Weqaa: An Intelligent Mobile Application for Real-Time Inspection of Fire Safety Equipment

Rehab Alidrisi
College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
rehabmh48@gmail.com

Ekram Feras
College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
ekramferasj@gmail.com

Shahad Aboukozzana
College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
shahadmoh232@gmail.com

Alaa Alomayri
College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
alaauniver20@gmail.com

Asmaa Alayed
College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
asayed@uqu.edu.sa (corresponding author)

Received: 11 March 2024 | Revised: 24 March 2024 and 27 March 2024 | Accepted: 30 March 2024

 DOI: https://doi.org/10.48084/etasr.7229

ABSTRACT

Fire safety is an important consideration, particularly in buildings where there are significant risks linked to a possible fire breakout. Therefore, it is crucial to implement procedures and regulations in buildings to minimize fire damage. Despite the installation of various pieces of Fire Safety Equipment (FSE), over time their effectiveness may be reduced due to factors, such as failure, damage, and insufficient maintenance. For this reason, the fire safety inspection process came to ensure the FSE availability and efficiency. Visual fire safety inspection conducted by civil defense is found to be time-consuming and inefficient, primarily due to manual procedures and difficulty in identifying defects, leading to inaccurate results and low performance. The purpose of this research is to enhance and automate fire safety inspection by implementing deep learning and computer vision techniques in a mobile application, thus addressing the challenges associated with visual inspection. Weqaa application allows the inspector to point their mobile phone camera at the fire extinguisher, then determine the condition of the extinguisher, document it, and report it to the relevant authority to quickly determine the appropriate action procedure. Interviews with expert inspectors were performed to outline the required functions of the application. The mobile application was developed using Flutter and being integrated with the detection model to permit the user to inspect fire extinguishers. Initial testing of the application has exhibited promising results, with inspectors noting its competence in detecting violations and improving inspection processes. The use of the particular application enabled the inspectors to perform the required functions faster, more accurately, and with fewer errors compared to the visual inspection deployment, indicating the application's effectiveness in detecting violations.

Keywords-fire safety inspection; Fire Safety Equipment (FSE); mobile application; visual inspection; deep learning; computer vision

I. INTRODUCTION

Fire safety has always been a priority, particularly after the rise in fire accidents. Buildings are the most frequent locations at risk of fires [1]. As reported by Saudi civil defense statistics, the majority of fire-related fatalities and injuries occurred in residential buildings, with 1480 deaths and injuries, a percentage of 75.6% [2]. Consequently, certain procedures and regulations should be implemented in buildings to reduce fire damage. As a procedure, a complete fire safety system should be installed in buildings, utilizing devices, such as smoke detectors, sprinklers, fire extinguishers, and fire doors [3]. This equipment provides outstanding fire safety and loss prevention measures. An example of such equipment is the fire extinguisher, which was the only firefighting item required to put out a fire in around 80% of all fire occurrences [4]. However, as time passes, this equipment is subject to failure, damage, and the subsequent lack of functionality [3]. As an outcome, regular maintenance is necessary. Therefore, the fire safety inspection process comes to ensure the availability and effectiveness of FSE [5]. In Saudi Arabia, such an operation is carried out in the facilities in compliance with the Saudi Building Code for Fire Protection (SBC 801) [6], and those who violate the rules encounter serious consequences since insufficient maintenance can cause significant losses. Approximately 150 people lost their lives and were injured in one of Saudi Arabia's worst hospital fires, which occurred in Jazan. According to information released by the Saudi Ministry of Health, a reason for these losses was the inadequate maintenance of safety equipment [7].

Fire safety inspection is performed by civil defense. A qualified inspector visits a location, visually inspects every piece of equipment, records any violations, and compiles a final report that is then sent to the safety manager to take appropriate actions. This approach requires experience and takes time. A survey was distributed to expert safety inspectors in the Civil Defense in Saudi Arabia and 74 responses were received [8]. Most of the respondents believe or tend to believe that the process of inspecting FSE using the current practices is difficult (Figure 1). Several reasons would contribute to this belief: Firstly, fire safety inspections are usually time-consuming. Secondly, issuing violations is a non-automated process. Thirdly, a lack of experience in the inspection process would lead to inaccurate results, errors, and low performance. Furthermore, the respondents pointed out the difficulty of following up on the updated safety regulations. Thus, due to the low level of automation and the difficulty in discovering defects, defect inspection in FSE continues to be problematic.

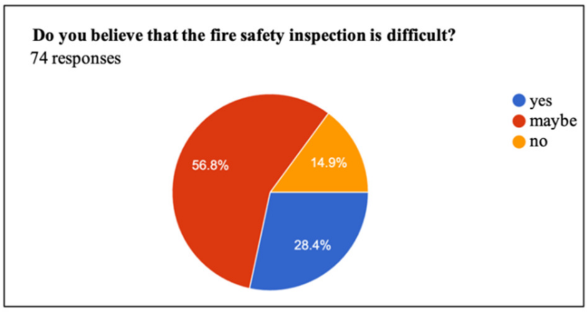


Fig. 1. Percentage of respondents finding inspection of FSE difficult with the current practices.

The era of technology has led to the digitization of many fields, which in turn improves productivity and speeds up manual tasks. Meanwhile, Artificial Intelligence (AI) has enabled applications to improve job accuracy and decrease human intervention [9]. Several mobile applications in various fields have been developed utilizing machine learning [10], deep learning, and computer vision [11]. The main goal of the current work is to facilitate, automate and improve the process of fire safety inspection by developing an intelligent mobile application called Weqaa. The proposed application gives the inspector the opportunity to point their mobile phone camera at the fire extinguisher, then determine the condition of the extinguisher by utilizing deep learning and computer vision techniques, document it, and report it to the relevant authority to quickly determine the appropriate action procedure.

The existing mobile applications in Saudi Arabic for reporting safety violations are Madani [12] and Salamti [13]. Although these applications have certain comparable functionalities, such as reporting safety violations, the Weqaa application differs in many ways. Weqaa introduces the innovative function of smart inspection, which provides intelligent violation inspection while avoiding receiving an incorrect violation owing to user inexperience. This function can exceed the capabilities of both Madani and Salamti, which need users to manually submit violation information. The inspection in Weqaa can be performed in real-time once the user points their mobile phone camera at the fire extinguisher. This feature is not available in the other existing applications. Furthermore, Weqaa provides excellent documentation of violations through images, a function that provides a better understanding of safety violations using visual evidence. This capability is noticeably absent from Salamti. Also, Weqaa's features extend to the automatic generation of inspection form, resulting in a quicker and more efficient operation that Madani and Salamti lack in their manual procedures. Table I compares the proposed application (Weqaa) with these applications.

TABLE I. COMPARISON BETWEEN WEQAA AND CURRENT APPLICATIONS

Features	Salamti [13]	Madani [12]	Weqaa
Reporting a safety violation	•	•	•
Intelligent inspection			•
Real-time inspection			•
Inspect without prior knowledge			•
Automatic violation report filling and generation			•
Document the violation by image	•		•
Better user experience		•	•

The current research aims to enhance the process of fire safety inspection by integrating AI into a mobile application. The application targets Hajj season as a case study, focusing on its specific types of facilities called camps. These camps are numerous whereas inspections need to be conducted in a short period of time.

II. METHODOLOGY

The Weqaa development process is illustrated in Figure 2. This process starts by defining the main functionalities of the application through conducting interviews with experts and exploring existing systems. After that, Weqaa was designed, developed, and integrated with the AI model. Finally, Weqaa was tested to ensure its effectiveness and usability. The following subsections clarify the discussed process.

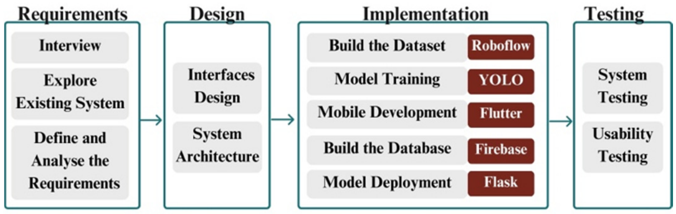


Fig. 2. The development process of Weqaa.

A. System Requirements

The first step was to explore and go in-depth in the procedure of fire safety inspection. An interview with two expert inspectors was conducted. According to the interviewees, the inspection process usually involves inspectors and safety managers, who periodically visit the inspection location, examine the equipment, and complete the report demanded. Safety managers are responsible of reviewing those reports and deciding on the appropriate procedure. After analyzing the inspection process that was explained and clarified by the interviewees, the functionalities of Weqaa were specified for both actors as portrayed in the use case diagram in Figure 3.

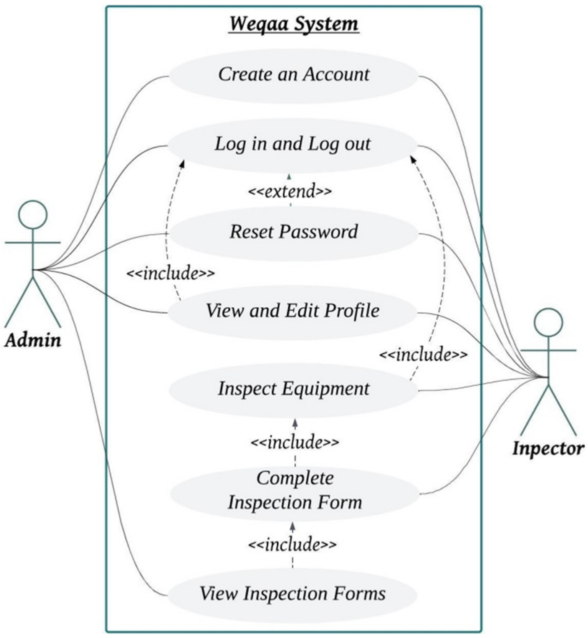


Fig. 3. The use case diagram for Weqaa application.

As depicted in Figure 3, there are common use cases for both actors, inspector and admin, including Create an Account, Log in and Log out, Reset Password, and View and Edit Profile. The inspector use case, Inspect Equipment is where the inspection is performed in real-time by pointing the mobile phone camera to the equipment, detect its defect, retrieve camp location, and finally store the violations -if any- in the database. In the inspector use case, Complete Inspection Form will display the detected violations and camp information and will allow the inspector to check out the completeness of the report. In the admin use case, the View Inspection Form allows the admin to view the completed inspection forms. To view a specific report, the admin can look up the camp and specify the year of inspection. Violation statistics are also included in the inspection form. The inspection forms can be converted to a pdf file format for sharing and printing.

B. System Architecture

Weqaa architecture is observed in Figure 4. The client-server, where the client remotely connects to server through the internet, was employed [14]. The client must have the application installed on their mobile phone. Internet access is essential to access the services related to the database or perform the inspection.

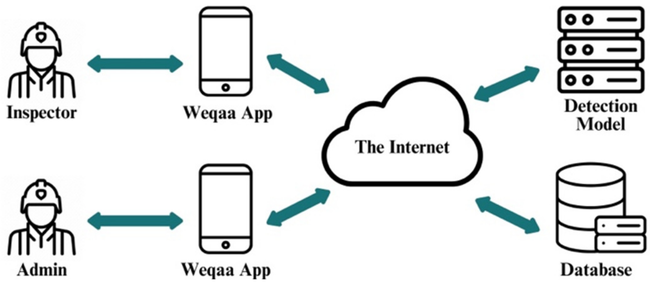


Fig. 4. Weqaa architecture.

C. Database

The database structure of the Weqaa application was designed deploying a Class Diagram in Unified Modeling Language (UML) Notation. The class diagram serves as a graphical representation illustrating the structural aspects of a database. Figure 5 shows the class diagram for the proposed application, which consists of 5 classes including Inspector and Admin users, Camp, InspectionForm, and Violation. Then Firebase [15] was put into service to build the database since it is a good choice for real-time applications.

D. Mobile Development and Implementaion

The mobile application was developed utilizing Flutter [16], a recent and effective mobile framework to build iOS and android applications from a single code base. Google API [17] was used to obtain camp location information.

E. Development of the AI Inspection Model

The intelligent inspection relies on a deep learning detection model that inspects specific types of fire extinguisher. Fire extinguisher images were collected, annotated, and augmented to create a dataset of 7,633 images with 16,092 labeled instances, and the model was developed and evaluated in [18]. The model inspects the extinguisher based on five criteria in accordance with SBC. These criteria are: checking the gauge indicator towards green region, verifying that the body is not rust, and hose, pin and expire date exist. Using You Only Look Once (YOLOv8n), the model achieved 87.2% of mAP0.5. The model size was 6.3 MB in Pytorch format. This model has been built to work in real-time and does not require any action from the user to start inspection. The model output consists of six classes: 'gauge_good', 'gauge_bad', 'rust', 'hose_exist', 'pin_exist', and 'expire_date'. The application shall process this output to display in the interface.

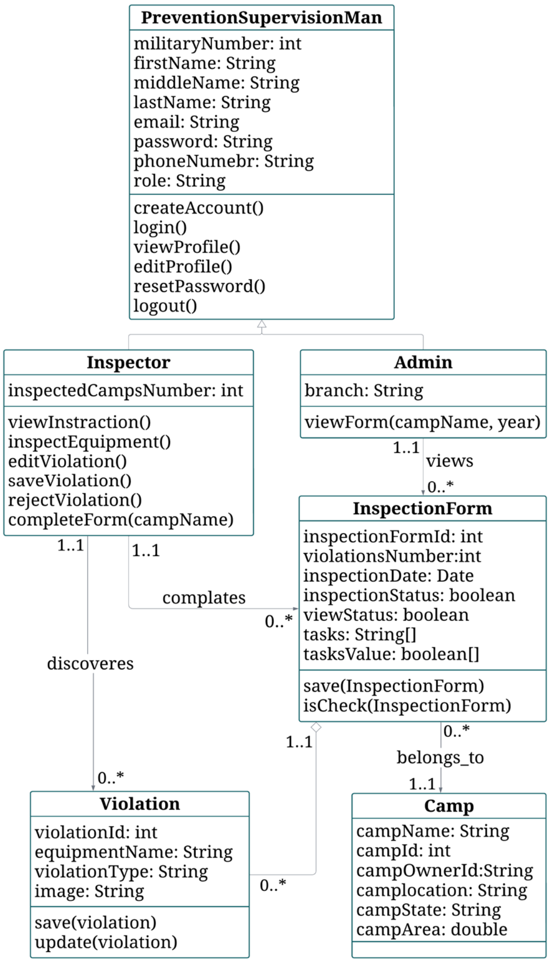


Fig. 5. UML Class diagram of Weqaa database.

F. *Model Deployment*

Following the completion of the mobile application, the integration process with the inspection model began. This integration posed some challenges because current flutter packages do not support YOLOv8. As a result, it was decided to develop a separate backend using Flask, a lightweight web framework in Python [19], where the model could operate. Then, a connection with the Weqaa application was established implementing an Application Programming Interface (API). The real-time detection process is carried out in several steps. First, the camera feed has to be processed by converting each frame into a format, such as the JPG format, which could be sent over the network. Next, an http request that contains the image is created and sent to the backend server. On the backend side, the system receives the request, makes the inference, and responds with the results in JavaScript Object Notation (JSON) format. Finally, the Flutter application receives, processes, and presents the results.

G. *Testing*

This phase ensures that the application functions properly. Therefore, several test types were applied. First, the unit testing was performed, in which each function was individually tested to determine whether it worked as expected. This utility allowed the identification and removal of bugs at an earlier stage. An integration test was also conducted to assess how the various components of the system interact with one another and work together. Furthermore, the application was evaluated for its usability and general performance through user testing. Some target users for both inspector and admin tested the application. Certain tasks were assigned to each participant who was observed during this test session. Total time and number of errors for each task were recorded. Finally, the participants were asked to fill out a questionnaire regarding the application usability, and overall experience deploying a 5-point Likert scale to answer the questions. Also, open questions for feedback and suggestions were asked.

III. RESULTS AND DISCUSSION

In this section, the results of developing Weqaa interfaces and testing are discussed. The evaluation results of the inspection model were addressed in our previous work [18].

A. *User Interface for Weqaa*

The main interfaces of Weqaa are described in this section. Since the target users of Weqaa are Arabic native speakers, the language of the developed interfaces was Arabic. The interfaces were designed with design principles in mind, including visibility, feedback, constraints, consistency, and affordance [20]. These concepts helped to create interfaces that are intuitive, user-friendly, and efficient.

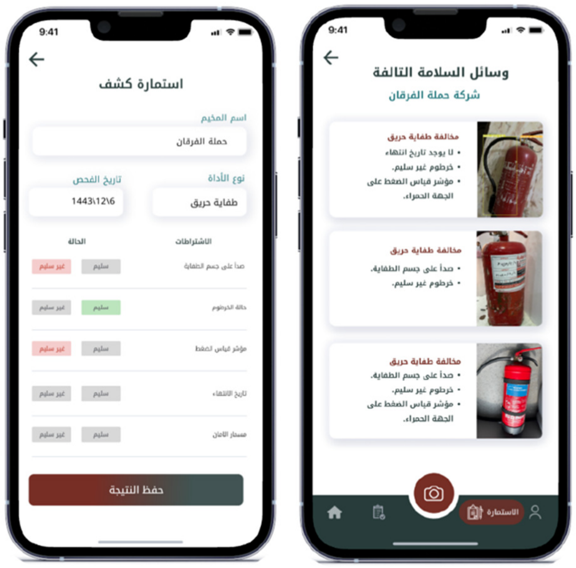
As displayed in Figure 6(a), upon launching the application, users are greeted with the application's logo followed by log in interface presenting options to create a new account, or log in with existing credentials. If the user logs in as an inspector, all their assigned inspection tasks will be presented as noticed in Figure 6(b).

Fig. 6. Weqaa interfaces: (a) Login interface, (b) inspector home interface.

When the camera button is pressed, the inspector can conduct inspection by pointing the camera at the equipment as detected in Figure 7(a), and see at the bottom part of the screen if it is defective or not in real time as observed in Figure 7(b). Then, when the documentation button is pressed, the inspection results will be presented for editing and confirmation, as spotted in Figure 8(a). Later, all violations are stored and displayed in the violation interface, as shown in Figure 8(b).

(a) (b)

Fig. 7. Weqaa interfaces: (a) Camera pointed at equipment, (b) detection results.



(a) (b)

Fig. 8. Weqaa interfaces: (a) Confirm inspection result, (b) all defective equipment at one location.

The inspectors can conveniently view all the camps awaiting inspection as exhibited in Figure 9(a). Weqaa allows the inspectors to smoothly move on to fill out inspection forms for each inspected camp as manifested in Figures 9(b) and 10. Each safety requirement on the inspection form must be clicked by the inspector to verify compliance with the sub-requirements for each camp – whether they are satisfied or not. After that, the inspector can see the status of the requirements, allowing them to proceed to fill out the next requirement, and so on, until the inspection form is complete as illustrated in Figure 10(b).

(a) (b)

Fig. 9. Weqaa interfaces: (a) All the camps awaiting inspection, (b) inspection form sections.

(a) (b)

Fig. 10. Weqaa interfaces: (a) Inspection form, (b) fill out inspection form.

Through the admin interfaces, admins can monitor the inspections conducted by inspectors, review violations, and track inspection statuses for each inspected camp. Admins can access and view inspection forms for the inspected location through a dedicated feature. The admin can enter the camp name and select the inspection year as showcased in Figure 11(a) in order to view the inspection form and approve it as spotted in Figure 11(b).

(a) (b)

Fig. 11. Weqaa Interfaces: (a) Review inspection forms, (b) approval inspection forms.

(a) (b)

Fig. 12. Weqaa Interfaces: (a) Violations statistics, (b) detected violations.

The statistics regarding violations can be presented to the admin with an option to view the inspection forms in a PDF format. This can be done by clicking the icon on the top right corner of Figure 12(a), which provides easy access to and sharing of inspection data with others. All the details of the violations, along with their images, can be displayed as shown in Figure 12(b).

B. Testing Results

The application was tested by ten users, including eight inspectors and two administrators. Table II depicts the demographics (age, working experience, qualification, and experience in technology) of the users. Only one among them was an expert in using technology, while the remaining participants believed they were at a medium level. Various tasks were assigned to each user according to their roles. Inspectors were asked to perform tasks that include sign-up, login, inspecting equipment, completing inspection forms, and editing profiles. Admins, on the other hand, were asked to perform tasks that involve viewing violations and inspection forms for each facility. The performance of the users was recorded during executing these tasks on their devices. The metrics employed to evaluate the users' performance were the average time taken to complete the task and error rate for each task. Considering the positive evaluations in Table III, it is evident that the tasks were accurately completed, with few errors, and in a low average completion time. This was particularly notable in real-time functions, such as inspecting equipment, where the model effectively detected violations with an excellent success rate.

TABLE II. PARTICIPANT DEMOGRAPHICS

Item	Category	Percentage of responses
Age	20 - 25	10%
	30 - 35	50%
	40 - 45	40%
Working experience (years)	1 or less	10%
	2 - 5	30%
	6 - 10	30%
	11- 15	10%
	16 - 20	10%
	25 or more	10%
Qualification	High School	0
	Bachelor	100%
	Master	0
Experience in using technology	Beqinner	0
	Medium	90%
	Expert	10%

TABLE III. INSPECTOR AND ADMIN TASK RESULTS

Task	Task success	Average error rate	Average time
Sign up	Completed	0	01:12 min
Log in	Completed	0	00:18 sec
Inspect the equipment	Completed	0.002%	00:25 sec
Complete inspection form	Completed	0.008%	01:18 min
Edit Profile	Completed	0	00:19 sec
View inspection form	Completed	0	00:22 sec
View violations	Completed	0	00:11 sec

In the post test satisfaction survey, users' experiences varied. While 80% of the participants found the application helpful and easy to navigate, 20% had difficulties understanding how to use it effectively to inspect equipment.

However, 70% of the participants thought that the application simplified the implementation of fire safety inspection services. Participants had varying opinions about the clarity and design of the interface. While 70% expressed satisfaction, the rest remained neutral. The neutrality could be due to factors such as individual preferences. Additionally, participants were asked to provide suggestions for improvement. One major suggestion was to expand the application's detection model to include the ability to detect the type of extinguisher (carbon dioxide gas extinguishers, water extinguishers, etc.).

Generally, participants reacted positively to the Weqaa application, noting its ability to facilitate the inspection process and improve its quality. These findings are encouraging and suggest that the application is well-liked by the users.

IV. CONCLUSION

This research presents a mobile application able to intelligently inspect fire safety equipment and report violations accurately, quickly, and without experience. A deep learning model was utilized in this application, which is the key feature among the existing systems. The mobile interface was developed to look friendly and easy to use. Flutter was deployed to build this application, allowing it to operate on Android and iOS. Several test cases were conducted to ensure the application efficiency. The results of usability testing show the level of satisfaction from the target users. The former express the usefulness of the application which facilitates the digital transformation of fire safety and accelerates the process.

The Weqaa application significantly contributes to the field of fire safety inspection. Wegaa is an intelligent application that integrates our artificial intelligence model to speed up the inspection process while achieving high-accuracy results. Therefore, Weqaa would minimize the need for specialized personnel to conduct inspections. Unlike similar applications, filling out violation information is automatic, which would reduce human intervention, thereby making the process simpler, more efficient, and less prone to errors.

Future improvements include the following features that were suggested by testers during the application testing:

- Saving all inspected extinguisher information, not just the violations, in order to track all of the equipment in the facilities.
- Improving the detection model to provide more specific details regarding the different types of extinguishers.

REFERENCES

[1] D. Rasbash, G. Ramachandran, B. Kandola, J. Watts, and M. Law, *Evaluation of Fire Safety*, 1st ed. Hoboken, N.J, USA: Wiley, 2004.

[2] "Annual Statistical Report," Civil Defense Directorate, Saudi Arabia, 2020.

[3] D. E. Della-Giustina, *Fire Safety Management Handbook*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2014.

[4] "New survey highlights the importance of fire extinguishers," *SecurityWorldMarket.com*, Jun. 19, 2022. https://www.securityworldmarket.com/me/News/Business-News/survey-finds-93-of-fires-put-out-with-portable-extinguishers.

[5] D. Dieken, "Inspection, testing and maintenance of fire protection systems at industrial plants," *Process Safety Progress*, vol. 18, no. 3, pp. 151–155, 1999, https://doi.org/10.1002/prs.680180306.

[6] Saudi Building Code National Committee, *Saudi Fire Protection Code SBC 801 - CR*. Saudi Arabia, 2018.

[7] "Statement on JGH Fire Investigations," *Ministry of Health in Saudi Arabia*. https://www.moh.gov.sa/Ministry/MediaCenter/News/Pages/News-2016-01-13-001.aspx.

[8] "Questionnaire about the Process of FSE Inspection," *Google Docs*. https://forms.gle/FfwETkfbhc5DxGJ18

[9] R. S. T. Lee, *Artificial Intelligence in Daily Life*. Springer, 2020.

[10] F. M. U. Baran, L. S. A. Alzughaybi, M. A. S. Bajafar, M. N. M. Alsaedi, T. F. H. Serdar, and O. M. N. Mirza, "Etiqa'a: An Android Mobile Application for Monitoring Teen's Private Messages on WhatsApp to Detect Harmful/Inappropriate Words in Arabic using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12012–12019, Dec. 2023, https://doi.org/10.48084/etasr.6174.

[11] L. Loyani and D. Machuve, "A Deep Learning-based Mobile Application for Segmenting Tuta Absoluta's Damage on Tomato Plants," *Engineering, Technology & Applied Science Research*, vol. 11, no. 5, pp. 7730–7737, Oct. 2021, https://doi.org/10.48084/etasr.4355.

[12] "Madani Application." Saudi Civil Defense, [Online]. Available: https://apps.apple.com/sa/app%D9%85%D8%AF%D9%86%D9%8A/id1596908770.

[13] "Salamti Application." Saudi Civil Defense, [Online]. Available: https://my.998.gov.sa/app/salamati.

[14] I. Sommerville, *Software Engineering*, 9th ed. Boston, MA, USA: Pearson, 2011.

[15] "Firebase | Google's Mobile and Web App Development Platform." Google, [Online]. Available: https://firebase.google.com/.

[16] "Flutter - Build apps for any screen." Google, [Online]. Available: https://flutter.dev/.

[17] "Google Maps Platform." Google, [Online]. Available: https://developers.google.com/maps.

[18] A. Alayed, R. Alidrisi, E. Feras, S. Aboukozzana, and A. Alomayri, "Real-Time Inspection of Fire Safety Equipment using Computer Vision and Deep Learning," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13290–13298, Apr. 2024, https://doi.org/10.48084/etasr.6753.

[19] "Welcome to Flask — Flask Documentation (3.0.x)," *Flask*. https://flask.palletsprojects.com/en/3.0.x/.

[20] H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 6th ed. Indianapolis, IN, USA: Wiley, 2023.