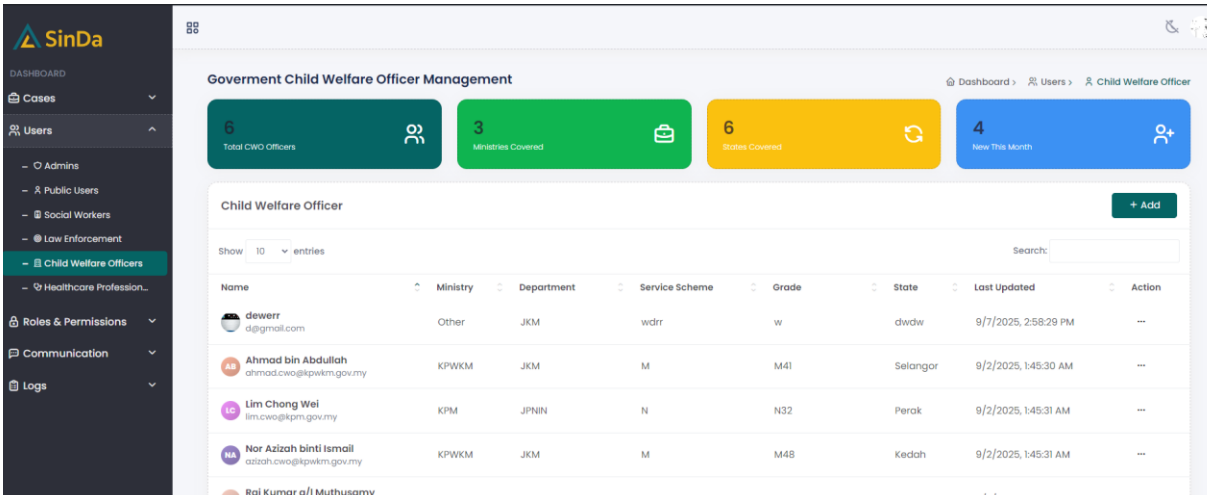


Figure 277: Government child welfare officer management page

The above page provides users with an organized interface to oversee government child welfare officers. At the top, summary cards display key statistics including the total number of officers, agencies represented, stations covered, and newly added officers, giving a quick overview of system records. Below the statistics, a detailed table lists officers along with their name, email, ministry, department, service scheme, grade, state, the date of the last update, and the actions dropdown are shown. Lastly on the right, the Add button allows users to quickly register new officers into the system.

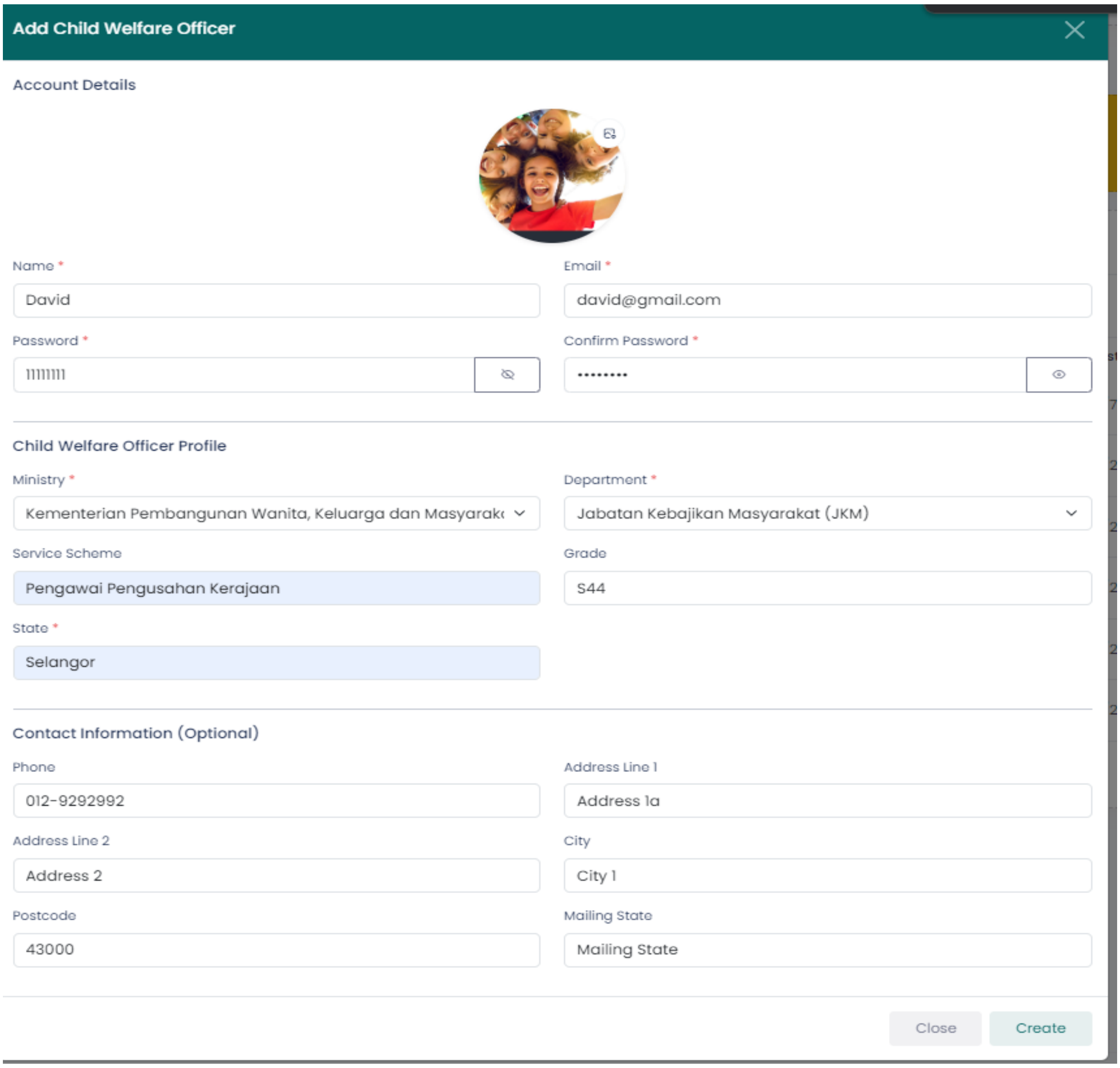


Figure 278: Add child welfare officer modal

Based on above figure, it is the form to add new child welfare officers into the system. At the top, the Account Details section requires essential inputs such as name, email, password, and confirm password to establish secure login credentials. The Child Welfare Officer Profile section collects key professional information including ministry, department, service scheme, grade, and state, which helps classify officers according to their roles and assignments. Additionally, the Contact Information section is optional, allowing for the entry of phone numbers, addresses, city, postcode, and mailing state to maintain complete records if necessary. If the user filled up the necessary field and to continue the creation, the user could finalize the addition by clicking the Create button, else if the user doesn't want too, users can click on the close button to close the form. As for the validation, like the rest this modal follow the two-tier validation process, where browser validation ensures the mandatory fields such as name, email address, password, confirm passwords and the others with asterisks are filled properly, while server-side check for unique credentials, input that are spaces, secure password confirmation and more.

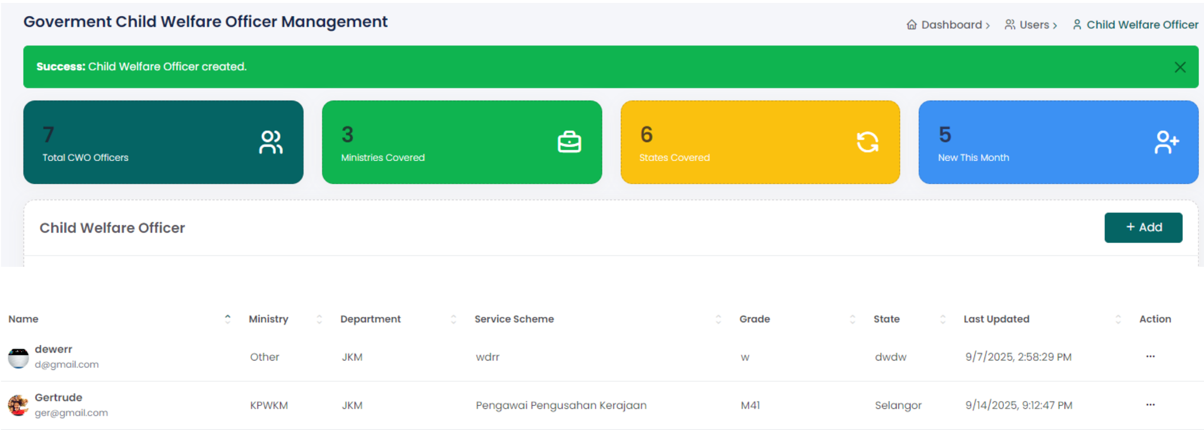


Figure 279: Add child welfare officer success message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above and the created child welfare officer will be in the table.

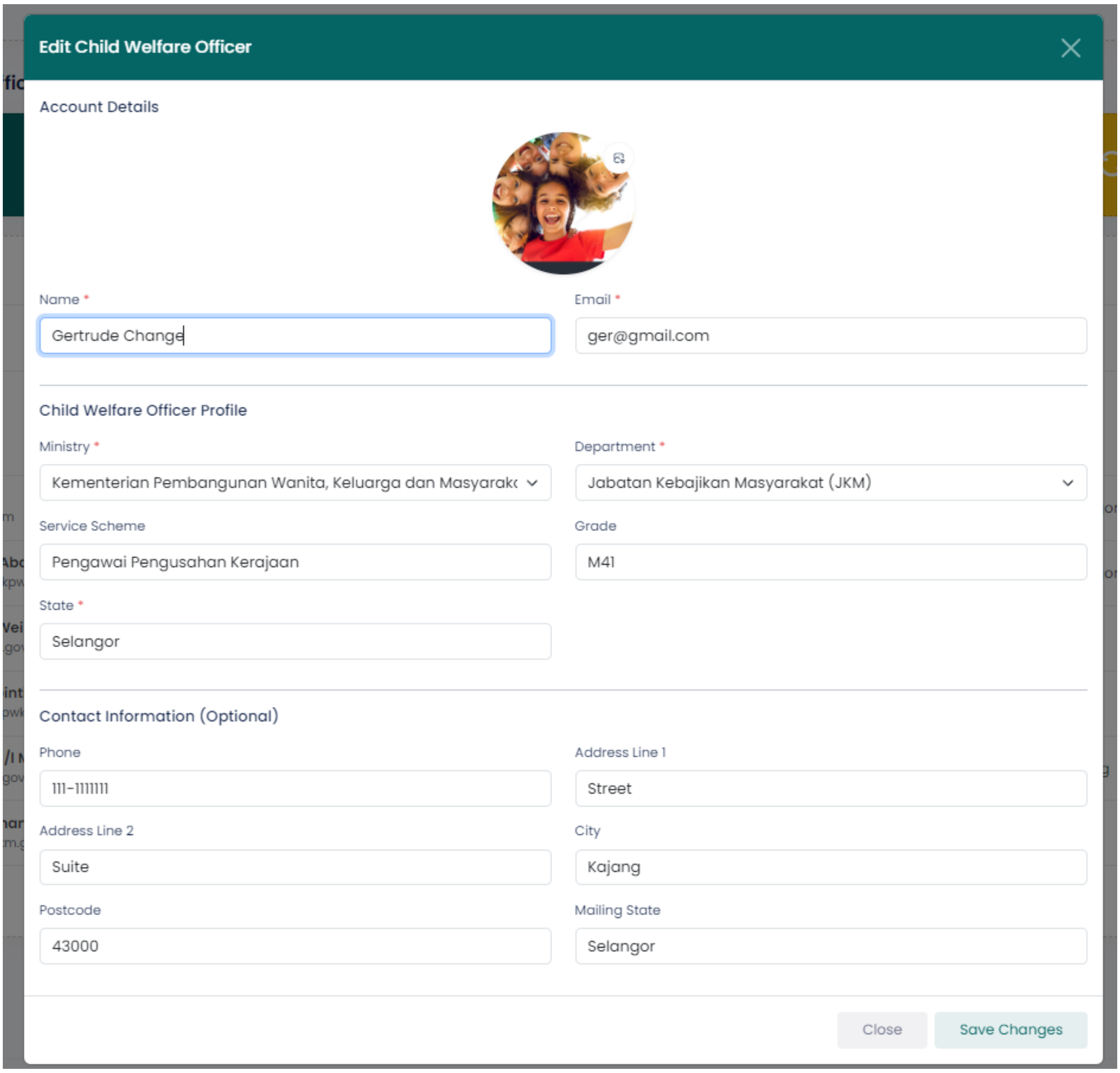


Figure 280: Edit child welfare officer modal

Above is the edit child welfare officer modal which are prefilled where it appears if users clicked on Edit button on any of the child welfare officers action dropdown. Like others, for this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button. Like other forms, this modal follows a two-tier validation process, where browser validation ensures required fields such as the name, email address, placement state and the rest that have asterisk are filled correctly before submission, while server-side validation enforces consistency and accuracy of the data across the system. This ensures that CWO records remain accurate, secure, and up to date.

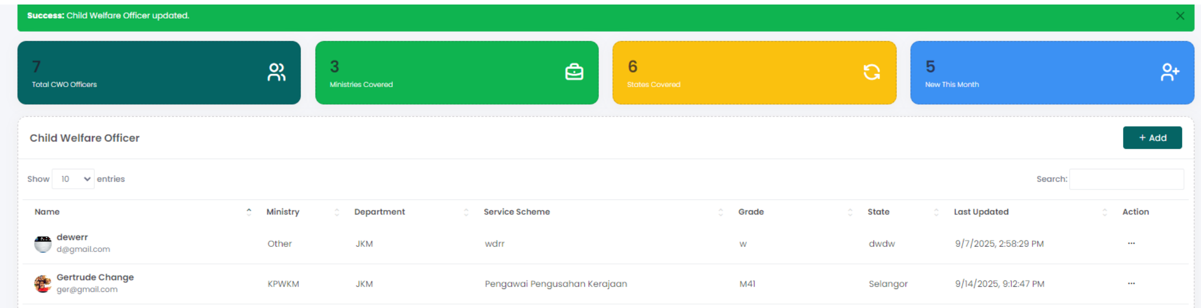


Figure 281: Child welfare officer changes successful message

Now if the user saves the changes correctly in the edit child welfare officer modal and click on Save Changes button, then there will be green successful message that will be displayed, and the changes can be seen based on figure above.

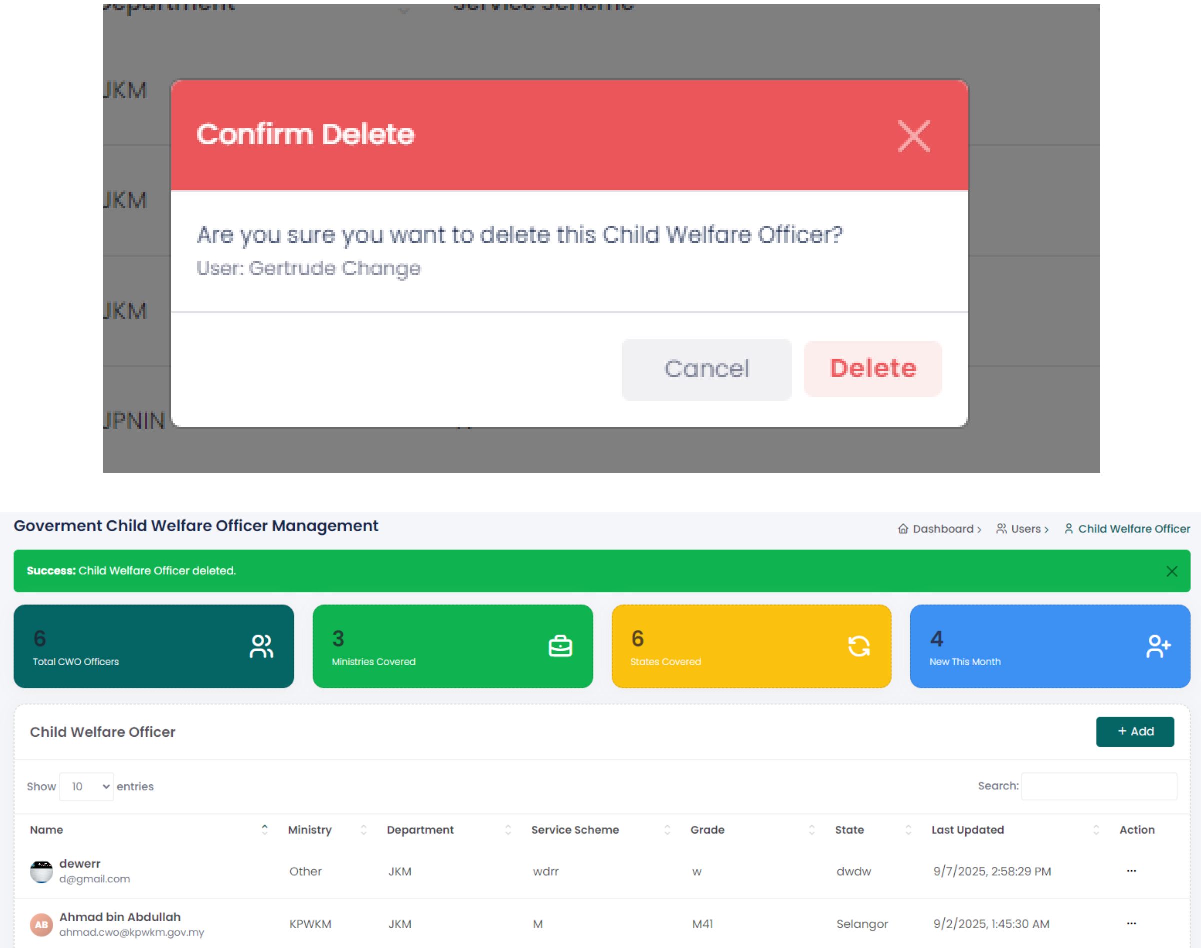


Figure 282: Delete confirmation modal and delete successful message

Like other user management functionalities, if user selected the Delete button form the action dropdown, the above figure modal will pop up to let user confirm the deletion. If user clicked on the Delete button, the green successful message will be displayed, and the user will get deleted from the system. However, if user clicked on Cancel, the modal would close, and nothing will happen to that selected child welfare officer.

4.6.1.17 Healthcare Professional Management

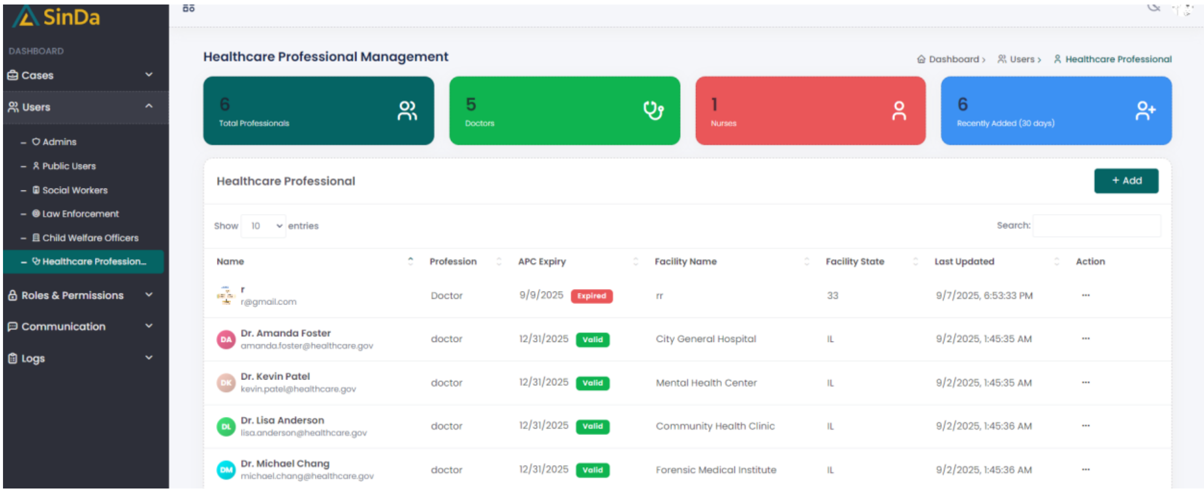


Figure 283: Healthcare professional management page

The Healthcare Professional Management page provides users with a structured view to manage doctors and nurses within the system. At the top, summary cards display key statistics, including the total number of professionals, the breakdown of doctors and nurses, and the number of recently added accounts within the last 30 days. Below, a detailed table lists each professional along with important details such as name, email, profession, APC (Annual Practicing Certificate) expiry status, facility name, facility state, the date of the last update and the actions. The APC expiry field is color-coded, showing Valid in green for active certificates and expired in red for lapsed ones, helping administrators quickly identify compliance status. On the right-hand side, an Add button is available to register new healthcare professionals.

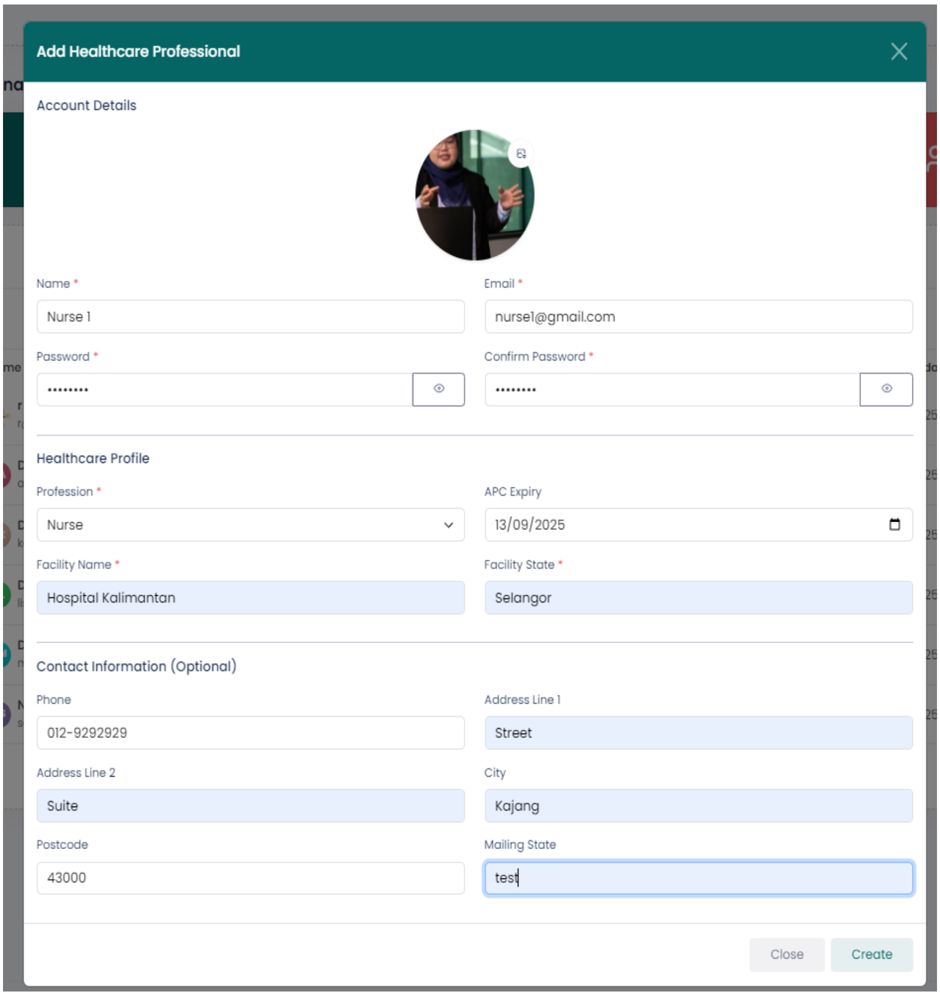


Figure 284: Add healthcare professionals' modal

Based on above figure, the form above is to add new healthcare professional into the system. Like always, it begins with account details, where the users enter the professional's name, email, and password with confirmation for security. Next, the Healthcare Profile section captures profession type (e.g., doctor or nurse), APC expiry date to track certification validity, facility name, and facility state to indicate the professional's workplace. Additionally, the Contact Information section is optional, allowing for the entry of phone numbers, addresses, city, postcode, and mailing state to maintain complete records if necessary. If the user filled up the necessary field and to continue the creation, the user could finalize the addition by clicking the Create button, else if the user doesn't want too, users can click on the close button to close the form. As for the validation, like the rest this modal follow the two-tier validation process, where browser validation ensures the mandatory fields such as name, email address, password, confirm passwords and the others with asterisks are filled properly, while server-side check for unique credentials, and secure password confirmation.

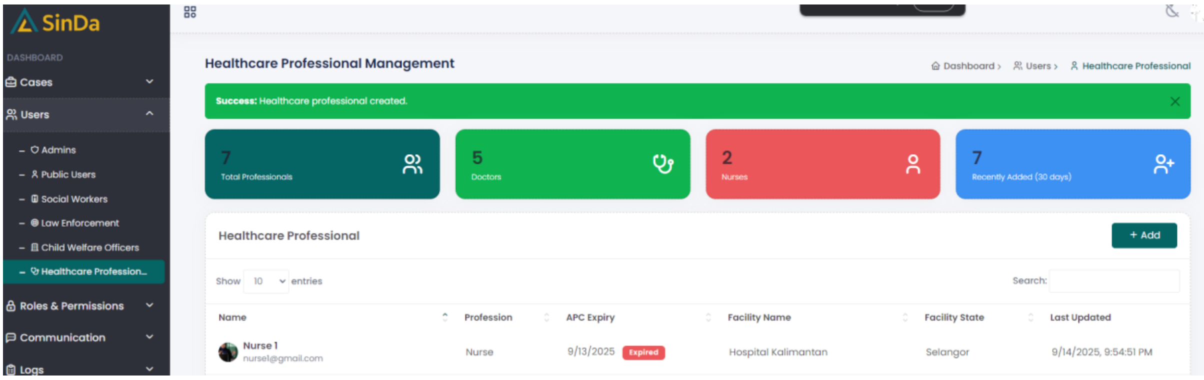


Figure 285: Add healthcare professional success message

If user fill up all required field correctly and press the create button, then the system will show the success message based on figure above and the created healthcare professional will be in the table.

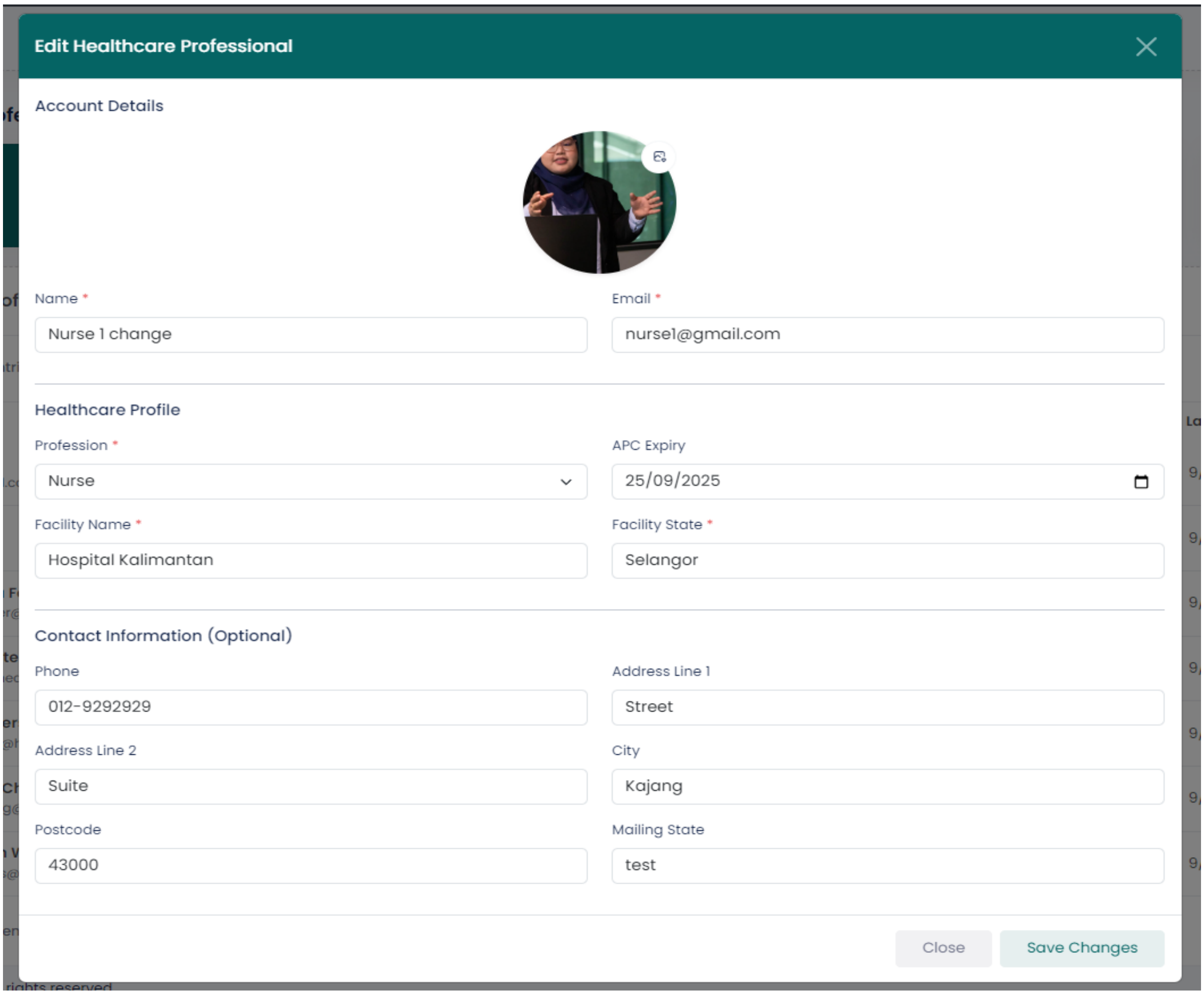


Figure 286: Edit healthcare professional modal

Above is the edit healthcare professional modal which are prefilled where it appears if users clicked on Edit button on any of the child welfare officers action dropdown. Like others, for this modal, user can make necessary updates and save them by pressing the Save Changes button or exit without applying changes by selecting the Close button. Like other forms, this modal follows a two-tier validation process, where browser validation ensures required fields such as the name, email address, placement state and the rest that have asterisk are filled correctly before submission, while server-side validation enforces consistency and accuracy of the data across the system. This ensures that healthcare professional records remain accurate, secure, and up to date.

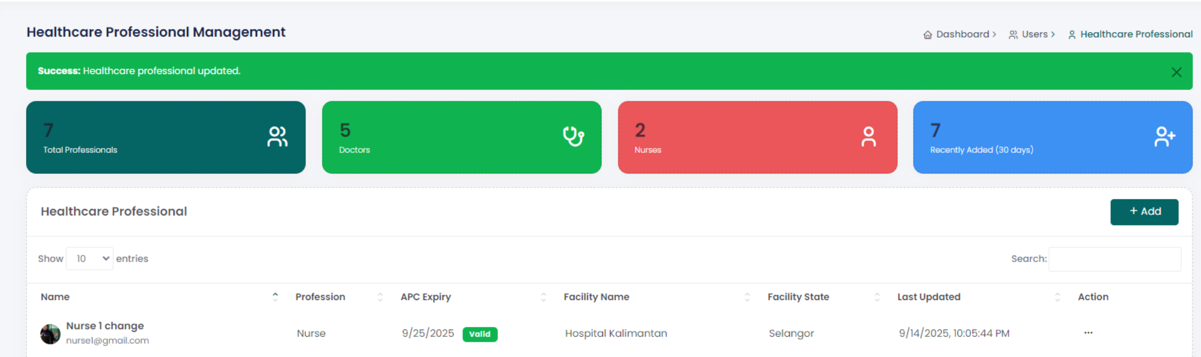
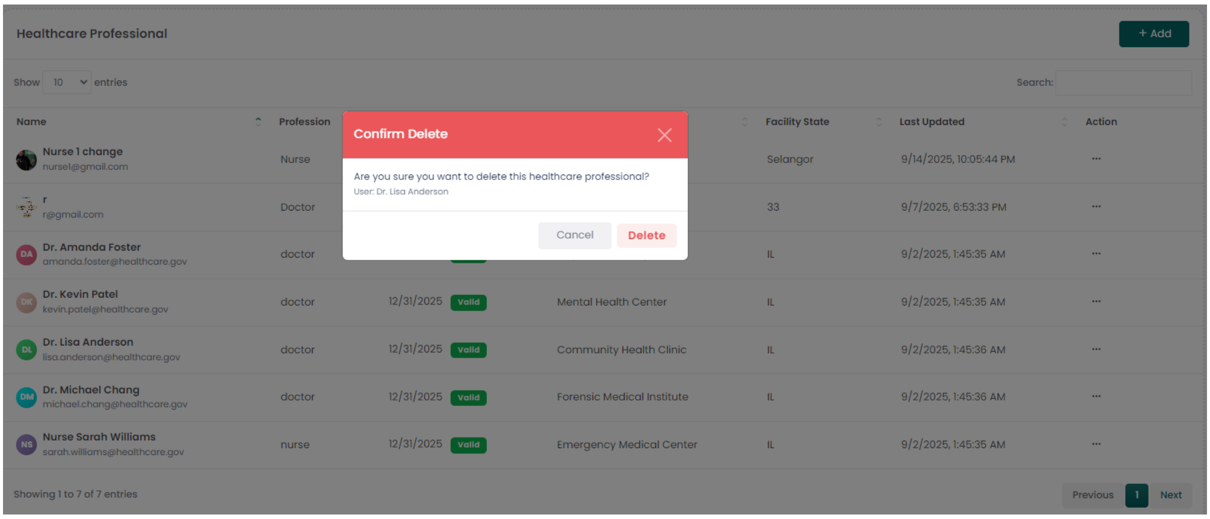


Figure 287: Healthcare professional changes successful message

Now if the user saves the changes correctly in the edit healthcare professional modal and click on Save Changes button, then there will be green successful message that will be displayed, and the changes can be seen based on figure above.



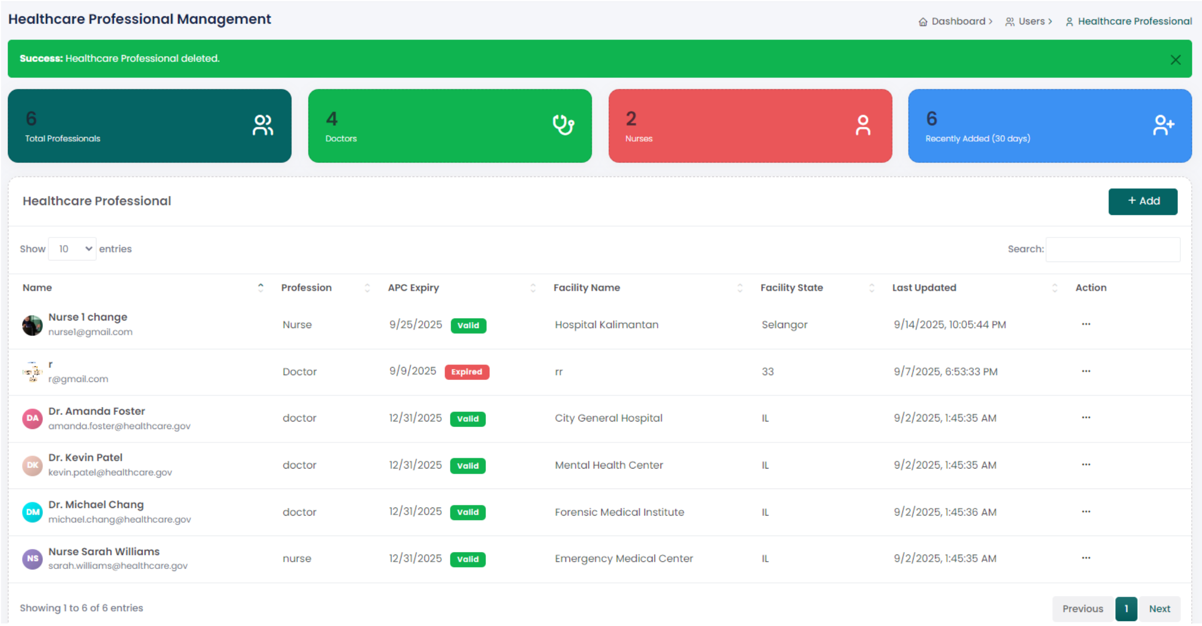


Figure 288: Delete confirmation modal and delete successful message

Like other user management functionalities, if user selected the Delete button form the action dropdown, the above figure modal will pop up to let user confirm the deletion. If user clicked on the Delete button, the green successful message will be displayed, and that user will get deleted from the system. However, if user clicked on Cancel, the modal would close, and nothing will happen to that selected healthcare professional.

4.6.1.18 Role Management

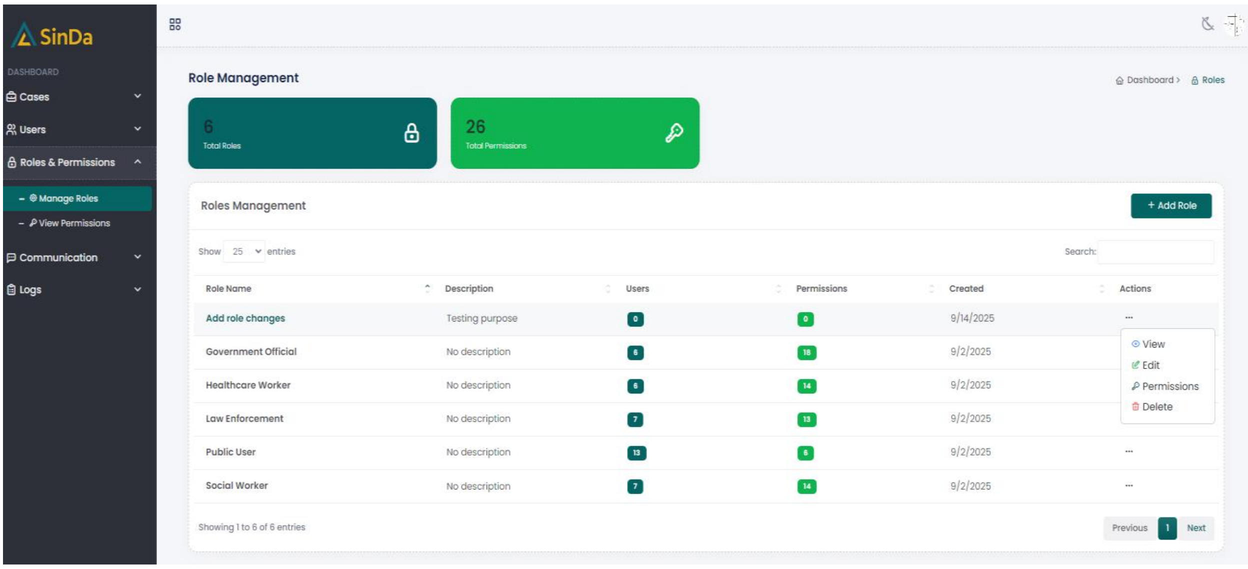


Figure 289: Role management page

Above figure is the page where user can oversee and control user roles within the system. At the top, summary cards show the total number of roles and the total permissions currently configured, giving a quick overview of the role structure. The main table lists each role along with its description, the number of users assigned, and the permissions granted, ensuring visibility of how responsibilities are distributed. Each entry also displays its creation date, with an Actions menu for editing or managing specific roles. Additionally, an Add Role button is available, allowing users to define new roles as the system evolves.

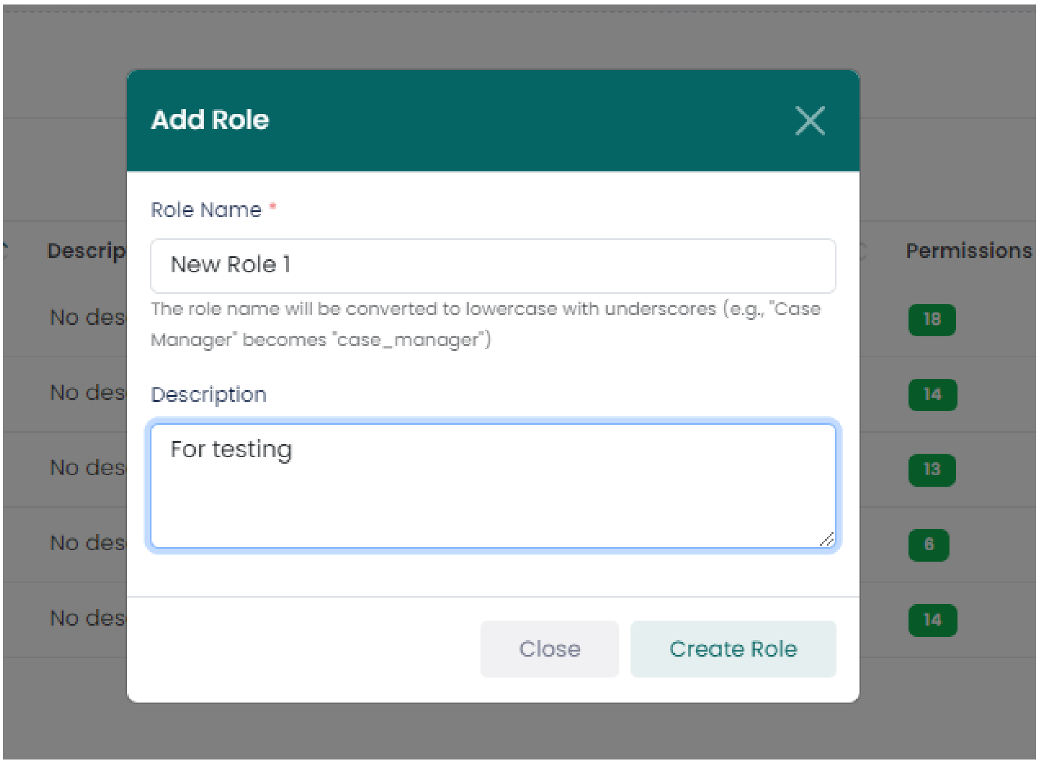


Figure 290: Add role modal

This figure is the Add Role modal when the suer clicked on the Add Role button. It allows administrators to create new roles that define user access levels and responsibilities. There are two field name which is Role Name and Description where Role Name is required to fill. If user are satisfied with the inputs, user can click on the Create Role button to save the new role into the system, else user can cancel the process by clicking the Close button.

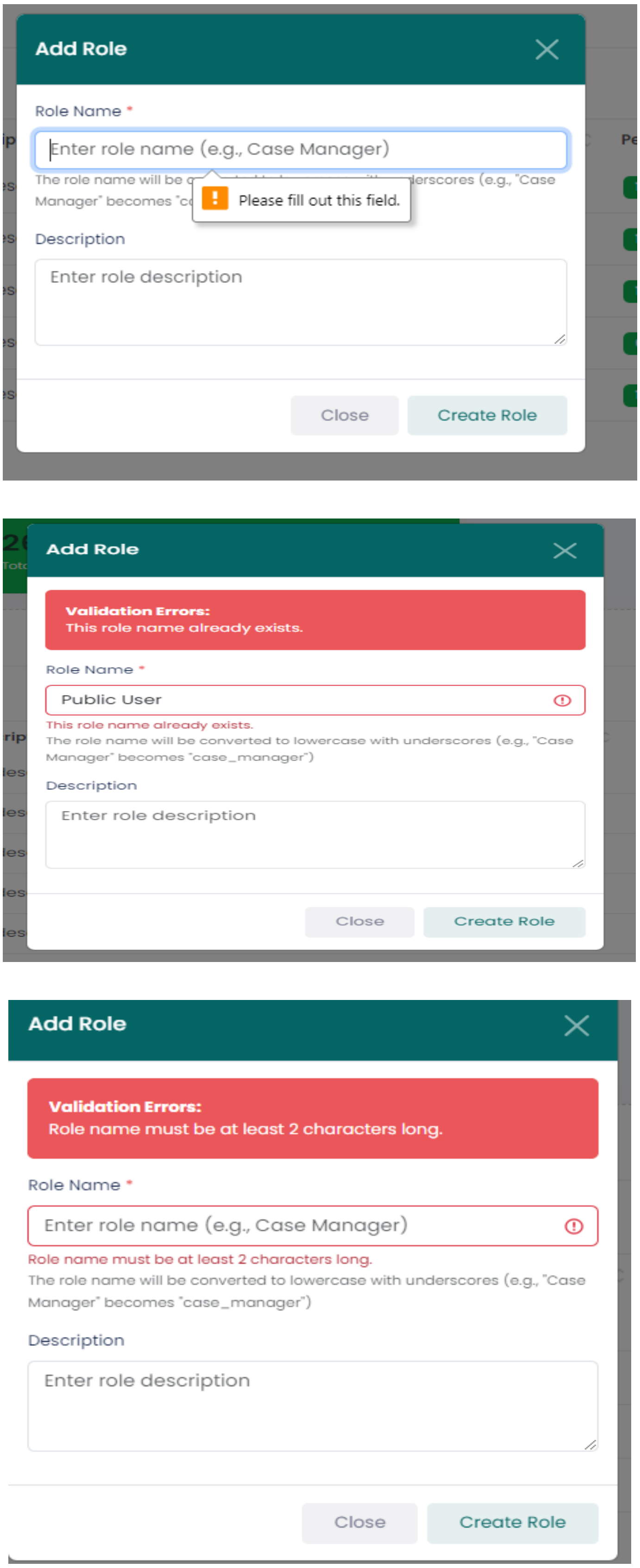


Figure 291: Add Role validations

Theres also validation check for creating role. Based on above figures, if the user didn't enter anything and clicked on the Create Role button, the browser validation will be prompted. Then if the user typed out and existing role, there's also server-side validation where it checked the inputted role with the database. Lastly, if the user entered spaces or have inputted 1 character, the system considers it invalid and shows the validation error message as well.

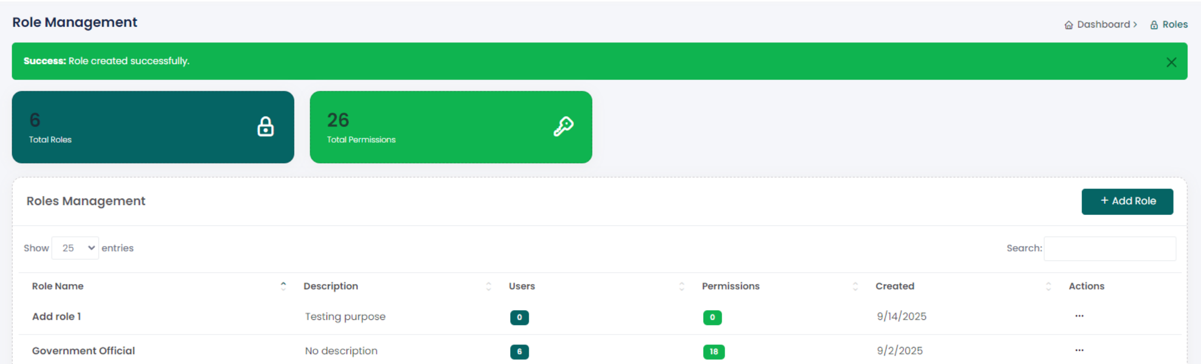


Figure 292: Role addition successful message

If the role entered are valid, then the green message will be displayed, and the role will be added to the system and displayed in the table.

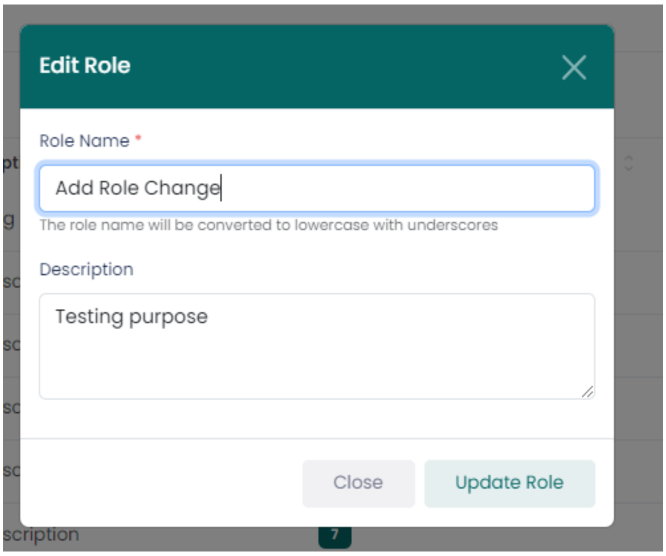


Figure 293: Edit role modal

Above figure shows the edit role modal where it appears if the user clicked on Edit button on the action dropdown. It allows updating the Role Name, which is automatically converted to lowercase with underscores for standardization, and the Description, which provides context about the role's purpose. So, if the user change something correctly and wanted to save, user can click on the Update Role button to save the changes or if the user wants to cancel the process, the user can click on the Close button.

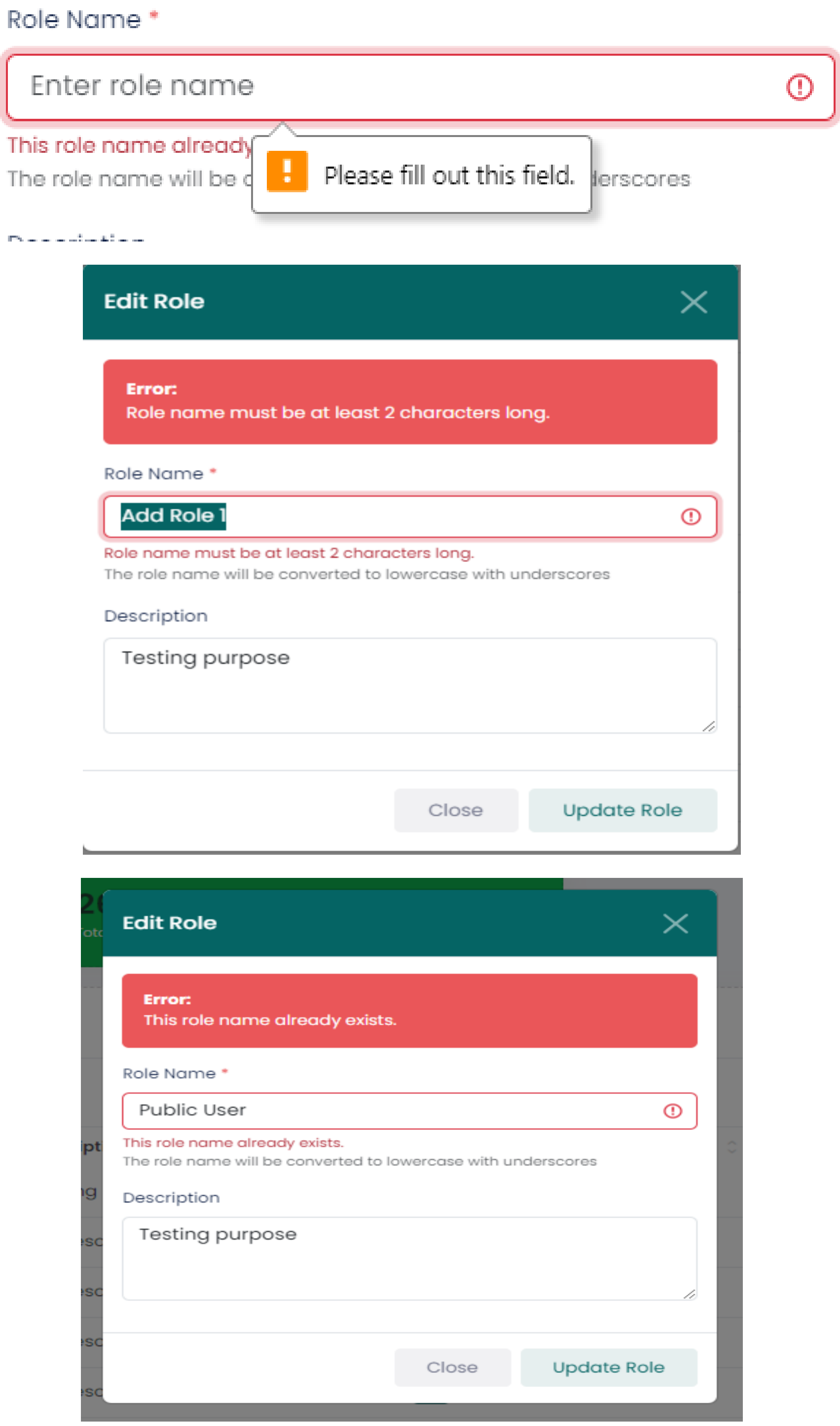


Figure 294: Edit role modal

Like the add role model, the validations are the same. For example, if the role name is empty, the browser validation came in, if the role name is not 2 characters long or more or the role name entered is existing, the server-side validation will catch on it and display the validation error message above.

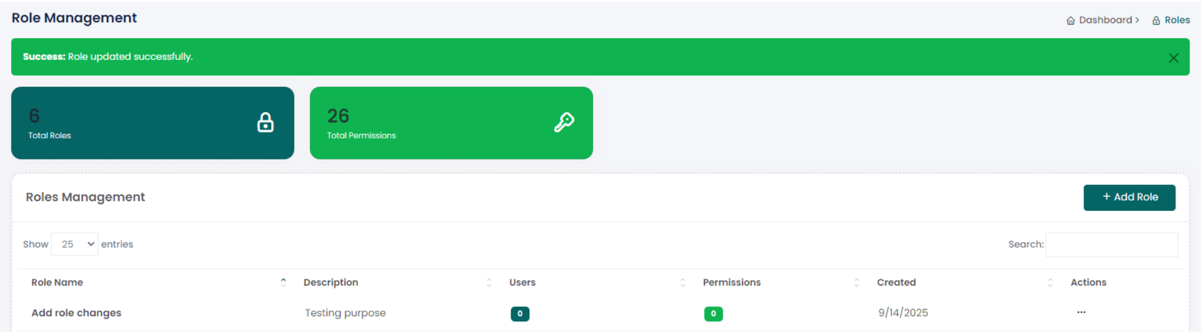


Figure 295: Role updated successfully message

If the role changes are valid, then there will be a successful message displayed based on the above figure.

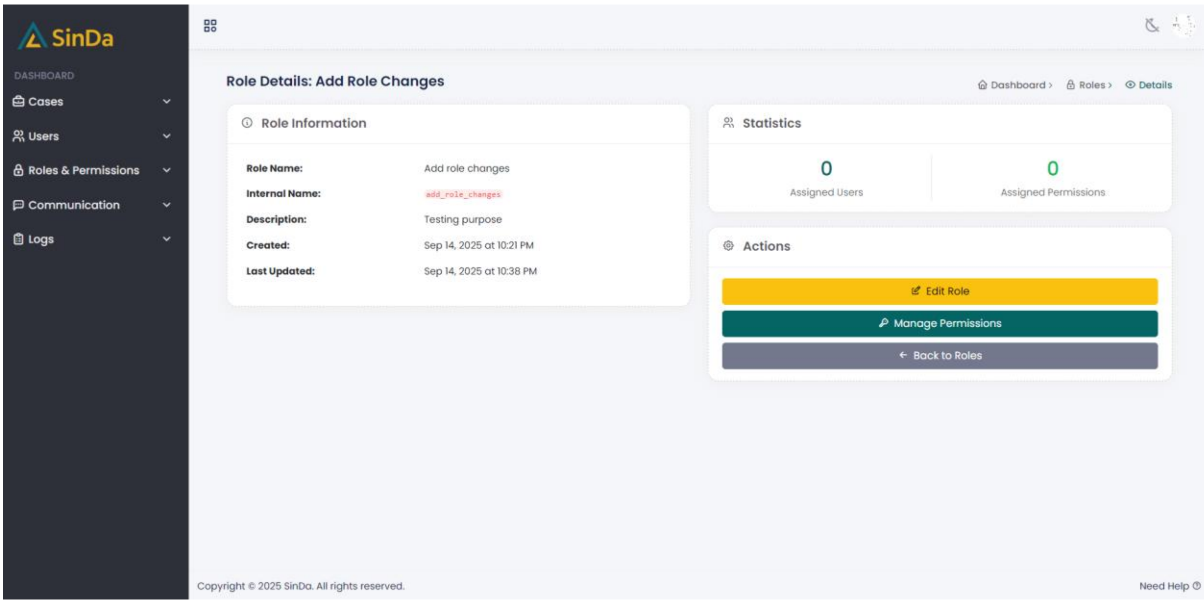


Figure 296: Role details page

This page is the Role Details page which opens when a user clicks the View button from the action dropdown in the Role Management page. This page provides a detailed description of a specific role based on the figure above and on the right, the Statistics panel which displays how many users are assigned to this role and how many permissions are linked to it. Below that, there's action button which shows three buttons which is Edit Role button where it have similar functionalities to the Edit Role button in the Role Management page, Manage Permissions page which will redirect the user to the Manage permission page, and Back to Roles button which redirects the user back to the Role Management page.

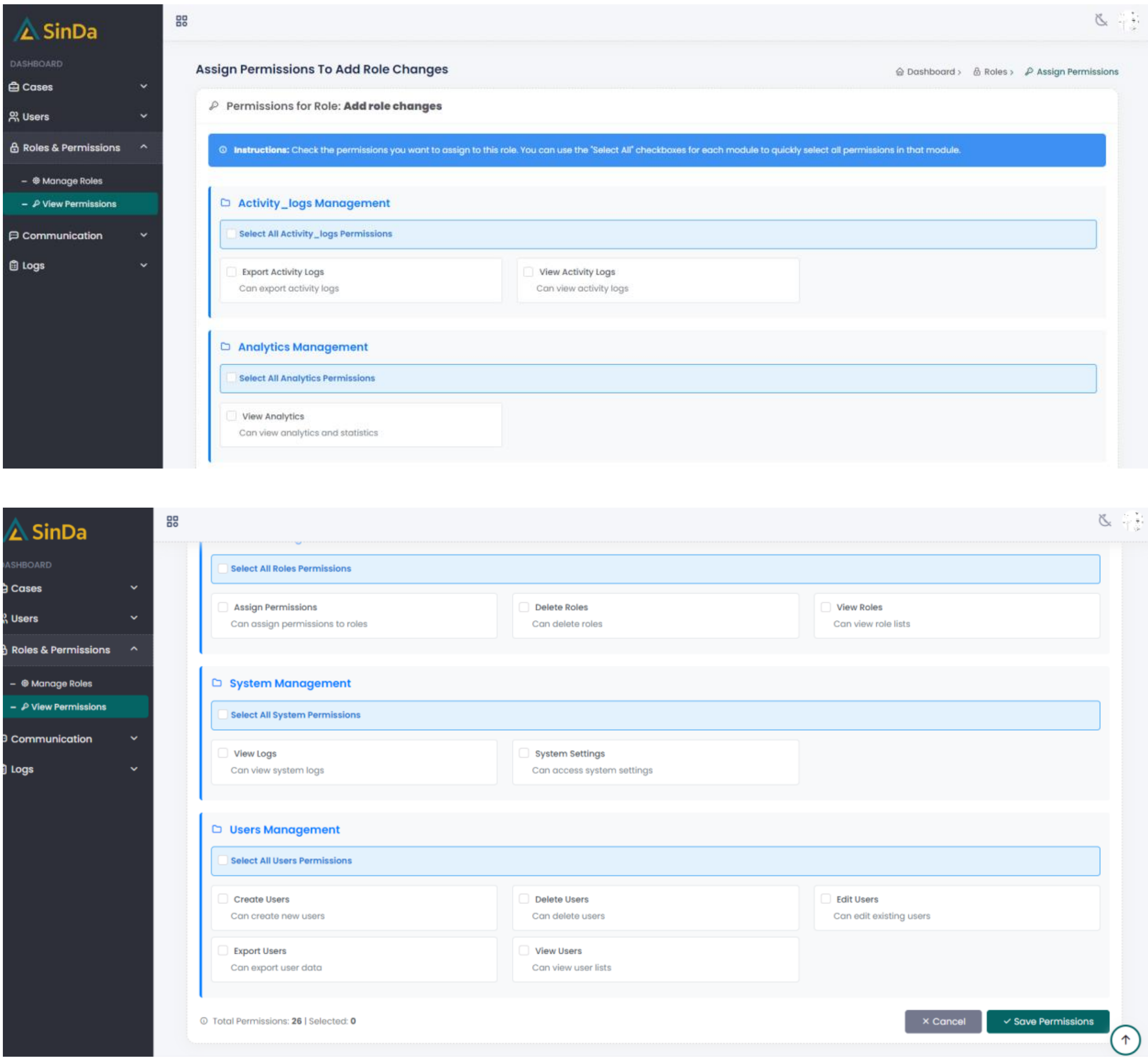


Figure 297: Assign permission page

This is the Assign Permission page, where users can allocate permissions to a selected role. To guide users, clear instructions are provided on how to use this module. Permissions can be assigned by checking individual boxes or by using the Select All option to grant all permissions within a module. At the bottom, a summary displays the total number of permissions and how many have been selected. Once the desired permissions are chosen, the user can click the Save Permissions button to apply the changes or use the Cancel button to exit without saving.

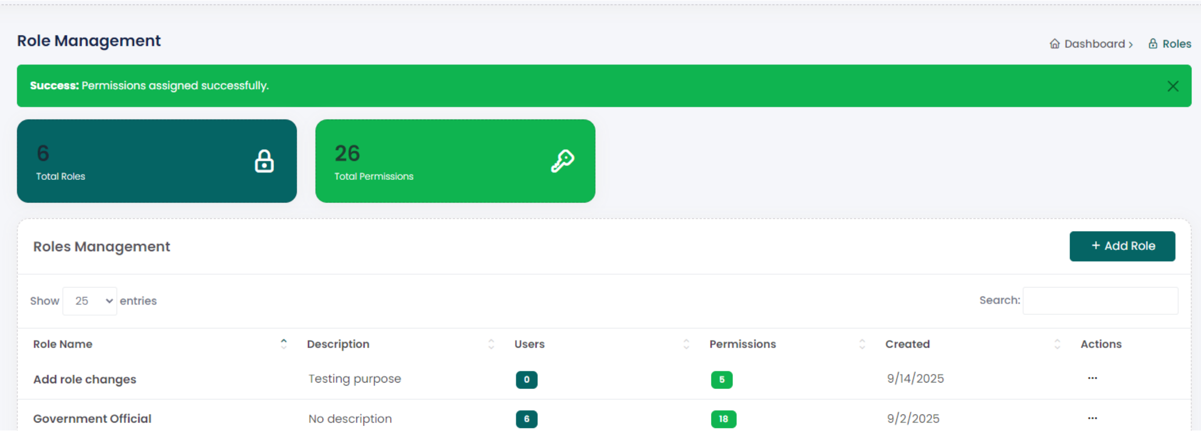


Figure 298: Permission assigned successfully message

If the user clicked save changes on the Assign Permission page, they will be redirected back to the Role Management page with a success message displayed and with the permissions counts updated as well.

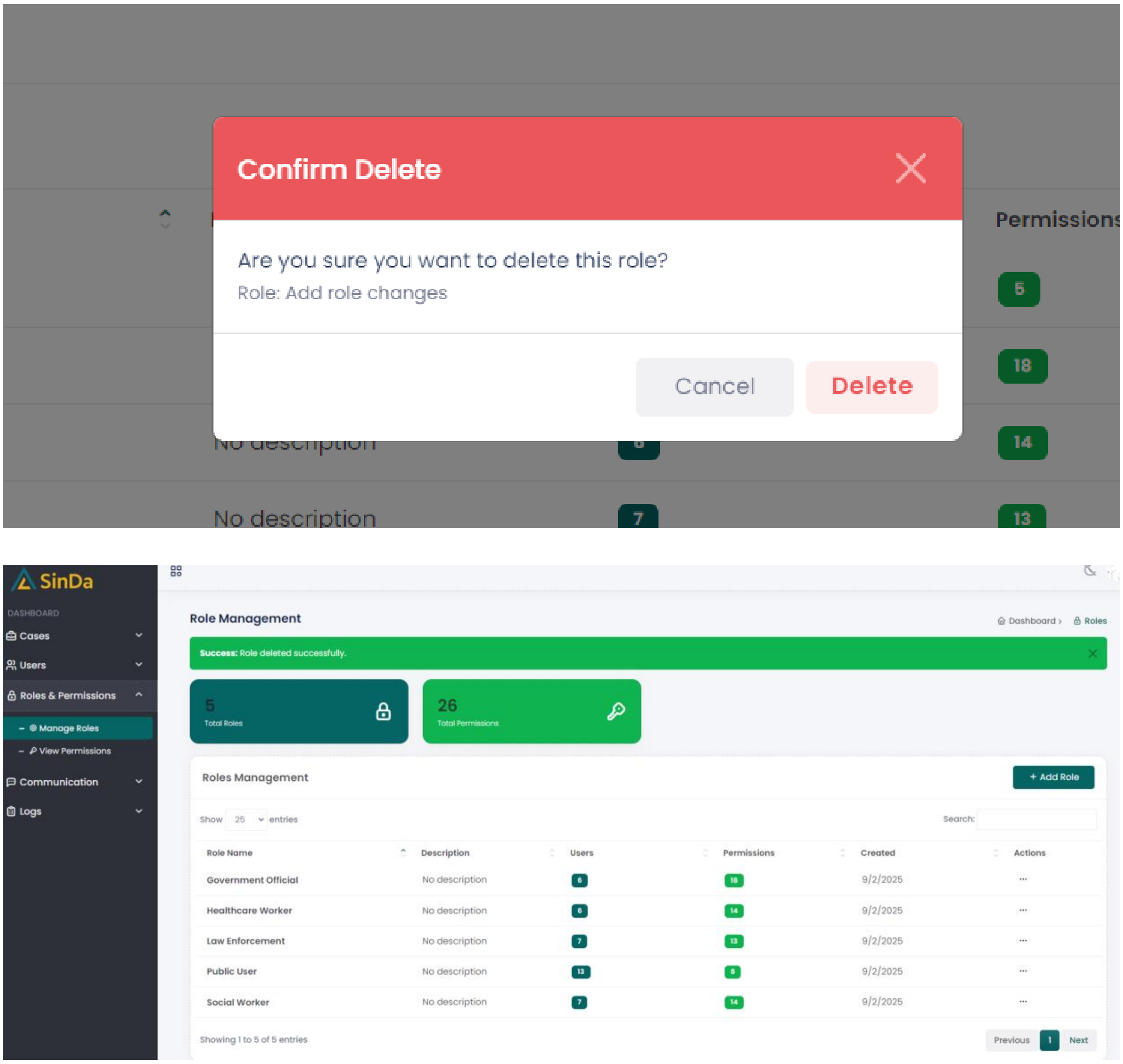


Figure 299: Delete role confirmation modal

Like other user management functionalities, if user selected the Delete button form the action dropdown, the above figure modal will pop up to let user confirm the deletion. If user clicked on the Delete button, the green successful message will be displayed, and that role will get deleted from the system. However, if user clicked on Cancel, the modal would close, and nothing will happen to that selected role.

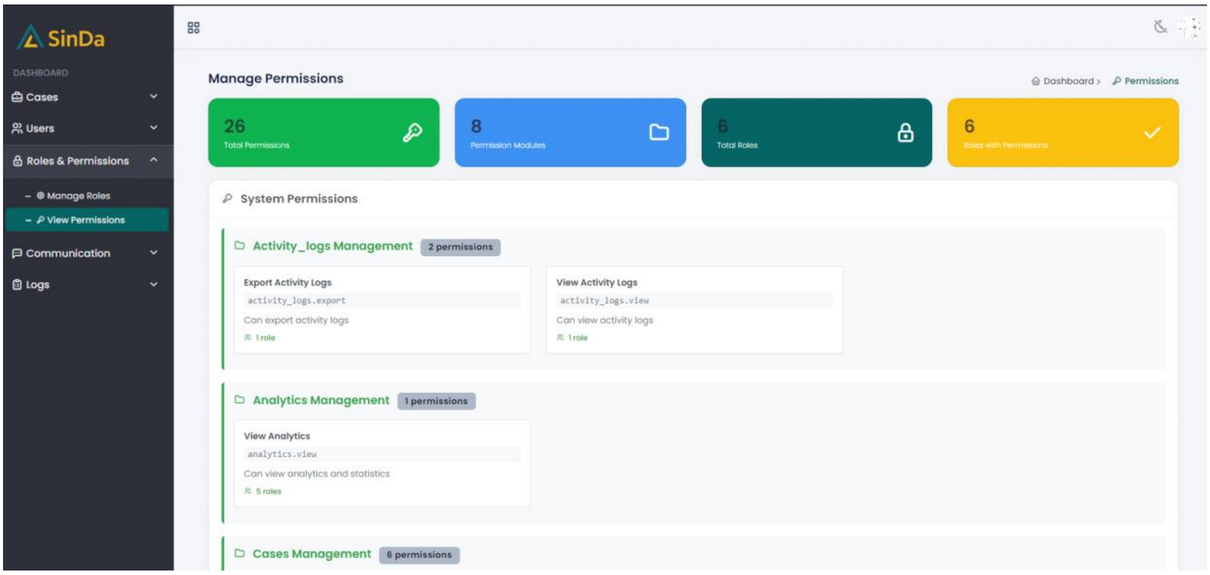


Figure 300: Manage Permissions

Above figure is the Manage Permissions page, which displays an overview of all system permissions.

4.6.1.19 Contact Queries Management

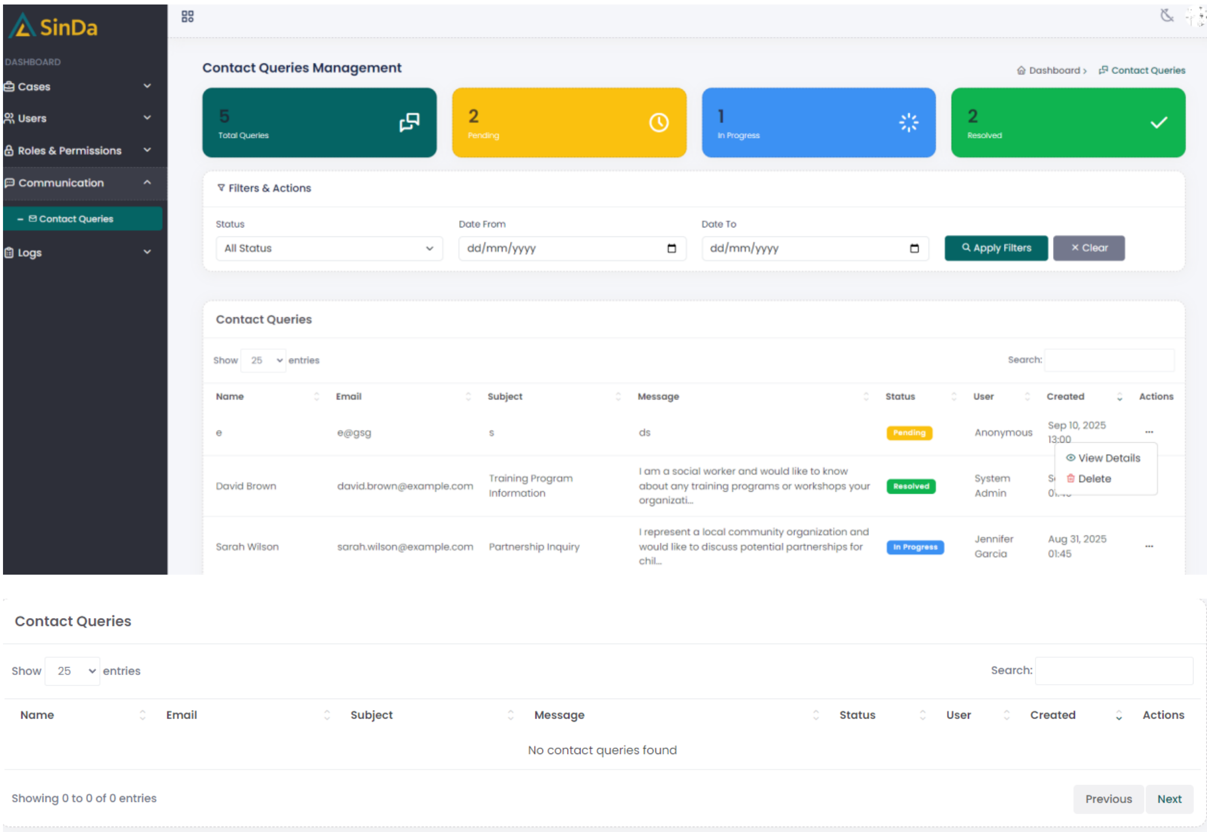


Figure 301: Contact queries management page

This page allows user to view and manage queries submitted by public users. Like usual, at the top, there's summary cards to help user see the status count. There also filters to help user to narrow results by status or date range with options to apply or clear filters. If there are not available result, there will be no contact queries found message based on figure above.

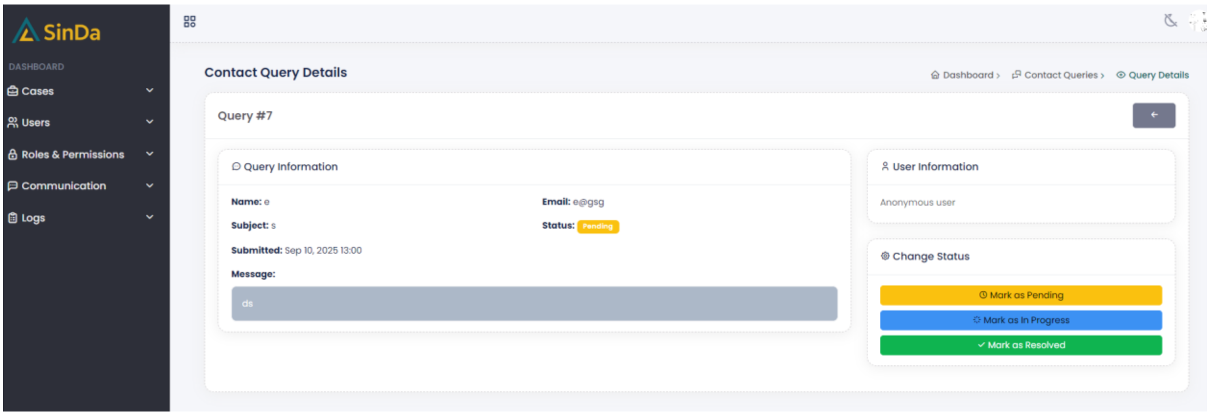


Figure 302: Contact query details page

This is the Contact Query Details page where users can see the information that was send by the public user. In this page, user can update the status of the query by selecting either Mark as Pending button, Mark as In Progress or Mark as Resolved button.

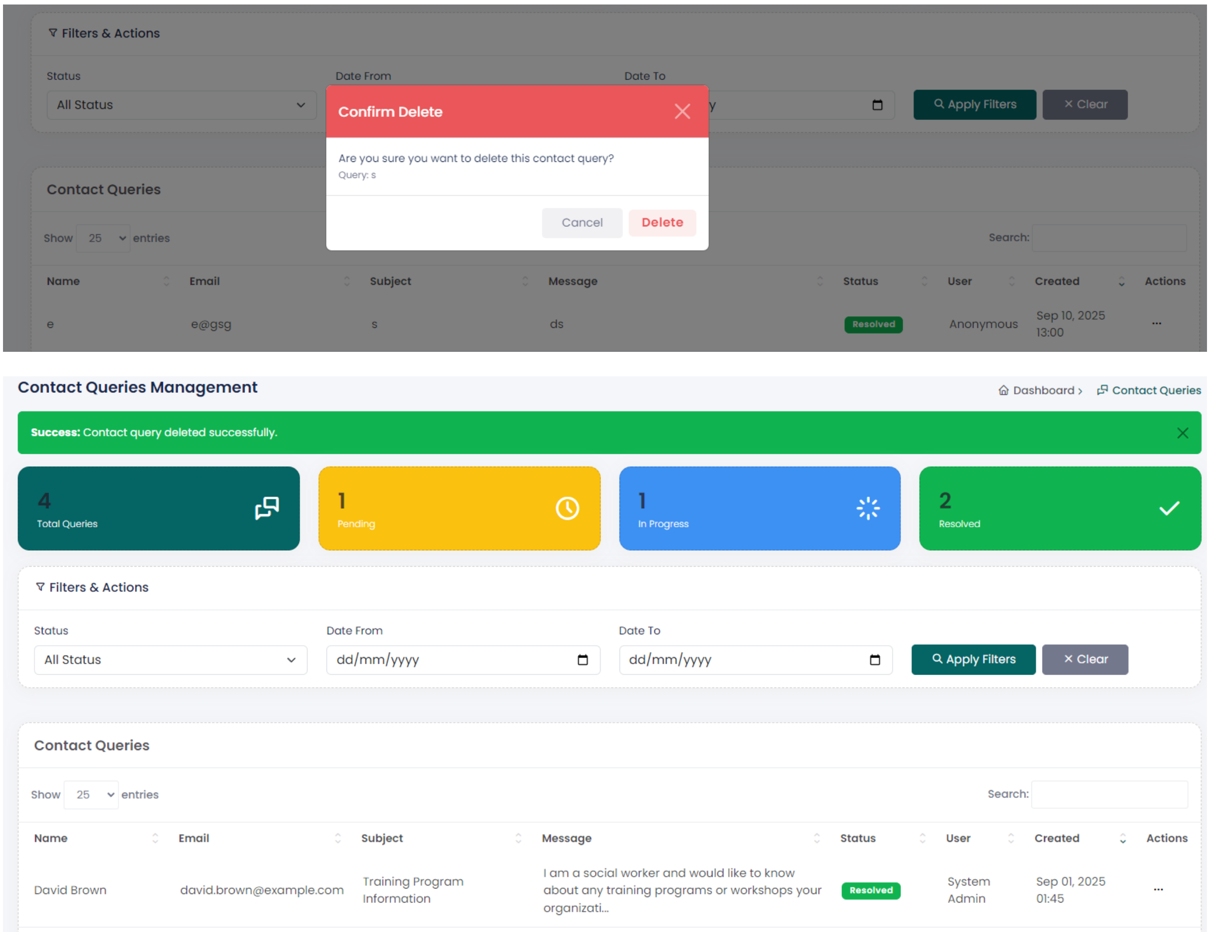


Figure 303: Confirm delete modal and contact query deletion successfully message

Above figure is the delete confirmation modal if user clicked on the Delete button on the action dropdown on one of the queries. Is user want to proceed with the process user may clicked on Delete button to delete the selected query else if user want to cancel the process user may clicked on Cancel button.

4.6.1.20 Session Logs page

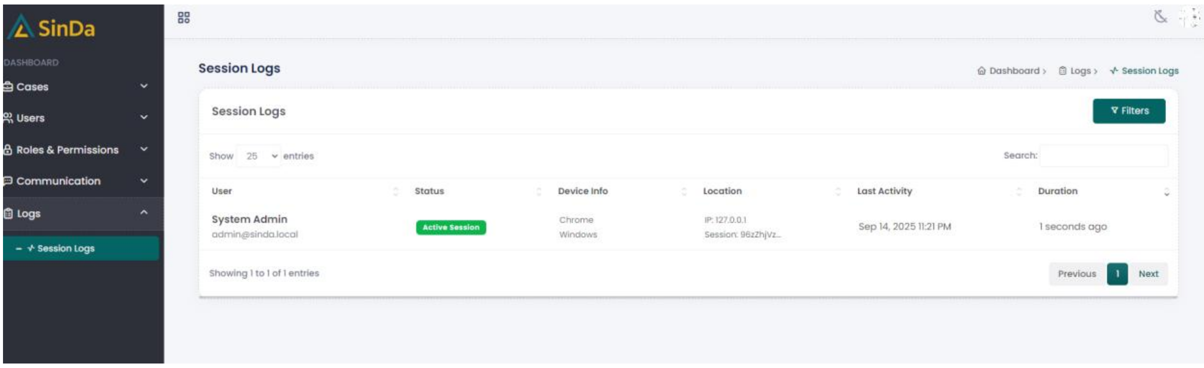
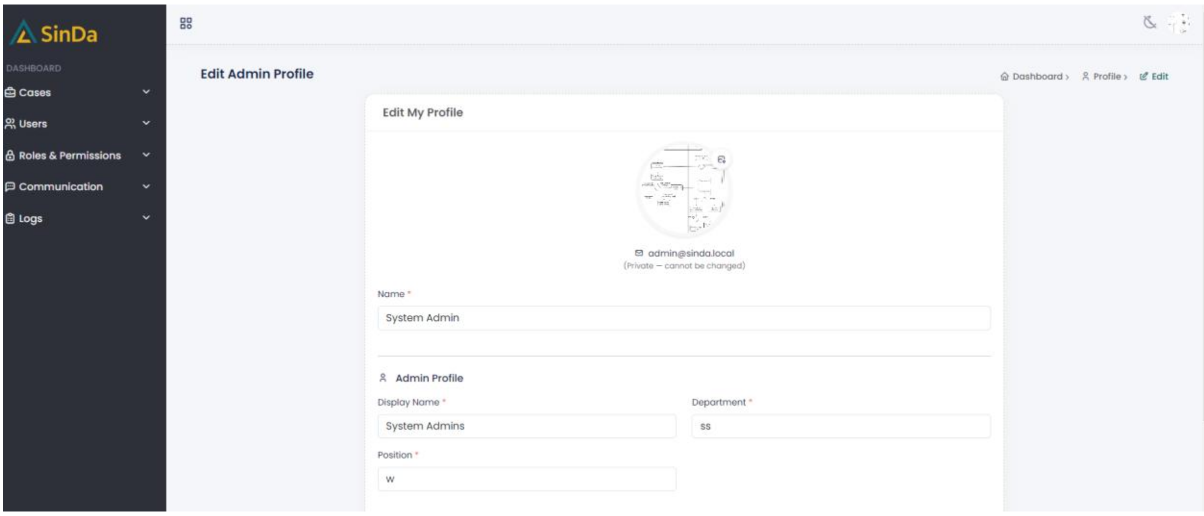


Figure 304: Session logs page

This is the Session Logs page which provides users with details of active and past user session. The table lists important information such as the user's name and email, session status, device and browser used, IP address, session ID, last activity time, and session duration. It also includes filter and search options to help locate specific session records easily.

4.6.1.21 Profile page

4.6.1.21.1 Edit Admin Profile page



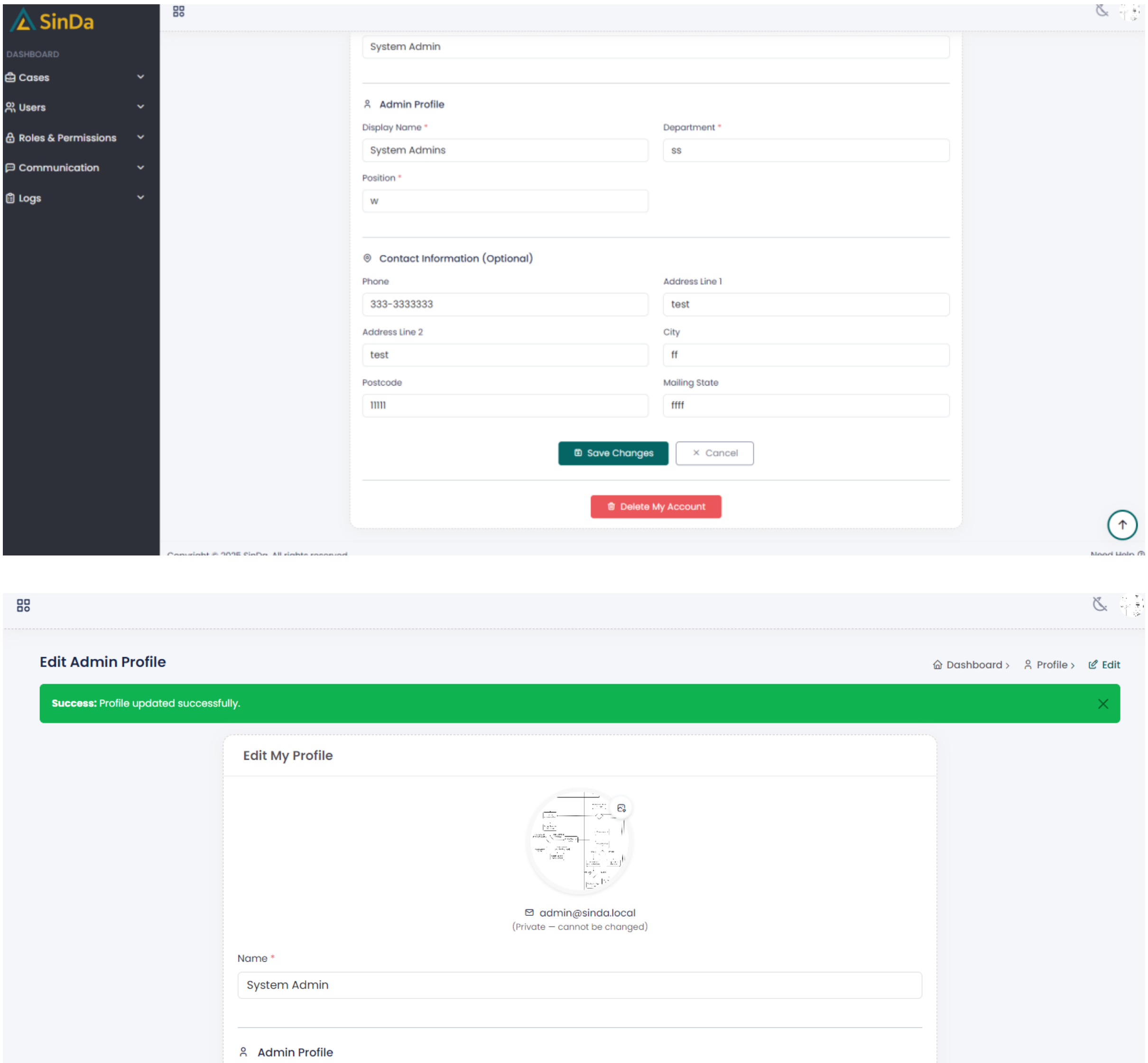


Figure 305: Edit my profile page (System Administrator)

Above is the profile page for the current logged in user for system administrator. Like the user management edit functions and validation, this form has similar validation rules to ensure required fields are filled correctly, while optional fields such as contact information can be left empty without causing errors. So, if user do some changes and clicked on the Save Changes button, there will be a successful message like the above figure.

4.6.1.21.2 Delete Account Confirmation modal

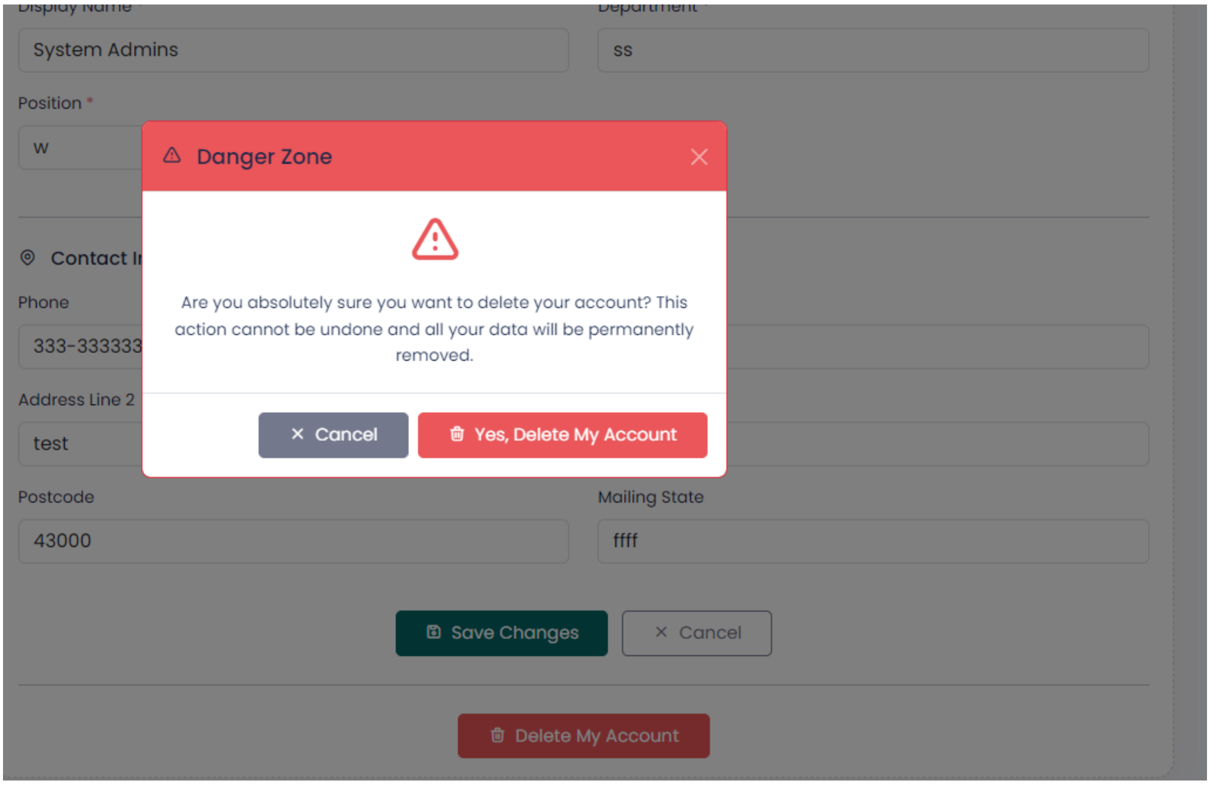


Figure 306: Delete Confirmation modal

Above figure is the Delete account confirmation modal, which appears when a user chooses to permanently remove their account. The user can click on Cancel to abort the process or Yes, Delete My Account button to proceed with deletion. If user clicked on it, user will be redirected to sign in page with their account deleted.

4.6.1.21.3 Edit Public User Profile page

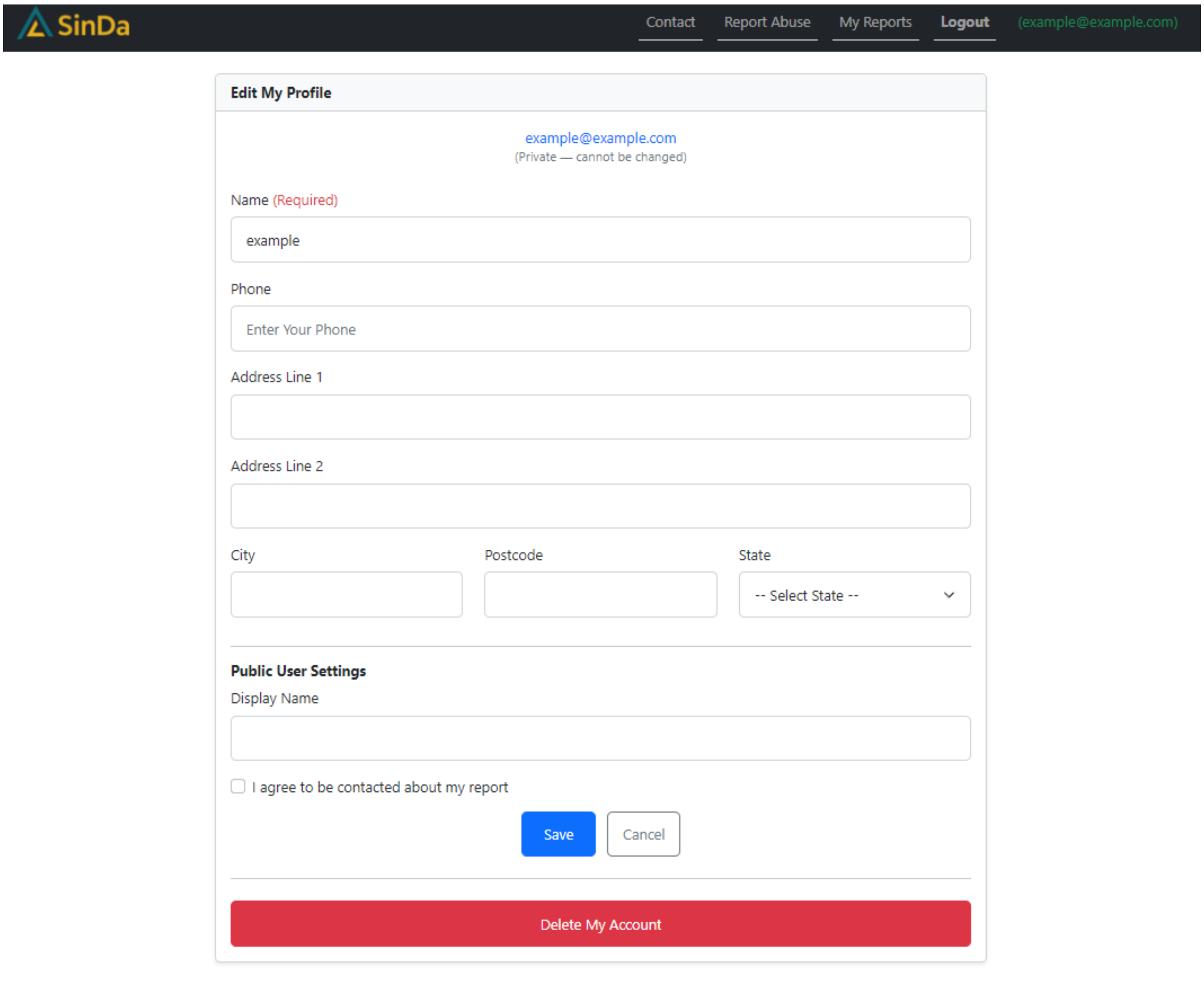


Figure 307: Public User Profile

This is the Edit My Profile page for public users, where they can update their personal details and preferences. The form includes fields for name, phone number, address, city, postcode, and state, along with a section for Public User Settings such as display name and an option to agree to be contacted regarding submitted reports. The user's email is displayed at the top as a fixed, undatable identifier. At the bottom, users can click Save to update their information, Cancel to discard changes, or Delete My Account to permanently remove their account.

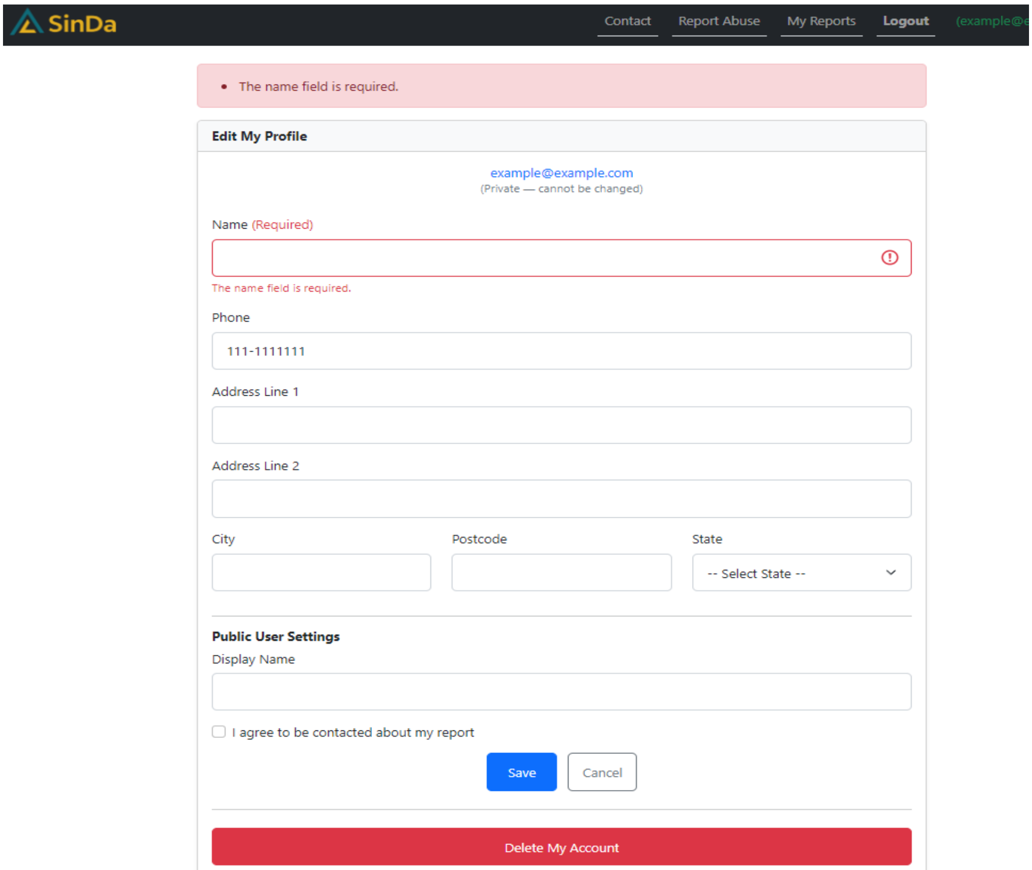
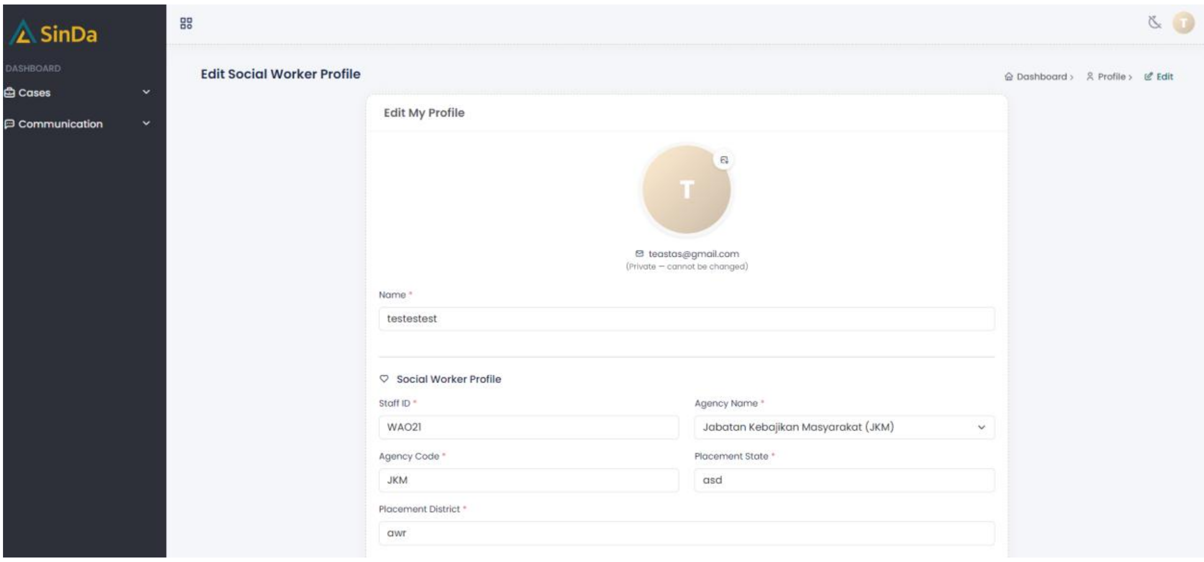


Figure 308: Edit public user profile validation

If the name field is empty which is required there will be validation error message based on figure above.

4.6.1.21.4 Edit Social Worker Profile page



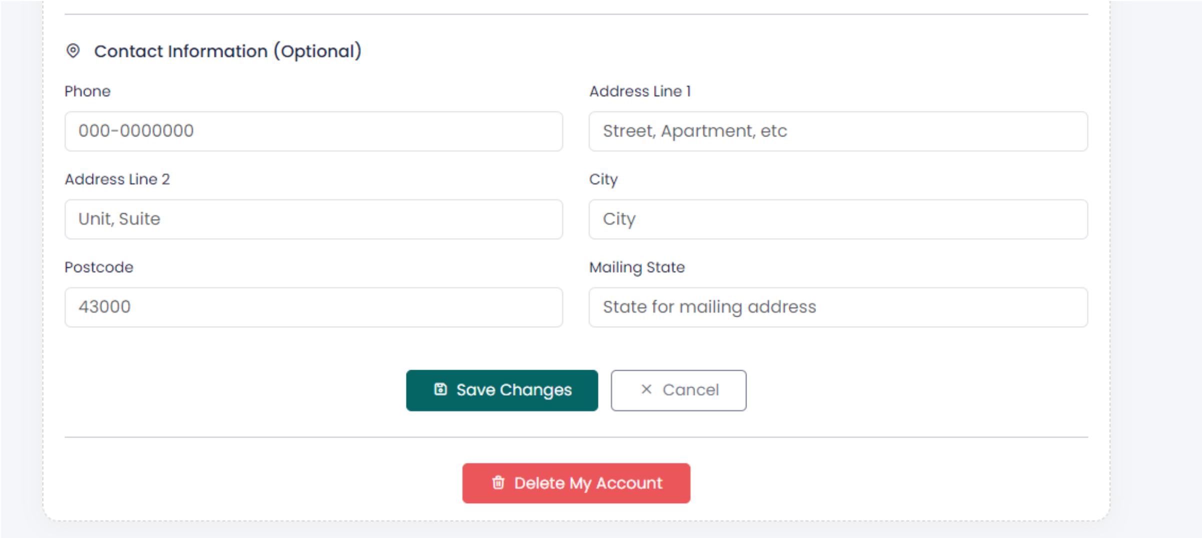


Figure 309: Edit social worker profile

Like the other profile page, the Edit Social Worker Profile form allows users to update their personal and professional details, including name, staff ID, agency information, placement details, and optional contact information. The email remains fixed as the primary identifier and cannot be changed. At the bottom, users can choose to save changes, cancel edits, or delete their account entirely.

4.6.1.21.5 Edit Law Enforcement Profile page

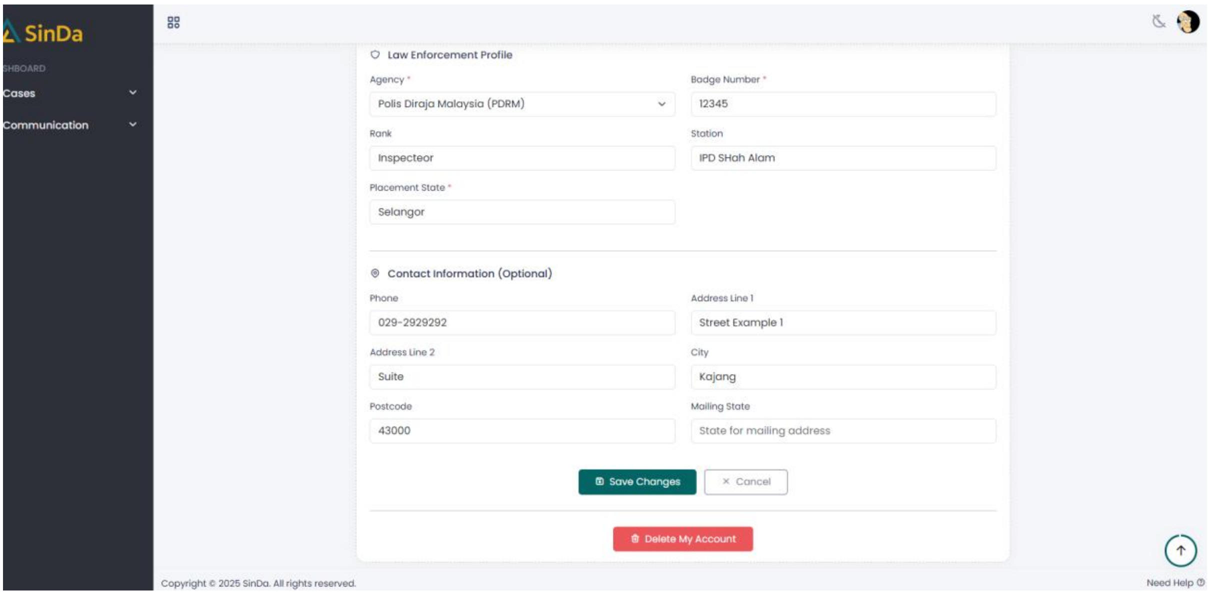
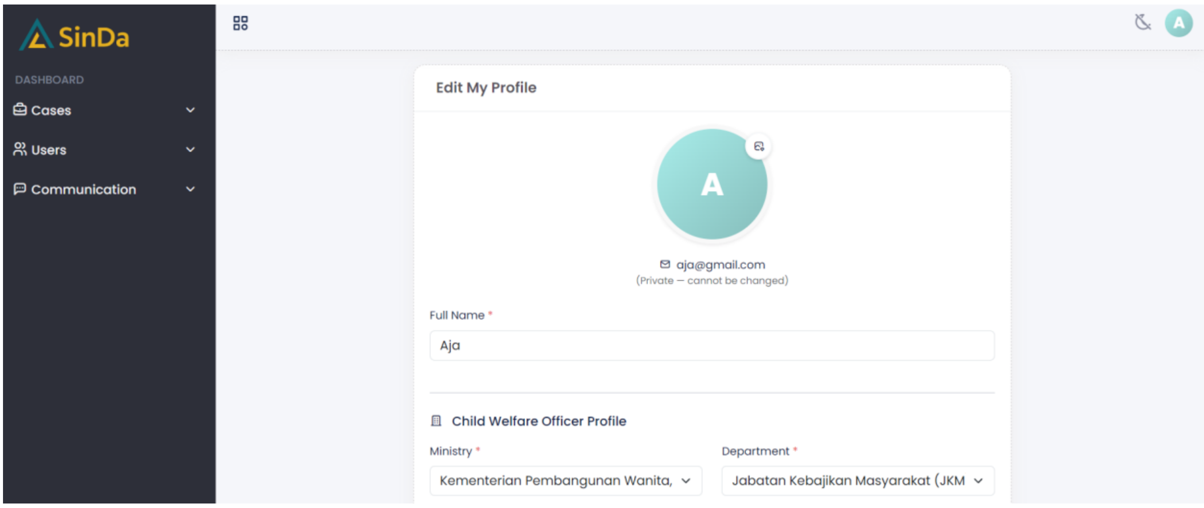


Figure 310: Edit law enforcement profile

Like the other profile page, the Edit Law Enforcement Profile form allows users to update their personal, law enforcement profile or contact information profile. The email remains fixed as the primary identifier and cannot be changed. At the bottom, users can choose to save changes, cancel edits, or delete their account entirely.

4.6.1.21.6 Edit Child Welfare Officer Profile page



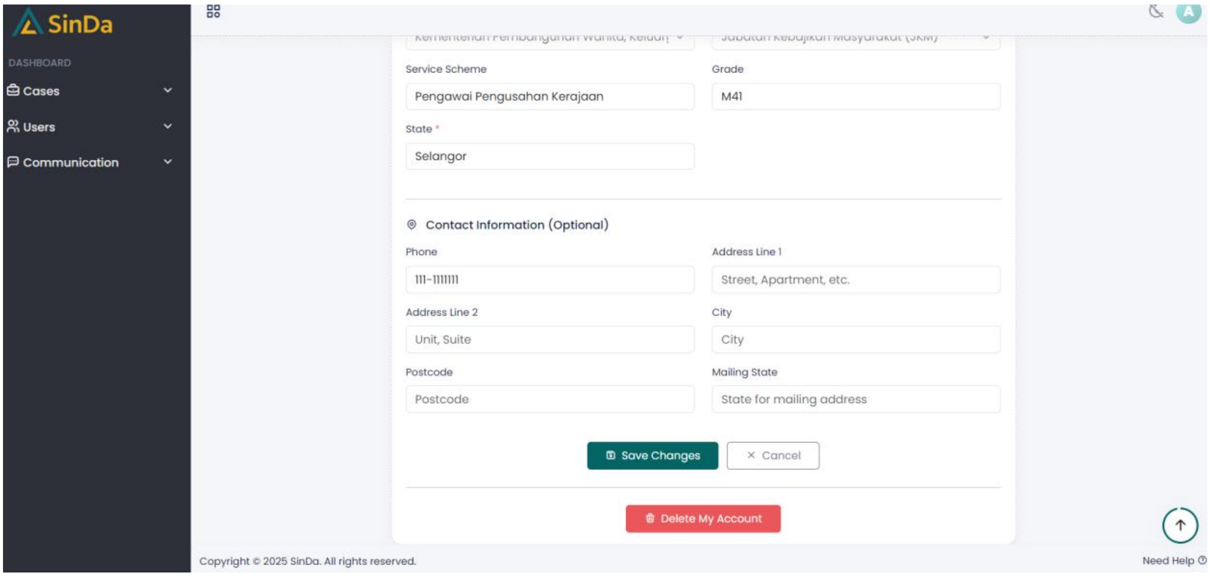
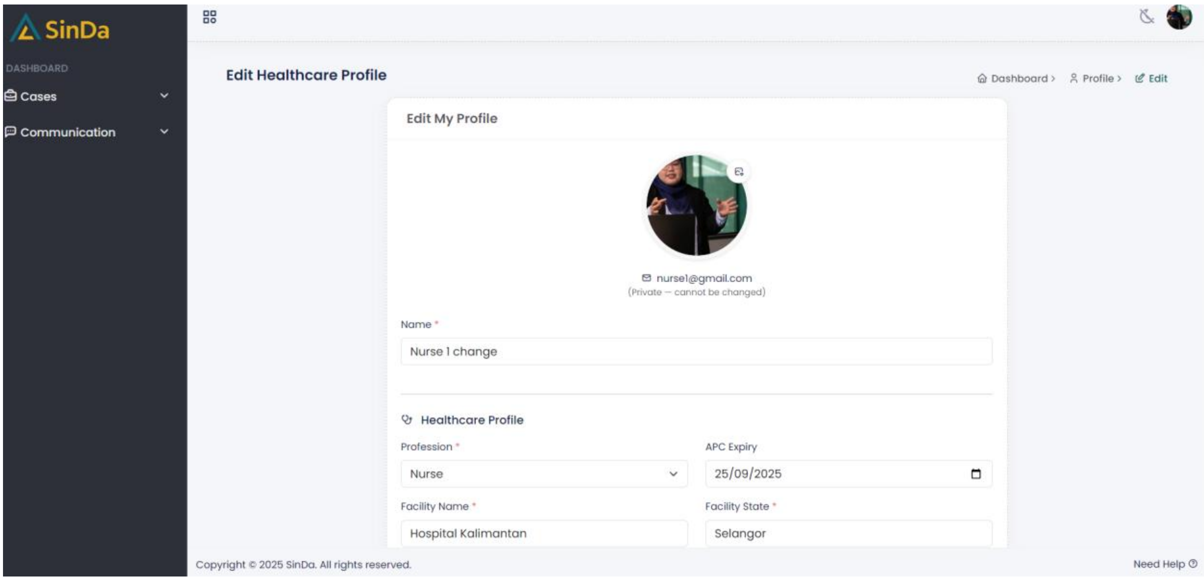


Figure 311: Child welfare officer profile

The Edit Child Welfare Officer Profile form allows users to update their personal, child welfare officers profile or contact information profile. The email remains fixed as the primary identifier and cannot be changed. At the bottom, users can choose to save changes, cancel edits, or delete their account entirely. As for the validation it offers the same as the user's management pages as well with two-tier validation process.

4.6.1.21.7 Edit Healthcare Profile page



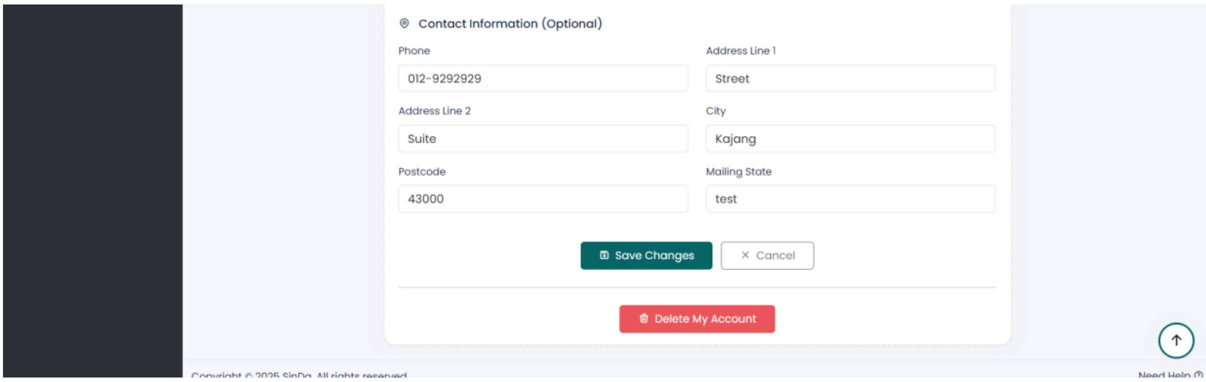


Figure 312: Edit Healthcare Profile page

The Edit Healthcare Profile form allows users to update their personal, Healthcare profile or contact information profile. The email remains fixed as the primary identifier and cannot be changed. At the bottom, users can choose to save changes, cancel edits, or delete their account entirely. As for the validation it offers the same as the user's management pages as well with two-tier validation process.

4.6.2 Sample Codes

4.6.2.1 Secure Messaging feature

```
<div class="card">
    <div class="card-header">
        <h6 class="mb-0"><i class="ti ti-messages"></i> Secure Messages</h
    </div>
    <div class="card-body">
        <!-- Check if case has assignees -->
        @if($assignees->count() > 0)
            <!-- Search Bar -->
            <div class="mb-3">
                <div class="input-group">
                    <span class="input-group-text">
                        <i class="ti ti-search"></i>
                    </span>
                    <input type="text" class="form-control" id="messageSea
                    <button class="btn btn-outline-secondary" type="button
                        <i class="ti ti-x"></i>
                    </button>
                </div>
            </div>

            <!-- Messages Thread -->
            <div id="messagesThread" class="mb-4" style="height:
600px; overflow-y: auto; border: 1px solid #dee2e6;
border-radius: 0.375rem; padding: 1rem;">
                <div class="text-center text-muted">
                    <i class="ti ti-loader ti-spin"></i> Loading messages.
                </div>
            </div>

            <!-- Compose Message Form -->
            <div id="composeMessage" class="border-top pt-3">
                <form id="messageForm">
                    @csrf
                    <div class="mb-3">
                        <label for="messageBody" class="form-label">New Me
                        <textarea
                            class="form-control"
                            id="messageBody"
                            name="body"
                            rows="3"
                            placeholder="Type your message here..."
                            maxlength="5000"
                        ></textarea>
                        <div class="form-text">
                            <span id="charCount">0</span>/5000 characters
                        </div>
                    </div>
                    <button type="submit" class="btn btn-primary" id="send
                        <i class="ti ti-send"></i> Send Message
                    </button>
                </form>
            </div>
        @else
            <!-- No Assignees Message -->
            <div class="text-center py-5">
                <div class="mb-3">
                    <i class="ti ti-users-off fs-1 text-muted"></i>
                </div>
                <h5 class="text-muted mb-3">No Assignees Found</h5>
                <p class="text-muted mb-4">
                    This case needs to have at least one assignee before m
                    Please assign case workers, law enforcement,
or healthcare professionals to this case.
                </p>
                <div class="alert alert-info">
                    <i class="ti ti-info-circle"></i>
                    Contact an administrator to assign
case workers to this case.
                </div>
            </div>
        @endif
    </div>
</div>
```

Figure 313: Secure Messaging UI Component

This code shows the user interface for the secure messaging within the case management. When the user accesses the messaging interface, the system displays a searchable message thread with a scrollable container for viewing conversation history. The system then will also provide a message composition form with character counting and validation which are implemented using java scripts. Furthermore, it also includes conditional rendering that shows an informative message when no case assignees are present which preventing messaging.

```
const caseId = '{{ $report->id }}';
let canPost = true;
let allMessages = []; // Store all messages for search functionality

// Load messages
function loadMessages() {
    $.get(`/cases/${caseId}/messages`, function(data) {
        canPost = data.can_post;
        allMessages = data.messages; // Store all messages
        renderMessages(allMessages);
        updateComposeForm();
        updateMessageCount(allMessages.length);

        // Update message count badge based on assignees
        if (!data.has_assignees) {
            $('#messagesCount').text('-');
        }
    }).fail(function() {
        $('#messagesThread').html(`
            <div class="alert alert-danger">
                <i class="ti ti-alert-circle"></i> Failed to load messages
            </div>
        `);
    });
}
```

```
// Render messages
function renderMessages(messages) {
    if (messages.length === 0) {
        $('#messagesThread').html(`
            <div class="text-center text-muted">
                <i class="ti ti-messages-off fs-1"></i>
                <p class="mt-2">No messages yet</p>
                <small>Start the conversation by sending the first message</small>
            </div>
        `);
        return;
    }

    // Reverse the messages array to show oldest first (chronological order)
    const chronologicalMessages = messages.slice().reverse();

    const messagesHtml = chronologicalMessages.map(message => {
        const isOwnMessage = message.sender_id === '{{ auth()->id() }}';
        const messageClass = isOwnMessage ? 'text-end' : 'text-start';
        const bubbleClass = isOwnMessage ? 'bg-primary text-white' : 'bg-light';

        return `
            <div class="mb-3 ${messageClass}">
                <div class="d-inline-block ${bubbleClass} rounded p-3" style="max-width: 70%;">
                    <div class="message-body">${escapeHtml(message.body)}</div>
                    <div class="message-meta mt-2">
                        <small class="${isOwnMessage 0' : 'text-muted'}">
                            <strong>${escapeHtml(mess Review next file )}</strong> •
```

```
                    ${new Date(message.created_at).toLocaleString()}
                </small>
            </div>
        </div>
    </div>
    `;
}).join('');

$('#messagesThread').html(messagesHtml);
$('#messagesThread').scrollTop($('#messagesThread')[0].scrollHeight);
}
```

```
// Handle message form submission
$('#messageForm').on('submit', function(e) {
    e.preventDefault();
    | Ctrl+L to chat, Ctrl+K to generate
    const messageBody = $('#messageBody').val().trim();
    if (!messageBody) return;

    const $btn = $('#sendMessageBtn');
    const originalText = $btn.html();

    $btn.prop('disabled', true).html('<i class="ti ti-loader ti-spin"></i> Sending...');

    $.ajax({
        url: `/cases/${caseId}/messages`,
        method: 'POST',
        data: {
            _token: $('meta[name="csrf-token"]').attr('content'),
            body: messageBody
        },
        success: function(response) {
            $('#messageBody').val('');
            $('#charCount').text('0');
            loadMessages(); // Reload messages to show the new one
        },
        error: function(xhr) {
            let errorMessage = 'Failed to send message';
            if (xhr.responseJSON && xhr.responseJSON. Review next file >
                errorMessage = xhr.responseJSON.error;
```

```
            const alertHtml = `
                <div class="alert alert-danger alert-dismissible fade show" role="alert">
                    <i class="ti ti-alert-circle"></i> ${errorMessage}
                    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
                </div>
            `;
            $('.container-fluid').prepend(alertHtml);

            setTimeout(() => {
                $('.alert').fadeOut(function() {
                    $(this).remove();
                });
            }, 5000);
        },
        complete: function() {
            $btn.prop('disabled', false).html(originalText);
        }
    });
});
```

```javascript
// Search functionality
$('#messageSearch').on('input', function() {
    const searchTerm = $(this).val().toLowerCase().trim();

    if (searchTerm === '') {
        // Show all messages
        renderMessages(allMessages);
        $('#clearSearch').hide();
    } else {
        // Filter messages
        const filteredMessages = allMessages.filter(message => {
            const messageText = message.body.toLowerCase();
            const senderName = message.sender.name.toLowerCase();
            return messageText.includes(searchTerm) || senderName.includes(searchTerm);
        });

        renderMessages(filteredMessages);
        $('#clearSearch').show();
    }
});

// Auto-refresh messages every 30 seconds
setInterval(loadMessages, 30000);
```

Figure 314: Secure Messaging JavaScript Implementation

These are the javascripts codes that works together with the user interface of the feature. First When the user loads the messaging interface, the system initializes by calling the `loadMessages()` function to fetch existing messages from the server and the `renderMessages()` function to display them in chronological order with proper styling for sent vs received messages. The system then handles form submission through AJAX using the `$('#messageForm').on('submit')` event handler, providing real-time feedback with loading states and error handling. If the message submission fails, the system displays appropriate error messages and automatically dismisses them after 5 seconds. The system also implements search functionality through the `$('#messageSearch').on('input')` event handler that filters messages by content or sender name using the `renderMessages()` function and includes an auto-refresh mechanism using `setInterval(loadMessages, 30000)` that updates the message thread every 30 seconds to ensure real-time communication.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Report;
use App\Models\Message;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;

class CaseMessageController extends Controller
{
    /**
     * Store a new message for a case
     */
    public function store(Request $request, Report $case)
    {
        // Check if user can post to this case
        $this->authorize('post', $case);

        // Validate request
        $request->validate([
            'body' => 'required|string|max:5000',
        ]);

        // Check if case is closed
        if ($case->report_status === 'Closed') {
            return response()->json([
                'error' => 'Cannot send messages to closed cases'
            ], 403);
        }

        // Check if case has assignees
        if ($case->assignees()->count() === 0) {
            return response()->json([
                'error' => 'Cannot send messages to cases without assignee
            ], 403);
        }

        try {
            DB::transaction(function () use ($request, $case) {
                // Create the message
                $message = Message::create([
                    'messageable_type' => Report::class,
                    'messageable_id' => $case->id,
                    'sender_id' => Auth::id(),
                    'body' => $request->body,
                    'attachments' => null, // For future implementation
                ]);

                // Update last_message_at on the case
                $case->update([
                    'last_message_at' => now()
                ]);
            });

            return response()->json([
                'success' => true,
                'message' => 'Message sent successfully'
            ]);

        } catch (\Exception $e) {
            return response()->json([
                'error' => 'Failed to send message'
            ], 500);
        }
    }
}
```

Figure 315: Secure Messaging Backend Code Snippet

Above figure is the Laravel backend code that does the works with the secure messaging feature. First it works where when the user attempts to send a message to a case, the system first performs authorization checks to ensure the suer have the permission to post messages to that specific case. If the user doesn't have the permission, then the system will return a 403 forbidden response else the codes continue. The system then validates the message content and checks if the case is still active and has assigned personnel. Based on the code, if the case status is 'Closed', the system returns an error message preventing the user to send messages. Then it continues where it checks whether if the case has any assigned personnel or not. If there are no assignees exist within that case, the system will return the error message as well based on the code. If all conditions are met, the system will then create the message within the database transaction, linking it to the specific case through a polymorphic relationship and updates the case's last message timestamp. Else if the transaction fail for some reason, the system will the return an error response.

4.6.2.2 Role-Based Permissions & Access Control

```
<nav class="dark-sidebar">
    <div class="app-nav" id="app-simple-bar">
        <ul class="main-nav p-0 mt-2">
            <!-- Cases -->
            @permission('cases.view')
            <li>
                <a data-bs-toggle="collapse" href="#caseManagement" aria-expanded="false">
                    <i class="ti ti-briefcase"></i>
                    Cases
                </a>
                <ul class="collapse" id="caseManagement">
                    <li><a href="{{ route('cases.index') }}"><i class="ti ti-list"></i> View Cases</a></li>
                </ul>
            </li>
            @endpermission
```

```
            <!-- Users -->
            @permission('users.view')
            <li>
                <a data-bs-toggle="collapse" href="#userManagement" aria-expanded="false">
                    <i class="ti ti-users"></i>
                    Users
                </a>
                <ul class="collapse" id="userManagement">
                    <li><a href="{{ route('users.admins') }}"><i class="ti ti-shield"></i> Admins</a></li>
                    <li><a href="{{ route('users.public') }}"><i class="ti ti-user"></i> Public Users</a></li>
                    <li><a href="{{ route('users.social') }}"><i class="ti ti-id-badge"></i> Social Workers</a></li>
                    <li><a href="{{ route('users.law') }}"><i class="ti ti-target"></i> Law Enforcement</a></li>
                    <li><a href="{{ route('users.cwo') }}"><i class="ti ti-building"></i> Child Welfare Officers</a></li>
                    <li><a href="{{ route('users.health') }}"><i class="ti ti-stethoscope"></i> Healthcare Professionals</a></
                    li>
                </ul>
            </li>
            @endpermission
```

```
            <!-- Roles & Permissions -->
            @permission('roles.view')
            <li>
                <a data-bs-toggle="collapse" href="#roleManagement" aria-expanded="false">
                    <i class="ti ti-lock"></i>
                    Roles & Permissions
                </a>
                <ul class="collapse" id="roleManagement">
                    <li><a href="{{ route('roles.index') }}"><i class="ti ti-settings"></i> Manage Roles</a></li>
                    <li><a href="{{ route('permissions.index') }}"><i class="ti ti-key"></i> View Permissions</a></li>
                </ul>
            </li>
            @endpermission
        </ul>
    </div>
```

Figure 316: Role-Based Permissions UI Component

Above figure is the code that implements conditional menu rendering using Laravel Blade directives to enforce role-based access control in the user interface. The `@permission('cases.view')` directive checks if the current authenticated user has the 'cases.view' permission before rendering the Cases menu section. If the permission check passes, it renders the menu item with Bootstrap collapse functionality using `data-bs-toggle="collapse"` and `href="#caseManagement"` attributes. The nested `<ul class="collapse" id="caseManagement">` creates a collapsible submenu that contains the "View Cases" link. Similarly, the `@permission('users.view')` directive conditionally renders the Users menu with multiple submenu items for different user types (Admins, Public Users, Social Workers, etc.), each with appropriate icons using the `ti ti-*` classes. The `@permission('roles.view')` directive controls access to the Roles & Permissions section. The `@endpermission` directive closes each permission block. This approach ensures that menu items are only rendered in the DOM if the user has the necessary permissions, preventing unauthorized users from seeing or accessing restricted functionality.

```
// User permissions for frontend permission checking
const userPermissions = @json($userPermissions ?? []);

// Helper function to check if user has permission
function hasPermission(permission) {
    return userPermissions.includes(permission);
}
```

```
$(function() {
    $('#rolesTable').DataTable({
        processing: true,
        ajax: {
            url: '{{ route('roles.data') }}',
            type: 'GET',
            dataSrc: 'data'
        },
        columns: [
            {
                data: 'name',
                render: function(data, type, row) {
                    const label = data || `ID ${row.id}`;
                    return `<span class="fw-bold">${label}</span>`;
                }
            },
            {
                data: 'description',
                render: function(data, type, row) {
                    return data || '<span class="text-muted">No description</span>';
                }
            },
            {
                data: 'users_count',
                render: function(data, type, row) {
                    return `<span class="badge bg-info">${data} users</span>`;
                }
```

```
                data: null,
                orderable: false,
                searchable: false,
                render: function(data, type, row) {
                    const label = row.name || `ID ${row.id}`;
                    return `
                        <div class="dropdown">
                            <button class="bg-none border-0" type="button" data-bs-toggle="dropdown" aria-expanded="false">
                                <i class="ti ti-dots"></i>
                            </button>
                            <ul class="dropdown-menu">
                                <li>
                                    <a class="dropdown-item" href="/roles/${row.id}">
                                        <i class="ti ti-eye text-info"></i> View
                                    </a>
                                </li>
                                <li>
                                    <button type="button" class="dropdown-item edit-btn"
                                            data-id="${row.id}"
                                            data-bs-toggle="modal"
                                            data-bs-target="#editRole">
                                        <i class="ti ti-edit text-success"></i> Edit
                                    </button>
                                </li>
                                <li>
                                    <a class="dropdown-item" href="/role Review next file › issions">
                                        <i class="ti ti-key text-primary"></i> Permissions
```

```
                                        <i class="ti ti-key text-primary"></i> Permissions
                                    </a>
                                </li>
                                ${row.users_count == 0 ? `
                                <li>
                                    <a class="dropdown-item delete-btn" href="javascript:void(0)"
                                       data-id="${row.id}" data-label="${label}">
                                        <i class="ti ti-trash text-danger"></i> Delete
                                    </a>
                                </li>
                                ` : `
                                <li>
                                    <span class="dropdown-item text-muted" style="cursor: not-allowed;">
                                        <i class="ti ti-trash text-muted"></i> Delete (has users)
                                    </span>
                                </li>
                                `}
                            </ul>
                        </div>
                    `;
                }
            }
        ]
    });
```

```
// Delete functionality
let deleteId = null;
const deleteModalEl = document.getElementById('deleteModal');
const deleteModal = new bootstrap.Modal(deleteModalEl);
const deleteRoleLabelEl = document.getElementById('deleteRoleLabel');

$(document).on('click', '.delete-btn', function() {
    deleteId = $(this).data('id');
    const label = $(this).data('label') || `ID ${deleteId}`;
    deleteRoleLabelEl.textContent = label;
    deleteModal.show();
});

$('#confirmDeleteBtn').on('click', function() {
    if (!deleteId) return;
    const form = document.getElementById('deleteForm');
    form.action = `/roles/${deleteId}`;
    form.submit();
});
```

```
// Edit functionality
$(document).on('click', '#rolesTable .edit-btn', function () {
    const table = $('#rolesTable').DataTable();
    let $tr = $(this).closest('tr');
    if ($tr.hasClass('child')) {
        $tr = $tr.prev('.parent');
    }

    const rowData = table.row($tr).data();
    if (!rowData) return;

    // Modal + form
    const $modal = $('#editRole');
    const $form = $modal.find('form');

    // Replace __ID__ in action template
    const actionTemplate = $form.attr('data-action-template') || $form.attr('action');
    if (actionTemplate) {
        $form.attr('action', actionTemplate.replace('__ID__', rowData.id));
    }
```

````
        // Modal + form
        const $modal = $('#editRole');
        const $form = $modal.find('form');

        // Replace __ID__ in action template
        const actionTemplate = $form.attr('data-action-template') || $form.attr('action');
        if (actionTemplate) {
            $form.attr('action', actionTemplate.replace('__ID__', rowData.id));
        }

        // Fill form fields
        $modal.find('input[name="name"]').val(rowData.name ? rowData.name.replace(/_/g, ' ').replace(/\b\w/g, l => l.
        toUpperCase()) : '');
        $modal.find('textarea[name="description"]').val(rowData.description === 'No description' ? '' : rowData.description);

        $modal.modal('show');
    });
});
```
````

Figure 317: Role-Based Permissions JavaScript Implementation
````

This code implements client-side permission checking and role management functionality. The userPermissions constant stores the user's permissions array from the server, and the hasPermission(permission) function checks if a permission exists using includes(). The DataTable configuration load’s role data via AJAX and renders columns with custom functions. The actions column uses conditional rendering to show active delete buttons only for roles with no users (row.users_count == 0). Event handlers manage delete confirmation modals and edit form population using $(document).on('click') for dynamically generated elements.

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class PermissionMiddleware
{
    public function handle(Request $request, Closure $next,
...$permissions): Response
    {
        $user = $request->user();

        // Guest → send to sign in (preserve intended URL)
        if (!$user) {
            return redirect()->guest(route('sign_in'))
                ->withErrors(['auth' => 'Please sign in first.']);
        }

        // Load role and permissions
        $user->loadMissing(['role.permissions']);

        // Check if user has any of the required permissions
        $hasPermission = false;

        if ($user->role) {
            foreach ($permissions as $permission) {
                if ($user->hasPermission($permission)) {
                    $hasPermission = true;
                    break;
                }
            }
        }

        if (!$hasPermission) {
            // JSON/AJAX (e.g. DataTables) → 403 JSON; normal page
//→ 403 view
            if ($request->ajax() || $request->wantsJson()
|| $request->expectsJson()) {
                return response()->json(['message' =>
'Insufficient permissions'], 403);
            }
            abort(403, 'You do not have
permission to access this resource.');
        }

        return $next($request);
    }
}
```

Figure 318: Role-Based Permissions Middleware Code Snippet

This code implements the PermissionMiddleware class that intercepts HTTP requests and enforces role-based access control. The handle() method receives the incoming request and a closure for the next middleware in the chain. It first checks if a user is authenticated by calling $request->user() - if no user exists, it redirects to the sign-in page using redirect()->guest() while preserving the intended URL. The middleware then loads the user's role and permissions relationship using $user->loadMissing(['role.permissions']) to avoid N+1 query problems. It iterates through the required permissions passed as variadic parameters and calls $user->hasPermission($permission) to check if the user's role contains any of the required permissions. If no permissions match, it returns different error responses based on the request type: JSON responses with 403 status for AJAX requests using $request->ajax(), $request->wantsJson(), or $request->expectsJson() checks, or an HTTP 403 abort for regular page requests. If permissions are satisfied, it calls $next($request) to continue processing the request.

4.6.2.3 Anonymous Reporting with Tracking

```
<div class="card">
    <div class="card-header">
        <h5>Report a Child Protection Concern</h5>
        <p class="text-muted mb-0">Your identity is optional. All reports are confidential and protected under PDPA.</p>
        <p class="text-muted mb-0"><small>* Required fields</small></p>
    </div>
    <div class="card-body">
        <form action="{{ route('report.store') }}" method="POST" class="report-form" enctype="multipart/form-data">
            @csrf
            <div class="row">
                <!-- Reporter Identity (Optional) -->
                <div class="col-md-6">
                    <div class="mb-3">
                        <label for="reporter_name" class="form-label">Your Name (Optional)</label>
                        <input type="text" name="reporter_name" id="reporter_name" class="form-control" placeholder="Enter Your Name" value="{{ old('reporter_name',
                        $prefillName) }}">
                    </div>
                </div>

                <div class="col-md-6">
                    <div class="mb-3">
                        <label for="reporter_email" class="form-label">Your Email (Optional)</label>
                        <input type="email" name="reporter_email" id="reporter_email" class="form-control" placeholder="Enter Your Email" value="{{ old('reporter_email',
                        $prefillEmail) }}" {{ $readonlyEmail ? 'readonly' : '' }}>
                    </div>
                </div>

                <div class="col-md-6">
                    <div class="mb-3">
                        <label for="phone" class="form-label">Your Contact Number (Optional)</label>
                        <input type="tel" name="reporter_phone" id="phone" class="form-control" placeholder="Enter Your Phone" value="{{ old('reporter_phone', $prefillPhone) }}
                        ">
                    </div>
                </div>
```

```
                <div class="col-md-6">
                    <div class="mb-3">
                        <label for="victim_age" class="form-label">Victim's Approximate Age *</label>
                        <input type="number" name="victim_age" id="victim_age" class="form-control" min="0" max="17" required value="{{ old('victim_age') }}" oninvalid="this.
                        setCustomValidity('Please enter an age between 0 and 17.')" oninput="this.setCustomValidity('')">
                    </div>
                </div>

                <!-- Upload Evidence -->
                <div class="col-md-12">
                    <div class="mb-3">
                        <label for="evidence" class="form-label">Upload Evidence (Optional)</label>
                        <input type="file" name="evidence[]" id="evidence" class="form-control" accept="image/*,video/*,application/pdf" multiple onchange="validateFileUpload
                        (this)">
                        <small class="form-text text-muted">You may upload photos, videos, or documents. Maximum 5 files allowed.</small>
                        <div id="file-validation-message" class="text-danger mt-1" style="display: none;"></div>
                    </div>
                </div>

                <!-- Consent Confirmation -->
                <div class="col-12">
                    <div class="mb-3 form-check">
                        <input type="checkbox" name="confirmed_truth" id="confirmed_truth" class="form-check-input" required>
                        <label class="form-check-label" for="confirmed_truth">I confirm that the information provided is accurate to the best of my knowledge. *</label>
                    </div>
                </div>

                <!-- Submit -->
                <div class="col-12">
                    <button type="submit" class="btn btn-primary btn-lg">
                        <i class="ti ti-send"></i> Submit Report
                    </button>
                </div>
            </div>
        </form>
    </div>
</div>
```

Figure 319: Anonymous Reporting UI Component

Above is the form for anonymous reporting with privacy-focused design elements. Based on the figure above, the form uses action="{{ route('report.store') }}" to submit to the backend route, method="POST" for data transmission, and enctype="multipart/form-data" to handle file uploads. The @csrf directive generates a CSRF token for security. Form fields use Bootstrap's grid system with col-md-6 classes for responsive layout. Personal information fields include name="reporter_name" with placeholder="Enter Your Name" and value="{{ old('reporter_name', $prefillName) }}" to preserve form data on validation errors. The victim age field uses type="number" with min="0" max="17" attributes and oninvalid="this.setCustomValidity('...')" for custom validation messages. The evidence upload field uses name="evidence[]" for multiple file selection, accept="image/*,video/*,application/pdf" to restrict file types, and onchange="validateFileUpload(this)" to trigger client-side validation. The consent checkbox uses name="confirmed_truth" with required attribute to enforce mandatory agreement before submission.

```javascript
function validateFileUpload(input) {
    const maxFiles = 5;
    const files = input.files;
    const messageDiv = document.getElementById('file-validation-message');

    if (files.length > maxFiles) {
        messageDiv.textContent = `Maximum ${maxFiles} files allowed. You selected ${files.length} files. Please select only ${maxFiles} files or fewer.`;
        messageDiv.style.display = 'block';
        messageDiv.className = 'text-danger mt-1';

        // Clear the input to prevent submission
        input.value = '';
        return false;
    } else if (files.length > 0) {
        // Show success message for valid selection
        messageDiv.textContent = `${files.length} file(s) selected. Maximum ${maxFiles} files allowed.`;
        messageDiv.style.display = 'block';
        messageDiv.className = 'text-success mt-1';
        return true;
    } else {
        // No files selected
        messageDiv.style.display = 'none';
        return true;
    }
}
```

```javascript
function validateIncidentDate(input) {
    const selectedDate = new Date(input.value);
    const today = new Date();
    today.setHours(23, 59, 59, 999); // Set to end of today to allow today's date

    if (selectedDate > today) {
        alert('Please select a date that is not in the future.');
        input.value = ''; // Clear the invalid date
        input.focus();
        return false;
    }
    return true;
}
```

```javascript
// Form validation before submit
document.querySelector('.report-form').addEventListener('submit', function(e) {
    const fileInput = document.getElementById('evidence');
    const dateInput = document.getElementById('incident_date');
    const abuseTypeCheckboxes = document.querySelectorAll('input[name="abuse_types[]"]:checked');
    const messageDiv = document.getElementById('file-validation-message');
    const submitBtn = document.getElementById('submitBtn');
    const submitText = document.getElementById('submitText');
    const submitSpinner = document.getElementById('submitSpinner');

    // Check if form is already being submitted
    if (submitBtn.disabled) {
        e.preventDefault();
        return false;
    }

    // Validate abuse types selection
    if (abuseTypeCheckboxes.length === 0) {
        alert('Please select at least one type of abuse.');
        e.preventDefault();
        return false;
    }

    // Validate file upload - check file count
    if (fileInput.files.length > 5) {
        messageDiv.textContent = 'Maximum 5 files allowed. Please select fewer files before submitting.';
        messageDiv.style.display = 'block';
        messageDiv.className = 'text-danger mt-1';
        e.preventDefault();
        return false;
    }

    // Validate file upload - check individual file validation
    if (!validateFileUpload(fileInput)) {
        e.preventDefault();
        return false;
    }
```

```javascript
    // Validate incident date
    if (dateInput.value && !validateIncidentDate(dateInput)) {
        e.preventDefault();
        return false;
    }

    // If all validations pass, disable submit button to prevent double submission
    submitBtn.disabled = true;
    submitText.textContent = 'Submitting...';
    submitSpinner.style.display = 'inline-block';
});
```

Figure 320: Anonymous Reporting JavaScript Implementation

```php
namespace App\Http\Controllers;

use App\Models\Report;
use Illuminate\Http\Request;
use Illuminate\Support\Str;

class ReportController extends Controller
{
    public function store(Request $request)
    {
        $validated = $request->validate([
            'reporter_name' => 'nullable|string|max:255',
            'reporter_email' => 'nullable|email|max:255',
            'reporter_phone' => 'nullable|string|max:20',
            'victim_age' => 'required|string|max:10',
            'victim_gender' => 'required|string|in:Male,Female,Other',
            'abuse_types' => 'required|array|min:1',
            'abuse_types.*' => 'required|string|in:Physical Abuse,Emotional Abuse,Sexual Abuse,Neglect,Exploitation',
            'incident_description' => 'required|string',
            'incident_location' => 'required|string',
            'incident_date' => 'required|date|before_or_equal:today',
            'suspected_abuser' => 'nullable|string|max:255',
            'evidence' => 'nullable|array|max:5',
            'evidence.*' => 'file|mimes:jpg,jpeg,png,mp4,pdf|max:20480',
            'confirmed_truth' => 'accepted'
        ]);

        $filePaths = [];
        if ($request->hasFile('evidence')) {
            foreach ($request->file('evidence') as $file) {
                $filePaths[] = $file->store('evidence', 'public');
            }
        }
```

```php
        Report::create([
            'id' => Str::uuid(), // Generate UUID
            'user_id' => auth()->check() ? auth()->id() : null, // Link if Logged in

            'reporter_name' => $validated['reporter_name'] ?? 'anonymous',
            'reporter_email' => $validated['reporter_email'] ?? 'anonymous@gmail.com',
            'reporter_phone' => $validated['reporter_phone'] ?? null,
            'victim_age' => $validated['victim_age'] ?? null,
            'victim_gender' => $validated['victim_gender'] ?? null,
            'abuse_types' => $validated['abuse_types'] ?? [],
            'incident_description' => $validated['incident_description'],
            'incident_location' => $validated['incident_location'],
            'incident_date' => $validated['incident_date'],
            'suspected_abuser' => $validated['suspected_abuser'] ?? null,
            'evidence' => $filePaths,
            'confirmed_truth' => true,
            'report_status' => 'Submitted',
            'priority_level' => 'Medium',
        ]);

        return redirect()->back()->with('success', 'Your report has been submitted successfully.');
    }
}
```

Figure 321: Anonymous Reporting Backend Code Snippet

This code implements the store() method in the ReportController class that handles anonymous report submissions. The method first validates incoming request data using Laravel's $request->validate() method with specific rules: reporter_name is nullable with a 255-character limit, victim_age is required as a string with 10-character limit, abuse_types must be an array with at least one element where each item must match predefined abuse types, incident_date must be a valid date not in the future using before_or_equal:today, and evidence files are limited to 5 files with specific MIME types and 20MB size limit. The code then processes file uploads using $request->hasFile('evidence') to check for uploaded files and $file->store('evidence', 'public') to store each file in the public disk. The Report::create() method creates a new report record with Str::uuid() generating a unique identifier, auth()->check() ? auth()->id() : null conditionally linking to a user account, and the null coalescing operator (??) setting default values for anonymous submissions like 'anonymous' for name and 'anonymous@gmail.com' for email. The method returns a redirect with a success message using redirect()->back()->with('success', ...).

4.7 Summary

The proposed systems design and execution were covered in this chapter. Diagrams such as use cases, activity, class and sequence diagrams are all covered in the design part to describe the system interactions, workflows and behaviour. An Entity Relationship diagram and database schema were used here to demonstrate the structure and relationship between each key module. Wireframes of the proposed system were also illustrated to show the interface design. Furthermore, to verify functionality and check on the user stratifications on later chapter, test case preparation has been made which included both unknit testing and user acceptance testing plans. Along with the executions to show the system screenshots, sample code snippets were also included to highlight important implementation logic and demonstrate how the framework and technologies were applied. In conclusion, this chapter provided a complete technical overview of the system’s design, development, and testing preparation, ensuring the system is ready for evaluation and practical deployment.

Chapter 5: Result and Discussion

5.1 Unit Testing

5.1.1 Sign In

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid email and password	- Login successful - Redirect to appropriate page	- Login successful - Redirect to appropriate page	Pass
TC2	User enters invalid email format (e.g., "test@")	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC3	User enters wrong password	-Display "Credentials do not match" error	-Display "Credentials do not match" error	Pass
TC4	User leaves email field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC5	User leaves password field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC6	User clicks "Remember me" checkbox	- Login successful with remember token set	- Login successful with remember token set	Pass

TC7	User clicks "Forgot Password" link	- Open password reset modal	- Open password reset modal	Pass
TC8	User clicks "Sign Up" link	- Redirect to registration page	- Redirect to registration page	Pass

Table 133: Sign In Unit Testing

5.1.2 Reset Password Unit Testing

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid email in forgot password form	- Send password reset email - Display "Reset link sent" message	- Send password reset email - Display "Reset link sent" message	Pass
TC2	User enters invalid email format in forgot password	- Display "Enter valid email" error - Keep modal open	- Display "Enter valid email" error - Keep modal open	Pass
TC3	User enters non-existent email in forgot password	- Display "Credentials do not match" error	- Display "Credentials do not match" error	Pass
TC4	User clicks reset link from email	- Display password reset form - Show new password and confirm fields	- Display password reset form - Show new password and confirm fields	Pass
TC5	User enters matching new passwords (8+ chars)	- Password reset successful - Redirect to login page - Display success message	- Password reset successful - Redirect to login page - Display success message	Pass

TC6	User enters passwords that don't match	- Display "Passwords do not match" error - Stay on reset form	- Display "Passwords do not match" error - Stay on reset form	Pass
TC7	User enters password less than 8 characters	- Display "Password must be at least 8 characters" error - Stay on reset form	- Display "Password must be at least 8 characters" error - Stay on reset form	Pass

Table 134 Reset Password Unit Testing

5.1.3 Sign Up

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid username, email, and matching passwords	- Account created successfully - Redirect to login page - Display "Account created successfully" message	- Account created successfully - Redirect to login page - Display "Account created successfully" message	Pass
TC2	User enters incomplete email format (e.g., "gtes@"	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC3	User enters already registered email	- Display "This email is already registered" error	- Display "This email is already registered" error	Pass

		- Keep form data for retry	- Keep form data for retry	
TC4	User leaves username field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC5	User leaves email field empty	- Browser prevents form submission - Display browser validation message- Display success message	- Browser prevents form submission - Display browser validation message- Display success message	Pass
TC6	User enters passwords that don't match	- Display "Passwords do not match" error - Stay on reset form	- Display "Passwords do not match" error - Stay on reset form	Pass
TC7	User enters password less than 8 characters	- Display "Password must be at least 8 characters" error - Stay on reset form	- Display "Password must be at least 8 characters" error - Stay on reset form	Pass
TC8	User leaves password field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass

TC9	User doesn't check Terms & Conditions checkbox	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC10	User clicks "Sign in" link	- Redirect to login page -Maintain guest session	- Redirect to login page -Maintain guest session	Pass
TC11	User clicks "Back to Home" link	- Redirect to landing page - Maintain guest session		Pass

Table 135: Sign Up Unit Testing

5.1.4 Create Report

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User enters valid username, email, and matching passwords	- Report submitted successfully - Display "Your report has been submitted successfully" message	- Report submitted successfully - Display "Your report has been submitted successfully" message	Pass
TC2	User leaves victim age field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass

TC3	User enters victim age greater than 17	- Browser prevents form submission - Display "Please enter an age between 0 and 17" message	- Browser prevents form submission - Display "Please enter an age between 0 and 17" message	Pass
TC4	User doesn't select victim gender	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC5	User doesn't select any abuse type	- Display "Please select at least one type of abuse" error - Stay on form	- Display "Please select at least one type of abuse" error - Stay on form	Pass
TC6	User leaves incident description empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC7	User leaves incident location empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC8	User leaves incident date empty	- Browser prevents form submission	- Browser prevents form submission	Pass

		- Display browser validation message	- Display browser validation message	
TC9	User doesn't check confirmation checkbox	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC10	User uploads more than 5 files	- Display "Maximum 5 files allowed" error - Clear file selection	- Display "Maximum 5 files allowed" error - Clear file selection	Pass
TC11	User uploads invalid file type	- Display file type validation error - Clear file selection	- Display file type validation error - Clear file selection	Pass
TC12	User uploads file larger than 20MB	- Display file size validation error - Clear file selection	- Display file size validation error - Clear file selection	Pass
TC13	User enters incomplete email format (e.g., "gtes@"	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC14	User clicks Reset button	- Clear all form fields - Reset to initial state	- Clear all form fields - Reset to initial state	Pass
TC15	Logged-in user accesses report form	- Pre-fill name, email, phone from profile	- Pre-fill name, email, phone from profile	Pass

		- Make email field readonly	- Make email field readonly	

Table 136: Create Report

5.1.5 Create Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User fills all required fields and submits valid query	- Query submitted successfully - Display "Thank you for your message. We will get back to you soon!" message - Redirect to same page	- Query submitted successfully - Display "Thank you for your message. We will get back to you soon!" message - Redirect to same page	Pass
TC2	User leaves name field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC3	User enters invalid email format (e.g., "test@")	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC4	User leaves email field empty	- Browser prevents form submission	- Browser prevents form submission	Pass

		- Display browser validation message	- Display browser validation message	
TC5	User leaves subject field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC6	User leaves message field empty	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC7	User enters subject with more than 15 words (e.g., 20 word sentence)	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC8	User enters message with more than 150 words (e.g., 200 word message)	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass

Table 137: Create Query

5.1.6 View Submitted Report

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	Logged-in user accesses "My Reports" page with existing reports	- Display reports table with all user's reports	- Display reports table with all user's reports	Pass
TC2	Logged-in user accesses "My Reports" page with no reports	- Display "No Reports Found" message - Show "Create Your First Report" button	- Display "No Reports Found" message - Show "Create Your First Report" button	Pass
TC3	User clicks "View Details" button on a report	- Open modal with full report details Display victim info, incident details, abuse types - Show assignees and timeline information	- Open modal with full report details Display victim info, incident details, abuse types - Show assignees and timeline information	Pass
TC4	User clicks "Export Report" button	- Generate PDF download - Display "PDF generated successfully" message - Download file with proper naming convention	- Generate PDF download - Display "PDF generated successfully" message - Download file with proper naming convention	Pass

TC5	User searches for specific report in table	- Filter table results based on search term - Update result count	- Filter table results based on search term - Update result count	Pass
TC6	User closes report detail modal	- Close modal and return to table view - Maintain table state and position - Preserve current filters/sort	- Close modal and return to table view - Maintain table state and position - Preserve current filters/sort	Pass

Table 138: View Submitted Report

5.1.7 Authorized users Dashboard

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	Administrator user accesses dashboard	- Display "SinDa Admin Dashboard" heading - Show all 4 statistics cards (Total Reports, Active Cases, Total Users, Contact Queries) - Display "Real-time overview of system activity and trends"	- Display "SinDa Admin Dashboard" heading - Show all 4 statistics cards (Total Reports, Active Cases, Total Users, Contact Queries) - Display "Real-time overview of system activity and trends"	Pass
TC2	Government Child Welfare Officer accesses dashboard	- Display "SinDa Government Official Dashboard" heading	- Display "SinDa Government Official Dashboard" heading	Pass

		- Show all 4 statistics cards - Display "Overview of all cases and system activity"	- Show all 4 statistics cards - Display "Overview of all cases and system activity"	
TC3	Social Worker accesses dashboard	- Display "SinDa Social Worker Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	- Display "SinDa Social Worker Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	Pass
TC4	Law Enforcement accesses dashboard	- Display "SinDa Law Enforcement Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	- Display "SinDa Law Enforcement Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	Pass
TC5	Healthcare Professional accesses dashboard	- Display "SinDa Healthcare Professional Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	- Display "SinDa Healthcare Professional Dashboard" heading - Show 2 statistics cards (My Cases, Active Assignments) - Only see assigned cases	Pass
TC6	User clicks "Export Reports"	- Export dropdown menu opens - Shows CSV Export and PDF Export options	- Export dropdown menu opens - Shows CSV Export and PDF Export options	Pass

		- Displays available export types (All Cases, All Users, All Queries, Session Logs)	- Displays available export types (All Cases, All Users, All Queries, Session Logs)	
TC7	Dashboard shows recent reports table	- Display table with latest 5 reports - Show reporter name, description, status - Include timestamps	- Display table with latest 5 reports - Show reporter name, description, status - Include timestamps	Pass
TC8	Dashboard handles empty data	- Show "No case data available" message - Display appropriate empty states - Handle zero counts gracefully	- Show "No case data available" message - Display appropriate empty states - Handle zero counts gracefully	Pass

Table 139: Authorized user Dashboard

5.1.8 Case Management

5.1.8.1 View Cases

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses case management page with different roles	- **Administrator/Child Welfare Officer**: Page loads with All cases and statistics	- **Administrator/Child Welfare Officer**: Page loads with All cases and statistics	Pass

		- **Social Worker/Law Enforcement/Healthcare**: Page loads with only assigned cases and filtered statistics - Data table shows cases based on user role	- **Social Worker/Law Enforcement/Healthcare**: Page loads with only assigned cases and filtered statistics - Data table shows cases based on user role	
TC2	User views case data table with existing cases	- Data table displays cases based on user role	- Data table displays cases based on user role	Pass
TC3	User views case data table with no accessible cases	- Empty data table displayed - "No data available" message shown	- Empty data table displayed - "No data available" message shown	Pass
TC4	User searches case by reporter name in data table (e.g., search "John")	- Filtered results displayed from user's accessible cases - Only matching cases shown	- Filtered results displayed from user's accessible cases - Only matching cases shown	Pass
TC5	User without create permission views case management page	- Page loads successfully - Add Case button not visible	- Page loads successfully - Add Case button not visible	Pass
TC6	User with create permission views case management page	- Page loads successfully - Add Case button visible	- Page loads successfully - Add Case button visible	Pass

TC7	User with create permission views case management page clicks "Add" button	- Add Case modal opens successfully - Modal displays all form fields	- Add Case modal opens successfully - Modal displays all form fields	Pass
TC8	User without view history permission views case management page	- Page loads successfully - View History buttons not visible in action dropdown	- Page loads successfully - View History buttons not visible in action dropdown	Pass
TC9	User with view history permission views case management page	- Page loads successfully - View History buttons visible in action dropdown	- Page loads successfully - View History buttons visible in action dropdown	Pass
TC10	User with view history permission clicks "View History" button	- History modal opens with loading state - Case history data loads and displays	- History modal opens with loading state - Case history data loads and displays	Pass
TC11	User without edit permission views case management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC12	User with edit permission views case management page	- Page loads successfully - Edit button visible in action dropdown	- Page loads successfully - Edit button visible in action dropdown	Pass

TC13	User with edit permission clicks "Edit" button	- Edit modal opens with loading state - Case data loads and populates form fields - All form fields are editable and pre-filled	- Edit modal opens with loading state - Case data loads and populates form fields - All form fields are editable and pre-filled	Pass
TC14	User without delete permission views case management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC15	User with delete permission views case management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass
TC16	User with delete permission clicks "Delete" button	- Delete confirmation modal opens - Case ID and reporter name displayed	- Delete confirmation modal opens - Case ID and reporter name displayed	Pass
TC17	User clicks "Cancel" button in delete modal	- Modal closes without action - Case remains in system - Return to cases list view	- Modal closes without action - Case remains in system - Return to cases list view	Pass
TC18	Admin clicks "Delete Case" button in modal	- Case deleted successfully - Display "Case deleted successfully" message	- Case deleted successfully - Display "Case deleted successfully" message	Pass

		- Remove case from list	- Remove case from list	
TC19	User without export permission views case management page	- Page loads successfully - Export PDF buttons not visible in action dropdown	- Page loads successfully - Export PDF buttons not visible in action dropdown	Pass
TC20	User with export permission clicks "Export PDF" button	- PDF generation starts with loading state - PDF file downloads automatically	- PDF generation starts with loading state - PDF file downloads automatically	Pass

Table 140: View Cases

5.1.8.2 Create Case

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Add Case" button	- Add Case modal opens - Form displays with all fields	- Add Case modal opens - Form displays with all fields	Pass
TC2	User fills all required fields and submits	- Case created successfully - Success message displayed - Case appears in table	- Case created successfully - Success message displayed - Case appears in table	Pass
TC3	User submits form with missing reporter name	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass

TC4	User submits form with missing reporter email	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC5	User submits form with invalid email format (e.g., "john@")	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC6	User submits form with missing incident description	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC7	User submits form with missing incident location	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC8	User submits form with missing incident date	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass

TC9	User submits form with missing status	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC10	User submits form with missing priority	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC11	User uploads more than 5 files (e.g., selects 6 images)	"You can only select up to 5 files" error message displayed - Files cleared from input	"You can only select up to 5 files" error message displayed - Files cleared from input	Pass

Table 141: Create Case

5.1.8.3 Edit Case Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User with edit permission clicks "Edit" button	- Edit modal opens with loading state - Case data loads and populates form fields	- Edit modal opens with loading state - Case data loads and populates form fields	Pass

TC2	User successfully updates case with valid data	- Case updated successfully - Success message displayed - Changes reflected in data table	- Case updated successfully - Success message displayed - Changes reflected in data table	Pass
TC3	User submits form with missing reporter name	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC4	User submits form with missing reporter email	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC5	User submits form with invalid email format (e.g., "john@")	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC6	User submits form with missing incident description	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC7	User submits form with missing incident location	- Browser prevents form submission	- Browser prevents form submission	Pass

		- Display browser validation message	- Display browser validation message	
TC8	User submits form with missing incident date	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC9	User submits form with missing status	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC10	User submits form with missing priority	- Browser prevents form submission - Display browser validation message	- Browser prevents form submission - Display browser validation message	Pass
TC11	User uploads more than 5 files (e.g., selects 6 images)	"You can only select up to 5 files" error message displayed - Files cleared from input	"You can only select up to 5 files" error message displayed - Files cleared from input	Pass
TC12	User creates case and cancels form	- Modal closes without saving - No case created - Form data cleared - No success/error messages	- Modal closes without saving - No case created - Form data cleared - No success/error messages	Pass

Table 142: Edit Case Details

5.1.8.4 View Case Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "View Case Details" button	- Case details page loads successfully - All case information displays correctly	- Case details page loads successfully - All case information displays correctly	Pass
TC2	Admin user views case details page	- Full case information displayed - All action buttons visible (Edit, Export, History)	- Full case information displayed - All action buttons visible (Edit, Export, History)	Pass
TC3	Government Child Welfare Officer views any case details	- Full case information displayed - All administrative functions available	- Full case information displayed - All administrative functions available	Pass
TC4	User views case with evidence files	- Evidence section displays correctly - File viewer modal opens when clicking evidence	- Evidence section displays correctly - File viewer modal opens when clicking evidence	Pass
TC5	User clicks "View History" button	- Case history modal opens - Historical actions displayed chronologically	- Case history modal opens - Historical actions displayed chronologically	Pass

TC6	User clicks "Export Case" button	- PDF download starts automatically	- PDF download starts automatically	Pass
TC7	User clicks "Edit Case" button	- Edit Case modal opens with loading state - Case data loads and populates form fields	- Edit Case modal opens with loading state - Case data loads and populates form fields	Pass
TC8	User clicks “Message”	- Page smoothly scrolls to messages section	- Page smoothly scrolls to messages section	Pass
TC9	User views case with no assignees and checks messages section	- "No Assignees Found" displayed - Message composition form disabled	- "No Assignees Found" displayed - Message composition form disabled	Pass
TC10	User views case with assignees and no messages	- "No messages yet" displayed - "Start the conversation by sending the first message" shown	- "No messages yet" displayed - "Start the conversation by sending the first message" shown	Pass
TC11	User views case with existing messages	- All messages displayed chronologically - Sender names and timestamps shown	- All messages displayed chronologically - Sender names and timestamps shown	Pass
TC12	User searches for messages using search bar	- Search results filtered correctly - Clear search button appears	- Search results filtered correctly - Clear search button appears	Pass

Table 143: View Case Details

5.1.9 Admin Management

5.1.9.1 View Admins

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses admin management page	- Page loads successfully with admin dashboard - Statistics cards display (Total Admins, Active Admins, Inactive Admins, New Admins) - Data table shows admin users list - Add button is visible	- Page loads successfully with admin dashboard - Statistics cards display (Total Admins, Active Admins, Inactive Admins, New Admins) - Data table shows admin users list - Add button is visible	Pass
TC2	User views admin data table with no admin users	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown - Statistics show zero values	Pass
TC3	User searches admin by name in data table (e.g., search "John")	- Filtered results displayed - Only matching admins shown	- Filtered results displayed - Only matching admins shown	Pass
TC4	User without create permission views admin management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass

TC5	User with create permission views case management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass
TC6	User without edit permission views admin management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC7	User with edit permission views admin management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views admin management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC9	User with delete permission views admin management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 144: View Admins

5.1.9.2 Create Admin

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User submits admin creation form with all required fields filled correctly	- Admin user successfully created - Success message displayed - Modal closes automatically - New admin appears in admin list	- Admin user successfully created - Success message displayed - Modal closes automatically - New admin appears in admin list	Pass
TC2	User submits admin creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits admin creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits admin creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	Pass
TC5	User submits admin creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass

TC6	User submits admin creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	Pass
TC7	User submits public user creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass
TC8	User submits admin creation form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC9	User submits admin creation form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC10	User creates admin and cancels form by clicking 'Close' button	- Modal closes without saving - No admin user created - Form data cleared	- Modal closes without saving - No admin user created - Form data cleared	Pass

Table 145: Create Admin

5.1.9.3 Edit Admin Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit admin form with all required fields updated correctly required fields filled correctly	- Admin user successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in admin list	- Admin user successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in admin list	Pass
TC2	User submits edit admin form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit admin form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits edit admin form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass

TC5	User edits admin and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to admin data - Form data cleared	- Modal closes without saving - No changes made to admin data - Form data cleared	Pass
TC6	User uploads new avatar	- Avatar previews review the selected avatar	- Avatar previews review the selected avatar	Pass

Table 146: Edit Admin Details

5.1.9.4 Delete Admin

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User confirms admin deletion by clicking "Delete" button in modal	- Admin user successfully deleted from database - Success message displayed - Admin removed from admin list - Modal closes automatically	- Admin user successfully deleted from database - Success message displayed - Admin removed from admin list - Modal closes automatically	Pass
TC2	User cancels admin deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to admin data - Admin remains in admin list	- Modal closes without deleting - No changes made to admin data - Admin remains in admin list	Pass
TC3	User cancels admin deletion by clicking	Modal closes without deleting - No changes made to admin data - Admin remains in admin list	Modal closes without deleting - No changes made to admin data - Admin remains in admin list	Pass

	"Close" (X) button in modal			

Table 147: Delete Admin

5.1.10 Public User Management

5.1.10.1 View Public Users

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses public user management page	- Page loads successfully with public user dashboard - Statistics cards display (Total Users, Contactable Users, Non-Contactable Users, New Users) - Data table shows public users list	- Page loads successfully with public user dashboard - Statistics cards display (Total Users, Contactable Users, Non-Contactable Users, New Users) - Data table shows public users list	Pass
TC2	User views public user data table with no public users	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown - Statistics show zero values	Pass
TC3	User searches public user by name in data table (e.g., search "John")	- Filtered results displayed - Only matching public users shown	- Filtered results displayed - Only matching public users shown	Pass

TC4	User without create permission views public user management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass
TC5	User with create permission views public user management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass
TC6	User without edit permission views public user management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC7	User with edit permission views public user management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views public user management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC7	User with delete permission views public user management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 148: View Public Users

5.1.10.2 Create Public User

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits public user creation form with all required fields filled correctly	- Public user successfully created - Success message displayed - Modal closes automatically - New public user appears in public user list	- Public user successfully created - Success message displayed - Modal closes automatically - New public user appears in public user list	Pass
TC2	User submits public user creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits public user creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits public user creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	Pass

TC5	User submits public user creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass
TC6	User submits public user creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	Pass
TC7	User submits public user creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass
TC8	User submits public user creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass
TC9	User creates public user and cancels form by clicking 'Close' button	- Modal closes without saving - No public user created - Form data cleared	- Modal closes without saving - No public user created - Form data cleared	Pass

Table 149: Create Public User

5.1.10.3 Edit Public User Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit public user form with all required fields updated correctly	- Public user successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in public user list	- Public user successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in public user list	Pass
TC2	User submits edit public user form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit public user form with missing display name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User edits public user and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to public user data - Form data cleared	- Modal closes without saving - No changes made to public user data - Form data cleared	Pass

TC5	User edits admin and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to admin data - Form data cleared	- Modal closes without saving - No changes made to admin data - Form data cleared	Pass

Table 150: Edit Public User Details

5.1.10.4 Delete Public User

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms public user deletion by clicking "Delete" button in modal	- Public user successfully deleted from database - Success message displayed - Public user removed from public user list - Modal closes automatically	- Public user successfully deleted from database - Success message displayed - Public user removed from public user list - Modal closes automatically	Pass
TC2	User cancels public user deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to public user data - Public user remains in public user list	- Modal closes without deleting - No changes made to public user data - Public user remains in public user list	Pass
TC3	User cancels public user deletion by clicking	Modal closes without deleting	Modal closes without deleting	Pass

	"Close" (X) button in modal	- No changes made to public user data - Public user remains in public user list	- No changes made to public user data - Public user remains in public user list	

Table 151: Delete Public User

5.1.11 Social Worker Management

5.1.11.1 View Social Workers

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses social worker management page	- Page loads successfully with social worker dashboard - Statistics cards display (Total Workers, Agencies Covered, States Covered, New Users) - Data table shows social worker users list	- Page loads successfully with social worker dashboard - Statistics cards display (Total Workers, Agencies Covered, States Covered, New Users) - Data table shows social worker users list	Pass
TC2	User views social worker data table with no social workers	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown	Pass

			- Statistics show zero values	
TC3	User searches social worker by name in data table (e.g., search "John")	- Filtered results displayed - Only matching social workers shown	- Filtered results displayed - Only matching social workers shown	Pass
TC4	User without create permission views social worker management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass
TC5	User with create permission views social worker management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass
TC6	User without edit permission views social worker management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC7	User with edit permission views social worker management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views social worker management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass

TC9	User with delete permission views social worker management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 152: View Social Worker

5.1.11.2 Create Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits social worker creation form with all required fields filled correctly	- Social worker successfully created - Success message displayed - Modal closes automatically - New social worker appears in social worker list	- Social worker successfully created - Success message displayed - Modal closes automatically - New social worker appears in social worker list	Pass
TC2	User submits social worker creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits social worker creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass

TC4	User submits social worker creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	Pass
TC5	User submits social worker creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass
TC6	User submits social worker creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	Pass
TC6	User submits public social worker form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass

TC7	User submits social worker creation form with missing staff ID field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC8	User submits social worker creation form with missing agency name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC9	User submits social worker creation form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC10	User submits social worker creation form with missing placement district field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC11	User selects "Other" for agency name but doesn't provide agency name other	- Form submission proceeds but server validation catches this - Display server validation error: "Please enter the agency name when 'Other' is chosen"	- Form submission proceeds but server validation catches this - Display server validation error: "Please enter the agency name when 'Other' is chosen"	Pass

TC12	User creates social worker and cancels form by clicking 'Close' button	- Modal closes without saving - No social worker created - Form data cleared	- Modal closes without saving - No social worker created - Form data cleared	Pass

Table 153: Create Social Worker

5.1.11.3 Edit Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit social worker form with all required fields updated correctly	- Social worker successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in social worker list	- Social worker successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in social worker list	Pass
TC2	User submits edit social worker form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit social worker form with missing staff ID field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

			- Display browser validation message: "Please fill out this field"	
TC4	User submits edit social worker form with missing agency name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits edit social worker form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC6	User submits edit social worker form with missing placement district field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC7	User edits social worker and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to social worker data - Form data cleared	- Modal closes without saving - No changes made to social worker data - Form data cleared	Pass
TC8	User uploads new avatar	- Avatar previews the selected avatar	- Avatar previews the selected avatar	Pass

TC9	User updates agency information (agency name, agency code)	- Agency information successfully updated - Success message displayed - Updated agency information reflected in social worker profile	- Agency information successfully updated - Success message displayed - Updated agency information reflected in social worker profile	Pass
TC10	User updates placement information (state, district)	Placement information successfully updated - Success message displayed - Updated placement information reflected in social worker profile	Placement information successfully updated - Success message displayed - Updated placement information reflected in social worker profile	Pass

Table 154: Edit Social Worker

5.1.11.4 Delete Social Worker

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms social worker deletion by clicking "Delete" button in modal	- Social worker user successfully deleted from database - Success message displayed - Social worker removed from social worker list - Modal closes automatically	- Social worker user successfully deleted from database - Success message displayed - Social worker removed from social worker list - Modal closes automatically	Pass

TC2	User cancels social worker deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list	Pass
TC3	User cancels social worker deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list	- Modal closes without deleting - No changes made to social worker data - Social worker remains in social worker list	Pass

Table 155: Delete Social Worker

5.1.12 Law Enforcement Management

5.1.12.1 View Law Enforcements

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses law enforcement management page	- Page loads successfully with law enforcement dashboard - Statistics cards display (Total Officers, Agencies Represented, Stations Covered, New Users)	- Page loads successfully with law enforcement dashboard - Statistics cards display (Total Officers, Agencies Represented, Stations Covered, New Users)	Pass

		- Data table shows law enforcement users list	- Data table shows law enforcement users list	
TC2	User views law enforcement data table with no law enforcement officers	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown - Statistics show zero values	Pass
TC3	User searches law enforcement by name in data table (e.g., search "John")	- Filtered results displayed - Only matching law enforcement officers shown	- Filtered results displayed - Only matching law enforcement officers shown	Pass
TC4	User without create permission views law enforcement management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass
TC5	User with create permission views law enforcement management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass
TC6	User without edit permission views law	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass

	enforcement management page			
TC7	User with edit permission views law enforcement management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views law enforcement management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC9	User with delete permission views law enforcement management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 156: View Law Enforcements

5.1.12.2 Create Law Enforcement

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits law enforcement creation form	- Law enforcement officer successfully created - Success message displayed	- Law enforcement officer successfully created - Success message displayed	Pass

	with all required fields filled correctly	- Modal closes automatically - New law enforcement officer appears in law enforcement list	- Modal closes automatically - New law enforcement officer appears in law enforcement list	
TC2	User submits law enforcement creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits law enforcement creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits law enforcement creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	Pass
TC5	User submits law enforcement creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass

TC6	User submits law enforcement creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	Pass
TC7	User submits law enforcement creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass
TC8	User submits law enforcement creation form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC9	User submits law enforcement creation form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC10	User submits law enforcement creation form	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

	with missing law enforcement state field		- Display browser validation message: "Please fill out this field"	
TC11	User submits law enforcement creation form with no agency selected (empty selection)	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC12	User submits law enforcement creation form with duplicate badge number for same agency	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"	Pass
TC13	User creates law enforcement and cancels form by clicking 'Close' button	- Modal closes without saving - No law enforcement officer created - Form data cleared	- Modal closes without saving - No law enforcement officer created - Form data cleared	Pass

Table 157: Create Law Enforcement

5.1.12.3 Edit Law Enforcement

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User submits edit law enforcement form with all required fields updated correctly	- Law enforcement officer successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in law enforcement list - Updated data reflected in social worker list	- Law enforcement officer successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in law enforcement list - Updated data reflected in social worker list	Pass
TC2	User submits edit law enforcement form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit law enforcement form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits edit law enforcement form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass

TC5	User submits edit law enforcement form with missing law enforcement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC6	User submits edit law enforcement form with no agency selected (empty selection)	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC7	User submits edit law enforcement form with duplicate badge number for same agency (different user)	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The badge number has already been taken"	Pass
TC8	User edits law enforcement and cancels form by clicking 'Close' button	- Modal closes without saving - No changes made to law enforcement data - Form data cleared	- Modal closes without saving - No changes made to law enforcement data - Form data cleared	Pass
TC9	User uploads new avatar	- Avatar previews the selected avatar	- Avatar previews the selected avatar	Pass
TC10	User updates agency information (agency,	- Agency information successfully updated	- Agency information successfully updated	Pass

	badge number, rank, station)	- Success message displayed - Updated agency information reflected in law enforcement profile	- Success message displayed - Updated agency information reflected in law enforcement profile	
TC11	User updates law enforcement state information	- Law enforcement state successfully updated - Success message displayed - Updated state information reflected in law enforcement profile	- Law enforcement state successfully updated - Success message displayed - Updated state information reflected in law enforcement profile	Pass
TC12	User updates contact information (phone, address, city, state, postcode)	- Contact information successfully updated - Success message displayed - Updated contact information reflected in user profile	- Contact information successfully updated - Success message displayed - Updated contact information reflected in user profile	Pass

Table 158: Edit Law Enforcement

5.1.12.4 Delete Law Enforcement

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User confirms law enforcement deletion by	- Law enforcement user successfully deleted from database - Success message displayed	- Law enforcement user successfully deleted from database - Success message displayed	Pass

	clicking "Delete" button in modal	- Law enforcement removed from law enforcement list - Modal closes automatically	- Law enforcement removed from law enforcement list - Modal closes automatically	
TC2	User cancels law enforcement deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to law enforcement data - Law enforcement remains in law enforcement list	- Modal closes without deleting - No changes made to law enforcement data - Law enforcement remains in law enforcement list	Pass
TC3	User cancels law enforcement deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to law enforcement data - Law enforcement remains in law enforcement list	- Modal closes without deleting - No changes made to law enforcement data - Law enforcement remains in law enforcement list	Pass

Table 159: Delete Law Enforcement

5.1.13 Child Welfare Officer Management

5.1.13.1 View Child Welfare Officers

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User accesses child welfare officer management page	- Page loads successfully with CWO dashboard - Statistics cards display (Total CWO Officers, Ministries Covered, States Covered, New This Month) - Data table shows CWO users list	- Page loads successfully with CWO dashboard - Statistics cards display (Total CWO Officers, Ministries Covered, States Covered, New This Month) - Data table shows CWO users list	Pass
TC2	User views CWO data table with no child welfare officers	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown - Statistics show zero values	Pass
TC3	User searches CWO by name in data table (e.g., search "John")	- Filtered results displayed - Only matching law enforcement officers shown	- Filtered results displayed - Only matching law enforcement officers shown	Pass
TC4	User without create permission views CWO management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass
TC5	User with create permission views CWO management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass

TC6	User without edit permission views CWO management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC7	User with edit permission views CWO management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views CWO management	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC9	User with delete permission views CWO management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 160: View CWOs

5.1.13.2 Create Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits CWO creation form with all required fields filled correctly	- Child welfare officer successfully created - Success message displayed - Modal closes automatically	- Child welfare officer successfully created - Success message displayed - Modal closes automatically	Pass

		- New CWO appears in CWO list	- New CWO appears in CWO list	
TC2	User submits CWO creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits CWO creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits CWO creation form with invalid email format (e.g. jon@)	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	- Browser prevents form submission - Display browser validation message: "Please enter a valid email address"	Pass
TC5	User submits CWO creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass
TC6	User submits CWO creation form with	- Browser prevents form submission	- Browser prevents form submission	Pass

	password less than 8 characters	- Display browser validation message: "Please lengthen this text to 8 characters or more"	- Display browser validation message: "Please lengthen this text to 8 characters or more"	
TC7	User submits CWO creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass
TC8	User submits CWO creation form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC9	User submits CWO creation form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC10	User submits CWO creation form with missing CWO state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass

TC11	User creates CWO and cancels form by clicking 'Close' button	- Modal closes without saving - No child welfare officer created - Form data cleared	- Modal closes without saving - No child welfare officer created - Form data cleared	Pass

Table 161: Create CWO

5.1.13.3 Edit Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit CWO form with all required fields updated correctly	- Child welfare officer successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in CWO list	- Child welfare officer successfully updated - Success message displayed - Modal closes automatically - Updated data reflected in CWO list	Pass
TC2	User submits edit CWO form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit CWO form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

			- Display browser validation message: "Please fill out this field"	
TC4	User submits edit CWO form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits edit CWO form with missing CWO state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC6	User edits CWO and cancels form by clicking 'Close' button	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC8	User uploads new avatar	- Avatar previews the selected avatar	- Avatar previews the selected avatar	Pass
TC9	User updates service scheme and grade information	- Service scheme and grade information successfully updated - Success message displayed - Updated information reflected in CWO profile	- Service scheme and grade information successfully updated - Success message displayed - Updated information reflected in CWO profile	Pass

TC10	User updates CWO state information	- CWO state successfully updated - Success message displayed - Updated state information reflected in CWO profile	- CWO state successfully updated - Success message displayed - Updated state information reflected in CWO profile	Pass

Table 162: Edit CWO

5.1.13.4 Delete Child Welfare Officer

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User confirms CWO deletion by clicking "Delete" button in modal	- Child welfare officer successfully deleted from database - Success message displayed - CWO removed from CWO list - Modal closes automatically	- Child welfare officer successfully deleted from database - Success message displayed - CWO removed from CWO list - Modal closes automatically	Pass
TC2	User cancels CWO deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to CWO data - Child welfare officer remains in CWO list	- Modal closes without deleting - No changes made to CWO data - Child welfare officer remains in CWO list	Pass
TC3	User cancels CWO deletion by	- Modal closes without deleting - No changes made to CWO data	- Modal closes without deleting - No changes made to CWO data	Pass

	clicking "Close" (X) button in modal	- Child welfare officer remains in CWO list	- Child welfare officer remains in CWO list	

Table 163: Delete CWO

5.1.14 Healthcare Professional Management

5.1.14.1 View Healthcare Professionals

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses healthcare professional management page	- Page loads successfully with healthcare dashboard - Statistics cards display (Total Professionals, Doctors, Nurses, Recently Added) - Data table shows healthcare users list	- Page loads successfully with healthcare dashboard - Statistics cards display (Total Professionals, Doctors, Nurses, Recently Added) - Data table shows healthcare users list	Pass
TC2	User views healthcare data table with no healthcare professionals	- Empty data table displayed - "No data available" message shown - Statistics show zero values	- Empty data table displayed - "No data available" message shown - Statistics show zero values	Pass
TC3	User searches healthcare professional by name in	- Filtered results displayed	- Filtered results displayed	Pass

	data table (e.g., search "John")	- Only matching law enforcement officers shown	- Only matching law enforcement officers shown	
TC4	User without create permission views healthcare management page	- Page loads successfully - Add button not visible	- Page loads successfully - Add button not visible	Pass
TC5	User with create permission views healthcare management page	- Page loads successfully - Add button visible	- Page loads successfully - Add button visible	Pass
TC6	User without edit permission views healthcare management page	- Page loads successfully - Edit buttons not visible in action dropdown	- Page loads successfully - Edit buttons not visible in action dropdown	Pass
TC7	User with edit permission views healthcare management page	- Page loads successfully - Edit buttons visible in action dropdown	- Page loads successfully - Edit buttons visible in action dropdown	Pass
TC8	User without delete permission views healthcare management page	- Page loads successfully - Delete buttons not visible in action dropdown	- Page loads successfully - Delete buttons not visible in action dropdown	Pass
TC9	User with delete permission views healthcare management page	- Page loads successfully - Delete buttons visible in action dropdown	- Page loads successfully - Delete buttons visible in action dropdown	Pass

Table 164: View Healthcare Professionals

5.1.14.2 Create Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits healthcare creation form with all required fields filled correctly	- Healthcare professional successfully created - Success message displayed - Modal closes automatically - New healthcare professional appears in healthcare list	- Healthcare professional successfully created - Success message displayed - Modal closes automatically - New healthcare professional appears in healthcare list	Pass
TC2	User submits healthcare creation form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits healthcare creation form with missing email field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits healthcare creation form with invalid email format (e.g. jon@)	- Browser prevents form submission	- Browser prevents form submission	Pass

		- Display browser validation message: "Please enter a valid email address"	- Display browser validation message: "Please enter a valid email address"	
TC5	User submits healthcare creation form with email that already exists in database	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	- Form submission proceeds but server validation catches this - Display server validation error: "The email has already been taken"	Pass
TC6	User submits healthcare creation form with password less than 8 characters	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	- Browser prevents form submission - Display browser validation message: "Please lengthen this text to 8 characters or more"	Pass
TC7	User submits healthcare creation form with password confirmation that doesn't match password	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	- Form submission proceeds but server validation catches this - Display server validation error: "The password confirmation does not match"	Pass
TC8	User submits healthcare creation form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please select an item in the list"	- Browser prevents form submission	Pass

			- Display browser validation message: "Please select an item in the list"	
TC9	User submits healthcare creation form with missing facility name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC10	User submits healthcare creation form with missing state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC11	User creates CWO and cancels form by clicking 'Close' button	- Modal closes without saving - No child welfare officer created - Form data cleared	- Modal closes without saving - No child welfare officer created - Form data cleared	Pass

Table 165: Create Healthcare Professional

5.1.14.3 Edit Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits edit healthcare form with all	- Healthcare professional successfully updated	- Healthcare professional successfully updated	Pass

	required fields updated correctly	- Success message displayed - Modal closes automatically - Updated data reflected in healthcare list	- Success message displayed - Modal closes automatically - Updated data reflected in healthcare list	
TC2	User submits edit healthcare form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC3	User submits edit healthcare form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits edit healthcare form with missing facility name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits edit healthcare form with missing state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass

TC6	User edits healthcare professional and cancels form by clicking 'Close' button	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC8	User uploads new avatar	- Avatar previews the selected avatar	- Avatar previews the selected avatar	Pass
TC9	User updates facility information (facility name, state)	- Facility information successfully updated - Success message displayed - Updated facility information reflected in healthcare profile	- Facility information successfully updated - Success message displayed - Updated facility information reflected in healthcare profile	Pass
TC10	User updates APC expiry date	- APC expiry date successfully updated - Success message displayed - Updated APC expiry reflected in healthcare profile	- APC expiry date successfully updated - Success message displayed - Updated APC expiry reflected in healthcare profile	Pass

Table 166: Edit Healthcare Professional

5.1.14.4 Delete Healthcare Professional

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User confirms healthcare deletion by clicking "Delete" button in modal	- Healthcare professional successfully deleted from database - Success message displayed - Healthcare professional removed from healthcare list - Modal closes automatically	- Healthcare professional successfully deleted from database - Success message displayed - Healthcare professional removed from healthcare list - Modal closes automatically	Pass
TC2	User cancels healthcare deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list	Pass
TC3	User cancels healthcare deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list	- Modal closes without deleting - No changes made to healthcare data - Healthcare professional remains in healthcare list	Pass

Table 167: Delete Healthcare Professional

5.1.15 Role and Permission Management

5.1.15.1 View Role Management

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User access' role management page	- Page loads successfully with role dashboard - Statistics cards display (Total Roles, Total Permissions) - Data table shows roles list (excluding admin role)	- Page loads successfully with role dashboard - Statistics cards display (Total Roles, Total Permissions) - Data table shows roles list (excluding admin role)	Pass
TC2	User clicks on a role name to view details	- Role details page opens successfully - Shows role information, assigned users, and permissions	- Role details page opens successfully - Shows role information, assigned users, and permissions	Pass
TC3	User clicks on actions dropdown for a role	- Dropdown menu appears with options: View, Edit, Permissions, Delete	- Dropdown menu appears with options: View, Edit, Permissions, Delete	Pass
TC4	User clicks "View" option from actions dropdown	- Role details page opens successfully - Shows role information, assigned users, and permissions	- Role details page opens successfully - Shows role information, assigned users, and permissions	Pass
TC5	User without create permission views role management page	- Page loads successfully - Add Role button not visible	- Page loads successfully - Add Role button not visible	Pass

TC6	User with create permission views role management page	- Page loads successfully - Add Role button visible	- Page loads successfully - Add Role button visible	Pass
TC7	User without edit permission views role management page	- Page loads successfully - Edit option not visible in actions dropdown	- Page loads successfully - Edit option not visible in actions dropdown	Pass
TC8	User with edit permission views role management page	- Page loads successfully - Edit option visible in actions dropdown	- Page loads successfully - Edit option visible in actions dropdown	Pass

Table 168: View Role Management

5.1.15.2 Create Role

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User submits role creation form with all required fields filled correctly	- Role successfully created - Success message displayed - Modal closes automatically - New role appears in roles list	- Role successfully created - Success message displayed - Modal closes automatically - New role appears in roles list	Pass
TC2	User submits role creation form with missing role name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

			- Display browser validation message: "Please fill out this field"	
TC3	User submits role creation form with role name less than 2 characters	- Form submission proceeds but server validation catches this - Display server validation error: "Role name must be at least 2 characters long"	- Form submission proceeds but server validation catches this - Display server validation error: "Role name must be at least 2 characters long"	Pass
TC4	User submits role creation form with role name containing invalid characters (e.g., "Role@123")	- Form submission proceeds but server validation catches this - Display server validation error: "Role name can only contain lowercase letters, numbers, and underscores"	- Form submission proceeds but server validation catches this - Display server validation error: "Role name can only contain lowercase letters, numbers, and underscores"	Pass
TC5	User submits role creation form with role name that already exists	- Form submission proceeds but server validation catches this - Display server validation error: "This role name already exists"	- Form submission proceeds but server validation catches this - Display server validation error: "This role name already exists"	Pass
TC6	User creates role and cancels form by clicking 'Close' button	- Modal closes without saving - No role created - Form data cleared	- Modal closes without saving - No role created - Form data cleared	Pass

Table 169: Create Role

5.1.15.3 Delete Role

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Delete" button for a role with no assigned users	- Delete confirmation modal opens successfully - Role name displayed in confirmation message - "Are you sure you want to delete this role?" message shown	- Delete confirmation modal opens successfully - Role name displayed in confirmation message - "Are you sure you want to delete this role?" message shown	Pass
TC2	User confirms role deletion by clicking "Delete" button in modal	- Role successfully deleted from database - Success message displayed - Role removed from roles list - Modal closes automatically	- Role successfully deleted from database - Success message displayed - Role removed from roles list - Modal closes automatically	Pass
TC3	User cancels role deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to role data - Role remains in roles list	- Modal closes without deleting - No changes made to role data - Role remains in roles list	Pass

TC4	User cancels role deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to role data - Role remains in roles list	- Modal closes without deleting - No changes made to role data - Role remains in roles list	Pass

5.1.15.4 Assign Permission

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Assign Permissions" button for a role	- Permission assignment page opens successfully - Shows all available permissions grouped by module - Current role permissions are pre-selected	- Permission assignment page opens successfully - Shows all available permissions grouped by module - Current role permissions are pre-selected	Pass
TC2	User assigns permissions to a role and submits form	- Permissions successfully assigned to role - Success message displayed - Redirected back to role management page	- Permissions successfully assigned to role - Success message displayed - Redirected back to role management page	Pass
TC3	User cancels permission assignment by clicking "Cancel" button	- Redirected back to roles index	- Redirected back to roles index	Pass

		- No changes made to role permissions - Original permissions remain unchanged	- No changes made to role permissions - Original permissions remain unchanged	
TC4	User selects/deselects individual permissions	- Selected permissions count updates dynamically - Individual permission checkboxes reflect selection state	- Selected permissions count updates dynamically - Individual permission checkboxes reflect selection state	Pass
TC5	User selects "Select All" for a module	- All permissions in that module are selected - Module's "Select All" checkbox shows checked state	- All permissions in that module are selected - Module's "Select All" checkbox shows checked state	Pass
TC6	User deselects "Select All" for a module	- All permissions in that module are deselected - Module's "Select All" checkbox shows unchecked state	- All permissions in that module are deselected - Module's "Select All" checkbox shows unchecked state	Pass

Table 170: Assign Permission

5.1.16 Contact Queries Management

5.1.16.1 View Contact Queries

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses contact queries management page	- Page loads successfully with contact queries dashboard - Statistics cards display (Total Queries, Pending, In Progress, Resolved) - Data table shows contact queries list - Filters section visible with status, date filters		Pass
TC2	User views contact queries data table with no queries	- Empty data table displayed - "No contact queries found" message shown - Statistics show zero values		Pass
TC3	User searches contact query by name/email/subject in data table	- Filtered results displayed - Only matching contact queries shown		Pass

TC4	User applies status filter (Pending, In Progress, Resolved)	- Data table filtered by selected status - Only queries with matching status displayed		Pass
TC5	User applies date range filter	- Data table filtered by date range - Only queries within date range displayed		Pass
TC6	User clicks "Clear" button to reset filters	- All filters reset to default values - Data table shows all contact queries		Pass
TC7	User clicks on actions dropdown for a contact query	- Dropdown menu appears with options: View, Edit, Delete		Pass
TC8	User clicks "View" option from actions dropdown	- Contact query details page opens successfully - Shows full query information including message content		Pass
TC9	User without view permission accesses contact queries page	- Access denied or limited view based on permissions		Pass
TC10	User with view permission views contact queries page	- Page loads successfully with all viewable elements		Pass

Table 171: View Contact Queries

5.1.16.2 View Contact Query Details

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User views contact query details page	- Contact query details displayed - Shows name, email, subject, message, status, creation date	- Contact query details displayed - Shows name, email, subject, message, status, creation date	Pass
TC2	User without edit permission views contact query details	- Contact query details displayed - Status change buttons not visible	- Contact query details displayed - Status change buttons not visible	Pass
TC3	User with edit permission views contact query details	- Contact query details displayed - Status change buttons visible (Mark as Pending, In Progress, Resolved)	- Contact query details displayed - Status change buttons visible (Mark as Pending, In Progress, Resolved)	Pass
TC4	User views contact query from anonymous user	- Contact query details displayed - User information section shows "Anonymous user"	- Contact query details displayed - User information section shows "Anonymous user"	Pass
TC5	User views contact query from registered user	- Contact query details displayed - User information shows username, email, role, member since date	- Contact query details displayed - User information shows username, email, role, member since date	Pass

Table 172: View Contact Query Details

5.1.16.3 Edit Contact Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Mark as Pending" status button	- Status successfully updated to "Pending" - Success message displayed - Page refreshes with updated status	- Status successfully updated to "Pending" - Success message displayed - Page refreshes with updated status	Pass
TC2	User clicks "Mark as In Progress" status button	- Status successfully updated to "In Progress" - Success message displayed - Page refreshes with updated status	- Status successfully updated to "In Progress" - Success message displayed - Page refreshes with updated status	Pass
TC3	User clicks "Mark as Resolved" status button	- Status successfully updated to "Resolved" - Success message displayed - Page refreshes with updated status	- Status successfully updated to "Resolved" - Success message displayed - Page refreshes with updated status	Pass
TC4	User without edit permission tries to change status	- Status change buttons not visible - No status change functionality available	- Status change buttons not visible - No status change functionality available	Pass

TC5	User with edit permission views contact query details	- Status change buttons visible - Can change status using the buttons	- Status change buttons visible - Can change status using the buttons	Pass

Table 173: Edit Contact Query

5.1.16.4 Delete Contact Query

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Delete" button for a contact query with valid permissions	- Delete confirmation modal opens successfully - Contact query subject displayed in confirmation message - "Are you sure you want to delete this contact query?" message shown	- Delete confirmation modal opens successfully - Contact query subject displayed in confirmation message - "Are you sure you want to delete this contact query?" message shown	Pass
TC2	User confirms contact query deletion by clicking "Delete" button in modal	- Contact query successfully deleted from database - Success message displayed - Contact query removed from contact queries list - Modal closes automatically	- Contact query successfully deleted from database - Success message displayed - Contact query removed from contact queries list - Modal closes automatically	Pass

TC3	User cancels contact query deletion by clicking "Cancel" button in modal	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list	Pass
TC4	User cancels contact query deletion by clicking "Close" (X) button in modal	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list	- Modal closes without deleting - No changes made to contact query data - Contact query remains in contact queries list	Pass

Table 174: Delete Contact Query

5.1.17 View Session Logs

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses session logs page	- Page loads successfully with "Session Logs" title - Data table with columns: User, Status, Device Info, Location, Last Activity, Duration - Filters button visible	- Page loads successfully with "Session Logs" title - Data table with columns: User, Status, Device Info, Location, Last Activity, Duration - Filters button visible	Pass

TC2	User views session logs data table with no logs	- Empty data table displayed - "No activity logs found" message shown	- Empty data table displayed - "No activity logs found" message shown	Pass
TC3	User clicks "Filters" button	- Filter section expands - Shows user filter dropdown, date from/to inputs - Apply Filters and Clear Filters buttons visible	- Filter section expands - Shows user filter dropdown, date from/to inputs - Apply Filters and Clear Filters buttons visible	Pass
TC4	User applies user filter	- Data table filtered by selected user - Only logs for selected user displayed	- Data table filtered by selected user - Only logs for selected user displayed	Pass
TC5	User applies date range filter	- Data table filtered by date range - Only logs within date range displayed	- Data table filtered by date range - Only logs within date range displayed	Pass
TC6	User clicks "Apply Filters" button	- Filters applied to data table - Table reloads with filtered results	- Filters applied to data table - Table reloads with filtered results	Pass
TC7	User clicks "Clear Filters" button	- All filters reset to default values - Data table shows all session logs	- All filters reset to default values - Data table shows all session logs	Pass
TC8	User without view permission accesses session logs page	- Access denied or limited view based on permissions	- Access denied or limited view based on permissions	Pass

TC9	User with view permission views session logs page	- Page loads successfully with all viewable elements	- Page loads successfully with all viewable elements	Pass

Table 175: View Session Log

5.1.18 Profiles

5.1.18.1 Edit Profile (System Administrator)

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User accesses admin profile edit page	- Page loads successfully with "Edit Admin Profile" title - Form pre-populated with current admin data - Avatar upload section visible	- Page loads successfully with "Edit Admin Profile" title - Form pre-populated with current admin data - Avatar upload section visible	Pass
TC2	User submits admin profile form with all required fields updated correctly	- Admin profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	- Admin profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	Pass
TC3	User submits admin profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

			- Display browser validation message: "Please fill out this field"	
TC4	User submits admin profile form with missing display name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits admin profile form with missing department field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC6	User submits admin profile form with missing position field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC7	User uploads new avatar image	- Avatar preview shows selected image	- Avatar preview shows selected image	Pass
TC8	User cancels profile edit by clicking "Cancel" button	- Redirected back to dashboard - No changes saved to profile	- Redirected back to dashboard - No changes saved to profile	Pass

TC9	User clicks "Delete My Account" button	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible	Pass

Table 176: Edit Profile (System Administrator)

5.1.18.2 Edit Profile (Law Enforcement)

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User accesses law enforcement profile edit page	- Page loads successfully with "Edit Law Enforcement Profile" title - Form pre-populated with current law enforcement data - Avatar upload section visible	- Page loads successfully with "Edit Law Enforcement Profile" title - Form pre-populated with current law enforcement data - Avatar upload section visible	Pass
TC2	User submits law enforcement profile form with all required fields updated correctly	- Law enforcement profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	- Law enforcement profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	Pass

TC3	User submits law enforcement profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits law enforcement profile form with missing agency field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits law enforcement profile form with missing badge number field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC6	User submits law enforcement profile form with missing placement state field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC7	User uploads new avatar image	- Avatar preview shows selected image	- Avatar preview shows selected image	Pass

TC8	User cancels profile edit by clicking "Cancel" button	- Redirected back to dashboard - No changes saved to profile	- Redirected back to dashboard - No changes saved to profile	Pass
TC9	User clicks "Delete My Account" button	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible	- Delete confirmation modal opens - Warning message displayed - "Yes, Delete My Account" and "Cancel" buttons visible	Pass

Table 177: Edit Profile (Law Enforcement)

5.1.18.3 Edit Profile (Social Worker)

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**
TC1	User accesses social worker profile edit page	- Page loads successfully with "Edit Social Worker Profile" title - Form pre-populated with current social worker data - Avatar upload section visible	- Page loads successfully with "Edit Social Worker Profile" title - Form pre-populated with current social worker data - Avatar upload section visible	Pass
TC2	User submits social worker profile form with all required fields updated correctly	- Social worker profile successfully updated - Success message displayed - Redirected back to dashboard	- Social worker profile successfully updated - Success message displayed - Redirected back to dashboard	Pass

		- Updated data reflected in profile	- Updated data reflected in profile	
TC3	User submits social worker profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC4	User submits social worker profile form with missing staff ID field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC5	User submits social worker profile form with missing agency name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC6	User selects “Other” from agency dropdown	- Additional text input field appears - “Other” input field becomes required	- Additional text input field appears - “Other” input field becomes required	Pass
TC7	User selects predefined agency from dropdown	- Agency code field auto-populates - Agency code field becomes readonly	- Agency code field auto-populates - Agency code field becomes readonly	Pass
TC8	User submits social worker	- Browser prevents form submission	- Browser prevents form submission	Pass

	profile form with missing placement state field	- Display browser validation message: “Please fill out this field”	- Display browser validation message: “Please fill out this field”	
TC9	User submits social worker profile form with missing placement district field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC10	User uploads new avatar image	- Avatar preview shows selected image	- Avatar preview shows selected image	Pass
TC11	User cancels profile edit by clicking “Cancel” button	- Redirected back to dashboard - No changes saved to profile	- Redirected back to dashboard - No changes saved to profile	Pass
TC12	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	Pass

Table 178: Edit Profile (Social Worker)

5.1.18.4 Edit Profile (Child Welfare Officer)

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**

TC1	User accesses CWO profile edit page	- Page loads successfully with “Edit CWO Profile” title - Form pre-populated with current CWO data - Avatar upload section visible	- Page loads successfully with “Edit CWO Profile” title - Form pre-populated with current CWO data - Avatar upload section visible	Pass
TC2	User submits CWO profile form with all required fields updated correctly	- CWO profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	- CWO profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	Pass
TC3	User submits CWO profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC4	User submits CWO profile form with missing ministry field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits CWO profile form with missing department field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission	Pass

			- Display browser validation message: “Please fill out this field”	
TC6	User submits CWO profile form with missing state field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC7	User uploads new avatar image	- Avatar preview shows selected image	- Avatar preview shows selected image	Pass
TC8	User cancels profile edit by clicking “Cancel” button	- Redirected back to dashboard - No changes saved to profile	- Redirected back to dashboard - No changes saved to profile	Pass
TC9	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	Pass

Table 179: Edit Profile (Child Welfare Officer)

5.1.18.5 Edit Profile (Healthcare Professional)

TC	**Test Condition**	**Expected Message/Output**	**Actual message/output**	**Result**

TC1	User accesses healthcare profile edit page	- Page loads successfully with "Edit Healthcare Profile" title - Form pre-populated with current healthcare data - Avatar upload section visible	- Page loads successfully with "Edit Healthcare Profile" title - Form pre-populated with current healthcare data - Avatar upload section visible	Pass
TC2	User submits healthcare profile form with all required fields updated correctly	- Healthcare profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	- Healthcare profile successfully updated - Success message displayed - Redirected back to dashboard - Updated data reflected in profile	Pass
TC3	User submits healthcare profile form with missing name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC4	User submits healthcare profile form with missing profession field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	Pass
TC5	User submits healthcare profile form with missing facility name field	- Browser prevents form submission - Display browser validation message: "Please fill out this field"	- Browser prevents form submission	Pass

			- Display browser validation message: “Please fill out this field”	
TC6	User submits healthcare profile form with missing facility state field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC7	User uploads new avatar image	- Avatar preview shows selected image	- Avatar preview shows selected image	Pass
TC8	User cancels profile edit by clicking “Cancel” button	- Redirected back to dashboard - No changes saved to profile	- Redirected back to dashboard - No changes saved to profile	Pass
TC9	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	- Delete confirmation modal opens - Warning message displayed - “Yes, Delete My Account” and “Cancel” buttons visible	Pass

Table 180: Edit Profile (Healthcare Professional)

5.1.18.6 Edit Profile (Social Worker)

TC	Test Condition	Expected Message/Output	Actual message/output	Result

TC1	User accesses public user profile edit page	- Page loads successfully with "Edit Profile" title - Form pre-populated with current public user data - Email field shows as readonly with note "Private — cannot be changed"	- Page loads successfully with "Edit Profile" title - Form pre-populated with current public user data - Email field shows as readonly with note "Private — cannot be changed"	Pass
TC2	User submits public user profile form with all required fields updated correctly	- Public user profile successfully updated - Success message displayed - Redirected back to landing page - Updated data reflected in profile	- Public user profile successfully updated - Success message displayed - Redirected back to landing page - Updated data reflected in profile	Pass
TC3	User submits public user profile form with missing name field	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	- Browser prevents form submission - Display browser validation message: “Please fill out this field”	Pass
TC4	User cancels profile edit by clicking "Cancel" button	- Redirected back to landing page - No changes saved to profile	- Redirected back to landing page - No changes saved to profile	Pass
TC5	User clicks “Delete My Account” button	- Delete confirmation modal opens - Warning message displayed	- Delete confirmation modal opens - Warning message displayed	Pass

		- “Yes, Delete My Account” and “Cancel” buttons visible	- “Yes, Delete My Account” and “Cancel” buttons visible	

Table 181: Edit Profile (Social Worker)

5.1.19 Delete My Account – All User Type

TC	Test Condition	Expected Message/Output	Actual message/output	Result
TC1	User clicks "Delete My Account" button (any user type)	- Delete confirmation modal opens - Warning message: "Are you absolutely sure you want to delete your account? This action cannot be undone and all your data will be permanently removed." - "Yes, Delete My Account" and "Cancel" buttons visible	- Delete confirmation modal opens - Warning message: "Are you absolutely sure you want to delete your account? This action cannot be undone and all your data will be permanently removed." - "Yes, Delete My Account" and "Cancel" buttons visible	Pass
TC2	User confirms account deletion by clicking "Yes, Delete My Account" button	- Account successfully deleted from database - Success message displayed - User logged out and redirected to appropriate page	- Account successfully deleted from database - Success message displayed - User logged out and redirected to appropriate page	Pass

		- All user data permanently removed	- All user data permanently removed	
TC3	User cancels account deletion by clicking "Cancel" button	- Modal closes without deleting - No changes made to account - User remains logged in	- Modal closes without deleting - No changes made to account - User remains logged in	Pass
TC4	User cancels account deletion by clicking "Close" (X) button	- Modal closes without deleting - No changes made to account - User remains logged in	- Modal closes without deleting - No changes made to account - User remains logged in	Pass

Table 182: Delete My Account - All User Type

5.2 User Acceptance Testing

5.2.1 UAT 1

Name of the Tester: Ahmad Faris Danial Bin Ahmad Razlan

Role: Public User

Date: 10 August 2025

Time Start: 10:43 am

Time End: 1:50 pm

Name of parameter/Testing Category	Excellent (5) points	Very Good (4) points	Fair (2-3) points	Poor (1-0) points
User Registration & Authentication (Public user registration, role-based login, password reset, session management)		✓		
Anonymous Child Abuse Reporting (Submit reports without revealing identity, incident details, evidence upload, truth confirmation)	✓			
Case Management & Assignment (Case creation, assignment to social workers/law enforcement/healthcare, status updates, priority levels)			✓	
Role-Based Dashboard Access (Admin, Government Official, Social Worker, Law Enforcement, Healthcare, Public User dashboards)	✓			

Interagency Case Messaging (Secure communication between assigned case workers, real-time messaging, message history)			✓	
Case History & Audit Trail (Comprehensive case tracking, status changes, activity logs, accountability)			✓	
Contact Query Management (Public inquiries, response tracking, query resolution, admin management)		✓		
Reports & Analytics (Case statistics, downloadable reports, monthly trends, system analytics, CSV/PDF exports)	✓			
Activity Logs & Security (Login/logout tracking, failed login monitoring, IP address logging, device information)		✓		
User Profile Management (Role-specific profile editing, avatar support, account management, account deletion)	✓			
System Performance & Responsiveness (Page load speed, real-time updates, error handling, database optimization)	✓			

User Interface & Navigation (Ease of use, responsive design, accessibility, clear layout, modern UI)	✓			

Feedback from the Tester: The reporting felt straightforward and easy to use. I also can see the status of the report if I'm sign in as well which is nice. As for the admin side, I don't know much about them, but I played with them, and I think it did its job perfectly.	**Response from the Developer:** Thank you for the positive feedback Mr.Faris. I'm glad to hear the reporting process and status tracking worked smoothly for you

Table 183: UAT 1

5.2.2 UAT 2

Name of the Tester: Ashraf Shahidan Bin Sahizan
Role: Public User
Date: 10 August 2025
Time Start: 10:43 am
Time End: 7:31 pm

Name of parameter/Testing Category	Excellent (5) points	Very Good (4) points	Fair (2-3) points	Poor (1-0) points
User Registration & Authentication (Public user registration, role-based login, password reset, session management)		✓		
Anonymous Child Abuse Reporting (Submit reports without revealing identity, incident details, evidence upload, truth confirmation)	✓			

Case Management & Assignment (Case creation, assignment to social workers/law enforcement/healthcare, status updates, priority levels)			✓	
Role-Based Dashboard Access (Admin, Government Official, Social Worker, Law Enforcement, Healthcare, Public User dashboards)		✓		
Interagency Case Messaging (Secure communication between assigned case workers, real-time messaging, message history)				✓
Case History & Audit Trail (Comprehensive case tracking, status changes, activity logs, accountability)			✓	
Contact Query Management (Public inquiries, response tracking, query resolution, admin management)			✓	
Reports & Analytics (Case statistics, downloadable reports, monthly trends, system analytics, CSV/PDF exports)				✓
Activity Logs & Security (Login/logout tracking, failed login monitoring, IP				✓

address logging, device information)				
User Profile Management (Role-specific profile editing, avatar support, account management, account deletion)		✓		
System Performance & Responsiveness (Page load speed, real-time updates, error handling, database optimization)	✓			
User Interface & Navigation (Ease of use, responsive design, accessibility, clear layout, modern UI)	✓			

Feedback from the Tester:	**Response from the Developer:**
The anonymous reporting form was simple and easy to fill in, and I liked that I could upload evidence quickly without needing to log in. After submitting, the “Track My Report” option gave me peace of mind since I could check the progress later when I signed in. Overall, I feel the system made it less intimidating to report something serious.	Thank you Mr.Shahidan for the encouraging feedback. I’m pleased to know the anonymous reporting and tracking features made the process easier for you, and I will continue refining the interface to keep it user-friendly and reassuring.

Table 184: UAT 2

5.2.3 UAT 3

Name of the Tester: Nur Aisyah Binti Rahman **Role: Government Child Welfare Officer** **Date: 15 August 2025** **Time Start: 9:10 am**

Time End: 2:27 pm				
Name of parameter/Testing Category	**Excellent (5) points**	**Very Good (4) points**	**Fair (2-3) points**	**Poor (1-0) points**
User Registration & Authentication (Public user registration, role-based login, password reset, session management)		✓		
Anonymous Child Abuse Reporting (Submit reports without revealing identity, incident details, evidence upload, truth confirmation)	✓			
Case Management & Assignment (Case creation, assignment to social workers/law enforcement/healthcare, status updates, priority levels)	✓			
Role-Based Dashboard Access (Admin, Government Official, Social Worker, Law Enforcement, Healthcare, Public User dashboards)	✓			
Interagency Case Messaging (Secure communication between assigned case workers, real-time messaging, message history)		✓		
Case History & Audit Trail	✓			

(Comprehensive case tracking, status changes, activity logs, accountability)				
Contact Query Management (Public inquiries, response tracking, query resolution, admin management)		✓		
Reports & Analytics (Case statistics, downloadable reports, monthly trends, system analytics, CSV/PDF exports)	✓			
Activity Logs & Security (Login/logout tracking, failed login monitoring, IP address logging, device information)	✓			
User Profile Management (Role-specific profile editing, avatar support, account management, account deletion)		✓		
System Performance & Responsiveness (Page load speed, real-time updates, error handling, database optimization)		✓		
User Interface & Navigation (Ease of use, responsive design, accessibility, clear layout, modern UI)	✓			

Feedback from the Tester:	**Response from the Developer:**
Thank you for letting me try the system. I like that the dashboard gave me a clear overview of all incoming and assigned cases, and the filters made it easy to focus on high-priority ones. Case assignment to social workers worked smoothly, and I could also review the timeline of updates in the audit trail. Messaging was functional within each case, but it would be more effective with a notification alert when a new message arrives. Reports were especially helpful during our interagency meeting since I could generate summaries quickly. Overall, a good job on the project.	Thank you, Ms. Aisyah, for the constructive feedback. I am glad that the system works properly.

Table 185: UAT 3

5.2.4 UAT 4

Name of the Tester: Ahmad Faiz bin Zulkifli **Role: Government Law Enforcement** **Date: 24 August 2025** **Time Start: 10:10 am** **Time End: 11:32 pm**				
Name of parameter/Testing Category	**Excellent (5) points**	**Very Good (4) points**	**Fair (2-3) points**	**Poor (1-0) points**
User Registration & Authentication **(Public user registration, role-based login, password reset, session management)**			✓	
Anonymous Child Abuse Reporting **(Submit reports without revealing identity, incident**	✓			

details, evidence upload, truth confirmation)				
Case Management & Assignment (Case creation, assignment to social workers/law enforcement/healthcare, status updates, priority levels)	✓			
Role-Based Dashboard Access (Admin, Government Official, Social Worker, Law Enforcement, Healthcare, Public User dashboards)		✓		
Interagency Case Messaging (Secure communication between assigned case workers, real-time messaging, message history)		✓		
Case History & Audit Trail (Comprehensive case tracking, status changes, activity logs, accountability)		✓		
Contact Query Management (Public inquiries, response tracking, query resolution, admin management)		✓		
Reports & Analytics (Case statistics, downloadable reports, monthly trends, system analytics, CSV/PDF exports)		✓		

Activity Logs & Security (Login/logout tracking, failed login monitoring, IP address logging, device information)	✓			
User Profile Management (Role-specific profile editing, avatar support, account management, account deletion)		✓		
System Performance & Responsiveness (Page load speed, real-time updates, error handling, database optimization)	✓			
User Interface & Navigation (Ease of use, responsive design, accessibility, clear layout, modern UI)		✓		
Feedback from the Tester: Accessing active cases was smooth, and the information displayed was detailed enough to support investigations. The history log was valuable for reviewing actions taken and understanding how the case had progressed. Generating reports worked well and will be useful for preparing summaries during operations. Messaging between agencies functioned correctly as well. That's all from me. Thank you.	**Response from the Developer:** Thank you, sir, for your observation, I'm glad some functionalities work properly and may help to support child abuse case properly.			

Table 186: UAT 4

5.3 Summary

Hence, in this chapter the system was tested and validated to confirm its overall dependability, performance, and functioning. Core features like, case reporting, case management, and user authentication were the first to undergo unit testing. By comparing the expected and actual results of each test case, problems might be found and fixed quickly. The findings demonstrated that most test cases passed, indicating that the workflow and system logic operated as planned.

In addition, User Acceptance Testing (UAT) was carried out alongside unit testing, with a chosen group of volunteers representing various system roles such as public users, a law enforcement and a government child welfare officer. Testers assessed the system's usability, goal attainment, security, and error-free operation. The comments emphasized that the site was easy to use, achieved its goals, and complied with security standards for managing private case data.

All things considered, the testing and discussion stage showed that the system is secure, reliable, and in line with the project's goals. The built platform's ability to facilitate interagency collaboration, structured case management, and anonymous reporting was validated by both unit testing and UAT. The outcomes gave assurance that the system is prepared for deployment and upcoming improvements by validating the technical implementation and design choices discussed in previous chapters.

Chapter 6: Conclusion

6.1 Critical Evaluation

The Integrated Child Welfare System (SinDa) has successfully achieved the primary objectives by creating a secure and centralized platform for reporting and managing child abuse cases in Malaysia. To address issues such as fear of retaliation, poor coordination, and lack of structured reporting, the system integrates key features including anonymous reporting, role-based access control, secure case tracking, and interagency communication. The system was developed with a structured methodology and validated through systematic testing, ensuring that its design remains both practical and adaptable to the sensitive context of child protection work.

By addressing significant issues in child welfare, including fragmented case handling, dangerous options for witnesses and victims, and reporting delays, this project significantly benefits the community. While professionals like welfare officials, healthcare providers, law enforcements benefit from coordinated processes that increase case visibility and accountability, the public gains a safe and easy platform to report cases anonymously and securely. Beyond its direct use, the system shows how digital platforms may modernize child protection services and is very compatible with SDG 16: Peace, Justice, and Strong Institutions because it encourages accountability, transparency, and more robust institutional responses.

Regarding the project's qualities, several benefits become apparent when contrasting it with the comparable systems that were previously addressed. First off, this system offers end-to-end case management, guaranteeing that every report is tracked from submission through resolution. In contrast, most current reporting platforms or hotlines simply permit basic case submission without structured follow-up. Second, this approach prioritizes anonymity and role-based access, which gives reporters confidence while shielding critical victim data from unauthorized access, in contrast to many general reporting platforms that lack robust privacy controls. Thirdly, there aren't many technologies that offer safe interagency communication; instead, they frequently rely on disorganized external channels like phone or email. On the other hand, this system incorporates a secure communication module that enables direct collaboration between various specialists within the platform. These advantages demonstrate that the system not only achieves its goals but also provides significant enhancements over current approaches, guaranteeing its capacity to have a long-lasting effect on bolstering child welfare services.

6.2 Limitation

Despite achieving its goals, the system has a number of limitations that need to be recognized. Time constraints and the academic aim of the research prevented full-scale implementation in cooperation with government agencies, limiting development to a prototype stage. Furthermore, real child abuse case data could not be accessible due to ethical and confidentiality issues. Instead, testing was done using simulated data, which made it impossible to assess the system in intricate real-world scenarios when sensitive data is handled by several authorities.

Besides that, although the system has included the basic safety measures, for example authentication and encryption, it has not been tested with advanced auditing such as penetration testing. This implies that the security level at present is satisfactory enough for demonstration purposes but has not yet reached the state of being adequate for deployment in a real-world environment where the protection of sensitive victim data is more firmly required. A further shortcoming is the lack of the integration of AI. The latter functionalities such as case prioritization, predictive analytics, or sentiment analysis of secure communication were not implemented due to the limited resources and time available. Hence, the system is not capable of providing even the basic proactive insights or the advanced decision-making support that it is programmed to do. This has, in turn, impacted its possible role in the professionals' efficient management of high-risk cases.

6.3 Recommendation for Future Enhancement

To strengthen the system and prepare it for real-world deployment, several enhancements are recommended. Firstly, implementing a notification system would significantly improve usability by sending real-time alerts to users for new case assignments, progress updates, or pending tasks. This would ensure that critical information reaches professionals without delay, reducing the risk of missed or late responses. In addition, an appointment scheduling module should be introduced to allow welfare officers, healthcare professionals, law enforcement, and victims or witnesses to coordinate meetings directly within the system. This would minimize dependence on external channels such as phone calls or emails and ensure that collaboration is both streamlined and documented.

Another important recommendation is to enhance system security. The system should undergo penetration testing, vulnerability assessments, and compliance reviews to ensure readiness for handling highly sensitive case data. Additional measures such as two-factor authentication and encrypted backups can further safeguard the platform against unauthorized access or data breaches.

In the future, including artificial intelligence (AI) will significantly increase the system's worth. AI may be used to create predictive insights to help with preventive interventions, analyse secure communication for risk indicators, and prioritize critical instances. By doing this, the system would change from a reactive case management platform to one that assists agencies in making decisions. Additionally, making the system available as a mobile application with offline submission capabilities would improve accessibility for users in remote or under connected locations, guaranteeing inclusion and wider uptake. The Integrated Child Welfare System can develop from a final year project system into a strong national platform that provides safe reporting, effective case management, and productive cooperation amongst all child protection stakeholders by putting these future improvements into practice.

References

Adegbuyi, F. (n.d.). *Kanban*. Retrieved from ttodoist: https://www.todoist.com/productivity-methods/kanban

Afrouz, R., & Vassos, S. (2024). Adolescents' Experiences of Cyber-Dating Abuse and the Pattern of Abuse Through Technologu, A Scoping Review. *TRAUMA, VIOLENCE & ABUSE, 25*(4), 2814-2828. https://doi.org/10.1177/15248380241227457

Alex, B., Taylor, L., Winch, R., Hillert, G., Grandja, J., & Bryant, J. (n.d.). *Spring Security Reference*. Retrieved from Spring: https://docs.spring.io/spring-security/site/docs/5.0.7.RELEASE/reference/htmlsingle/

Ali, I. (13 December, 2024). *A Comprehensive Guide to Integrating Laravel Cache in Your Web Application*. Retrieved from Cloudways: https://www.cloudways.com/blog/integrate-laravel-cache/

Alsharif, M., Mishra, S., & Alshehri, M. (2022). Impact of Human Vulnerabilities on Cybersecurity. *Computer Systems Science & Engineering, 40*(3), 1153-1166. https://doi.org/10.32604/csse.2022.019938

Altunpul, N., & Kocturk, N. (2024). Technology-Based Child Abuse and Neglect Prevention Programs: A Systematic Review. *Psikiyatride Güncel Yaklaşımlar-Current Approaches in Psychiatry, 16*(3), 466-484. https://doi.org/10.18863/pgy.1321465

Avatour. (n.d.). *How to Implement Kanban for Continuous Improvement*. Retrieved from Avatour: https://avatour.com/article/how-to-implement-kanban-for-continuous-improvement#:~:text=Continuous%20Improvement%20with%20Kanban,continuous%20improvement%20in%20your%20organization.

Bagheri, F., Abbasi, F., Sadeghi, M., & Khajouei, R. (2023). Evaluating the usability of a cancer registry system using Cognitive Walkthrough, and assessing user agreement with its problems. *BMC Medical Informatics and Decision Making, 23*. https://doi.org/10.1186/s12911-023-02120-8

Barvaliya, P. (13 Feburary, 2024). *12 Benefits Of Laravel For Enterprise Development*. Retrieved from RejoiceHub: https://rejoicehub.com/blogs/benefits-of-laravel-for-enterprise-development

Bastian, C., Dunk-West, P., & Wendt, S. (2021). Being child-centred: Factors that facilitate professional judgement and decision-making in child protection. *Child & Family Social Work, 27*(2), 91-99. https://doi.org/10.1111/cfs.12855

Berger, I., & Torres, T. (10 September, 2024). *What is Trust and Safety? A Comprehensive Guide*. Retrieved from ActiveFence: https://www.activefence.com/what-is-trust-and-safety/#:~:text=For%20users%2C%20a%20lack%20of,financial%20fines%20and%20regulatory%20measures.

Bhandari, P. (15 July, 2021). *Questionnaire Design | Methods, Question Types & Examples*. Retrieved from Scribbr: https://www.scribbr.com/methodology/questionnaire/

Bishop, T. (4 Ocotber, 2021). *Windows 11 is here: Key points to know as Microsoft rolls out its new operating system*. Retrieved from GeekWire: https://www.geekwire.com/2021/windows-11-key-points-know-microsoft-rolls-new-operating-system/

Casebook. (2024). *Casebook Child Welfare Software*. Retrieved from Casebook: https://www.casebook.net/casebook-child-welfare-software/

CaseBook. (2024). *Casebook for Child, Youth, and Family Services*. Retrieved from Casebook for Child, Youth, and Family Services: https://www.casebook.net/casebook-for-child-youth-and-family-service/

Casebook. (2024). *Getting Started With Casebook* . Retrieved from Casebook Knowledge Base: https://gohub.casebook.net/knowledge/user-roles-and-permissions-in-casebook

CPIMS+. (2024). *CPIMS+*. Retrieved from CPIMS+: https://www.cpims.org/

DeFranzo, S. E. (n.d.). *Advantages and disadvantages of using surveys for research*. Retrieved from Snap Survey: https://www.snapsurveys.com/blog/advantages-disadvantages-surveys/

Digitai.ai. (n.d.). *Agile Development Process Models in Software Engineering*. Retrieved from digital.ai: https://digital.ai/catalyst-blog/agile-development-process-software-engineering/

Doelman, E. H., Horst, F. C., Luijk, M. P., & Steketee, M. J. (2024). Parents perspectives on the results of mandatory child protection support: A qualitative study. *Child & Family Social Work*, 928-938. https://doi.org/10.1111/cfs.13144

Doh, D., Kamara, J., Galukande, M., & Renzaho, A. (2021). Evaluating the impact of a community-based livelihood intervention on child protection: A mixed method approach. *Child & Family Social Work, 27*(2), 217-235. https://doi.org/10.1111/cfs.12874

Erickson, J. (29 August, 2024). *MySQL: Understanding What It Is and How It's Used*. Retrieved from Oracle: https://www.oracle.com/mysql/what-is-mysql/#:~:text=MySQL%20is%20an%20open%20source%20relational%20database.,NoSQL%20databases%2C%20such%20as%20MongoDB.

Garstang, J., Debelle, G., Anand, I., Armstrong, J., Botcher, E., Chaplin, H., . . . Taylor, J. (2020). Effect of COVID-19 lockdown on child protection medical assessments: a retrospective observational study in Birmingham, UK. *BMJ Open, 10*, 1-6. https://doi.org/10.1136/bmjopen-2020-042867

GeeksforGeeks. (20 June, 2022). *Advantages and Disadvantages of PHP*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/

GeeksforGeeks. (9 November, 2022). *Disadvantages of Using MySQL*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/disadvantages-of-using-mysql/

GeeksforGeeks. (5 September, 2024). *PHP Data Types*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/php-data-types/

Ghani, R. A., Rosman, A., & Muhammad, A. (2019). The Suspected Child Abuse and Neglect (SCAN) Programme in Malaysia: From Inception to Present. *Global Journal of Health Science, 11*(7), 148-153. https://doi.org/10.5539/gjhs.v11n7p148

Giardino, A., & De Jesus, O. (14 August, 2023). *Case Mangement*. Retrieved from National Library of Machine: https://www.ncbi.nlm.nih.gov/books/NBK562214/

Griffin, T., Williams, J. H., & Kadleck, C. (2023). AMBER Alert Effectivess Reexamined. *Crimnal Justice Policy Review, 33*(1), 1-21. https://doi.org/10.1177/08874034211026366

Hakoune, R. (24 October, 2023). *Kanban methodology: everything you need to know*. Retrieved from monday blog: https://monday.com/blog/rnd/kanban-project-management/

Halim, S., Taib, F., & Rahman, A. A. (2023). Non-compliance with Outpatient Follow-up in Child Abuse Cases. *Journal of Clinical and Health Sciences, 8*(1), 52-61. https://doi.org/10.24191/jchs.v8i1.16944

Hanif, F. (26 June, 2023). *Understanding the MVC Architecture in Laravel: A Comprehensive Guide*. Retrieved from Meidum: https://fkrihnif.medium.com/understanding-the-mvc-architecture-in-laravel-a-comprehensive-guide-8f620cc139b6

Herbert, J. L., & Bromfield, L. (2019). Multi-disciplinary teams responding to child abuse: Common features and assumptions. *Children and Youth Services Review, 106*. https://doi.org/10.1016/j.childyouth.2019.104467

Howard, N., Roelen, K., Ton, G., Hermoza, M. E., Mamun, S. A., Chowdhury, K., . . . Huq, L. (2025). A new approach to children's work that prioritises resilience, well-being and agency: emerging findings from a 'cash plus' intervention in Bangladesh. *BMJ Paediatrics Open, 9*, 1-8. https://doi.org/0.1136/bmjpo-2023-002422

Ibrahim, S. (11 January, 2024). *Laravel Fundamentals: Understanding the MVC Architecture*. Retrieved from Linkedin: https://www.linkedin.com/pulse/laravel-fundamentals-understanding-mvc-architecture-saheed-ibrahim--mgoyf

Indeed. (4 March, 2025). *Advantages And Disadvantages Of Interviews (And Useful Tips)*. Retrieved from Indeed: https://in.indeed.com/career-advice/interviewing/advantages-and-disadvantages-of-interviews

Indeed. (28 March, 2025). *What Is Document Analysis? (Definition, Steps, and Benefits)*. Retrieved from indeed: https://ca.indeed.com/career-advice/career-development/document-analysis

InfoStride. (3 March, 2022). *InfoStride*. Retrieved from Top 15 Websites Built with PHP: https://infostride.com/websites-built-with-php/#:~:text=its%20market%20usage.-,What%20is%20PHP?,work%20well%20with%20modern%20technologies.

Jaafar, S. M., Arifin, W. N., & Rahman, A. A. (2024). Factors Associated with Intention to Report Child Abuse among Early Childcare Providers in North-eastern Preschools, Malaysia. *IIUM Medical Journal Malaysia, 23*(2), 62-68. https://doi.org/10.31436/imjm.v23i02.2438

K&C. (n.d.). *Kanban vs Scrum: 7 reasons why Kanban might suit you better than Scrum*. Retrieved from K&C: https://kruschecompany.com/why-kanban-might-suit-better-than-scrum/#Flexibility

Kamal, M. H. (5 January, 2024). *Navigate Mid-Sprint Changes - Sprint Goal Adjustments: A Guide for Scrum Masters (Scenarios, Solutions & Decision Frameworks)*. Retrieved from Linkedin: https://www.linkedin.com/pulse/navigate-mid-sprint-changes-sprint-goal-adjustments-kamal-swxmf#:~:text=Changing%20the%20Sprint%20Goal%20during,exception%20and%20not%20a%20norm.

Khakhkhar, K. (5 February, 2025). *Guide to pick the right software development methodologies for your next project*. Retrieved from peerbits: https://www.peerbits.com/blog/how-to-choose-software-development-methodology.html

Khan, M. (26 January, 2025). *Practical Guide to Security Middleware in Django*. Retrieved from Medium: https://medium.com/django-unleashed/practical-guide-to-security-middleware-in-django-77b213fa944b#:~:text=12-,Practical%20Guide%20to%20Security%20Middleware%20in%20Django,measures%2C%20across%20the%20entire%20project.

Krasteva, I. (n.d.). *The 12 Most Popular Project Management Methodologies*. Retrieved from https://businessmap.io/project-management/methodologies#:~:text=Project%20Management%20Methodology?-,A%20project%20management%20methodology%20is%20a%20systematic%20approach%20that%20guides,achieved%20within%20the%20defined%20constraints.

Kwon, T., Jeong, M., Ko, E.-S., & Lee, Y. (2022). Captivate! Contextual Language Guidance for Parent–Child Interaction. 1-17. https://doi.org/10.1145/3491102.3501865

Laravel. (n.d.). *Authentication*. Retrieved from Laravel: https://laravel.com/docs/12.x/authentication

Laravel. (n.d.). *Build and ship software with tools crafted for productivity*. Retrieved from Laravel: https://laravel.com/

Laravel. (n.d.). *CSRF Protection*. Retrieved from Laravel: https://laravel.com/docs/12.x/csrf

Laravel. (n.d.). *Database: Migrations*. Retrieved from Laravel: https://laravel.com/docs/12.x/migrations

Laravel. (n.d.). *Eloquent: Getting Started*. Retrieved from Laravel: https://laravel.com/docs/12.x/eloquent

Laravel. (n.d.). *Release Notes*. Retrieved from Laravel: https://laravel.com/docs/12.x/releases

Lee, D., Lee, J. H., Kwak, Y. H., Suh, D., Kwon, H., Kim, D. K., . . . Kim, J. H. (2024). The Barriers to Recognizing and Reporting Child Physical Abuse by Emergency Physicians and Associated Factors. *Pediatric Emergency Care, 40*(9), 675-680. https://doi.org/10.1097/PEC.0000000000003146

Machpheerson, J. M., Gray, S. R., Ip, P., McCallum, M., Hanlon, P., Welsh, P., . . . Ho, F. K. (2021). Child maltreatment and incident mental disorders in middle and older ages: a retrospective UK Biobank cohort study. *The Lancet Regional Health, 11*, 1-9. https://doi.org/10.1016/j.lanepe.2021.100224

Martins, J. (19 January, 2025). *What is Kanban? Free Kanban template, with examples*. Retrieved from Asana: https://asana.com/resources/what-is-kanban

mdn web docs. (n.d.). *CSS: Cascading Style Sheets*. Retrieved from mdn web docs: https://developer.mozilla.org/en-US/docs/Web/CSS

mdn web docs. (n.d.). *What is Javascript*. Retrieved from mdn web docs: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript

Munnangi, S. (2024). Empowering Access to Justice through Automated Legal Case Management. *Research, 6*(6), 1. https://doi.org/10.36948/ijfmr.2024.v06i06.31915

Mydin, F., Othman, S., Choo, W. Y., Hairi, N., Hairi, F., Ali, Z., & Aziz, S. (2021). Primary care nurses' challenges, oppurtunities, and facilitators of elder abuse intervention in Malaysia. *Public Health Nursing, 39*, 3-14. https://doi.org/10.1111/phn.12961

MySQL Database. (n.d.). Retrieved from Systems Group: https://systems.cs.odu.edu/database/mysql/

Nwaimo, C. S., Adegbola, A. E., & Adegbola, M. D. (2024). Data-driven strategies for enhancing user engagement in digital platforms. *International Journal of Management*

& Entrepreneurship Research, 6(6), 1854-1868. https://doi.org/10.51594/ijmer.v6i6.1170

Odikwa, N. H., Ikenna, U., & Bright, A. (2022). Enhanced Child Abuse Information Management System - The Nigeria Context. *International Journal of Research and Scientific Innovation, 9*(1), 151-157. https://doi.org/10.51244/IJRSI.2022.9106

Oliveira, A. d. (30 Novemebr, 2023). *What Is Kanban in Lean Manufacturing and How to Use it*. Retrieved from Proaction: https://blog.proactioninternational.com/en/kanban-in-lean-manufacturing-how-to-use-it

Paki, N. P., King, P. T., Lewis, L., Walker, S., Mackay, H. T., & Anderson, D. (2023). Identifying key dimensions of indigenous led child welfare services: A qualitative literature review. *Child & Family School Work*, 465-481. https://doi.org/10.1111/cfs.13100

PFordten, D. (26 August, 2024). *Child neglect and abuse on the rise in Malaysia*. Retrieved from The Star: https://www.thestar.com.my/news/nation/2024/08/26/multimedia-child-neglect-and-abuse-on-the-rise-in-malaysia

Pinjatela, R. (2024). FACTORS INFLUENCING THE NON-USE AND ABANDONMENT OF ASSISTIVE TECHNOLOGY. *Research in Education and Rehabilitation, 7*(1), 56-66. https://doi.org/10.51558/2744-1555.2024.7.1.56

planview. (n.d.). *KANBAN VS. SCRUM: WHAT ARE THE DIFFERENCES?* Retrieved from planview: https://www.planview.com/resources/guide/introduction-to-kanban/kanban-vs-scrum/

Prasanna. (21 October, 2024). *Advantages and Disadvantages of Observation Method | Merits and Demerits of Observation in Marketing*. Retrieved from AplusTopper: https://www.aplustopper.com/advantages-and-disadvantages-of-observation-method/

Primero. (June, 2024). *Get to Know Primero v2*. Retrieved from https://uploads.strikinglycdn.com/files/3b743c4a-4b75-4a7c-9a4c-41259770c81a/Get%20to%20Know%20Primero%20v2.pdf

Rafael, R. (9 October, 2023). *7 Ways to Make Your Laravel App Seamlessly Faster*. Retrieved from Meidum: https://rafaelogic.medium.com/7-ways-to-make-your-laravel-app-seamlessly-faster-089f76797250

Randawar, D. K., Ikhsan, I. B., & Khan, A. R. (2024). Mandatory Reporting System for Domestic Violence Cases: A Comparative Legal Analysis. *International Journal of Academic Research in Business and Social Science, 14*(4), 1635-1646. https://doi.org/10.6007/IJARBSS/v14-i4/20941

Raymond, D. (10 January, 2024). *Top 10 Pros and Cons of Windows 11*. Retrieved from AceProject: https://www.aceproject.com/blog/top-10-pros-and-cons-of-windows-11-9387815/

Roche, S., & Flynn, C. (2020). Local child protection in the Philippines: Acase study of actors, processes and key risks forchildren. *Asia Pac Policy Stud*, 367-383. https://doi.org/10.1002/app5.332

S, B. (30 April, 2024). *What is Kanban? A Beginner's Guide!* Retrieved from Nimble: https://www.nimblework.com/kanban/what-is-kanban/#kanban-method

Seddighi, H., Salmani, I., Javadi, M. H., & Seddighi, S. (2021). Child Abuse in Natural Disasters and Conflicts: A Systematic Review. *Trauma, Violence, & Abuse, 22*(1), 176-185. https://doi.org/10.1177/152483801983597

Sharmal, S. (15 April, 2023). *Common Advantages and Disadvantages of Multithreading in Java Trainin*. Retrieved from Multisoft Virtual Academy: https://www.multisoftvirtualacademy.com/blog/common-advantages-and-disadvantages-of-multithreading-in-java#:~:text=Multithreading%20is%20a%20widely%20used,performance%20and%20responsiveness%20of%20applications.

Shetty, A. (21 March, 2025). *Customer service automation: Benefits and examples*. Retrieved from yellow.ai: https://yellow.ai/customer-service-automation/

strongdm. (2 January, 2025). *The Definitive Guide to Role-Based Access Control (RBAC)*. Retrieved from strongdm: https://www.strongdm.com/rbac#:~:text=Role%2Dbased%20access%20control%20(RBAC)%20is%20a%20security%20approach,execute%20commands%20or%20delete%20information.

StudySmarter. (n.d.). *StudySmarter*. Retrieved from Focus Group: https://www.studysmarter.co.uk/explanations/marketing/marketing-information-management/focus-groups/

SurveyMonkey. (n.d.). *Demographic survey question guide: Tips, examples, & best practices*. Retrieved from SurveyMonkey: https://www.surveymonkey.com/mp/gathering-demographic-information-from-surveys/

SurveyMonkey. (n.d.). *Why are surveys important in research?* Retrieved from surveymonkey: https://www.surveymonkey.com/learn/survey-best-practices/why-are-surveys-important-in-research/

Tahir, M., & Cobigo, V. (2025). 'I Don't Think I Have Ever Worked Harder on a Case': Needs of Canadian Child Protection Workers and Parents With Intellectual Disabilities. *Journal of Applied Research in Intelletual Disabilities, 38*(1). https://doi.org/10.1111/jar.70006

TiDB Team. (4 September, 2024). *Mastering Laravel Migrations with Expert Tips*. Retrieved from TiDB: https://www.pingcap.com/article/mastering-laravel-migrations-expert-tips/

Towhid, I. I., Nuzhat, R., & Haque, M. A. (2023). Protecting children from online sexual abuse in Bangladesh: A plocy brief. *Bangabandhu Sheikh Mujib Medical University Journal, 16*(2), 111-117. https://doi.org/10.3329/bsmmuj.v16i2.67209

UNICEF. (2020). *Violence against Children in South Asia: A systematic review of evidence since 2015*. Unicef.

UNICEF. (4 April, 2022). *UNICEF and partners launch digital tool for child protection case management in Sudan*. Retrieved from unicef: https://www.unicef.org/sudan/press-releases/unicef-and-partners-launch-digital-tool-child-protection-case-management-sudan

Uspenski, A. (25 April, 2023). *Why VS Code remains a developer favorite, year after year*. Retrieved from ShiftMag: https://shiftmag.dev/vs-code-171/

w3schools. (n.d.). *AJAX Introduction*. Retrieved from w3schools: https://www.w3schools.com/Asp/asp_ajax_intro.asp w3schools. (n.d.). *HTML Introduction*. Retrieved from w3schools: https://www.w3schools.com/html/html_intro.asp

Wharton, K. C. (n.d.). *The 7 principles of privacy by design*. Retrieved from ontrust: https://www.onetrust.com/blog/principles-of-privacy-by-design/#:~:text=Principle%207:%20Respect%20for%20user%20privacy%20%E2%80%93%20Keep%20it%20user%2D,their%20engagement%20in%20the%20process.

Yadav, R. (23 December, 2022). *Laravel(PHP) Vs Spring(JAVA) Vs Django(PYTHON)*. Retrieved from Cracode: https://cracode.com/laravelphp-vs-springjava-vs-djangopython/

Yob, Z., Shaari, M. S., Esquivias, M. A., Nangle, B., & Muhammad, Z. W. (2022). The Impacts of Poverty, Unemployment, and Divorce on Child Abuse in Malaysia: ARDL Approach. *Economies, 10*(11), 1-14. https://doi.org/10.3390/economies10110291

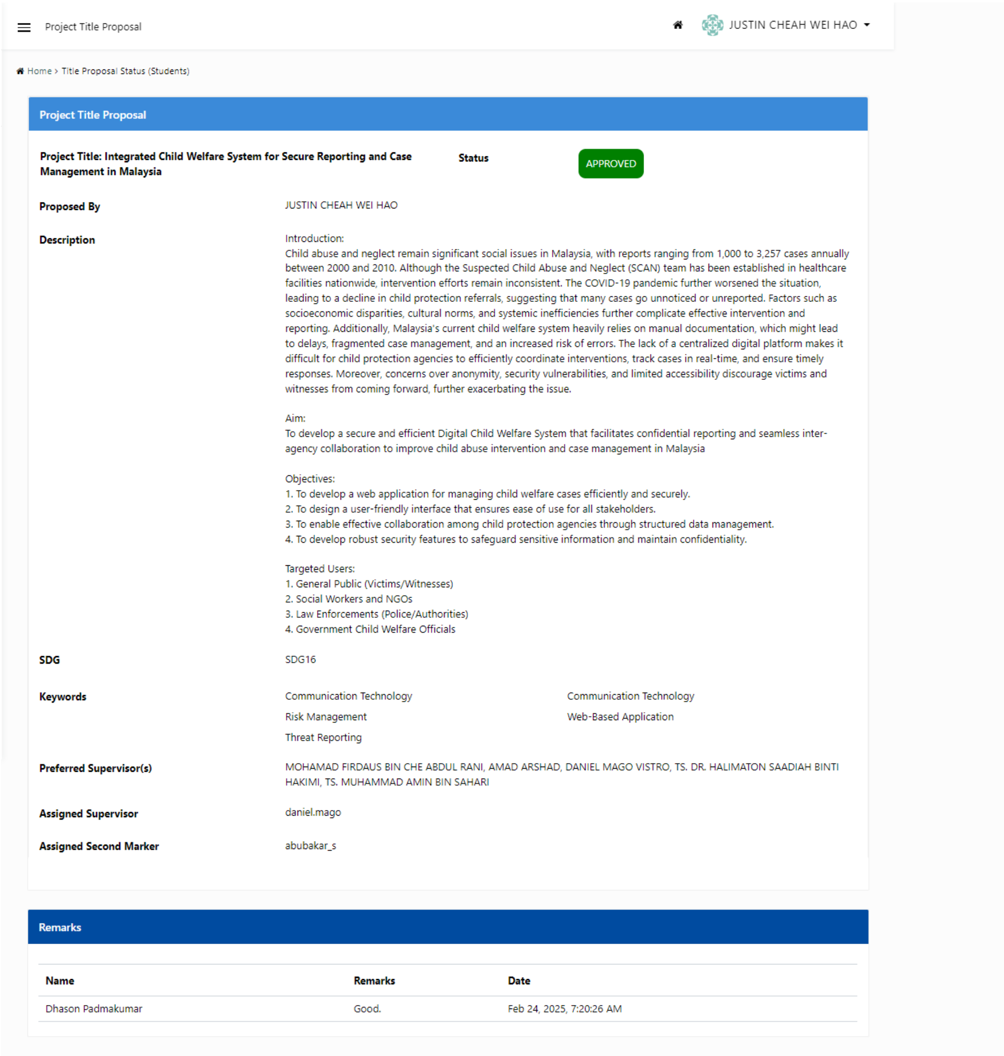
Yong, C. (28 August, 2021). *MVC pattern in Django*. Retrieved from Meidum: https://medium.com/@iamwankang/mvc-pattern-in-django-ef1ca4d1d64e

Yosef, G. (2 April, 2025). *API Security Best Practices: 16 Ways to Secure Your APIs*. Retrieved from pynt: https://www.pynt.io/learning-hub/api-security-guide/api-security-best-practices

Yu, J., Bekerian, D. A., & Osback, C. (2024). Navigating the Digital Landscape: Challenges and Barriers to Effective Information Use on the Internet. *Encyclopedia, 4*(4), 1665-1680. https://doi.org/10.3390/encyclopedia4040109

Appendices

Appendix A: FYP Title



Project Title Proposal

JUSTIN CHEAH WEI HAO

Home > Title Proposal Status (Students)

Project Title Proposal

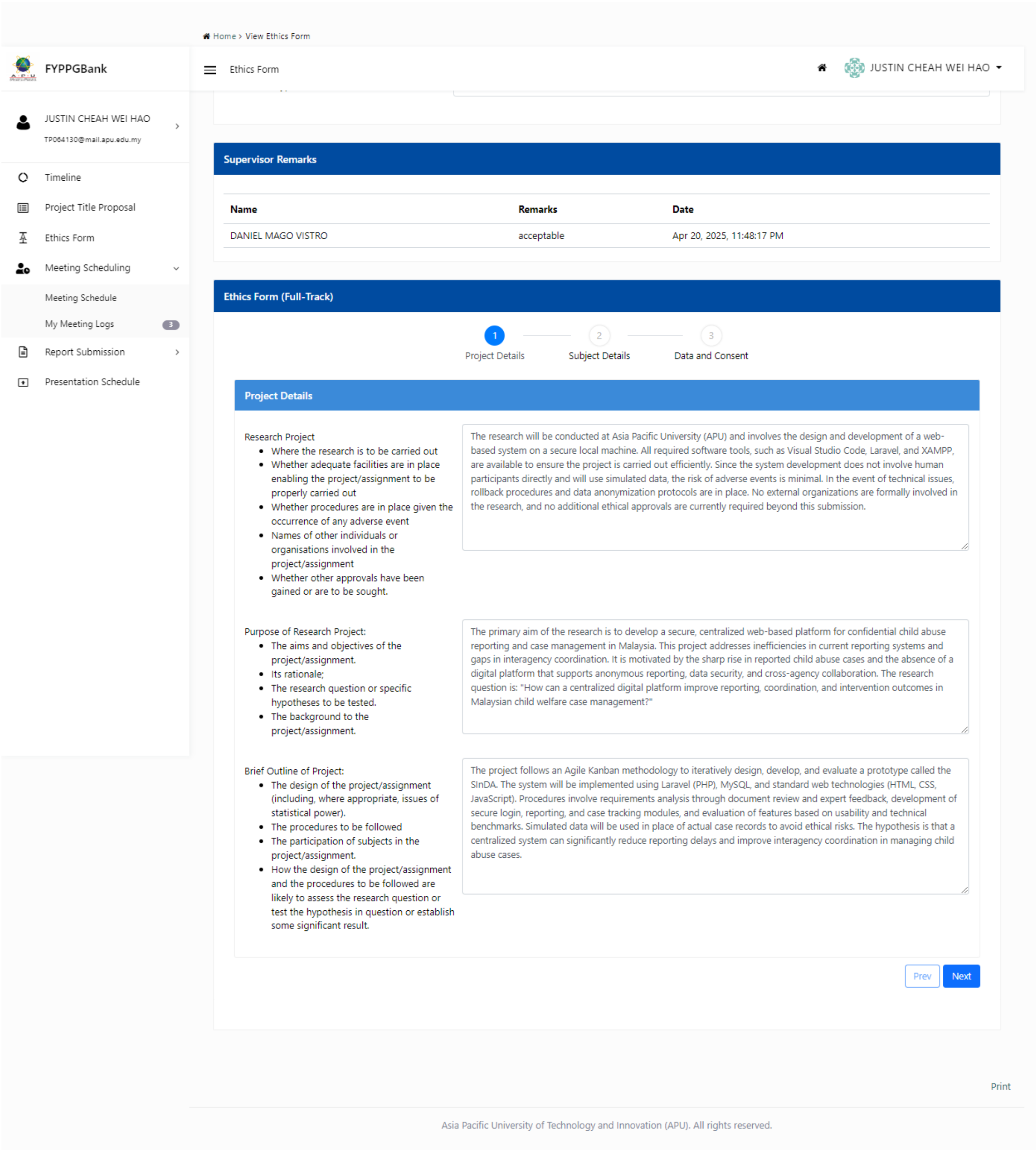
Project Title: Integrated Child Welfare System for Secure Reporting and Case Management in Malaysia	**Status** APPROVED
Proposed By	JUSTIN CHEAH WEI HAO
Description	Introduction: Child abuse and neglect remain significant social issues in Malaysia, with reports ranging from 1,000 to 3,257 cases annually between 2000 and 2010. Although the Suspected Child Abuse and Neglect (SCAN) team has been established in healthcare facilities nationwide, intervention efforts remain inconsistent. The COVID-19 pandemic further worsened the situation, leading to a decline in child protection referrals, suggesting that many cases go unnoticed or unreported. Factors such as socioeconomic disparities, cultural norms, and systemic inefficiencies further complicate effective intervention and reporting. Additionally, Malaysia's current child welfare system heavily relies on manual documentation, which might lead to delays, fragmented case management, and an increased risk of errors. The lack of a centralized digital platform makes it difficult for child protection agencies to efficiently coordinate interventions, track cases in real-time, and ensure timely responses. Moreover, concerns over anonymity, security vulnerabilities, and limited accessibility discourage victims and witnesses from coming forward, further exacerbating the issue. Aim: To develop a secure and efficient Digital Child Welfare System that facilitates confidential reporting and seamless inter-agency collaboration to improve child abuse intervention and case management in Malaysia Objectives: 1. To develop a web application for managing child welfare cases efficiently and securely. 2. To design a user-friendly interface that ensures ease of use for all stakeholders. 3. To enable effective collaboration among child protection agencies through structured data management. 4. To develop robust security features to safeguard sensitive information and maintain confidentiality. Targeted Users: 1. General Public (Victims/Witnesses) 2. Social Workers and NGOs 3. Law Enforcements (Police/Authorities) 4. Government Child Welfare Officials
SDG	SDG16
Keywords	Communication Technology Risk Management Threat Reporting Communication Technology Web-Based Application
Preferred Supervisor(s)	MOHAMAD FIRDAUS BIN CHE ABDUL RANI, AMAD ARSHAD, DANIEL MAGO VISTRO, TS. DR. HALIMATON SAADIAH BINTI HAKIMI, TS. MUHAMMAD AMIN BIN SAHARI
Assigned Supervisor	daniel.mago
Assigned Second Marker	abubakar_s

Remarks

Name	Remarks	Date
Dhason Padmakumar	Good.	Feb 24, 2025, 7:20:26 AM

Figure 322: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_FYP-TITLE

Appendix B: Ethics Form (Full Track)


Home > View Ethics Form

FYPPGBank

Ethics Form

JUSTIN CHEAH WEI HAO

JUSTIN CHEAH WEI HAO
TP064130@mail.apu.edu.my

- Timeline
- Project Title Proposal
- Ethics Form
- Meeting Scheduling
  - Meeting Schedule
  - My Meeting Logs
- Report Submission
- Presentation Schedule

Supervisor Remarks

Name	Remarks	Date
DANIEL MAGO VISTRO	acceptable	Apr 20, 2025, 11:48:17 PM

Ethics Form (Full-Track)

1 Project Details — 2 Subject Details — 3 Data and Consent

Project Details

Research Project
- Where the research is to be carried out
- Whether adequate facilities are in place enabling the project/assignment to be properly carried out
- Whether procedures are in place given the occurrence of any adverse event
- Names of other individuals or organisations involved in the project/assignment
- Whether other approvals have been gained or are to be sought.

The research will be conducted at Asia Pacific University (APU) and involves the design and development of a web-based system on a secure local machine. All required software tools, such as Visual Studio Code, Laravel, and XAMPP, are available to ensure the project is carried out efficiently. Since the system development does not involve human participants directly and will use simulated data, the risk of adverse events is minimal. In the event of technical issues, rollback procedures and data anonymization protocols are in place. No external organizations are formally involved in the research, and no additional ethical approvals are currently required beyond this submission.

Purpose of Research Project:
- The aims and objectives of the project/assignment.
- Its rationale;
- The research question or specific hypotheses to be tested.
- The background to the project/assignment.

The primary aim of the research is to develop a secure, centralized web-based platform for confidential child abuse reporting and case management in Malaysia. This project addresses inefficiencies in current reporting systems and gaps in interagency coordination. It is motivated by the sharp rise in reported child abuse cases and the absence of a digital platform that supports anonymous reporting, data security, and cross-agency collaboration. The research question is: "How can a centralized digital platform improve reporting, coordination, and intervention outcomes in Malaysian child welfare case management?"

Brief Outline of Project:
- The design of the project/assignment (including, where appropriate, issues of statistical power).
- The procedures to be followed
- The participation of subjects in the project/assignment.
- How the design of the project/assignment and the procedures to be followed are likely to assess the research question or test the hypothesis in question or establish some significant result.

The project follows an Agile Kanban methodology to iteratively design, develop, and evaluate a prototype called the SInDA. The system will be implemented using Laravel (PHP), MySQL, and standard web technologies (HTML, CSS, JavaScript). Procedures involve requirements analysis through document review and expert feedback, development of secure login, reporting, and case tracking modules, and evaluation of features based on usability and technical benchmarks. Simulated data will be used in place of actual case records to avoid ethical risks. The hypothesis is that a centralized system can significantly reduce reporting delays and improve interagency coordination in managing child abuse cases.

Prev Next

Print

Asia Pacific University of Technology and Innovation (APU). All rights reserved.

Figure 323: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_ETHICS_FORM_1

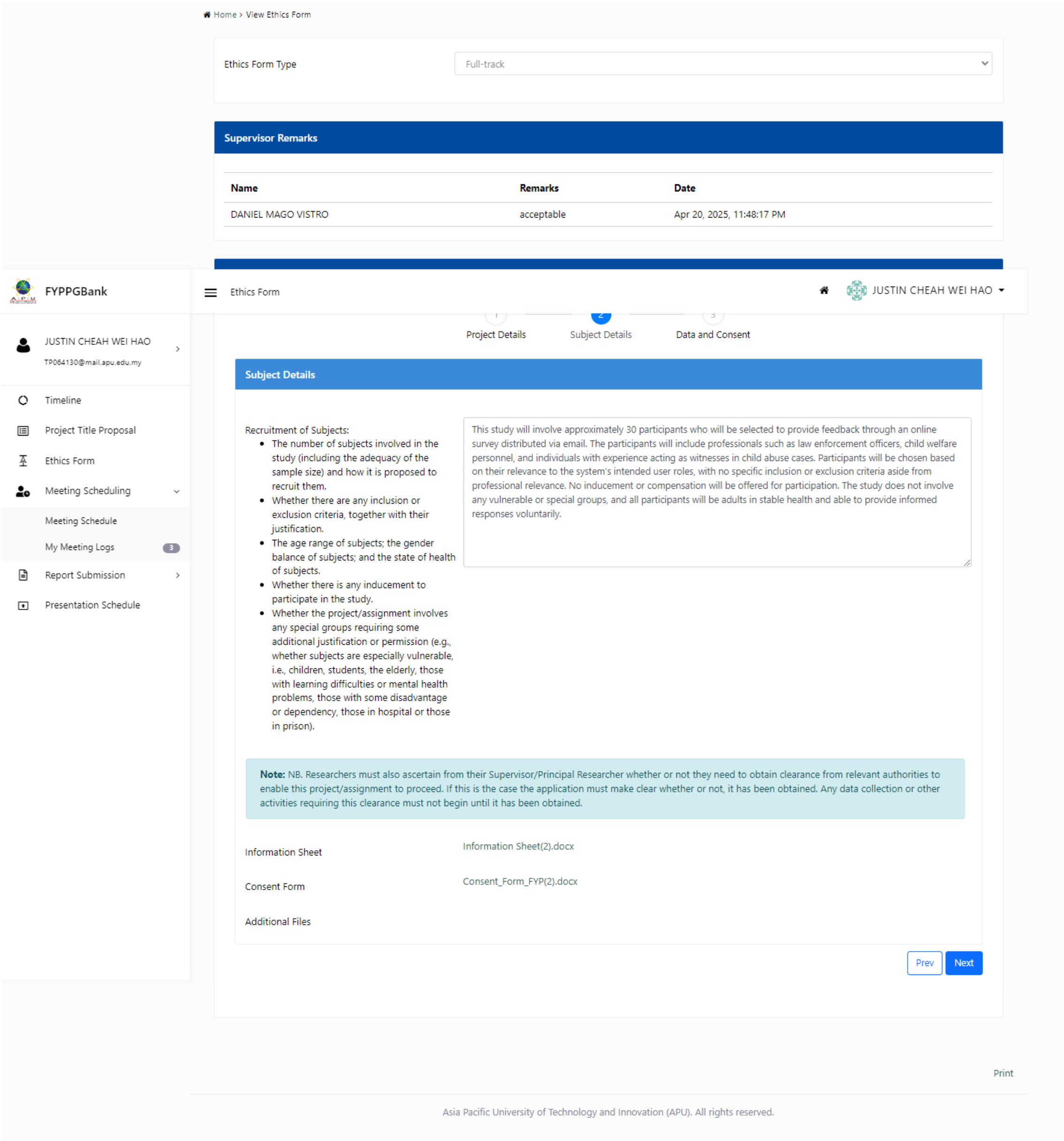
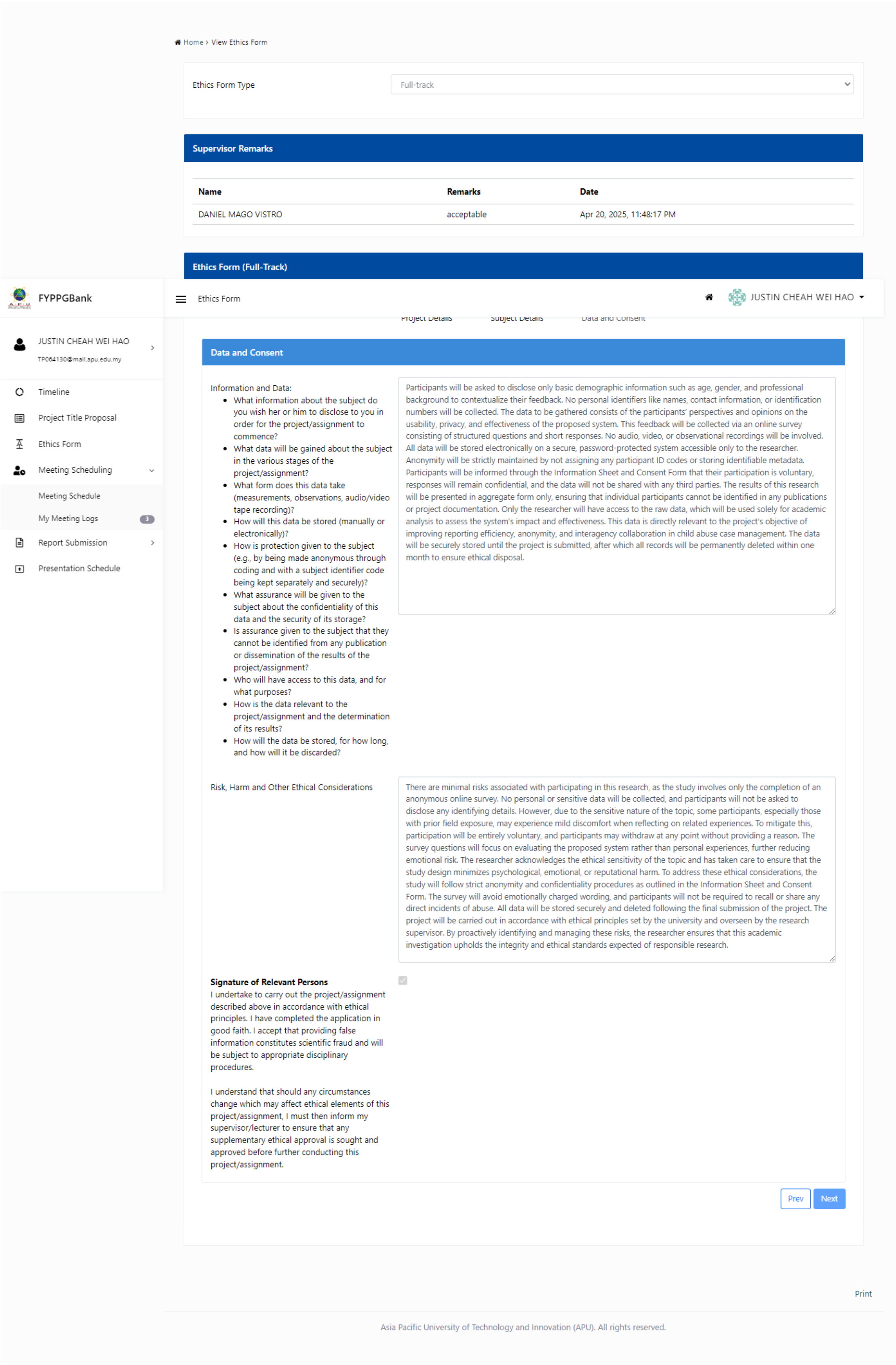


Figure 324: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_ETHICS_FORM_2



Home > View Ethics Form

Ethics Form Type: Full-track

Supervisor Remarks

Name	Remarks	Date
DANIEL MAGO VISTRO	acceptable	Apr 20, 2025, 11:48:17 PM

Ethics Form (Full-Track)

FYPPGBank

Ethics Form

JUSTIN CHEAH WEI HAO

JUSTIN CHEAH WEI HAO
TP064130@mail.apu.edu.my

- Timeline
- Project Title Proposal
- Ethics Form
- Meeting Scheduling
  - Meeting Schedule
  - My Meeting Logs 3
- Report Submission
- Presentation Schedule

Project Details Subject Details Data and Consent

Data and Consent

Information and Data:

- What information about the subject do you wish her or him to disclose to you in order for the project/assignment to commence?
- What data will be gained about the subject in the various stages of the project/assignment?
- What form does this data take (measurements, observations, audio/video tape recording)?
- How will this data be stored (manually or electronically)?
- How is protection given to the subject (e.g., by being made anonymous through coding and with a subject identifier code being kept separately and securely)?
- What assurance will be given to the subject about the confidentiality of this data and the security of its storage?
- Is assurance given to the subject that they cannot be identified from any publication or dissemination of the results of the project/assignment?
- Who will have access to this data, and for what purposes?
- How is the data relevant to the project/assignment and the determination of its results?
- How will the data be stored, for how long, and how will it be discarded?

Participants will be asked to disclose only basic demographic information such as age, gender, and professional background to contextualize their feedback. No personal identifiers like names, contact information, or identification numbers will be collected. The data to be gathered consists of the participants' perspectives and opinions on the usability, privacy, and effectiveness of the proposed system. This feedback will be collected via an online survey consisting of structured questions and short responses. No audio, video, or observational recordings will be involved. All data will be stored electronically on a secure, password-protected system accessible only to the researcher. Anonymity will be strictly maintained by not assigning any participant ID codes or storing identifiable metadata. Participants will be informed through the Information Sheet and Consent Form that their participation is voluntary, responses will remain confidential, and the data will not be shared with any third parties. The results of this research will be presented in aggregate form only, ensuring that individual participants cannot be identified in any publications or project documentation. Only the researcher will have access to the raw data, which will be used solely for academic analysis to assess the system's impact and effectiveness. This data is directly relevant to the project's objective of improving reporting efficiency, anonymity, and interagency collaboration in child abuse case management. The data will be securely stored until the project is submitted, after which all records will be permanently deleted within one month to ensure ethical disposal.

Risk, Harm and Other Ethical Considerations

There are minimal risks associated with participating in this research, as the study involves only the completion of an anonymous online survey. No personal or sensitive data will be collected, and participants will not be asked to disclose any identifying details. However, due to the sensitive nature of the topic, some participants, especially those with prior field exposure, may experience mild discomfort when reflecting on related experiences. To mitigate this, participation will be entirely voluntary, and participants may withdraw at any point without providing a reason. The survey questions will focus on evaluating the proposed system rather than personal experiences, further reducing emotional risk. The researcher acknowledges the ethical sensitivity of the topic and has taken care to ensure that the study design minimizes psychological, emotional, or reputational harm. To address these ethical considerations, the study will follow strict anonymity and confidentiality procedures as outlined in the Information Sheet and Consent Form. The survey will avoid emotionally charged wording, and participants will not be required to recall or share any direct incidents of abuse. All data will be stored securely and deleted following the final submission of the project. The project will be carried out in accordance with ethical principles set by the university and overseen by the research supervisor. By proactively identifying and managing these risks, the researcher ensures that this academic investigation upholds the integrity and ethical standards expected of responsible research.

Signature of Relevant Persons

I undertake to carry out the project/assignment described above in accordance with ethical principles. I have completed the application in good faith. I accept that providing false information constitutes scientific fraud and will be subject to appropriate disciplinary procedures.

I understand that should any circumstances change which may affect ethical elements of this project/assignment, I must then inform my supervisor/lecturer to ensure that any supplementary ethical approval is sought and approved before further conducting this project/assignment.

Prev Next

Print

Figure 325: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_ETHICS_FORM_3

Appendix C: Log Sheets

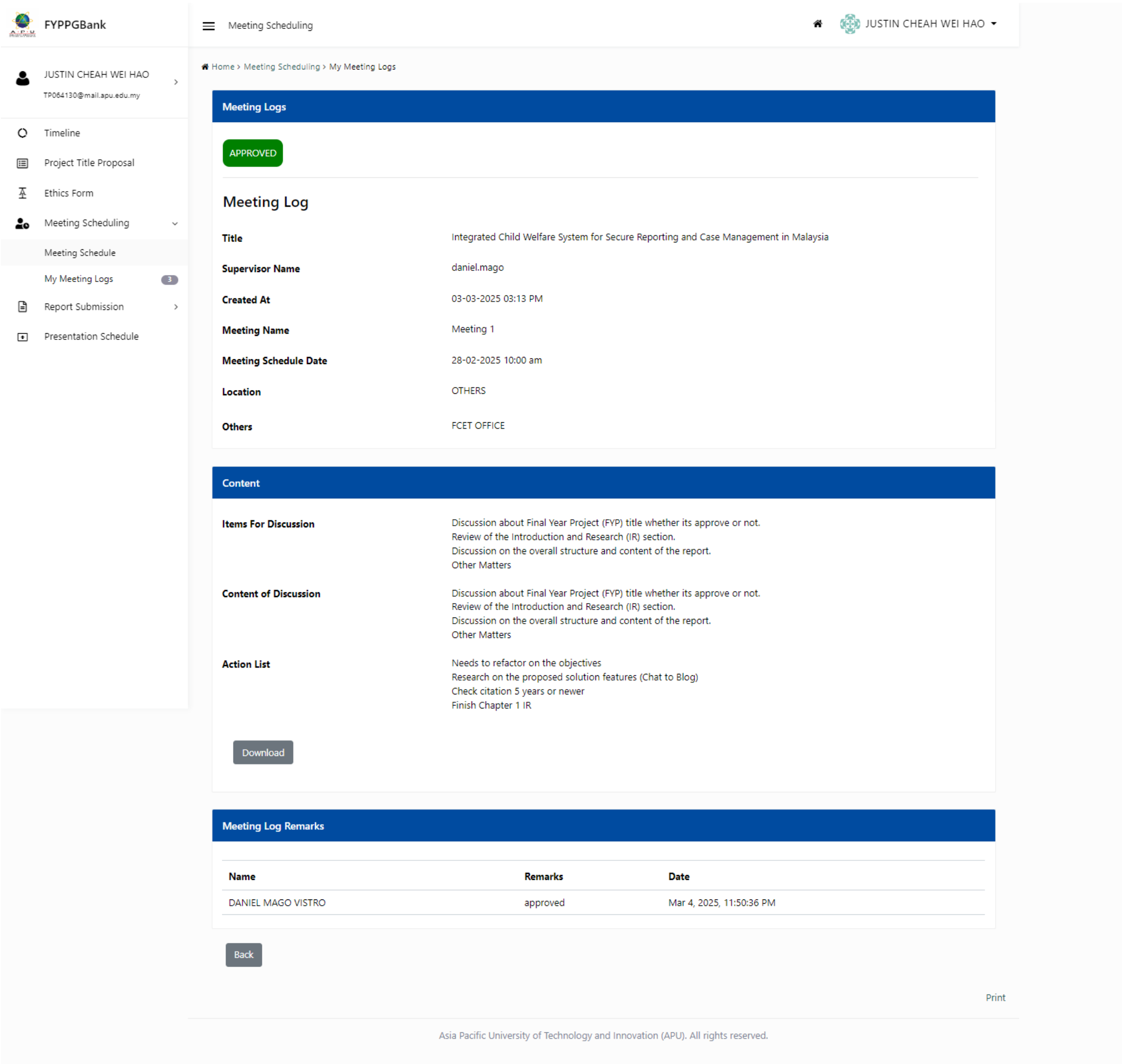


Figure 326: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_1

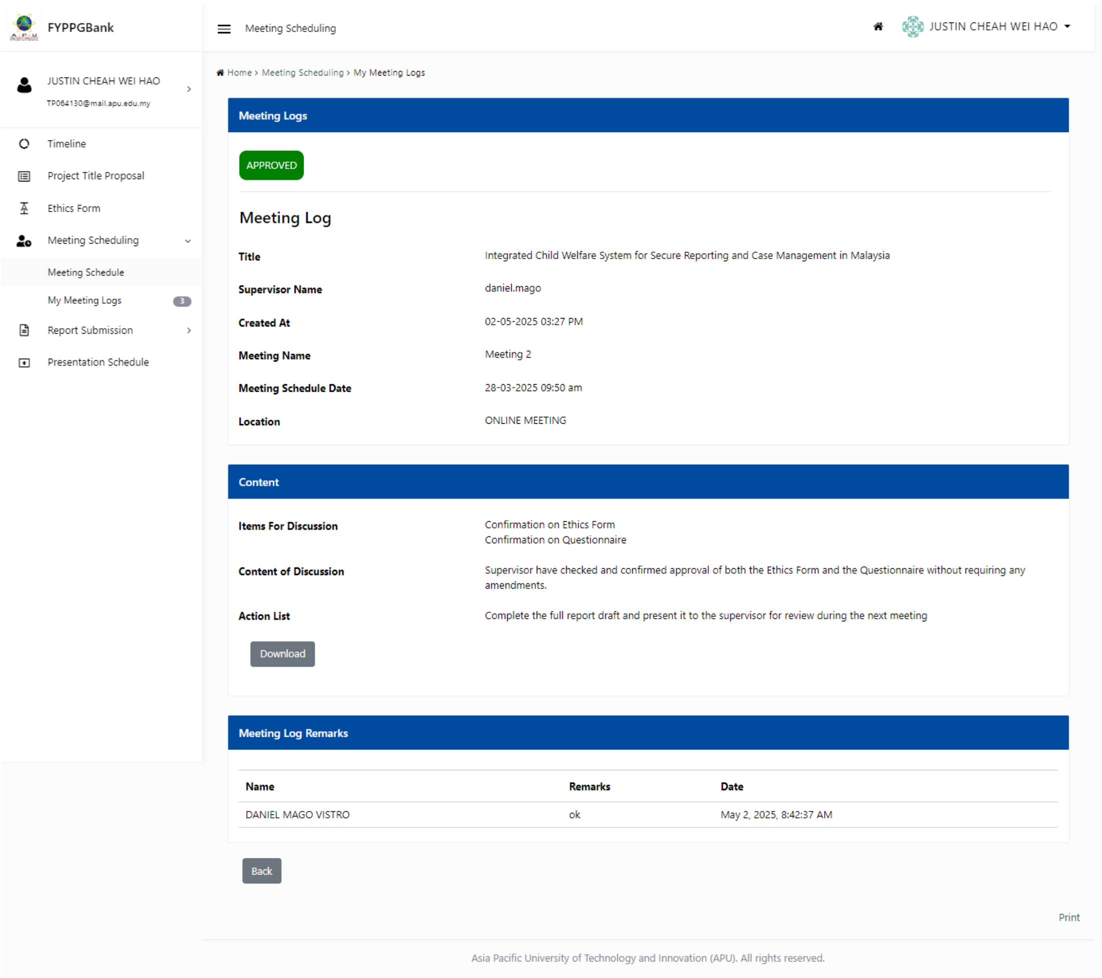


Figure 327: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_2

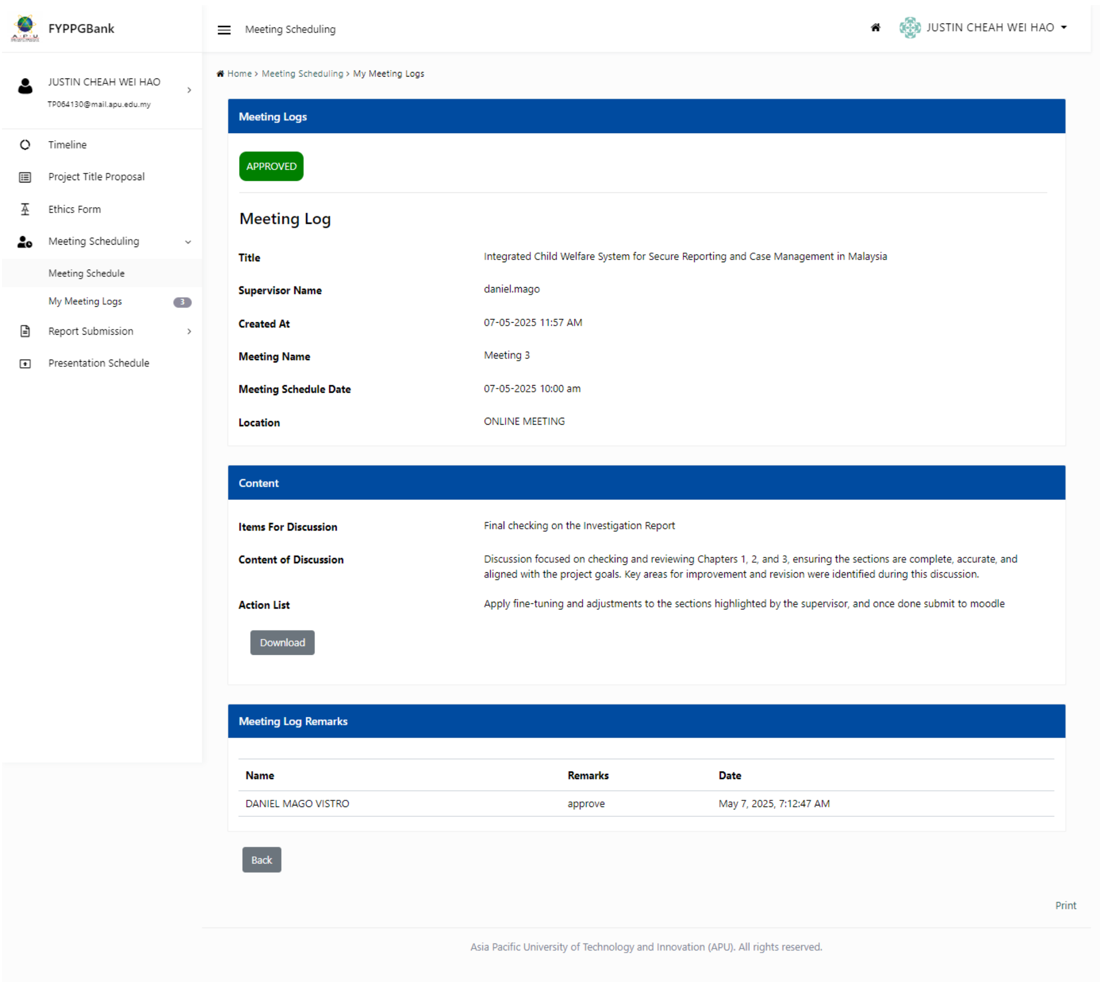


Figure 328: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_3

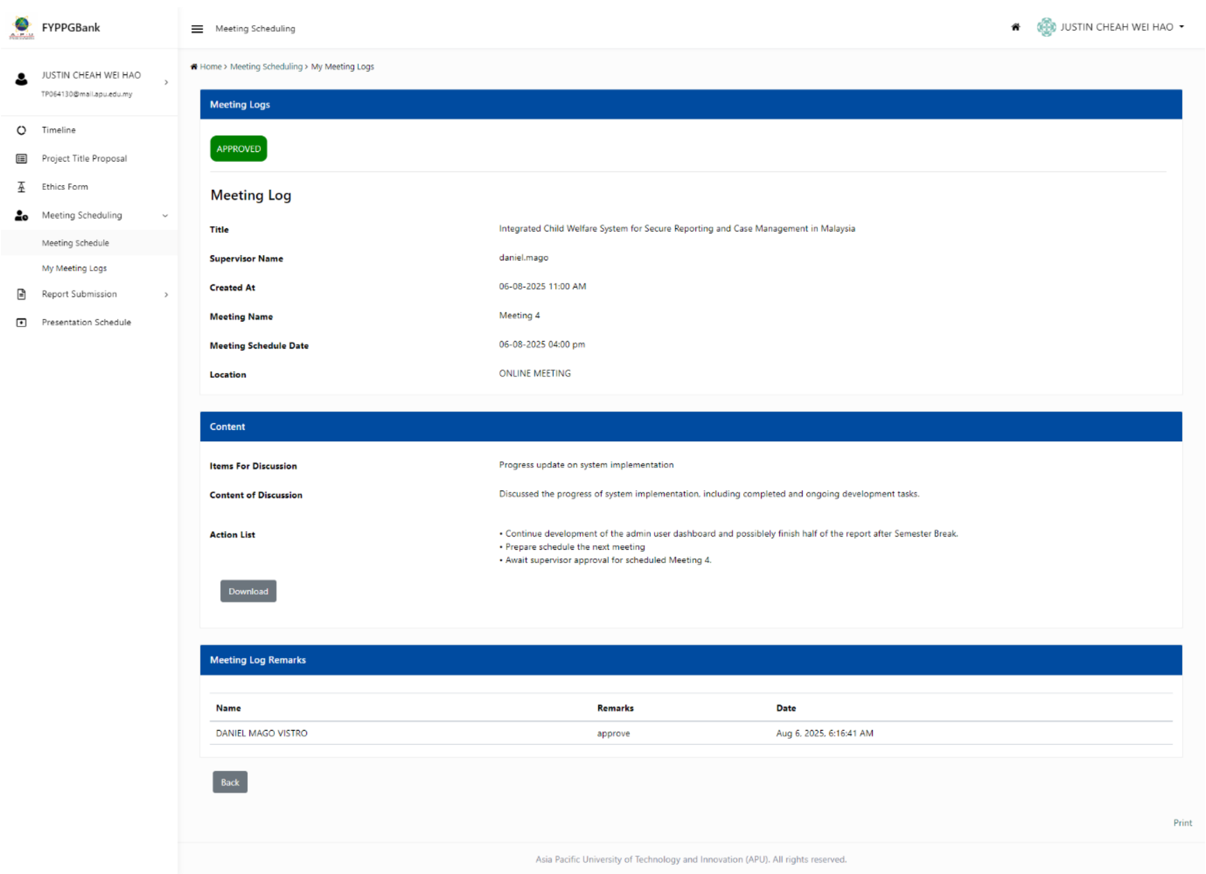


Figure 329: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_4

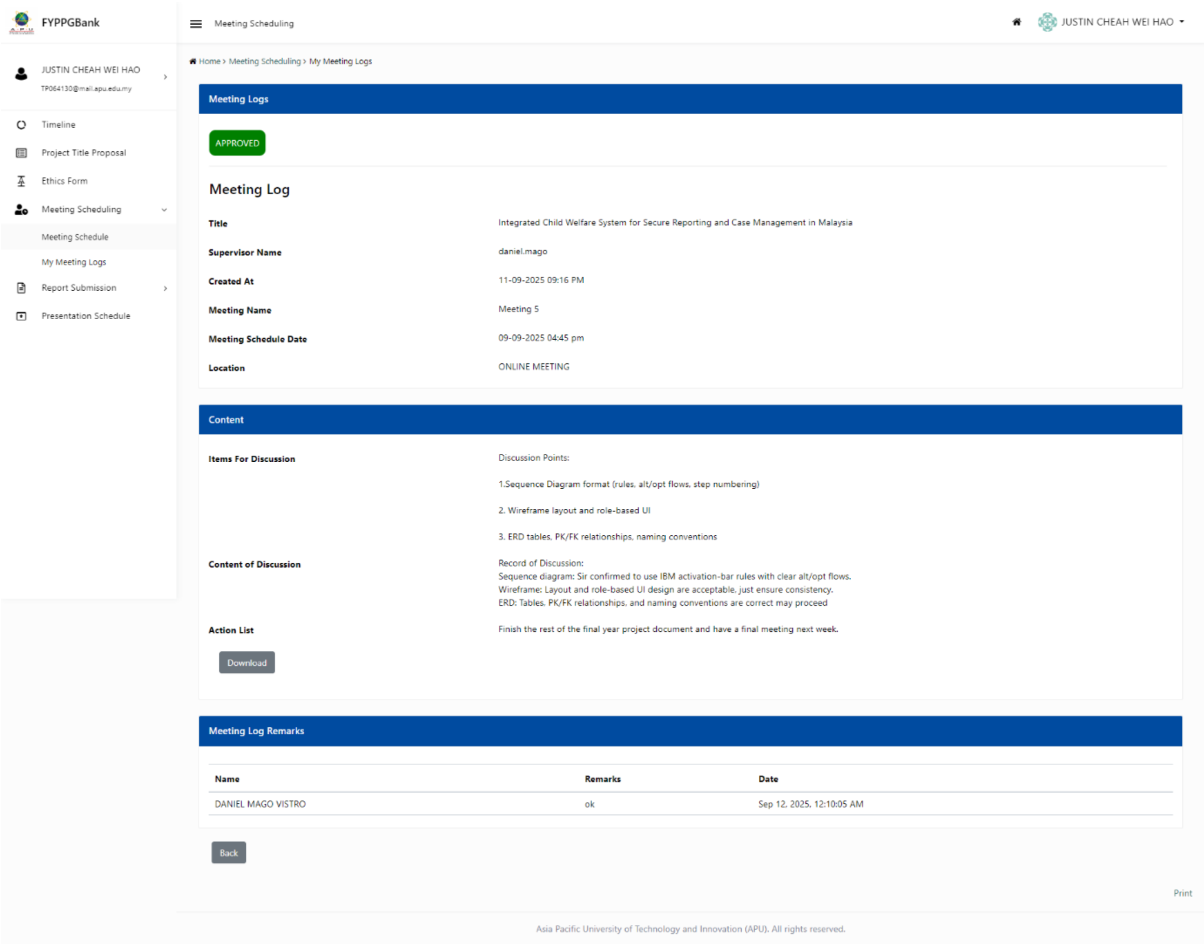


Figure 330: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_5

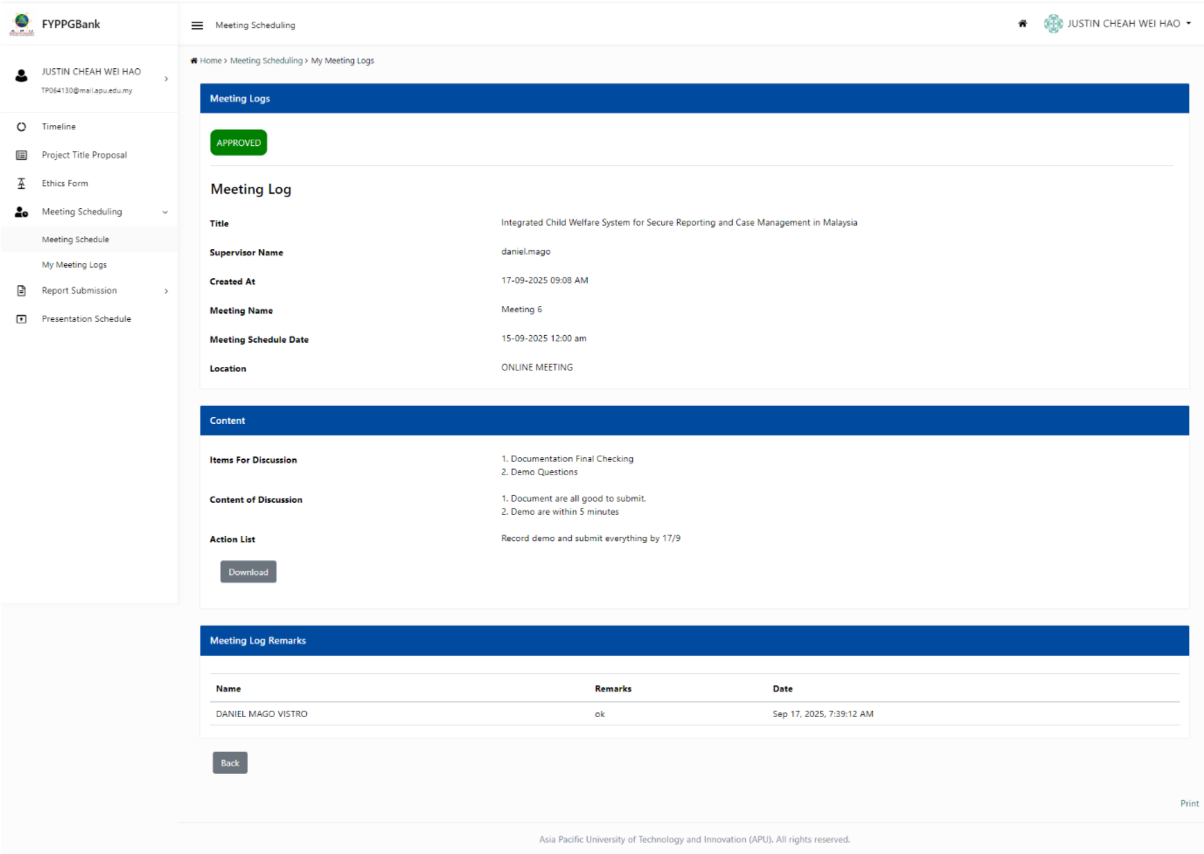


Figure 331: MR_JUSTIN_CHEAH_WEI_HAO_TP064130_APU3F2502_SE_LOG_SHEET_6

Appendix D: Poster

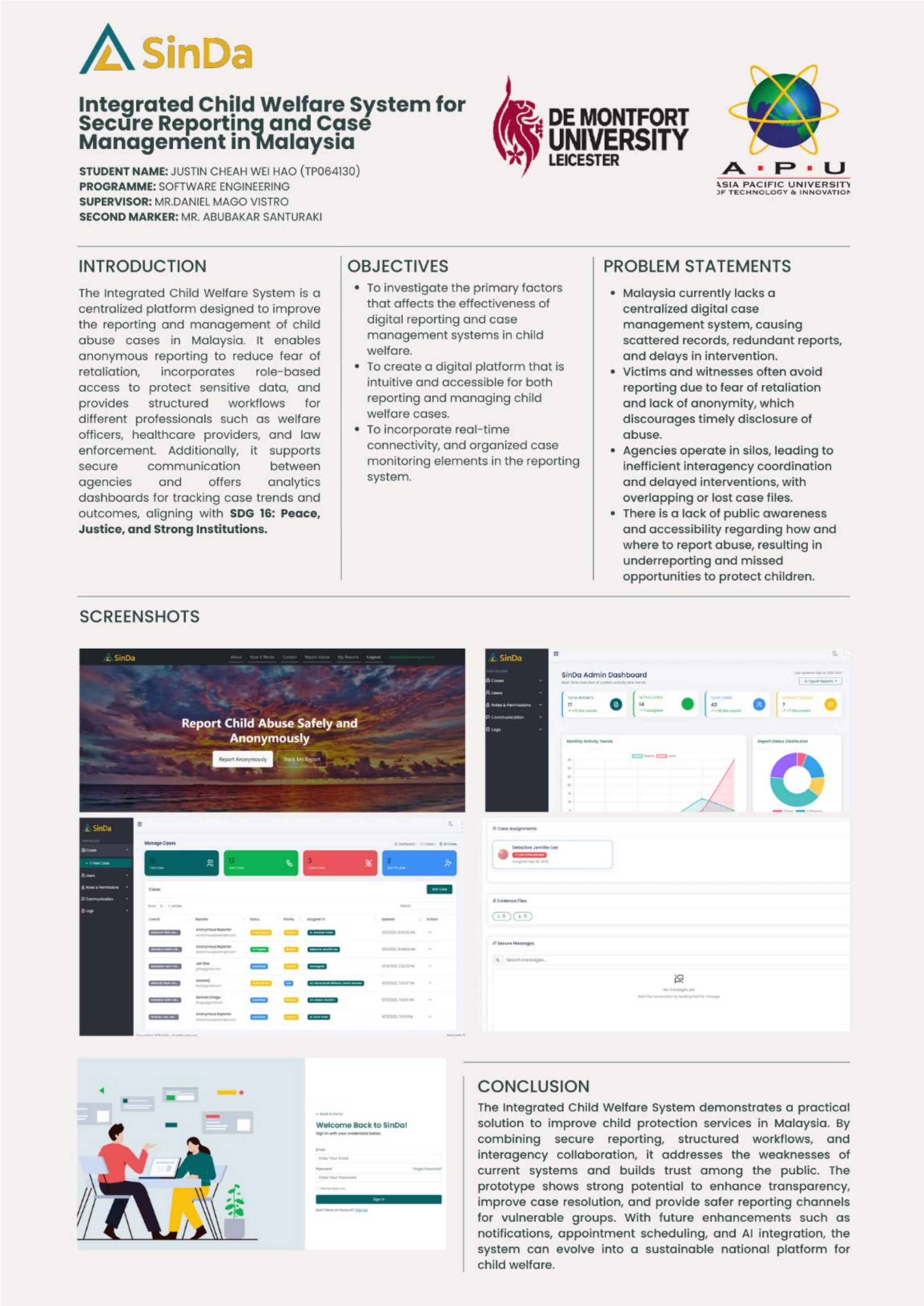


Figure 332: SinDa Poster

Appendix E: Gantt Chart

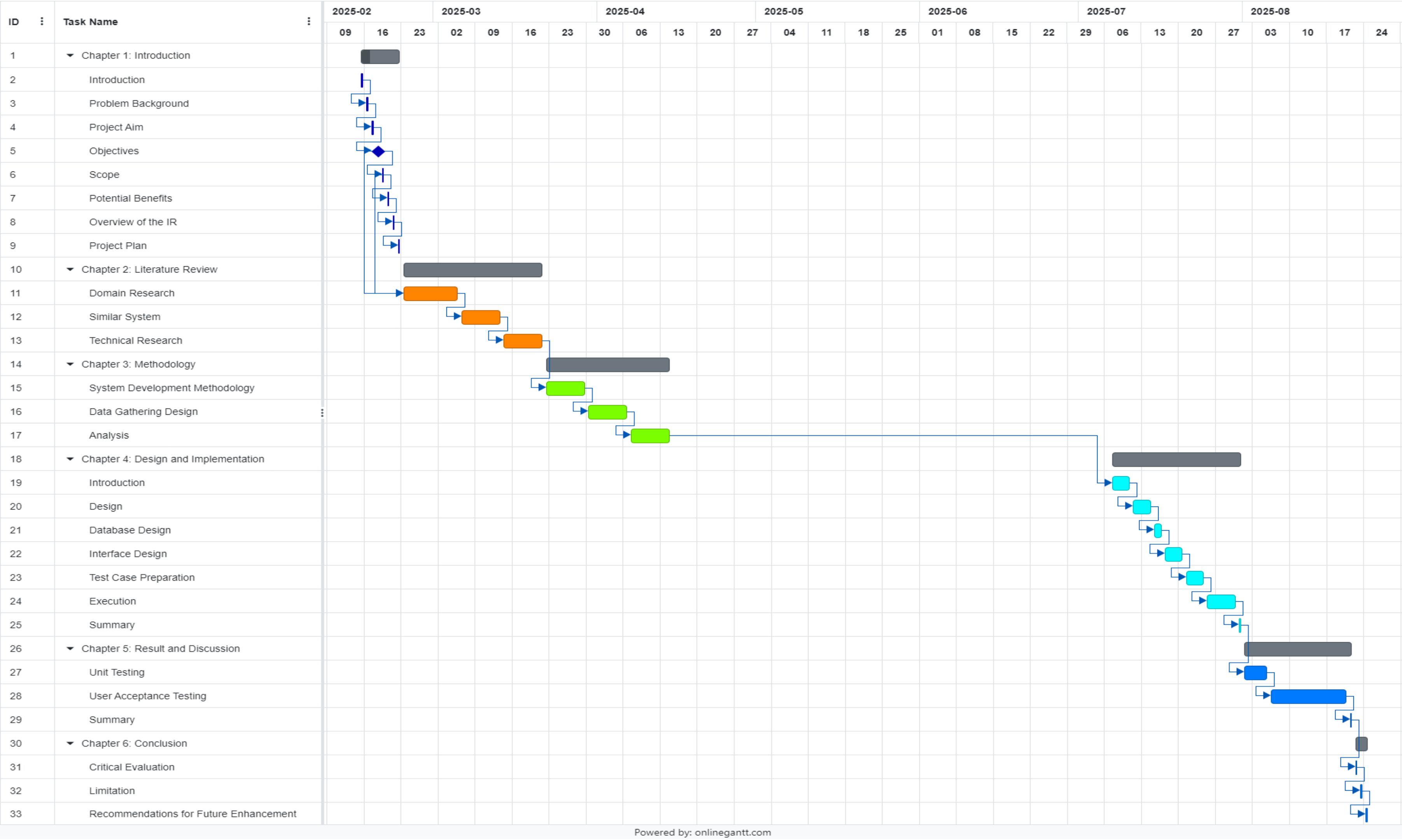


Figure 333: Gantt Chart

Appendix F: Respondence Demographic Profile

No	Gender	Age Group	Role/Occupation
1	Male (Lelaki)	45 and above	Witness or Member of the Public (Saksi / Orang Awam)
2	Male (Lelaki)	Below 25	Student
3	Male (Lelaki)	Below 25	Student
4	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
5	Female (Perempuan)	Below 25	Healthcare Professional (Doktor / Jururawat / Pegawai Sosial Perubatan)
6	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
7	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
8	Male (Lelaki)	25 - 34	Student
9	Male (Lelaki)	35 - 44	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
10	Male (Lelaki)	35 - 44	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
11	Female (Perempuan)	25 - 34	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
12	Male (Lelaki)	35 - 44	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
13	Male (Lelaki)	25 - 34	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
14	Female (Perempuan)	45 and above	Police / Law Enforcement Officer (Pegawai Polis / Penguatkuasa)
15	Female (Perempuan)	25 - 34	Social Worker (Pekerja Sosial)
16	Male (Lelaki)	35 - 44	Social Worker (Pekerja Sosial)
17	Female (Perempuan)	Below 25	Social Worker (Pekerja Sosial)

18	Female (Perempuan)	25 - 34	NGO Staff (Kakitangan NGO)
19	Male (Lelaki)	35 - 44	NGO Staff (Kakitangan NGO)
20	Prefer not to say	Below 25	NGO Staff (Kakitangan NGO)
21	Female (Perempuan)	35 - 44	Witness or Member of the Public (Saksi / Orang Awam)
22	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
23	Female (Perempuan)	25 - 34	Witness or Member of the Public (Saksi / Orang Awam)
24	Female (Perempuan)	25 - 34	Healthcare Professional (Doktor / Jururawat / Pegawai Sosial Perubatan)
25	Female (Perempuan)	35 - 44	Witness or Member of the Public (Saksi / Orang Awam)
26	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
27	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
28	Male (Lelaki)	25 - 34	Social Worker (Pekerja Sosial)
29	Male (Lelaki)	25 - 34	Student
39	Male (Lelaki)	Below 25	Witness or Member of the Public (Saksi / Orang Awam)
31	Female (Perempuan)	Below 25	Student

Table 187: Respondence Demographic Profile