CONSTRUCTION SITE SAFETY HELMET DETECTION IN MOBILE APPLICATION

ANG SUZANNE

BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING) WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : ANG SUZANNE

Date of Birth

Title : CONSTRUCTION SITE SAFETY HELMET DETECTION IN MOBILE

APPLICATION

Academic Session : 2022/2023

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

____
(Student's Signature)

____
(Supervisor's Signature)

____
New IC/Passport Number
Date: 8th February 2023

DR ANIS FARIHAN BINTI MAT RAFFEI
____
Name of Supervisor
Date: 8th February 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR’S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) with Honours.

(Supervisor’s Signature)

Full Name : DR ANIS FARIHAN BINTI MAT RAFFEI
Position : SUPERVISOR
Date : 8th February 2023

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student’s Signature)

Full Name : ANG SUZANNE

ID Number : CB19103

Date : 8th February 2023

CONSTRUCTION SITE SAFETY HELMET DETECTION IN MOBILE APPLICATION

ANG SUZANNE

Thesis submitted in fulfillment of the requirements
for the award of the
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

First and foremost, I like to praise and thank the Almighty God for the countless blessing and good health, so that I am able to accomplish the thesis. Words cannot express my gratitude to my supervisor, Dr. Anis Farihan Binti Mat Raffei for her constant support and feedback which has helped me complete my project effectively and easily on time.

I am also grateful to all of my friends who have always been there by my side. I would love to specifically thank Desmond Ling, who has provided me with constructive solutions and emotional support throughout the time of development of this project. Without each of their moral support, I wouldn’t have completed the project in a short amount of time.

Last but not least, I must express my profound gratitude to my parents for their faith in me has maintained my spirits and motivation strong throughout this process.

ABSTRAK

Industri pembinaan mempunyai kadar kemalangan yang sangat tinggi berbanding dengan industri lain. Walau bagaimanapun, kebanyakan kecederaan dan kemalangan ini boleh dielakkan jika pekerja memakai peralatan perlindungan diri (PPE) yang sesuai seperti topi keselamatan, jaket, but, atau cermin mata keselamatan. Ketiadaan PPE telah dikenal pasti sebagai isu penting yang menjejaskan keselamatan tapak pembinaan dalam beberapa kajian penyelidikan. Setiap peralatan dalam satu set PPE memainkan peranan penting dalam meminimumkan risiko kecederaan. Oleh itu, pekerja di tapak mesti mengambil kira langkah berjaga-jaga ini apabila bekerja dalam persekitaran yang berbahaya. Tidak mustahil bagi majikan untuk mengawasi pekerja mereka 24/7 yang kebanyakannya terletak di kawasan terpencil. Untuk mengatasi batasan ini, penggunaan teknologi bantuan penglihatan komputer menjadi semakin biasa. Oleh itu, projek ini bertujuan untuk membangunkan aplikasi untuk pengesanan topi keselamatan tapak pembinaan menggunakan teknologi berasaskan penglihatan komputer. Aplikasi ini menerima makluman mengenai ketidakpatuhan topi keselamatan terhadap pekerja yang dikesan menggunakan teknologi ini dan menghantar makluman kepada majikan untuk tindakan selanjutnya diambil.

ABSTRACT

The construction industry has had exceptionally high accident rates compared to other industries. However, most of these injuries and accidents can be avoided if workers wore appropriate personal protective equipment (PPE) like helmets, vests, boots, or safety glasses. The absence of PPE has been identified as a significant issue affecting construction site safety in several research studies. Each piece of equipment in a set of PPE plays a vital role in minimizing the risk of injuries. Therefore, onsite workers must take into consideration this precaution when working in dangerous environments. It is impossible for employers to keep an eye on their workers 24/7 that are most of the time located in remote areas. To overcome this limitation, the use of computer vision-assisted technology is becoming increasingly common. Therefore, this project is aimed at developing an application which focuses on construction site safety helmet detection using computer vision-based technology. The application receives alerts on helmet non-compliance on workers detected using the technology and sends the alerts to the employer for further actions to be taken.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

SRS	Software Requirements Specification
SDD	Software Design Document
UAT	User Acceptance Testing
ConSite	Construction Site Safety Helmet Detection Application

CHAPTER 1

INTRODUCTION

1.1 Introduction

A construction site is a plot of land in which all operations associated with the placement of a structure occur(*What Is a Construction Site? - LetsBuild*, 2019). A single construction site faces a multitude of obstacles, all of which are tied to the environmental conditions, rules, and the structure type that is being built. For that reason, construction is one of the highly hazardous industries that encompasses a wide variety of operations such as building, altering, and/or repairing structures. This statement is proven to be true as according to the major findings in the latest statistical data of the International Labour Organization (ILO) on occupational accidents and diseases, the construction sector has a vastly disproportionate rate of recorded accidents (*World Statistic*, 2011).

Construction workers are often exposed to tasks that involve serious risks, including falling from roofs, unprotected machinery, getting injured by large construction equipment, asbestos, electrocutions, and silica dust. In Malaysia, the number of accidents in the construction sector was 3958 cases as of 2020, which led to the deaths of 81 workers. The ILO reported in 2019 that falls, crush injuries, caught-between injuries, and electrocution were the leading causes of occupational deaths on construction sites (Warrier, 2019). Nevertheless, during a construction project, one of the most overlooked aspects is construction site safety. In most workplaces, accidents are a pain and a hassle for employees. On the other hand, accidents on construction sites have the potential to be fatal. However, most of these accidents and injuries can be avoided if workers wore appropriate personal protective equipment (PPE) like helmets, vests, boots, or safety glasses. Each piece of equipment in a set of PPE plays a vital role in minimizing the risk of injuries (see Figure 1.1). Therefore, onsite workers must take into consideration this precaution when working in dangerous environments. They have to comply with gearing up appropriately by wearing the proper equipment. In contrast, employers should always be fully aware of the workers' compliance with the guidelines and rules set by constantly monitoring them.

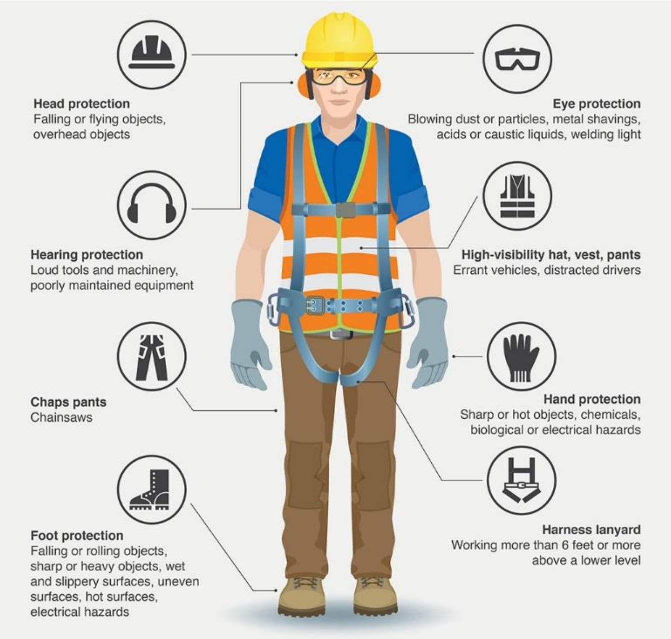

Head protection
Falling or flying objects, overhead objects
Eye protection
Blowing dust or particles, metal shavings, acids or caustic liquids, welding light
Hearing protection
Loud tools and machinery, poorly maintained equipment
High-visibility hat, vest, pants
Errant vehicles, distracted drivers
Chaps pants
Chainsaws
Hand protection
Sharp or hot objects, chemicals, biological or electrical hazards
Foot protection
Falling or rolling objects, sharp or heavy objects, wet and slippery surfaces, uneven surfaces, hot surfaces, electrical hazards
Harness lanyard
Working more than 6 feet or more above a lower level

Figure 1.1 Personal Protection Equipment in Workplace Safety

It is impossible for employers to keep an eye on their workers 24/7 that are most of the time located in remote areas. To overcome this limitation, the use of computer vision-assisted technology is becoming increasingly common. Computer vision is a branch of artificial intelligence (AI) that allows computers and systems to extract useful information from photos, videos, as well as other visual inputs to conduct any action based on that information. A computer vision-based approach is a cost-effective choice for remote non-invasive construction site safety monitoring. In recent applications, this technique has demonstrated its potential to handle a variety of construction management challenges.

This project is aimed at developing an application for detecting construction site safety helmet, one of the PPE, using computer vision-based technology. The primary purpose of safety helmet detection is to enhance construction safety by measuring the conformance of safety. Utilizing this technology allows for the automated collection and detection of workers' improper dress codes, which in this case, the safety helmet, on construction sites. By making use of videos, computer vision technology may improve employers' comprehension of situations at construction sites. While on the job, trained object recognition models can determine whether workers are wearing their helmet. This technology can detect non-compliant workers, allowing the management to intervene before an Occupational Safety and Health Administration (OSHA) fine or accident occurs. A well-fitted and intact helmet might mean the difference between life and death in the case of an accident. Hence, all these functions enable employers to stay informed and improve the overall productivity in managing safety-related tasks.

1.2 Problem Statement

Generally, the construction industry has had exceptionally high accident rates compared to other industries. According to the Department of Safety and Health Malaysia (DOSH) data on fatal accidents in 2020 (*Department of Statistics Malaysia Official Portal*, 2020), the Malaysian construction industry had the greatest number of deaths throughout the research period. For a long time, project managers have been concerned about the safety of construction employees. Despite the fact that construction site safety is recognised as the top priority, the absence of proper monitoring procedures further complicates the matter. Taking pre-emptive steps will significantly lower accident rates in construction sites. PPE for construction workers is one such important step. However, construction workers' compliance with PPE use is not always assured despite its effectiveness. Also, the absence of PPE has been identified as a significant issue affecting construction site safety in several research studies (Chi et al., 2005; Choudhry & Fang, 2008; Tam et al., 2004). Therefore, effective monitoring towards detecting the lack of PPE, which this project focuses on safety helmet, on construction workers becomes one of construction safety management's important aspect.

In spite of that, effective monitoring by humans themselves is not without risk, as there are significant obstacles that are beyond human ability. One of the obstacles, for instance, is time-consuming. Employers would have to closely monitor and observe the behaviour of each of their workers every second since the probability of a worker going against the rules of not wearing PPE is unpredictable. Without PPE, there is a higher chance of accidents occurring. However, manually checking PPE for proper usage frequently requires the dedication of the inspector, which not only adds to the expense, but also exposes the potential to human error. The employer has no capability to be constantly available onsite and be informed first-hand of any of the unforeseen situations mentioned that arise. As a result, automated ways of monitoring and identifying unusual situations are increasingly attracting the interest of researchers. Automated systems that can monitor safety and are both real-time and reliable can be quite beneficial in this field.

1.3 Objective

The objectives of this project are:

i. To identify the limitation in existing applications of construction site PPE detection.

ii. To develop a construction site safety helmet detection mobile application using computer vision-based technology.

iii. To evaluate the functionalities of the developed application.

1.4 Scope

The scopes of this project are as below:

i. The application is targeted towards users within the construction management.

ii. The application focuses on one type of PPE detection, which is the safety helmet.

iii. The application is Android-based where users will receive notifications on non-compliance workers.

iv. The application is developed using Android Studio for developing a mobile application and TensorFlow for machine learning.

v. The model used in the application's object detection is the MobileNetV2 SSD FPN model which supports TFLite conversion.

1.5 Significance of Project

i. Employers are able to be notified first hand whenever workers do not comply with wearing helmet.

ii. The proposed application provides a feature on PPE inventory management where users can record the number of PPE in use, or newly added PPE quantity into the system.

iii. The proposed application enables users to navigate through the system within a mobile application and receive real-time notifications.

1.6 Report Organization

This report consists of five chapters. CHAPTER 1 discusses about the introduction of construction site safety and the use of computer vision towards the matter, where the introduction, problem statement, objective, scope, significance of the project and report organization are also included.

CHAPTER 2 briefly discusses about the literature reviews on three existing construction site safety systems, each of their description and the comparison among them.

CHAPTER 3 focuses on the methodology used to develop the application. The functional and non-functional requirements, as well as the constraints and limitations of the proposed system are discussed in detail too.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews three existing systems available in the market regarding the detection of PPE in construction sites. Then, an analysis review of comparison among the three systems are explored in terms of their features, advantages, and disadvantages and so on. This chapter aims to get an overview of the functions and aspects of these related applications, in order to have a better understanding on how the proposed application will be developed.

2.2 Review of Existing Systems

The three existing systems will be reviewed, then analysed and discussed here in this part. The discussion aims to provide a guide for developing the proposed project where the features and performance of the existing systems can then be adapted into the proposed project. The three existing systems chosen for this study are PPE Detection by viAct, EasyFlow, and V-App.

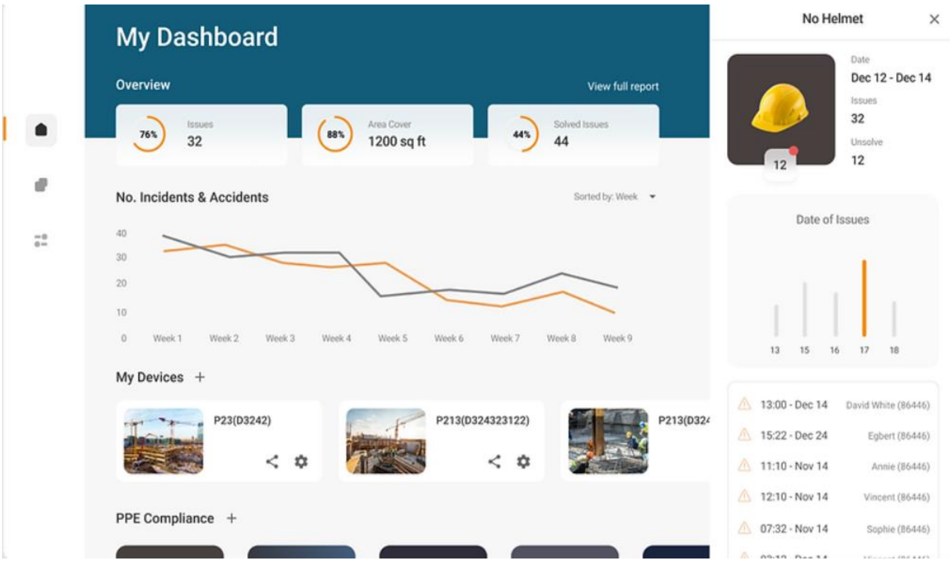
2.2.1 PPE Detection by viAct

The PPE detection solution provided by viAct comes in two forms, a web-based application, and a mobile application (*ViAct | PPE Detection Solution for Construction Jobsite |Construction Management*, n.d.). The system provides safety inspectors with real-time data and verification of breaches, allowing them to do their jobs better and, as a consequence, prevent workplace accidents. It combines advanced computer vision and artificial intelligence to identify when personal protective equipment (PPE) isn't being utilised, or isn't being used properly, and alerts the employee or necessary personnel to promptly resolve PPE concerns. To get on board the system, 3 simple steps are required:

capture with available cameras, process it with their AI technology, and lastly use cloud or edge for monitoring.

The system provides 24/7 onsite PPE inspection and monitoring service, which analyses more than 100 cameras simultaneously. One useful feature the system has is that it enables the safety inspector to monitor the entire site's safety remotely and receive safety reports through dashboards from numerous ends. If PPE non-compliance is detected, an instant alert will be delivered to the safety inspector along with the footage evidence. All previously compliance records will be automatically stored to AI cloud, and corresponding PPE breach footage will be available for incident analysis (See Figure 2.3).

The system is not free. However, viAct provides a 14-days free trial if potential clients are interested with their product. To access the 14-days free trial, clients would have to fill up the registration form and be contacted by their customer service via email.



My Dashboard

Overview

View full report

76% Issues 32

88% Area Cover 1200 sq ft

44% Solved Issues 44

No. Incidents & Accidents

Sorted by: Week

My Devices +

P23(D3242)

P213(D324323122)

P213(D324

PPE Compliance +

No Helmet

12

Date: Dec 12 - Dec 14

Issues: 32

Unsolve: 12

Date of Issues

13:00 - Dec 14	David White (86446)
15:22 - Dec 24	Egbert (86446)
11:10 - Nov 14	Annie (86446)
12:10 - Nov 14	Vincent (86446)
07:32 - Nov 14	Sophie (86446)

Figure 2.1 Accidents and PPE Compliance Dashboard

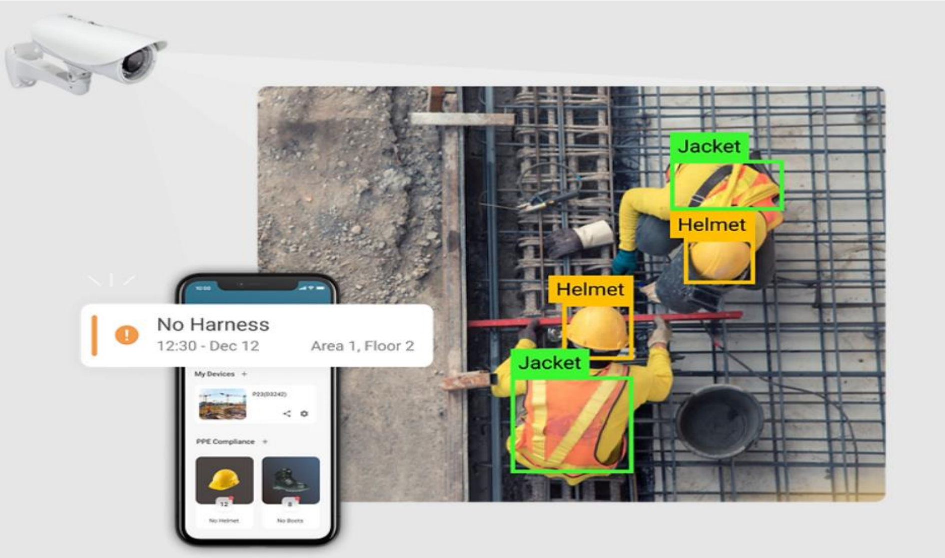


Figure 2.2 Real-Time Alert Notification in Mobile App

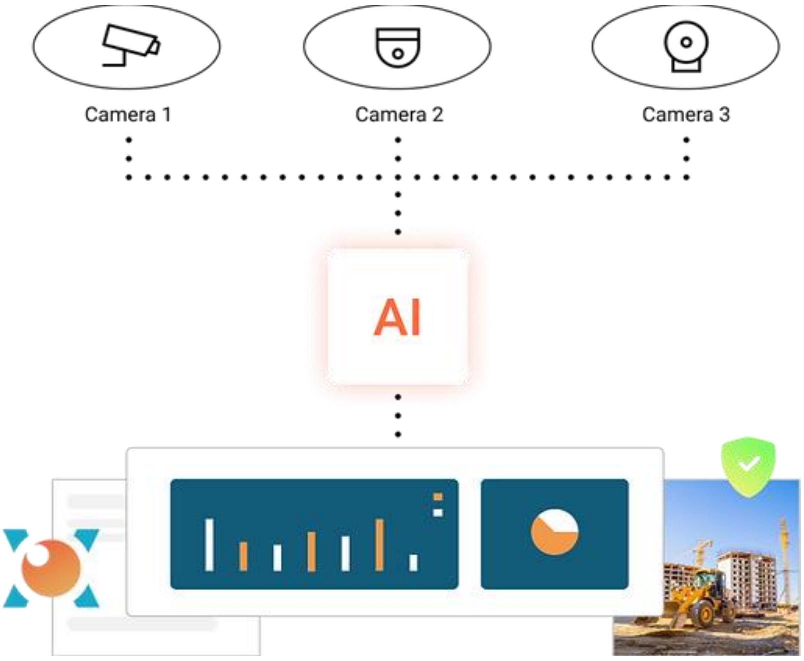


Figure 2.3 Incidents Captured From Cameras Sent to AI Cloud Data for Traceback and Analysis

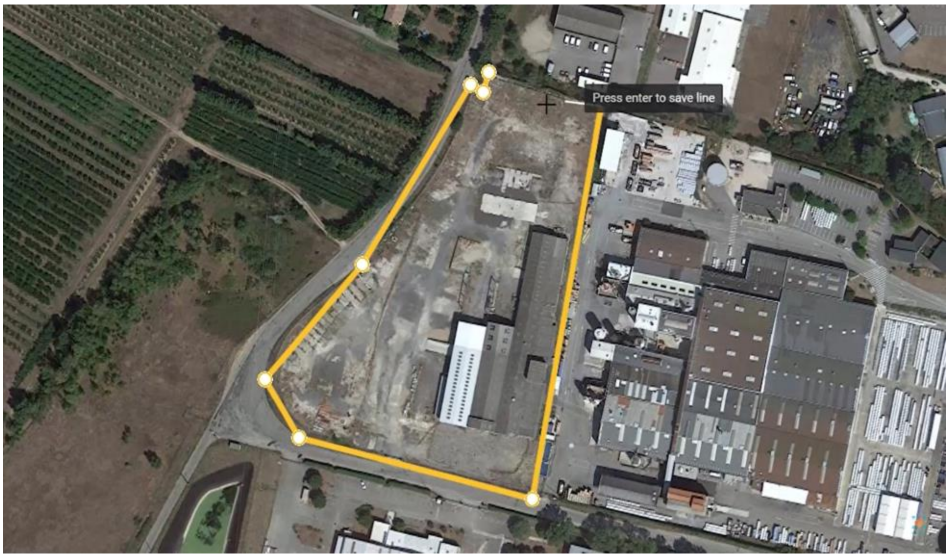


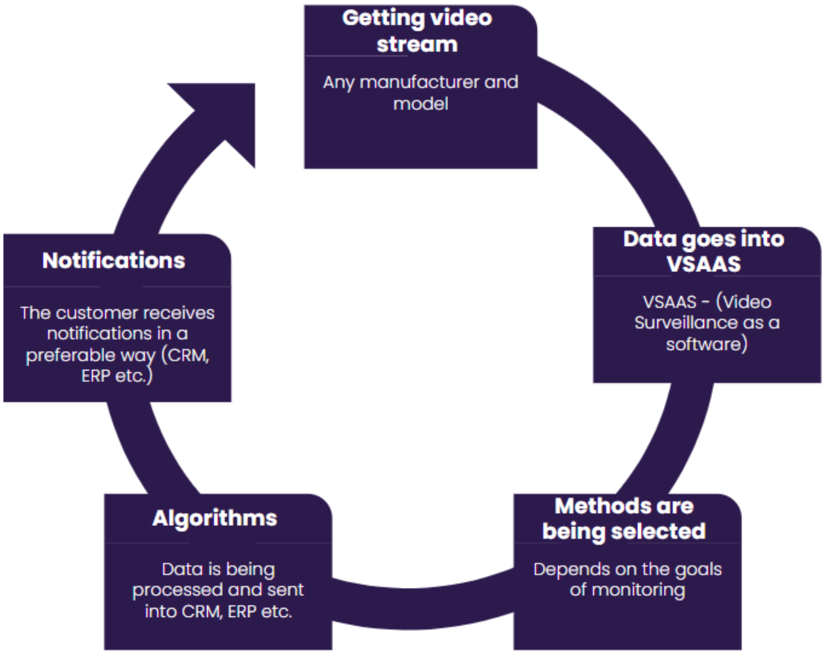
Figure 2.4 Site Plan Drawing for Detection

2.2.2 PPE Detection by EasyFlow

EasyFlow real-time PPE detection is a web-based service that detects hardhats, high-visibility vests, protective gloves, eyeglasses, and other needed PPE equipment in real time (*EasyFlow | PPE Detection*, n.d.). The technology provides real time data and evidence of breaches to safety inspectors, allowing them to do their job better and reducing workplace injuries as a consequence. EasyFlow PPE detection operates 24 hours a day, seven days a week on constant monitoring to assure compliance. When a violation of the PPE policy is detected, the platform instantly alerts the on-duty security officer and sends the relevant video evidence. It also offers corresponding PPE breach footage for incident analysis. Besides monitoring PPE non-compliance, the system allows configurable rules to be assigned for special monitoring zones, for instance, protective eyewear is needed at one location but not at others.

The platform works by utilising video stream from onsite security cameras. Their analytical dashboard is available for their clients to access. It also allows aerial video footage captured by drones to be utilised for site inspections.

EasyFlow provides a 30-days free trial for clients to access the platform. Until then, clients should get in touch with their representative and might need to pay for it to further use the platform.



Getting video stream
Any manufacturer and model
Data goes into VSAAS
VSAAS - (Video Surveillance as a software)
Methods are being selected
Depends on the goals of monitoring
Algorithms
Data is being processed and sent into CRM, ERP etc.
Notifications
The customer receives notifications in a preferable way (CRM, ERP etc.)

Figure 2.5 How It Works

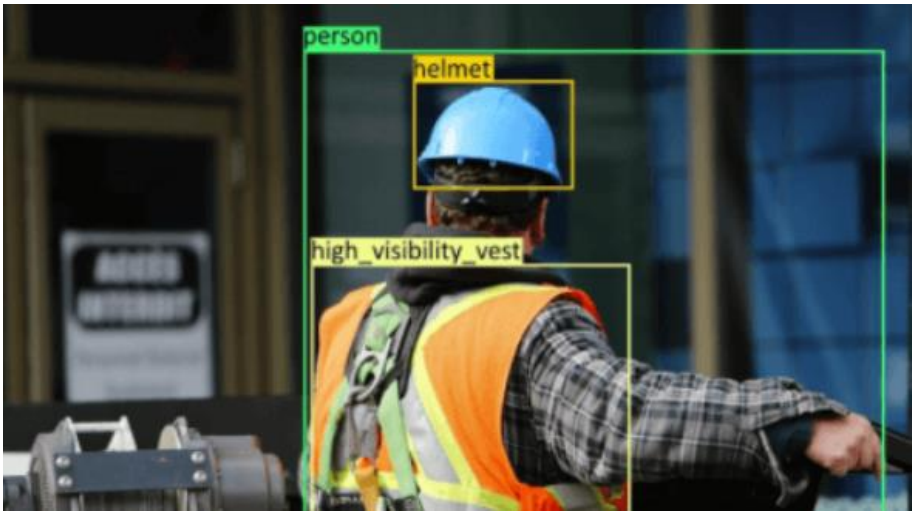


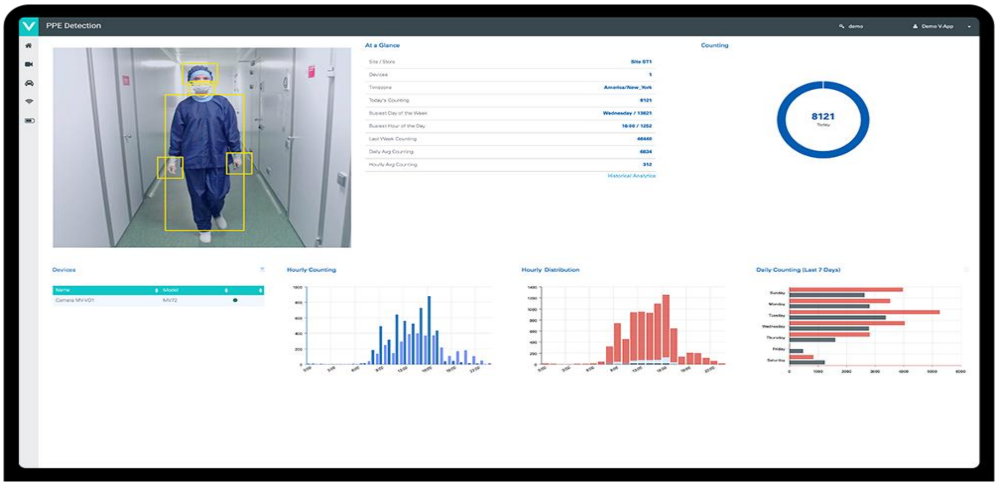
Figure 2.6 PPE Detection on Cameras

2.2.3 PPE Detection by V-APP

V-App is a web-based application that uses a real-time video and image analysis technology for PPE correct usage detection, which can be used to see if construction workers are wearing head protection properly, or medical staffs are wearing face and hand covers (*PPE Detection | V-App*, n.d.). The application can send notifications to the Safety Team when it detects the lack of PPE or their improper use of it and promptly limit employee exposure to dangers as soon as possible. The types of PPE that can be detected by V-App are face covers such as N95, surgical, and masks made of clothes, and goggles; hand cover like surgical gloves and safety gloves; and head cover such as hardhats, helmets, and earmuffs.

The system automatically detects PPE on persons in the form of images provided by Meraki cameras. Meraki smart cameras are able to provide the input image to obtain summary information on people who are wearing the required PPE, are not wearing it, or are uncertain.

Just like the two previous related systems, V-App is not a free system. Nonetheless, free trial is given to interested individuals by contacting their evaluation support team.



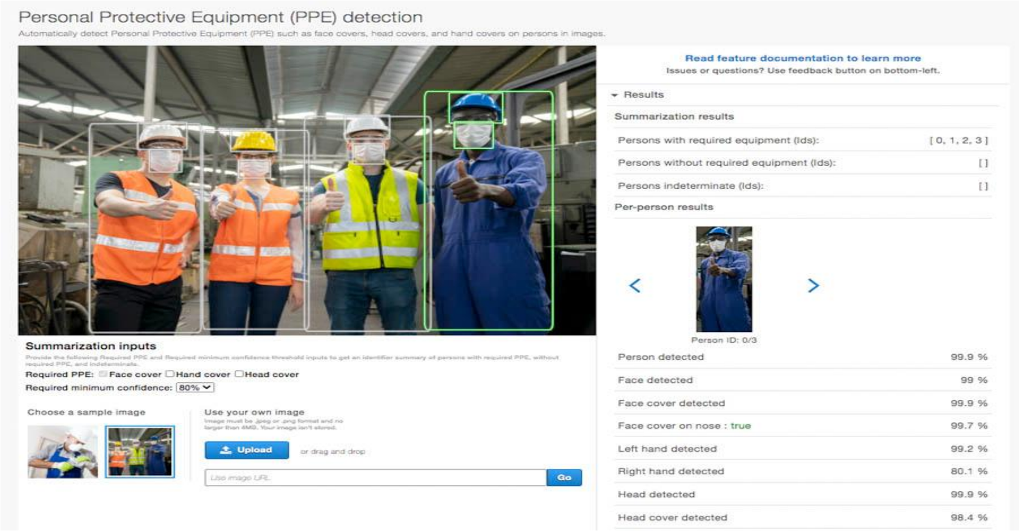
At a Glance

Site / Store	Site ST1
Devices	1
Timezone	America/New_York
Today's Counting	8121
Busiest Day of the Week	Wednesday / 13621
Busiest Hour of the Day	16:00 / 1252
Last Week Counting	46440
Daily Avg Counting	6634
Hourly Avg Counting	312

Historical Analytics

Name	Model
Camera MV-V01	MV72

Figure 2.7 V-App PPE Detection Dashboard



Personal Protective Equipment (PPE) detection

Automatically detect Personal Protective Equipment (PPE) such as face covers, head covers, and hand covers on persons in images.

Summarization inputs

Provide the following Required PPE and Required minimum confidence threshold inputs to get an identifier summary of persons with required PPE, without required PPE, and indeterminate.

Required PPE: ☑ Face cover □ Hand cover □ Head cover

Required minimum confidence: 80%

Choose a sample image

Use your own image

Image must be .jpeg or .png format and no larger than 4MB. Your image isn't stored.

Read feature documentation to learn more

Issues or questions? Use feedback button on bottom-left.

Results

Summarization results

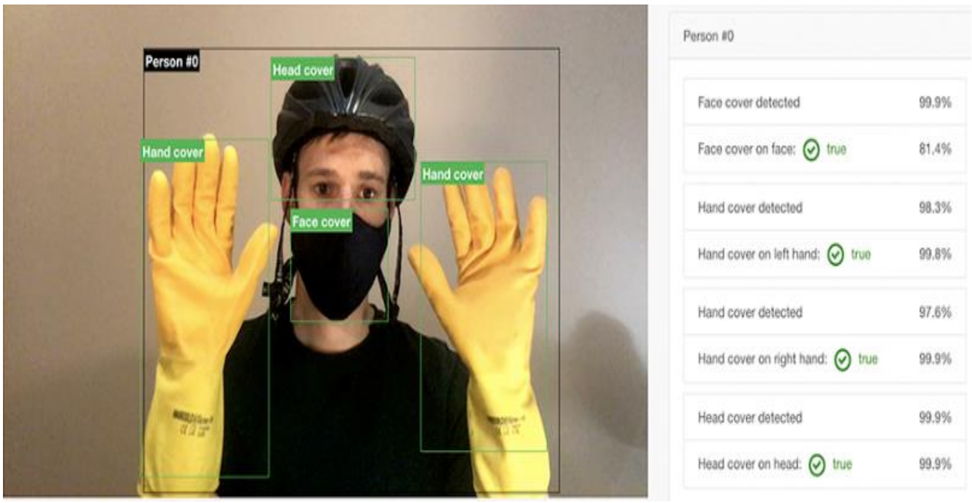
Persons with required equipment (Ids):	[0, 1, 2, 3]
Persons without required equipment (Ids):	[ ]
Persons indeterminate (Ids):	[ ]

Per-person results

Person ID: 0/3

Person detected	99.9 %
Face detected	99 %
Face cover detected	99.9 %
Face cover on nose : true	99.7 %
Left hand detected	99.2 %
Right hand detected	80.1 %
Head detected	99.9 %
Head cover detected	98.4 %

Figure 2.8 PPE Monitoring on Multiple Workers



Person #0

Face cover detected	99.9%
Face cover on face: ✓ true	81.4%
Hand cover detected	98.3%
Hand cover on left hand: ✓ true	99.8%
Hand cover detected	97.6%
Hand cover on right hand: ✓ true	99.9%
Head cover detected	99.9%
Head cover on head: ✓ true	99.9%

Figure 2.9 Type of PPE Detected

2.3 System Comparison

This section explores and compares the three existing systems with the proposed system as stated above based on different aspects. Their comparisons are discussed in the table below.

Table 2.1 Comparison of Existing Systems and Proposed System

Category	viAct	EasyFlow	V-App	Proposed System
Platform	Web and Mobile	Web	Web	Mobile
Targeted user	Construction safety inspectors and workers	Construction safety inspectors and workers	Construction safety inspectors and workers	Construction safety inspectors and workers
Trial period	14 days	30 days	To be requested from provider	Free to use
Technical support	24/7	Not stated	Not stated	24/7
Features	Detects PPE not being used Real time alert and alarms 24/7 monitoring service Safety report through dashboard	Detects PPE not being used Real time alert 24/7 monitoring service Configurable rules for special monitoring zones Analytical dashboard available	Detects PPE not being used Real time alert Charts and graphs through dashboard	Detect safety helmet not being used Real time alert 24/7 monitoring service PPE inventory management Helmet breach records Manage authorized users that can access the app
Advantages	Constant 24/7 monitoring Provides video evidence of PPE breach Offers PPE breach footage for incident analysis	Constant 24/7 monitoring Enables PPE customization for special construction zones Provides video evidence of PPE breach Offers PPE breach footage for incident analysis	Constant 24/7 monitoring Provides demo version Customizable summarization inputs of PPE	Constant 24/7 monitoring Provides a PPE inventory management platform Free to use with no charge imposed
Disadvantages	Cannot access trial instantly Limited trial period No facial recognition for identifying detected person	Cannot access trial instantly Limited trial period No facial recognition for identifying detected person	Cannot access trial instantly Limited trial period No facial recognition for identifying detected person	No facial recognition for identifying detected person No capturing of workers for breach record

2.4 Conclusion

According to the comparison and analysis done among the three existing systems, each of them has their distinct features, which can be applied in the proposed system. To summarise, the focus of all the existing systems is to detect any non-compliance workers towards PPE use and to send alerts when detected. Besides adding the common functionalities onto the proposed system, additional features should also be added to make the proposed system stand out among the similar systems, which includes allowing the admin to manage the information of authorized users that can access the company account in the application, providing a platform for PPE inventory management as well as allowing the helmet breaches to be viewed.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses about the methodology implemented in this proposed project. Among the many software development methodologies most frequently used in the market, Agile methodology will be selected as the development framework here. Further explanation on the chosen framework will be discussed in detail, as well as the functional and non-functional requirements, constraints, and limitations of the proposed project.

3.2 Project Management Framework

The Agile methodology divides a project into stages to make it easier to manage. The framework aims to produce the product through incremental delivery of small pieces of functionality iteratively. Agile is able to respond quickly to user's requirements and also able to change course as circumstance requires. This approach can minimise the risk of developing products that do not or no longer fulfil the market or customer demands. In this framework, long delivery cycle like the older waterfall framework is divided into shorter intervals known as sprints or iterations. Each sprint involves the process of planning, designing, building, testing, and finally, reviewing. The framework is illustrated in the figure below (DataVaults, 2021).

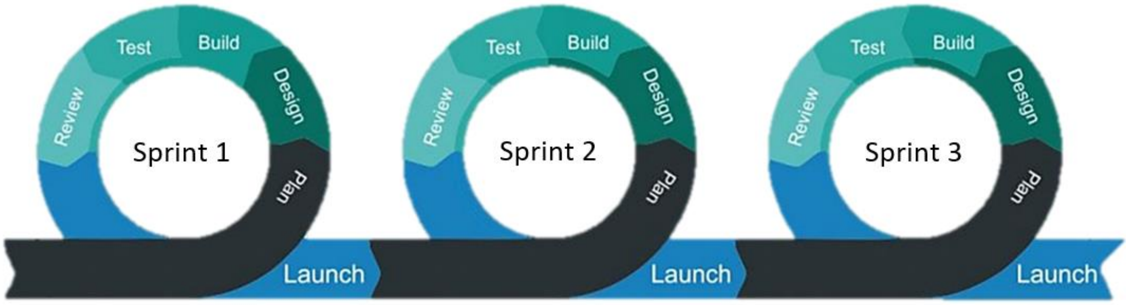


Figure 3.1 Agile Methodology

As a result, the Agile methodology framework enables delivery times to be shorten but delivering frequently, in order to ensure that smaller chunks of the product reach the market sooner, allowing feedbacks to be provided and ensuring the product meets user's needs. Because of its iterative nature, this framework is widely recommended in the market.

3.2.1 Planning Phase

The first phase in the sprint is the project planning phase. Before beginning the development process, requirements of the Construction Site Safety Helmet Detection Mobile Application (ConSite) are gathered first. Requirements to be gathered are categorised into two parts, functional and non-functional requirements, in which functional requirements describes the features needed to be included in the final product, and non-functional requirements that defines the final product's attributes like reliability and usability. The features to be implemented in the proposed application for safety helmet detection, along with how the interface design of the application will look like are documented into the Software Requirements Specification (SRS) document. Information regarding the flow on how the employers and workers will access the functions of the application are collected to provide ease of use on the application.

3.2.2 Designing Phase

In the designing phase of the sprint, the design of the proposed system is outlined by referring to the requirements defined previously. The necessary tools such as programming language to be used in this project are examined. Java programming language will be used to design the interface of the application. The detailed architecture design of the proposed application is documented inside Software Design Document (SDD).

3.2.3 Building Phase

The building phase in the agile sprint is also known as the development phase. This is the phase where developing the application starts, in accordance with the defined requirements and approved guidelines. The implementation of coding happens here, where the written documentation of the ConSite application is converted into an actual working application. The backend of the application where codes to detect helmet on workers are written. Therefore, this stage will be the most time-consuming stage.

3.2.4 Testing Phase

In the testing phase, the developed features of the ConSite application are tested to ensure it is bug-free. So, series of tests will be run to confirm that the product is completely working and achieves the defined requirements. If any possible faults or flaws are discovered, they must be fixed right away. A user acceptance test (UAT) and documentation of the results are completed to validate the developed features in the application and discover any issues before deployment.

3.2.5 Deployment Phase

During the deployment phase, the working product of the ConSite application is delivered and is available for stakeholders after it has gone through the testing phase. Continuous support is offered to ensure that the application continues to work properly and that any new any new defects are addressed.

3.2.6 Reviewing Phase

Reviewing is the last phase after all the previous phases are completed. The outcome achieved in fulfilling the defined requirements for the proposed application are presented after finishing all the previous stages of development. This is where feedbacks are gathered from the stakeholders in order to define requirements and any information and incorporate them into the next sprint.

3.3 Project Requirement

The following are the functional and non-functional requirements gathered for the proposed application. The constraints and limitations of the system are also discovered and explained here in detail.

3.3.1 Functional Requirements

i. The system shall be able to monitor the workers' compliance towards wearing safety helmet 24/7.

ii. The system shall detect any non-compliance of helmet on construction workers.

iii. The system shall send notifications to users to alert them on non-compliance workers.

iv. The system shall allow the user to add new PPE, update PPE information, and delete PPE in the PPE management system.

v. The system shall allow the admin to view and update a profile of himself or profile of other supervisors.

vi. The system shall allow the user to view all the records of the helmet breaches.

3.3.2 Non-Functional Requirements

Table 3.1 Non-Functional Requirements

Requirement	**Justification**
Performance	The system should send a notification within 3 seconds from the time a helmet non-compliance is detected.
Compatibility	The system must support Android devices running on Android version 7 to 12.
Availability	The system must be available for use to the user 98 percent of the time every day.

Security	The system must provide a login function for authorized users to access the system.
Learnability	The ability for the user to navigate through the system to the desired feature must not exceed 3 seconds.
Satisfaction	The system dashboard must be minimalist and clean with no charts and graphs cluttered together.

3.3.3 Constraints

The constrains of the project are as follows:

i. The need for internet connection – The user is unable to receive notifications from the system with the wifi/data off. The user would need to have internet connection 24/7 in order to receive real-time notifications anytime.

ii. High cost of installing multiple cameras – Multiple cameras are needed to be installed to achieve an overall effective monitoring on workers at different angles in different construction sites. However, this would need higher cost in purchasing cameras and installing each of them. This might exceed the budget allocated.

3.3.4 Limitations

The limitations of the project are as follows:

i. The lack of professionals – There is a lack of highly trained professionals and manpower with advanced knowledge in AI and machine learning technologies in order to train computer vision-based systems. More professionals are needed to help design this technology of the future.

ii. The need of regular monitoring – When the computer vision-based system experiences technical problems or fails, it can result in significant financial loss for companies. As a result, there needs to be a dedicated team to monitor and analyse the system.

iii. Incapability of warning non-compliance workers instantly – As the employer receives notifications on workers not complying to the rules of wearing proper PPE, the employer is unable to warn the workers directly face to face. There is a need to contact the workers directly by installing horn speakers around the construction sites for employers to communicate directly with them.

3.4 Proposed Design

3.4.1 Context Diagram

Error! Reference source not found. shows the context diagram of the proposed application. A context diagram depicts how the system interacts with external factors or actors with whom it is supposed to engage. Here, the external factors of the application are the admin, which will be the main user of the application from a company, supervisor, and the camera. The data flowing from the users towards the system and vice versa are shown inside the diagram.

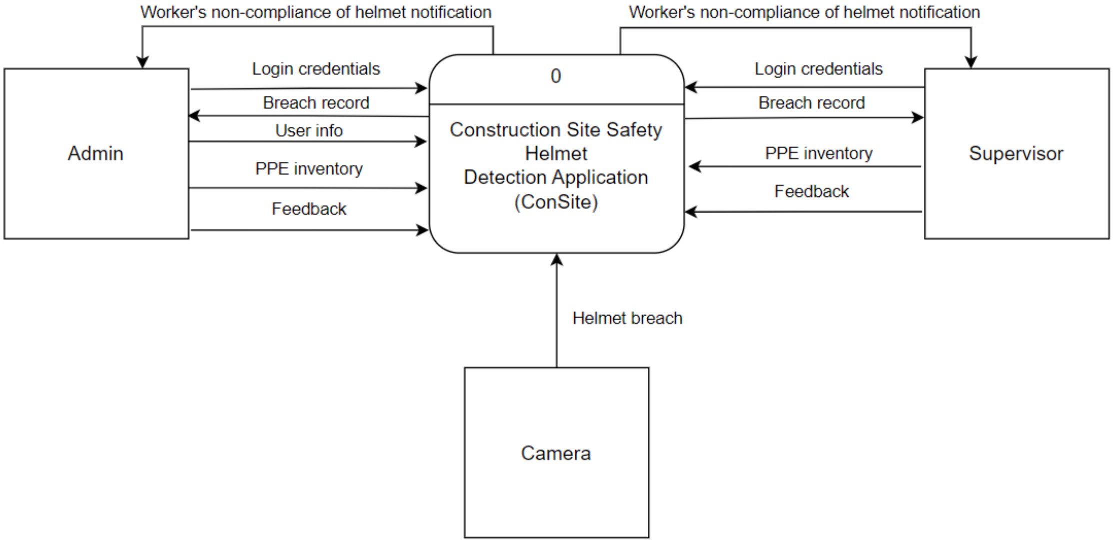

Worker's non-compliance of helmet notification
Worker's non-compliance of helmet notification
Admin
Login credentials
Breach record
User info
PPE inventory
Feedback
0
Construction Site Safety Helmet Detection Application (ConSite)
Login credentials
Breach record
PPE inventory
Feedback
Supervisor
Helmet breach
Camera

Figure 3.2 Context Diagram

From the context diagram above, the entities involved in the application are the admin, supervisor, and camera. Firstly, login credentials from both the users, in this case the admin and supervisor are needed to log into the application. After logging in, the admin may record and manage users' info who are able to access the application. Both users can manage PPE inventory and send feedback of the application. The application will send notifications of any non-compliance of helmet worn by workers. Breach records of absence of helmet on workers generated from the application can also be accessed by the users. The camera entity provides the helmet breach information to the application.

3.4.2 Use Case Diagram and Description

The figure below describes the use case diagram of the proposed application. A use case diagram helps to outline the specifics of the system's actors and how they interact with it. There are a total of 6 main modules in which the user has access to it, which are login, manage user data, access breach records, receive notifications, manage PPE inventory, and add feedback. Table 3.2 describes the functions of each module in general.

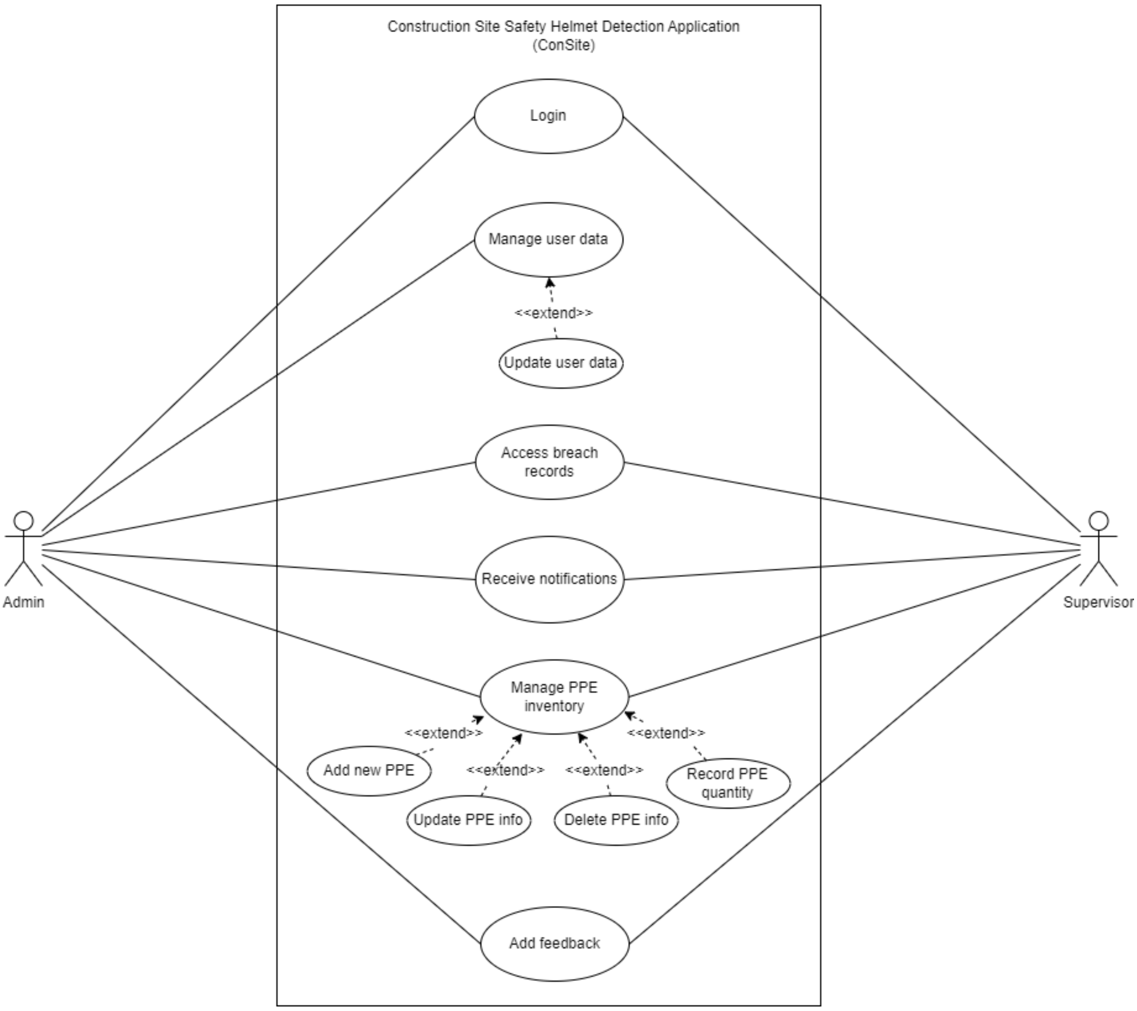


Figure 3.3 Use Case Diagram

Table 3.2 Use Case Description

Module	Function

Login	The user can log into the system.
Manage user data	The employer can manage other supervisor’s data.
Access breach records	The user can view the records of helmet breach.
Receive notifications	The user can receive notifications on absence of helmet on workers.
Manage PPE inventory	The user can manage the inventory of PPE where adding, updating, and deleting the inventory is allowed.
Add feedback	The user can submit feedback on the issues met or improvements to be suggested.

3.4.3 Data Flow Diagram

Figure 3.4 shows the data flow diagram of the proposed application, which is used to depict the flow of the data graphically in a business information system. It focuses on information flow, including where data originates, where it flows, and how it is kept.

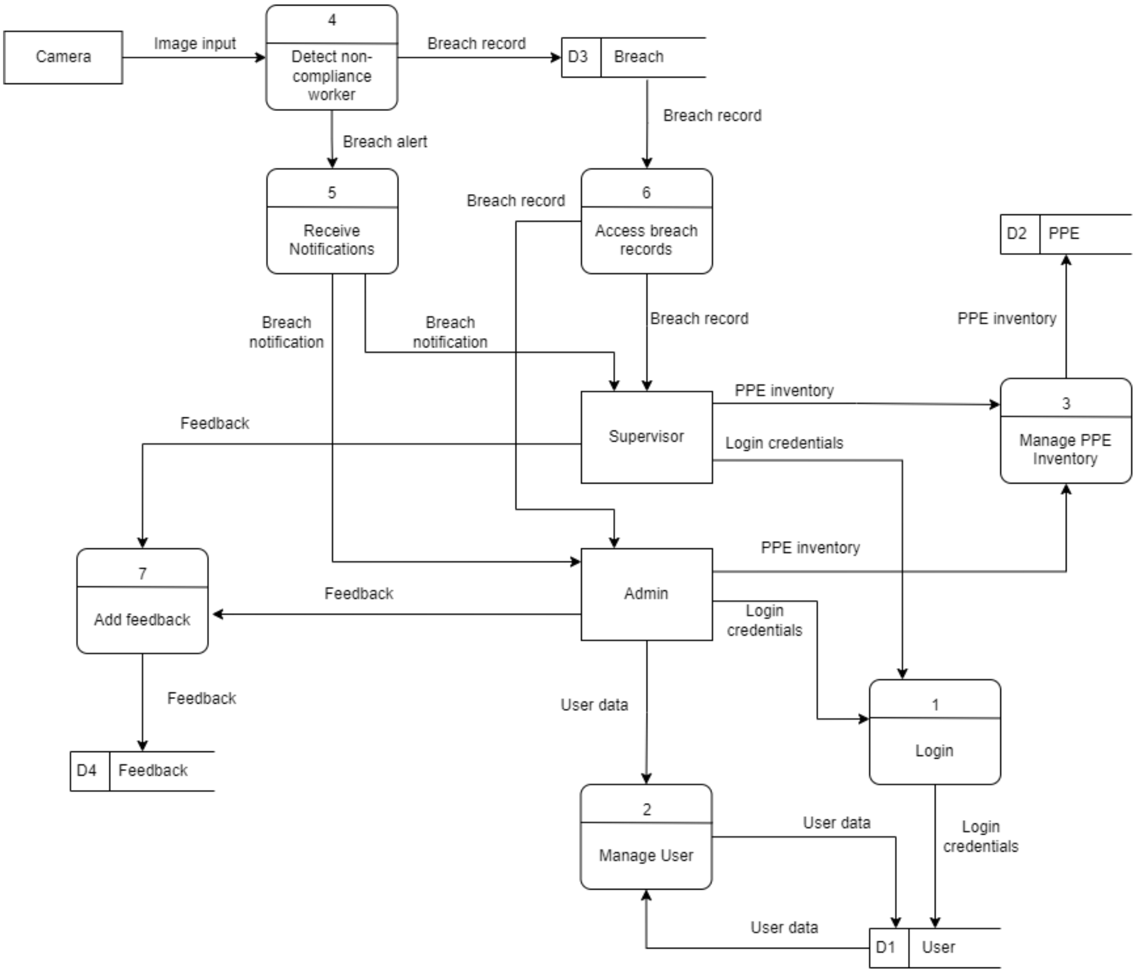


Figure 3.4 Data Flow Diagram

3.4.3.1 Process

Table 3.3 Process Description

Process	Description
Login	Process for user to log into the application.
Manage User	Process for admin to manage user's info who are granted access to the application.
Manage PPE Inventory	Process for user to manage PPE inventory that are available.

Detect non-compliance worker	Process for the camera on detecting inexistence of helmet on workers.
Receive Notifications	Process for user to receive notifications on helmet non-compliance issues.
Access breach records	Process for user to access the records of helmet breach.
Add feedback	Process for user to submit feedback on the issues met or improvements to be suggested.

3.4.3.2 External Entities

Table 3.4 External Entities Description

Process	Description
Admin	The admin is the main user of the application, who represents his company to manage the application.
Supervisor	The supervisor is one of the stakeholders involved in the application.
Camera	The camera captures the helmet breach.

3.4.3.3 Data Store

Table 3.5 Data Store Description

Process	**Description**
User	Data store that stores the users' info.
PPE	Data store that stores the PPE inventory info.
Breach	Data store that stores the records of helmet breach.

Feedback	Data store that stores the feedbacks given by the users.

3.4.3.4 Data Flow

Table 3.6 Data Flow Description

Process	Description
Login credentials	User enters the login credentials to log into the application.
User data	Admin enters the user data into the application.
PPE inventory	User enters the PPE inventory info into the application for record purpose.
Image input	Camera sends the scene that is captured.
Breach record	Application saves the breach record into the database.
Breach alert	Application sends alert on the helmet breach.
Breach notification	Users receive notifications on the helmet breach.
Feedback	Users submit feedback on the comments about the application.

3.4.4 Activity Diagram

The following diagram below is concerned about the activity diagram of the proposed application, which illustrates the application’s flow of activities. The actors of the application are placed at the left and right of the system representation in the diagram.

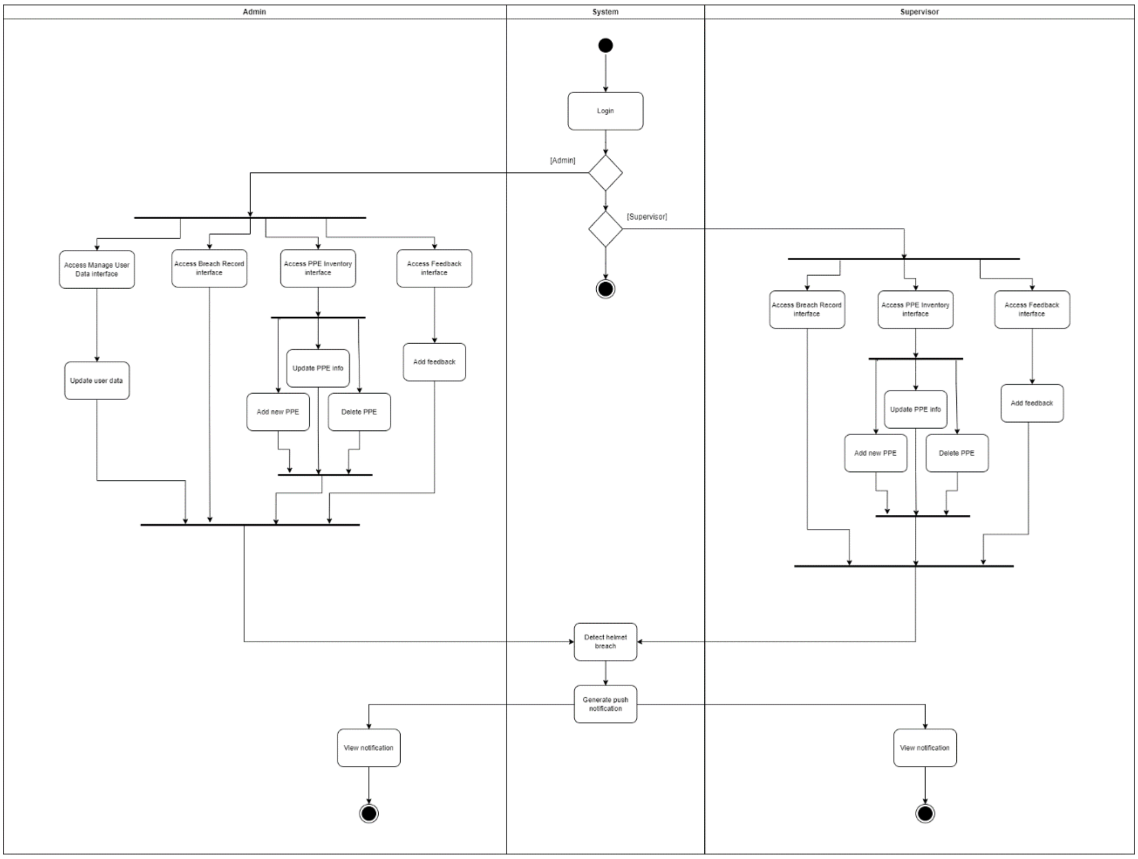

Admin
System
Supervisor
Login
[Admin]
[Supervisor]
Access Manage User Data interface
Access Breach Record interface
Access PPE Inventory interface
Access Feedback interface
Update user data
Update PPE info
Add new PPE
Delete PPE
Add feedback
Access Breach Record interface
Access PPE Inventory interface
Access Feedback interface
Update PPE info
Add new PPE
Delete PPE
Add feedback
Detect helmet breach
Generate push notification
View notification
View notification

Figure 3.5 Activity Diagram

3.5 Data Design

Figure 3.6 describes the Entity Relationship Diagram (ERD) of the application. The ERD is used to illustrate how every object relates to one another in the application. There are 4 objects in the ERD, each of them having their own attributes.

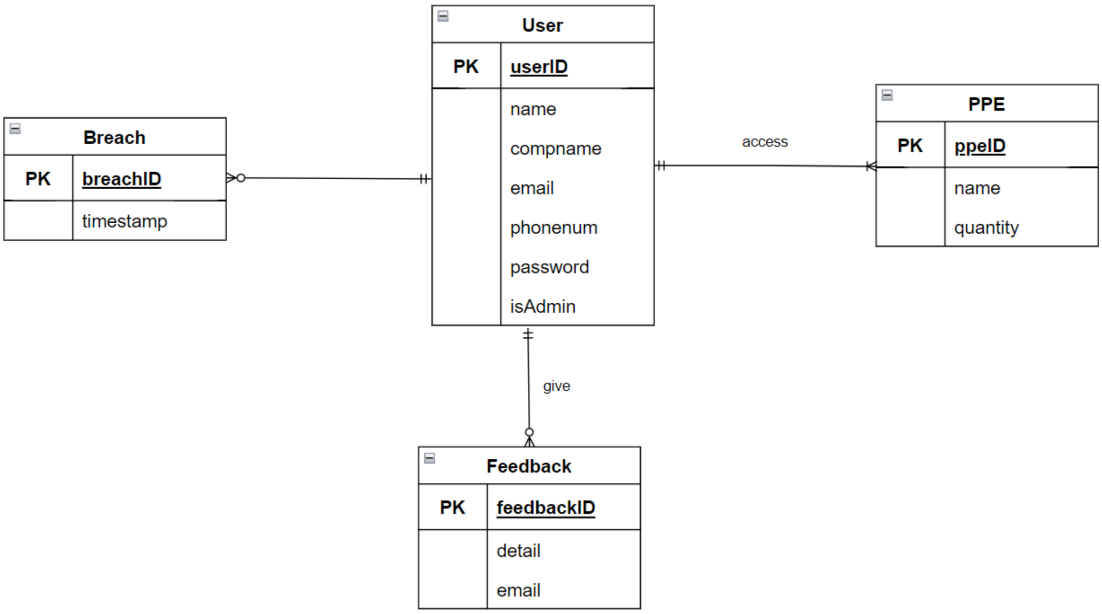


Figure 3.6 Entity Relationship Diagram

3.5.1 Database Dictionary

Table 3.7 User

Field Name	Description	Data Type	Constraint
userID	User ID	VARCHAR(255)	PK
name	User’s name	VARCHAR(255)	
compname	Company name	VARCHAR(255)	
email	User’s email	VARCHAR(255)	
phonenum	Phone number	VARCHAR(255)	
password	Password	VARCHAR(255)	
isAdmin	If user is admin	VARCHAR(255)	

Table 3.8 PPE

Field Name	Description	Data Type	Constraint
ppeID	PPE ID	VARCHAR(255)	PK
name	PPE name	VARCHAR(255)	
quantity	PPE quantity	INT	

Table 3.9 Breach

Field Name	Description	Data Type	Constraint
breachID	PPE breach ID	VARCHAR(255)	PK
timestamp	Date and time of breach	TIMESTAMP	

Table 3.10 Feedback

Field Name	Description	Data Type	Constraint
feedbackID	Feedback ID	VARCHAR(255)	PK
detail	Detail of feedback	VARCHAR(255)	
Email	Email of user	VARCHAR(255)	

3.6 Object Detection

Object detection is a computer vision technology which helps locate and identify things in an image or video. To be more precise, object detection creates bounding boxes around the objects that are detected, allowing us to determine their location inside (or how they move across) a scene.

In general, machine learning-based and deep learning-based techniques to object detection may be distinguished. In more conventional ML-based techniques, groupings of pixels that may belong to an object are identified by looking at various aspects of an image, such as the colour histogram or edges, using computer vision techniques. The location of the item and its label are then predicted using these attributes as input into a regression model. Convolutional neural networks (CNNs) are used in deep learning-based systems to do end-to-end, unsupervised object recognition, which eliminates the requirement for separate feature definition and extraction.

3.6.1 TensorFlow

TensorFlow is an open source, Python-compatible toolkit for numerical computation that accelerates and simplifies developing neural networks and machine learning algorithms. TensorFlow enables one to create dataflow graphs, which are structures that depict how data flows across a graph, or a collection of processing nodes. A mathematical operation is represented by each node in the graph, and each edge between nodes is a multidimensional data array, or tensor.

Abstraction is the single most important advantage TensorFlow offers for machine learning development. Overall of the application logic can be focused rather than worrying about the minute details of implementing algorithms or working out how to connect the result of one function to the input of another. The details are handled in the background via TensorFlow. Besides, TensorFlow provides extra conveniences if debugging and gaining insight are required in TensorFlow applications. Instead of creating and evaluating the complete graph as a single opaque object, each graph action may be evaluated and updated independently and openly.

3.6.2 MobileNetV2 SSD FPN Model

The model used in this project is the pre-trained MobileNetV2 SSD FPN-Lite 320x320 model. The COCO 2017 dataset, with pictures scaled to 320x320 resolution, was used to train the model.

Based network:

MobileNet is built using neural networks. High-level characteristics for categorization or detection are provided by the base network. The figure below shows an illustration of a network with several convolutional layers. Each training picture receives filtering at various resolutions, and the result of each convolved image serves as the input for the following layer.

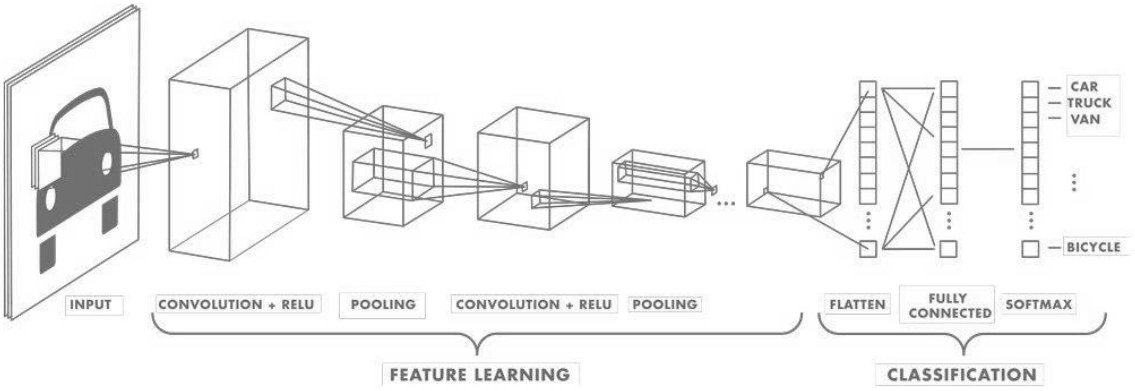


Figure 3.7 An illustration of a network with several convolutional layers.

Detection network:

Single Shot Detection (SSD) and RPN are the most popular detection networks (Regional Proposal Network). When employing SSD, many objects in the image may be detected with just one shot. On the other hand, systems based on regional proposal networks (RPNs), like the R-CNN series, require two shots: one for producing regional proposals and one for identifying the targets of each proposal.

As a result, SSD is significantly quicker than RPN-based methods, although precision is sometimes sacrificed for real-time processing speed. They frequently struggle to identify items that are too far away or too tiny.

Feature pyramid network:

It might be difficult to find things of various sizes, especially little ones. A feature extractor called the Feature Pyramid Network (FPN) was created using the feature pyramid concept to increase accuracy and speed.

3.6.3 Dataset

The dataset to be used for training the model for object detection is an open-source dataset called SafetyHelmetWearing-Dataset (SHWD). The dataset is prepared for the human head and safety helmet wearing detection and it contains 7581 photos with 111514 normal head objects and 9044 human safety helmet wearing objects (positive) (not wearing or negative). 5571 images from the dataset are taken for the use in this project. Multiple bugs are identified in this dataset and are fixed for the training of detection model. The format of the annotations for each image is in PASCAL VOC. The two object class name in the annotations for the dataset, namely “hat” for positive object and “person” for negative object, are changed to “helmet” and “no_helmet” respectively.

There is no constraint on the colour of safety helmet. All colours of the safety helmet are detected as positive. One limitation of the dataset is that there is no images in the dataset which categorize humans with their heads covered as negative. This indicates that if the person covers his head, the model still detects it as wearing a safety helmet. There are lots of improvement on the dataset for future development.

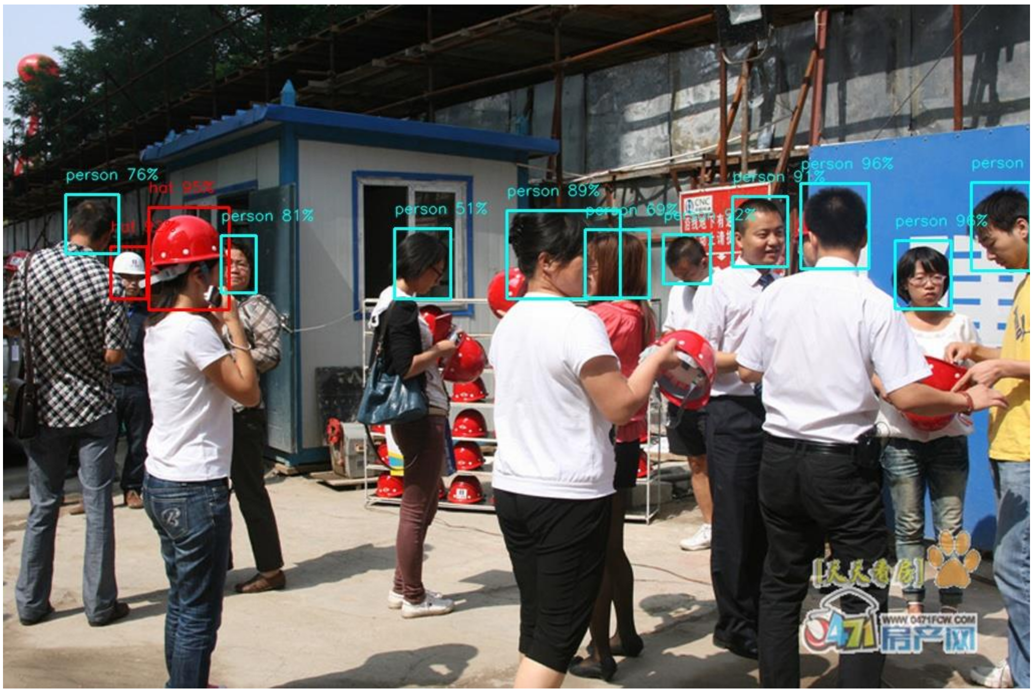


Figure 3.8 Dataset Example

3.7 Storyboard

This section discusses about the storyboard of the whole application. The storyboard is an approach to sketch out the general structure and user experience across various panels and tasks that are included in the application. The following figures represent the storyboard of each user, namely the admin and the supervisor.

3.7.1.1 Admin

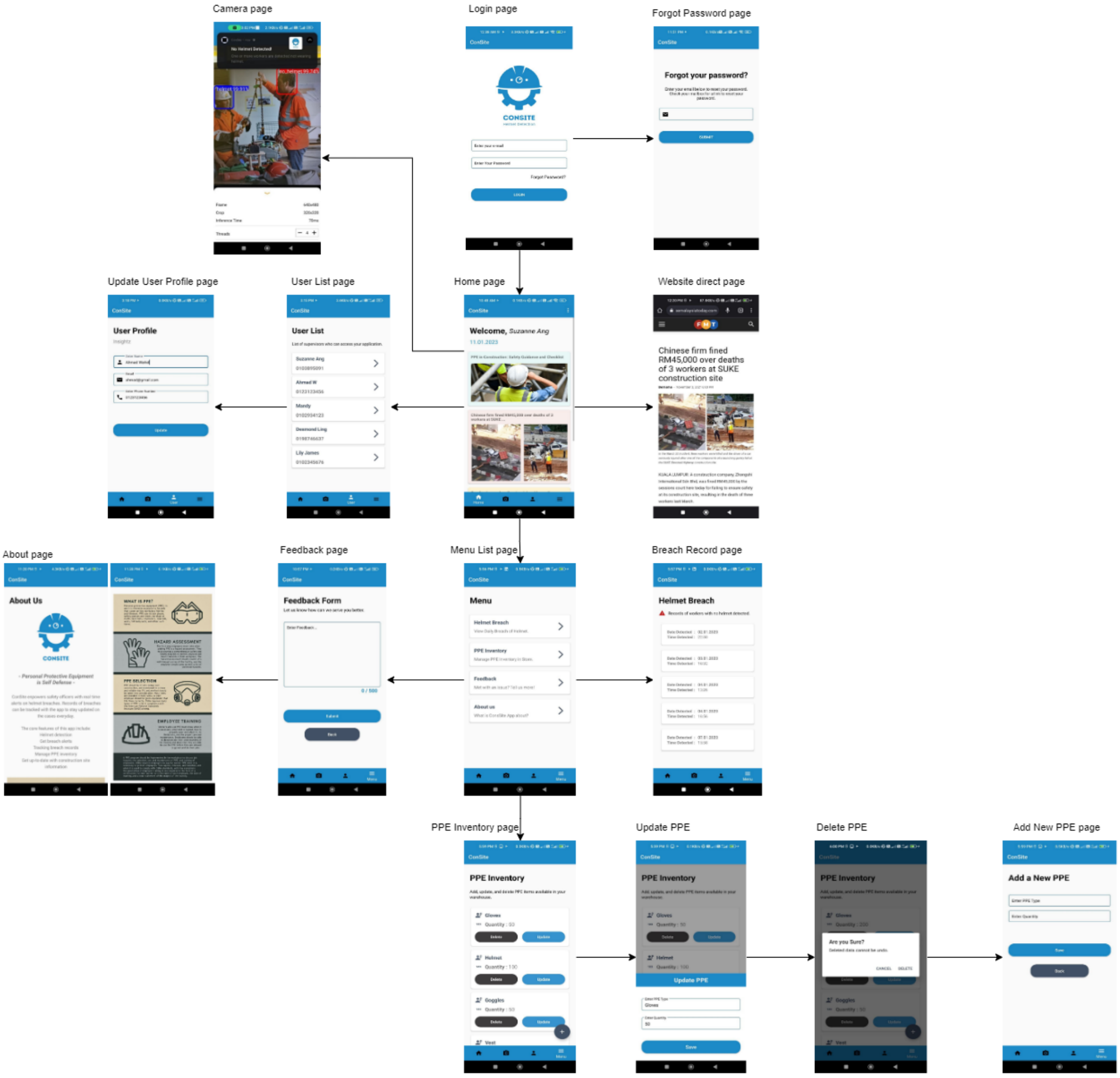


Figure 3.9 Admin Storyboard

3.7.1.2 Supervisor

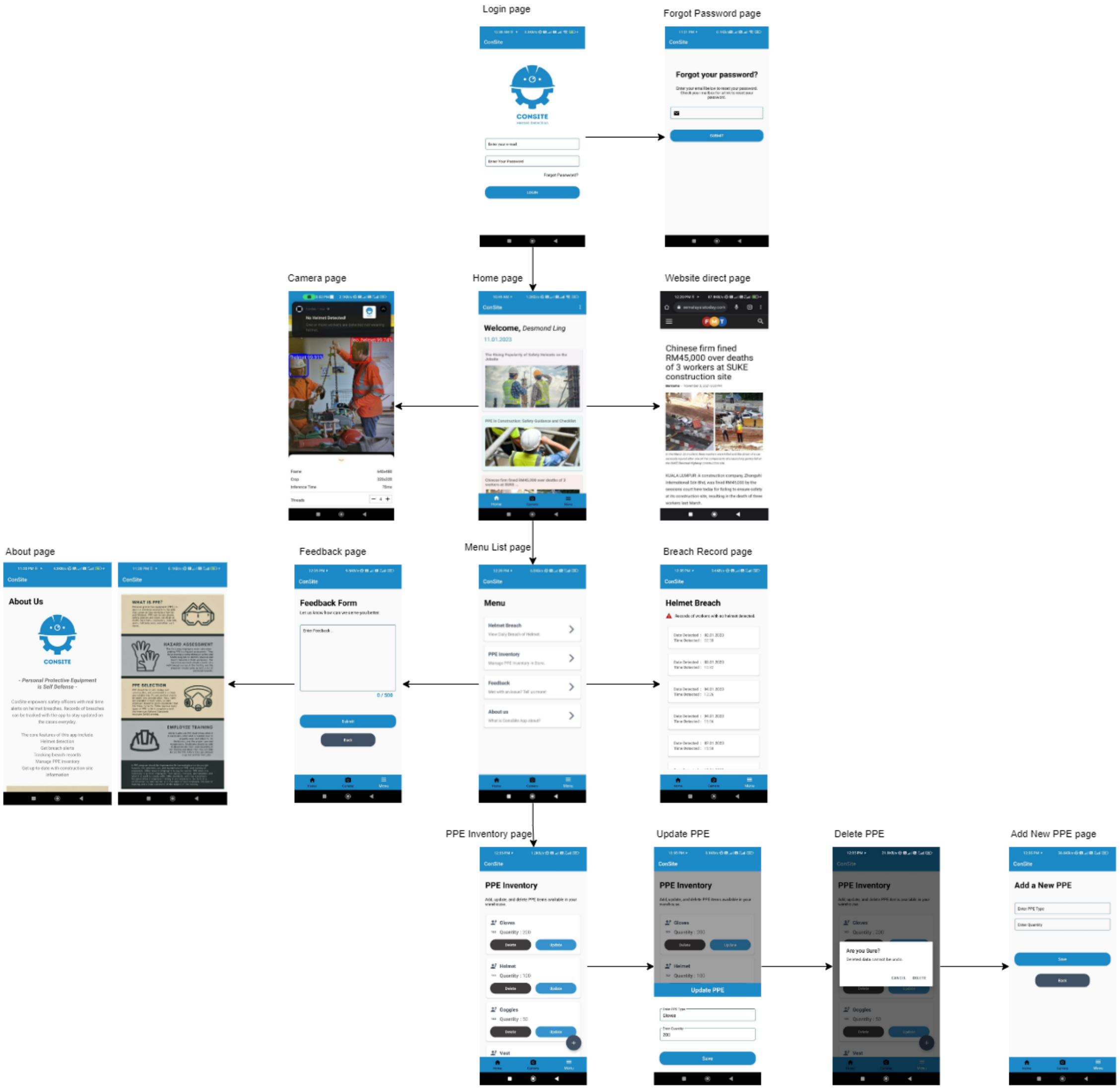


Figure 3.10 Supervisor Storyboard

3.8 Testing Plan

In the testing phase, User Acceptance Testing (UAT) is chosen and will be conducted after the application is developed. The purpose of performing the test is to ensure and validate the application fully meets the requirements as stated in the Software Requirements Specification (SRS).

Table 3.11 Testing Plan

No.	Module	Activity	Pass/Fail	Comment
1	Login	User login		
2	Login	Forgot Password		
3	Login	Invalid email or password		
4	Login	Reset password		
5	Manage user data	View list of supervisors		
6	Manage user data	Update user data		
7	Access breach records	View list breach records		
8	Receive notifications	Receive non-compliance workers notification		
9	Manage PPE inventory	View list of PPE inventory		
10	Manage PPE inventory	Add new PPE		
11	Manage PPE inventory	Update PPE info		

12	Manage PPE inventory	Delete PPE info		

3.9 Gantt Chart

The Gantt Charts for PSM 1 and PSM 2 are attached in Appendix A and Appendix B respectively.

3.10 Potential Use of Proposed Solution

The purpose of developing this application is to achieve an individual’s safety in a construction site by ensuring every one of them comply to the rules of wearing a safety helmet through the use of safety helmet detection. The application will be used in construction sites for safety helmet detection on workers. Employers can then utilize the application to monitor the conditions in different sites. If a worker who is not wearing proper helmet is detected by the camera, notification on the issue will be sent to the employer. Through this, constant monitoring physically on the workers is not necessarily needed, more so time and energy will not be wasted on ensuring the workers’ compliance towards wearing proper safety helmet.

Besides, the application enables its users to manage PPE. This way, users of the application can keep track of the amount of PPE inventory to avoid PPE shortage.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter discusses about the implementation, as well as results of the project. The method of implementation refers to the development of the proposed application. Besides, the functionalities of the application will be mentioned in this chapter.

4.2 Development Process

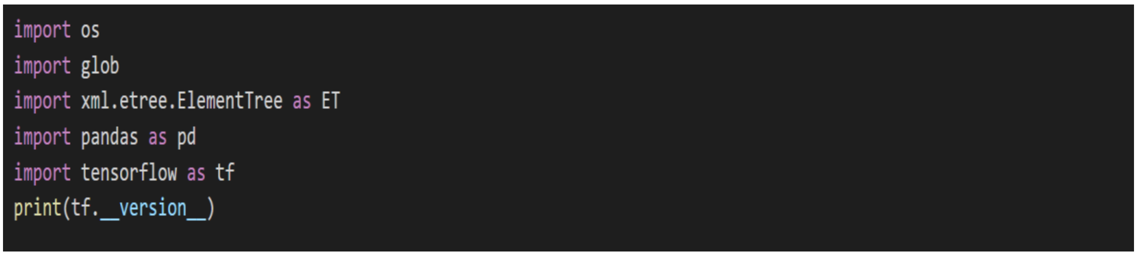
4.2.1 Object Detection Implementation

Figure 4.1 TensorFlow

TensorFlow, as shown in Figure 4.1, is an open-source library for large-scale machine learning and numerical computing. TensorFlow combines a variety of deep learning and machine learning models and techniques, also known as neural network. It is chosen to be used here because it offers for machine learning development. Developers may concentrate on the overarching logic of the program rather than solve on implementing algorithms or working out how to connect the result of one function to the input of another. The details are handled in the background via TensorFlow.

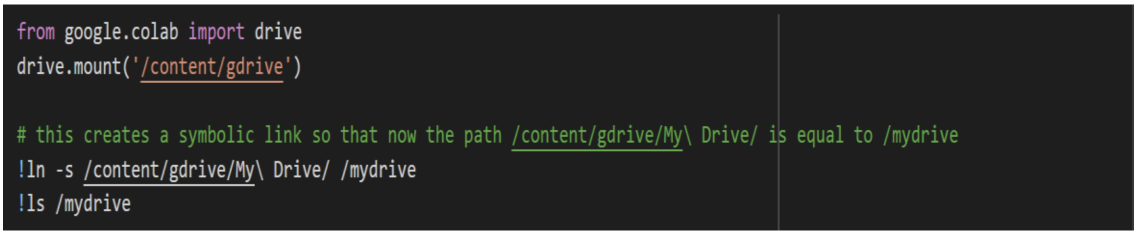
Building a computer vision model from scratch is particularly challenging since a large range of input data is needed to make the model generalize effectively. Therefore, transfer learning is utilized to make constructing our model simpler and faster. By merely retraining the top layers of a neural network, we can build upon a model that has already been trained and create far more trustworthy models that can train quickly and use much fewer datasets.

Here, the MobileNetV2 SSD FPN-Lite 320x320 pre-trained model is used. The COCO 2017 dataset, with pictures scaled to 320x320 resolution, was used to train the model. The reason this model was selected for this project is that TensorFlow Lite, often known as TFLite, an open-source toolkit for delivering machine learning models to edge devices like Android and iOS mobile, does not support all models for the time being. Only SSD models are supported by TFLite for now, excluding EfficientDet.



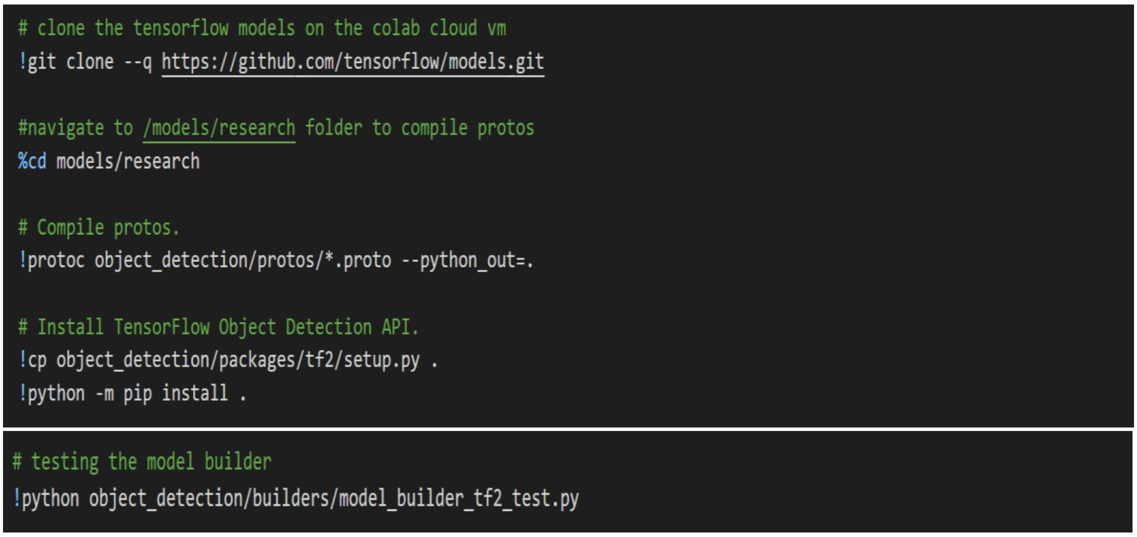
```
import os
import glob
import xml.etree.ElementTree as ET
import pandas as pd
import tensorflow as tf
print(tf.__version__)
```

Figure 4.2 Import libraries



```
from google.colab import drive
drive.mount('/content/gdrive')

# this creates a symbolic link so that now the path /content/gdrive/My\ Drive/ is equal to /mydrive
!ln -s /content/gdrive/My\ Drive/ /mydrive
!ls /mydrive
```

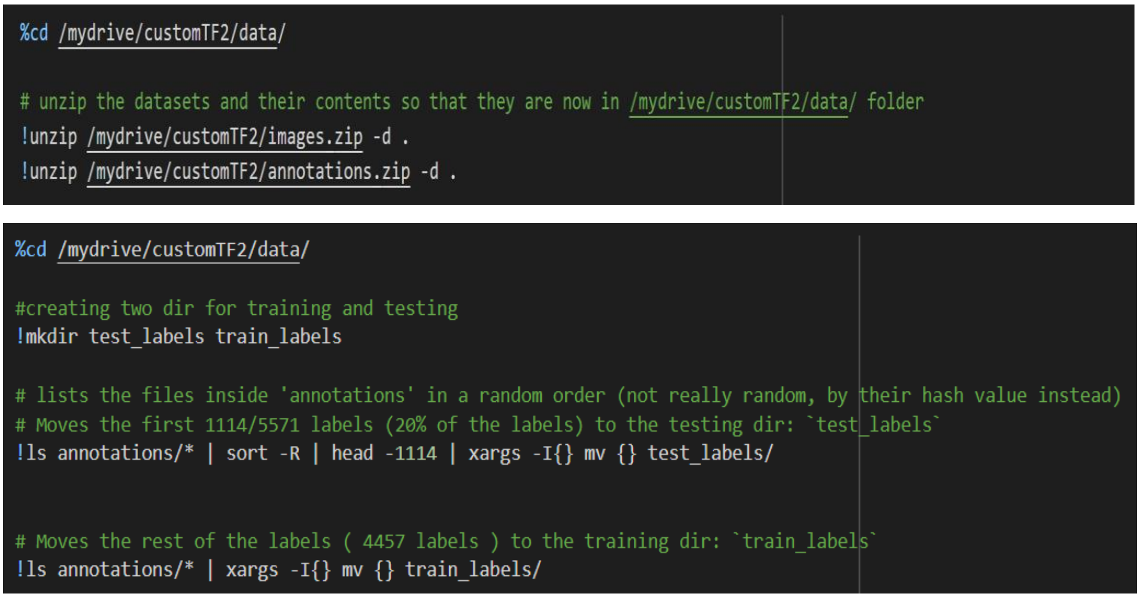
Figure 4.3 Mount drive and link to folder containing dataset images and annotations



```
# clone the tensorflow models on the colab cloud vm
!git clone --q https://github.com/tensorflow/models.git

#navigate to /models/research folder to compile protos
%cd models/research

# Compile protos.
!protoc object_detection/protos/*.proto --python_out=.

# Install TensorFlow Object Detection API.
!cp object_detection/packages/tf2/setup.py .
!python -m pip install .
```

```
# testing the model builder
!python object_detection/builders/model_builder_tf2_test.py
```

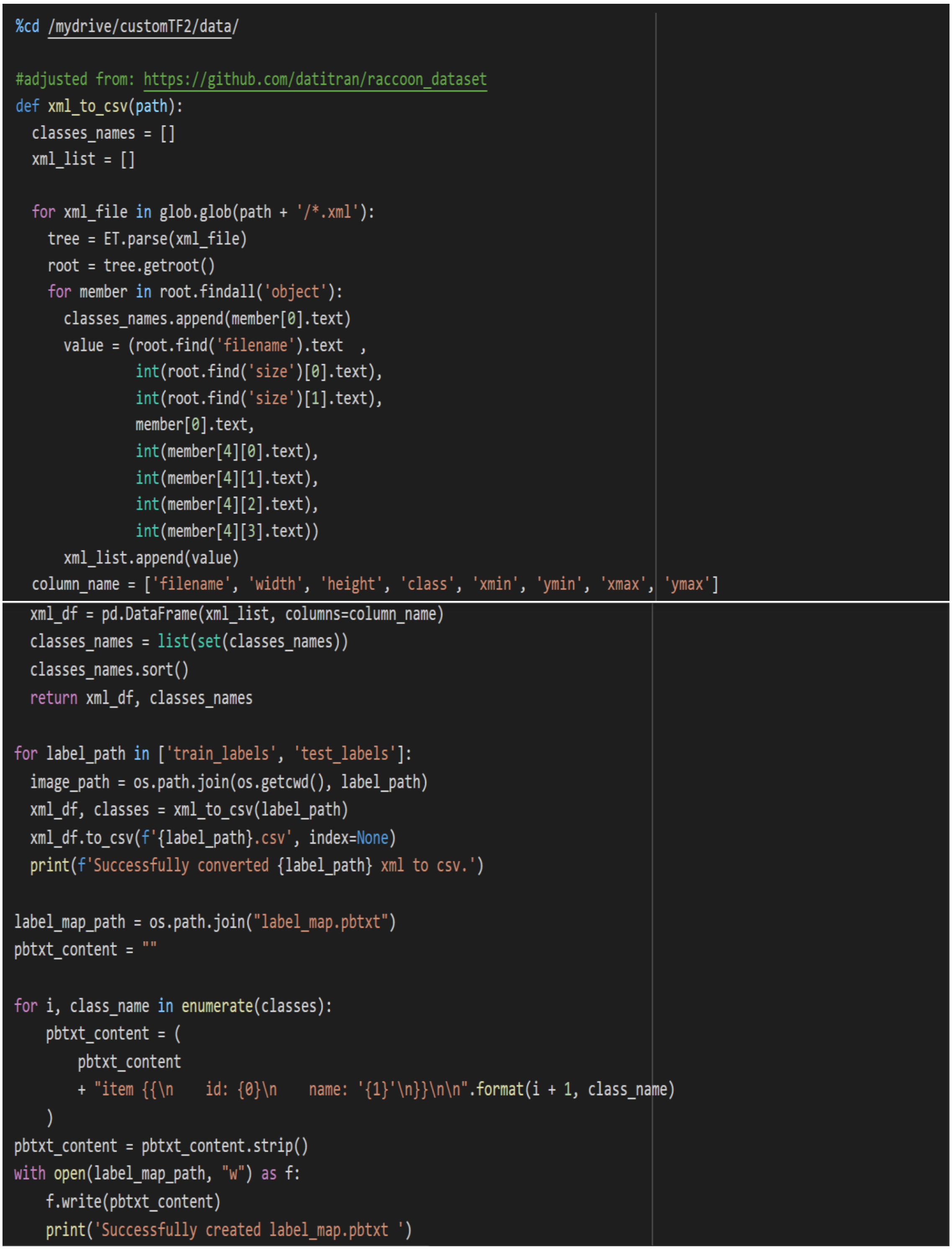
Figure 4.4 Clone the TensorFlow models git repository & Install TensorFlow Object Detection API



```
%cd /mydrive/customTF2/data/

# unzip the datasets and their contents so that they are now in /mydrive/customTF2/data/ folder
!unzip /mydrive/customTF2/images.zip -d .
!unzip /mydrive/customTF2/annotations.zip -d .
```

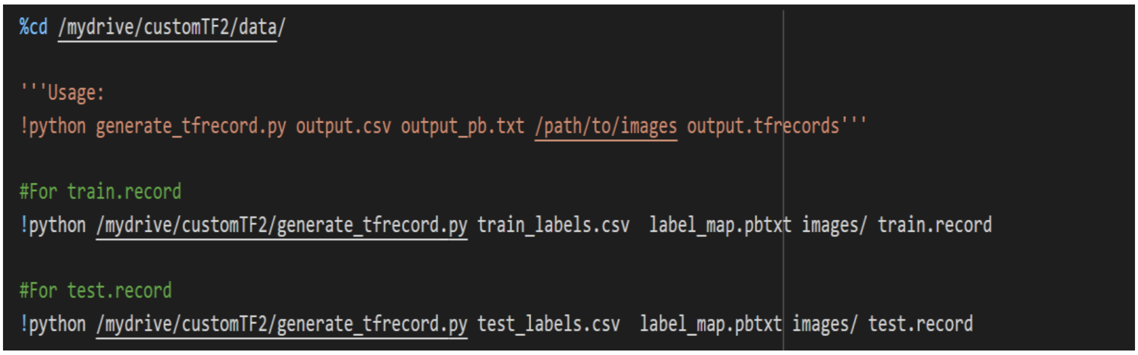
```
%cd /mydrive/customTF2/data/

#creating two dir for training and testing
!mkdir test_labels train_labels

# lists the files inside 'annotations' in a random order (not really random, by their hash value instead)
# Moves the first 1114/5571 labels (20% of the labels) to the testing dir: `test_labels`
!ls annotations/* | sort -R | head -1114 | xargs -I{} mv {} test_labels/


# Moves the rest of the labels ( 4457 labels ) to the training dir: `train_labels`
!ls annotations/* | xargs -I{} mv {} train_labels/
```

Figure 4.5 Create test labels & train labels from the dataset saved in Drive



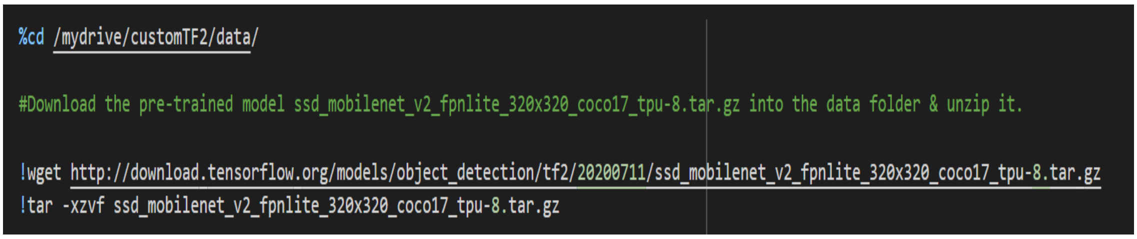
```
%cd /mydrive/customTF2/data/

#adjusted from: https://github.com/datitran/raccoon_dataset
def xml_to_csv(path):
  classes_names = []
  xml_list = []

  for xml_file in glob.glob(path + '/*.xml'):
    tree = ET.parse(xml_file)
    root = tree.getroot()
    for member in root.findall('object'):
      classes_names.append(member[0].text)
      value = (root.find('filename').text  ,
               int(root.find('size')[0].text),
               int(root.find('size')[1].text),
               member[0].text,
               int(member[4][0].text),
               int(member[4][1].text),
               int(member[4][2].text),
               int(member[4][3].text))
      xml_list.append(value)
  column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
  xml_df = pd.DataFrame(xml_list, columns=column_name)
  classes_names = list(set(classes_names))
  classes_names.sort()
  return xml_df, classes_names

for label_path in ['train_labels', 'test_labels']:
  image_path = os.path.join(os.getcwd(), label_path)
  xml_df, classes = xml_to_csv(label_path)
  xml_df.to_csv(f'{label_path}.csv', index=None)
  print(f'Successfully converted {label_path} xml to csv.')

label_map_path = os.path.join("label_map.pbtxt")
pbtxt_content = ""

for i, class_name in enumerate(classes):
    pbtxt_content = (
        pbtxt_content
        + "item {{\n    id: {0}\n    name: '{1}'\n}}\n\n".format(i + 1, class_name)
    )
pbtxt_content = pbtxt_content.strip()
with open(label_map_path, "w") as f:
    f.write(pbtxt_content)
    print('Successfully created label_map.pbtxt ')
```

Figure 4.6 Create the CSV files of the train and test labels and the "label_map.pbtxt" file containing the classes



```
%cd /mydrive/customTF2/data/

'''Usage:
!python generate_tfrecord.py output.csv output_pb.txt /path/to/images output.tfrecords'''

#For train.record
!python /mydrive/customTF2/generate_tfrecord.py train_labels.csv  label_map.pbtxt images/ train.record

#For test.record
!python /mydrive/customTF2/generate_tfrecord.py test_labels.csv  label_map.pbtxt images/ test.record
```

Figure 4.7 Create train record and test record files by using a python script that trains an SSD model for object-detection using TensorFlow 2.x



```
%cd /mydrive/customTF2/data/

#Download the pre-trained model ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz into the data folder & unzip it.

!wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz
!tar -xzvf ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz
```

Figure 4.8 Download pre-trained model checkpoint and make changes according to project specifications

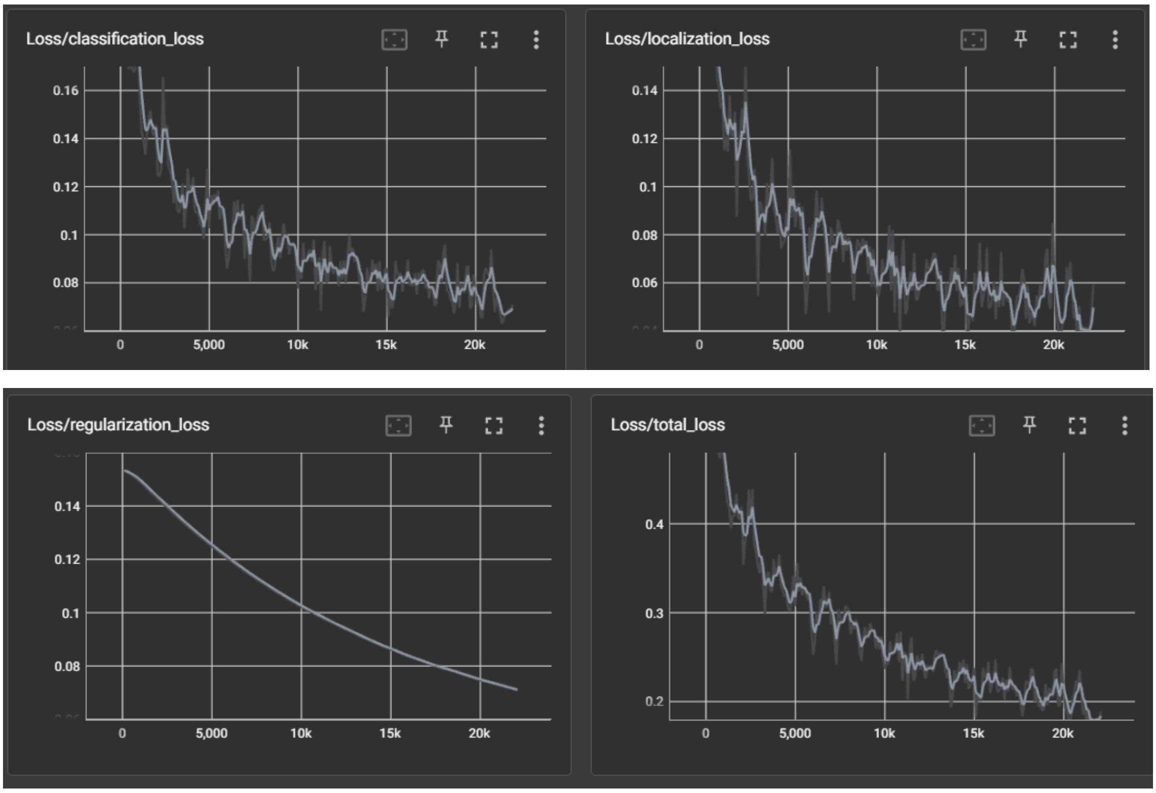
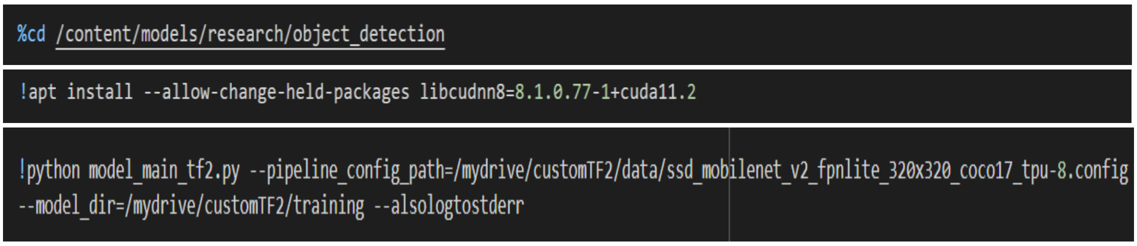


Figure 4.9 Loss visualization on TensorBoard

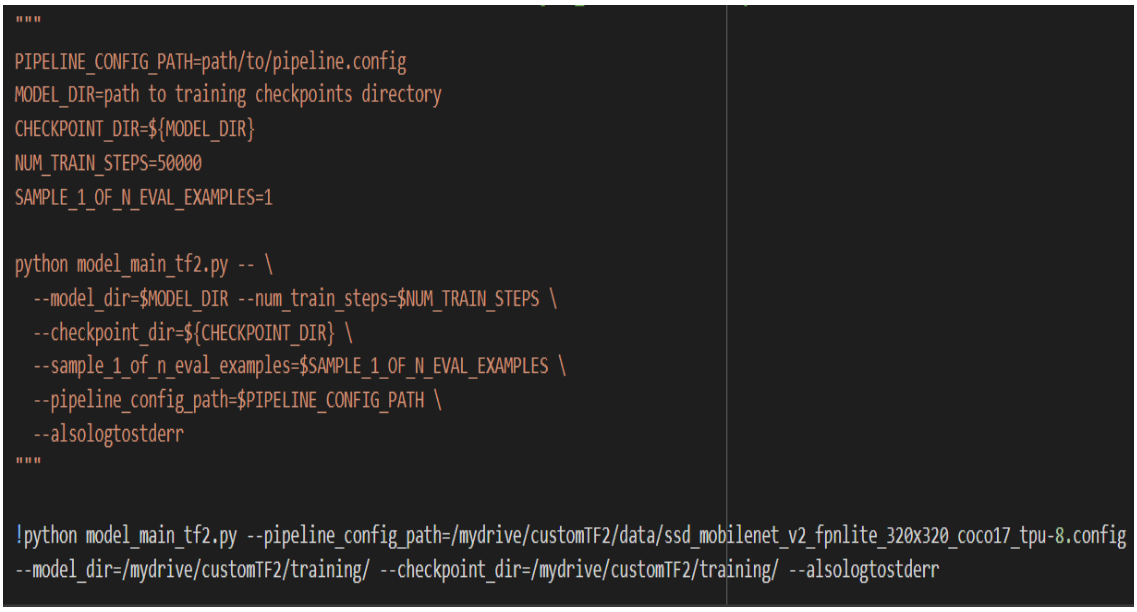


```
%cd /content/models/research/object_detection
```

```
!apt install --allow-change-held-packages libcudnn8=8.1.0.77-1+cuda11.2
```

```
!python model_main_tf2.py --pipeline_config_path=/mydrive/customTF2/data/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config
--model_dir=/mydrive/customTF2/training --alsologtostderr
```

Figure 4.10 Training using the model “model_main_tf2.py” from TensorFlow GitHub repository



```
"""
PIPELINE_CONFIG_PATH=path/to/pipeline.config
MODEL_DIR=path to training checkpoints directory
CHECKPOINT_DIR=${MODEL_DIR}
NUM_TRAIN_STEPS=50000
SAMPLE_1_OF_N_EVAL_EXAMPLES=1

python model_main_tf2.py -- \
  --model_dir=$MODEL_DIR --num_train_steps=$NUM_TRAIN_STEPS \
  --checkpoint_dir=${CHECKPOINT_DIR} \
  --sample_1_of_n_eval_examples=$SAMPLE_1_OF_N_EVAL_EXAMPLES \
  --pipeline_config_path=$PIPELINE_CONFIG_PATH \
  --alsologtostderr
"""

!python model_main_tf2.py --pipeline_config_path=/mydrive/customTF2/data/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config
--model_dir=/mydrive/customTF2/training/ --checkpoint_dir=/mydrive/customTF2/training/ --alsologtostderr
```

Figure 4.11 Evaluation on trained model

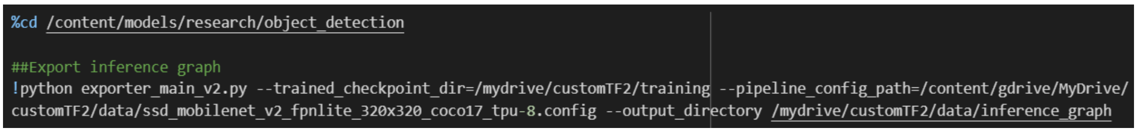
The results of the evaluation on the trained model after running the code in Figure 4.11 above are shown below:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.307
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.560
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.282
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.021
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.263
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.489
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.179
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.384
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.410
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.112
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.382
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.587
```

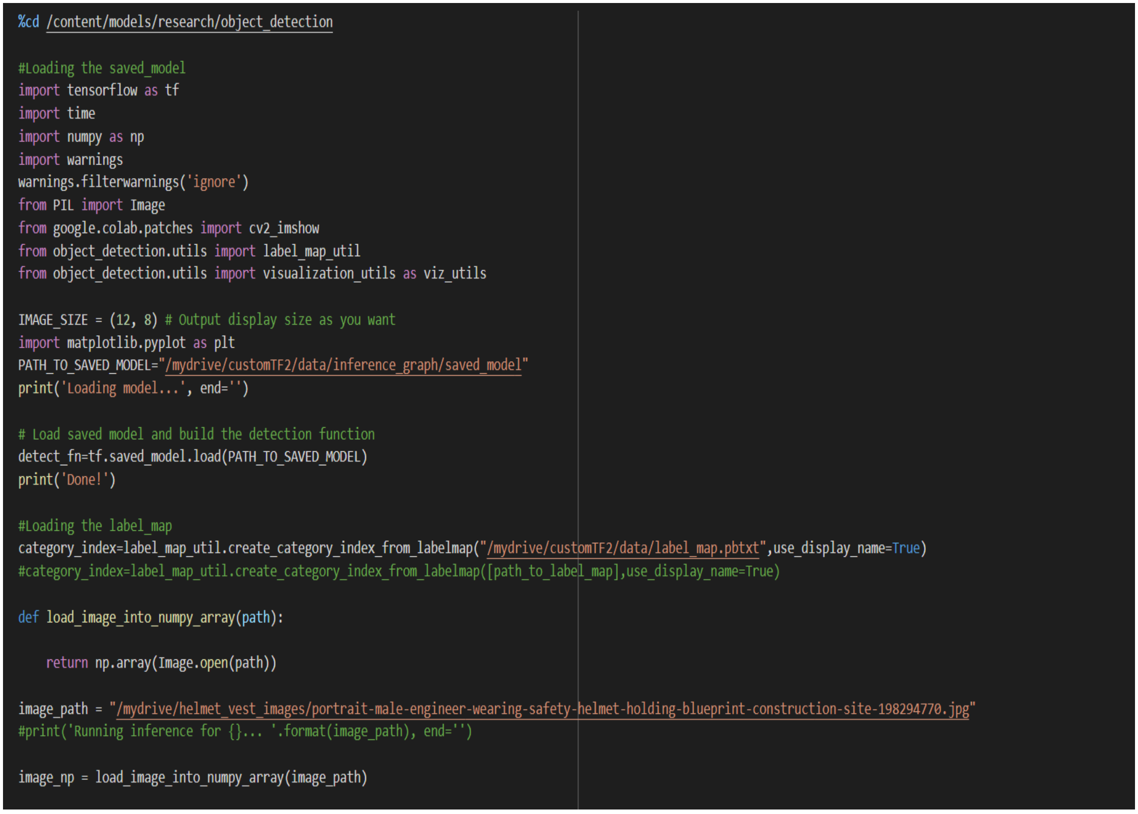
```
INFO:tensorflow:Eval metrics at step 50000
I1126 17:03:48.779237 140461284616064 model_lib_v2.py:1015] Eval metrics at step
50000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP: 0.306742
I1126 17:03:48.793360 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP: 0.306742
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.50IOU: 0.559943
I1126 17:03:48.795145 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP@.50IOU: 0.559943
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.75IOU: 0.282237
I1126 17:03:48.796798 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP@.75IOU: 0.282237
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (small): 0.021290
I1126 17:03:48.798462 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP (small): 0.021290
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (medium): 0.262669
I1126 17:03:48.800071 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP (medium): 0.262669
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (large): 0.489496
I1126 17:03:48.801797 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Precision/mAP (large): 0.489496
INFO:tensorflow:        + DetectionBoxes_Recall/AR@1: 0.178646
I1126 17:03:48.803462 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@1: 0.178646
INFO:tensorflow:        + DetectionBoxes_Recall/AR@10: 0.384194
I1126 17:03:48.805212 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@10: 0.384194
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100: 0.410218
I1126 17:03:48.806829 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@100: 0.410218
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (small): 0.112220
I1126 17:03:48.808360 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@100 (small): 0.112220
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (medium): 0.381655
I1126 17:03:48.809901 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@100 (medium): 0.381655
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (large): 0.587071
I1126 17:03:48.812524 140461284616064 model_lib_v2.py:1018]         +
DetectionBoxes_Recall/AR@100 (large): 0.587071
INFO:tensorflow:        + Loss/localization_loss: 0.690351
I1126 17:03:48.813660 140461284616064 model_lib_v2.py:1018]         +
Loss/localization_loss: 0.690351
INFO:tensorflow:        + Loss/classification_loss: 0.780754
I1126 17:03:48.814881 140461284616064 model_lib_v2.py:1018]         +
Loss/classification_loss: 0.780754
INFO:tensorflow:        + Loss/regularization_loss: 0.050073
I1126 17:03:48.816085 140461284616064 model_lib_v2.py:1018]         +
Loss/regularization_loss: 0.050073
```

```
INFO:tensorflow:          + Loss/total_loss: 1.521178
I1126 17:03:48.817217 140461284616064 model_lib_v2.py:1018]          +
Loss/total_loss: 1.521178
```

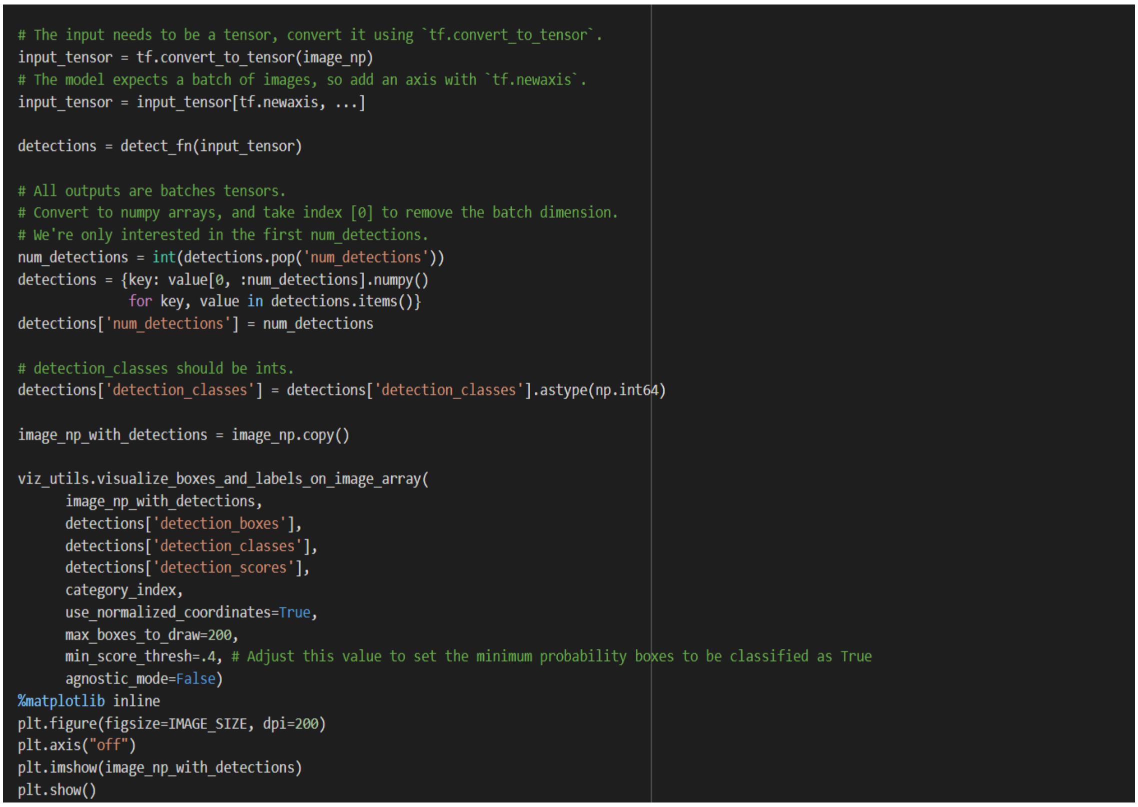
The total loss achieved by the trained model is 1.521178, where total loss is the sum of localization loss, classification loss, and regularization loss. Localization loss achieved a result of 0.690351, classification loss has a result of 0.780754, whereas regularization loss is at 0.050073.



```
%cd /content/models/research/object_detection

##Export inference graph
!python exporter_main_v2.py --trained_checkpoint_dir=/mydrive/customTF2/training --pipeline_config_path=/content/gdrive/MyDrive/
customTF2/data/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config --output_directory /mydrive/customTF2/data/inference_graph
```

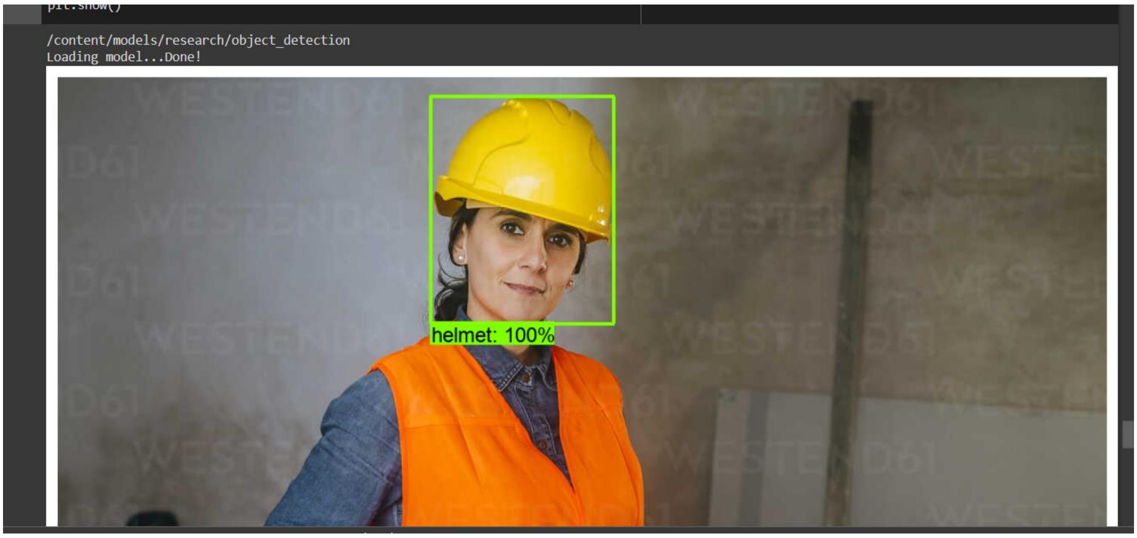


```
%cd /content/models/research/object_detection

#Loading the saved_model
import tensorflow as tf
import time
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from PIL import Image
from google.colab.patches import cv2_imshow
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils

IMAGE_SIZE = (12, 8) # Output display size as you want
import matplotlib.pyplot as plt
PATH_TO_SAVED_MODEL="/mydrive/customTF2/data/inference_graph/saved_model"
print('Loading model...', end='')

# Load saved model and build the detection function
detect_fn=tf.saved_model.load(PATH_TO_SAVED_MODEL)
print('Done!')

#Loading the label_map
category_index=label_map_util.create_category_index_from_labelmap("/mydrive/customTF2/data/label_map.pbtxt",use_display_name=True)
#category_index=label_map_util.create_category_index_from_labelmap([path_to_label_map],use_display_name=True)

def load_image_into_numpy_array(path):

    return np.array(Image.open(path))

image_path = "/mydrive/helmet_vest_images/portrait-male-engineer-wearing-safety-helmet-holding-blueprint-construction-site-198294770.jpg"
#print('Running inference for {}... '.format(image_path), end='')

image_np = load_image_into_numpy_array(image_path)
```



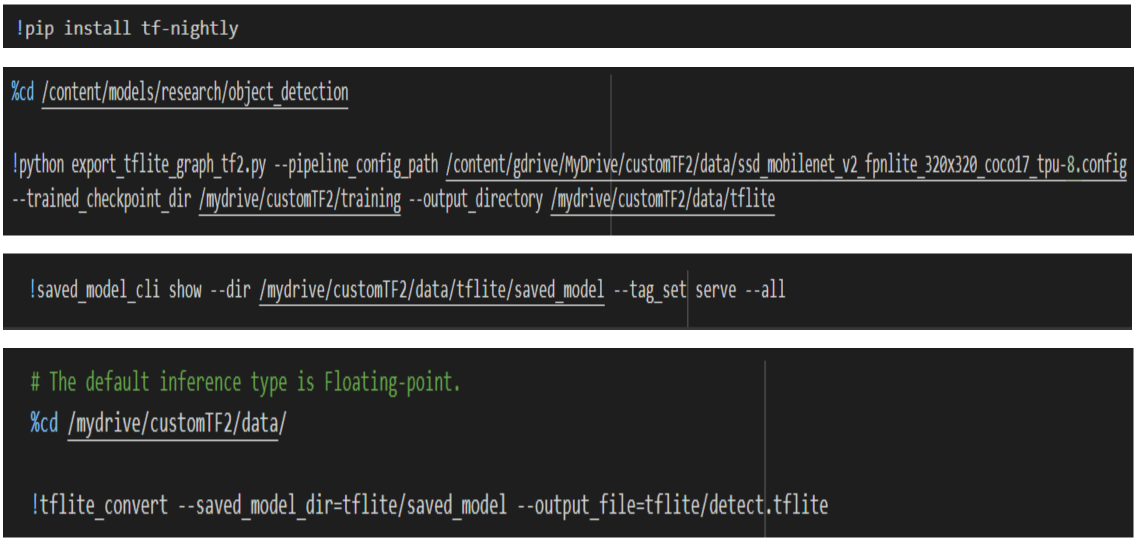
```
# The input needs to be a tensor, convert it using `tf.convert_to_tensor`.
input_tensor = tf.convert_to_tensor(image_np)
# The model expects a batch of images, so add an axis with `tf.newaxis`.
input_tensor = input_tensor[tf.newaxis, ...]

detections = detect_fn(input_tensor)

# All outputs are batches tensors.
# Convert to numpy arrays, and take index [0] to remove the batch dimension.
# We're only interested in the first num_detections.
num_detections = int(detections.pop('num_detections'))
detections = {key: value[0, :num_detections].numpy()
              for key, value in detections.items()}
detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
      image_np_with_detections,
      detections['detection_boxes'],
      detections['detection_classes'],
      detections['detection_scores'],
      category_index,
      use_normalized_coordinates=True,
      max_boxes_to_draw=200,
      min_score_thresh=.4, # Adjust this value to set the minimum probability boxes to be classified as True
      agnostic_mode=False)
%matplotlib inline
plt.figure(figsize=IMAGE_SIZE, dpi=200)
plt.axis("off")
plt.imshow(image_np_with_detections)
plt.show()
```

Figure 4.12 Test the trained model on images



```
plt.show()
/content/models/research/object_detection
Loading model...Done!
```

Figure 4.13 Helmet detected



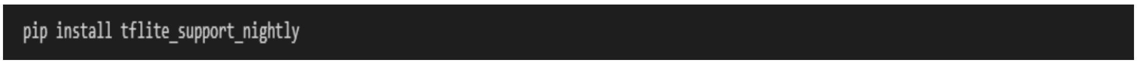
```
!pip install tf-nightly
```

```
%cd /content/models/research/object_detection

!python export_tflite_graph_tf2.py --pipeline_config_path /content/gdrive/MyDrive/customTF2/data/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config
--trained_checkpoint_dir /mydrive/customTF2/training --output_directory /mydrive/customTF2/data/tflite
```

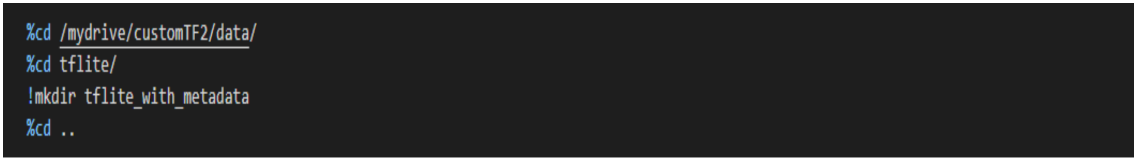
```
!saved_model_cli show --dir /mydrive/customTF2/data/tflite/saved_model --tag_set serve --all
```

```
# The default inference type is Floating-point.
%cd /mydrive/customTF2/data/

!tflite_convert --saved_model_dir=tflite/saved_model --output_file=tflite/detect.tflite
```

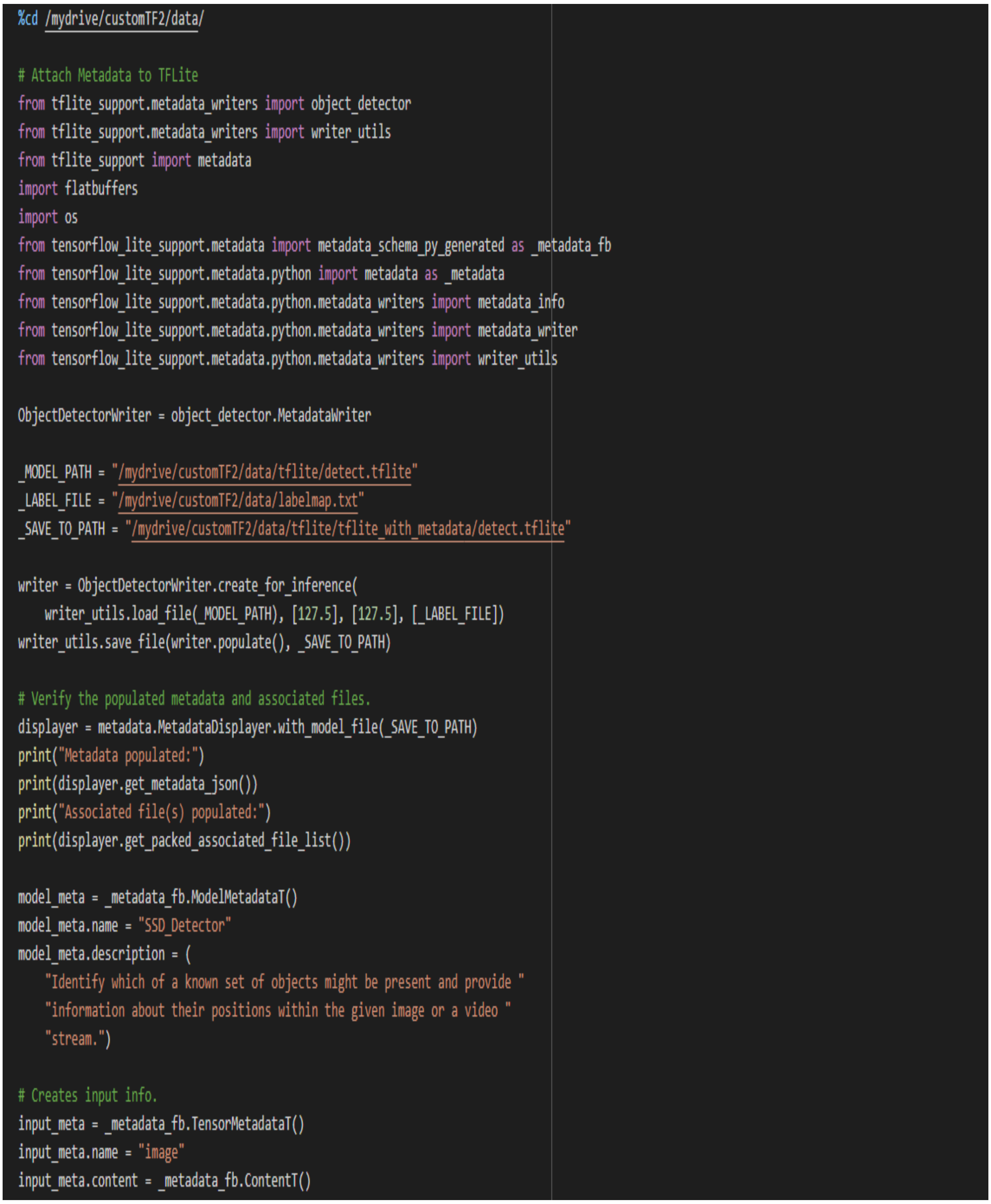
Figure 4.14 Export SSD TFLite graph and convert TF saved model to TFLite model



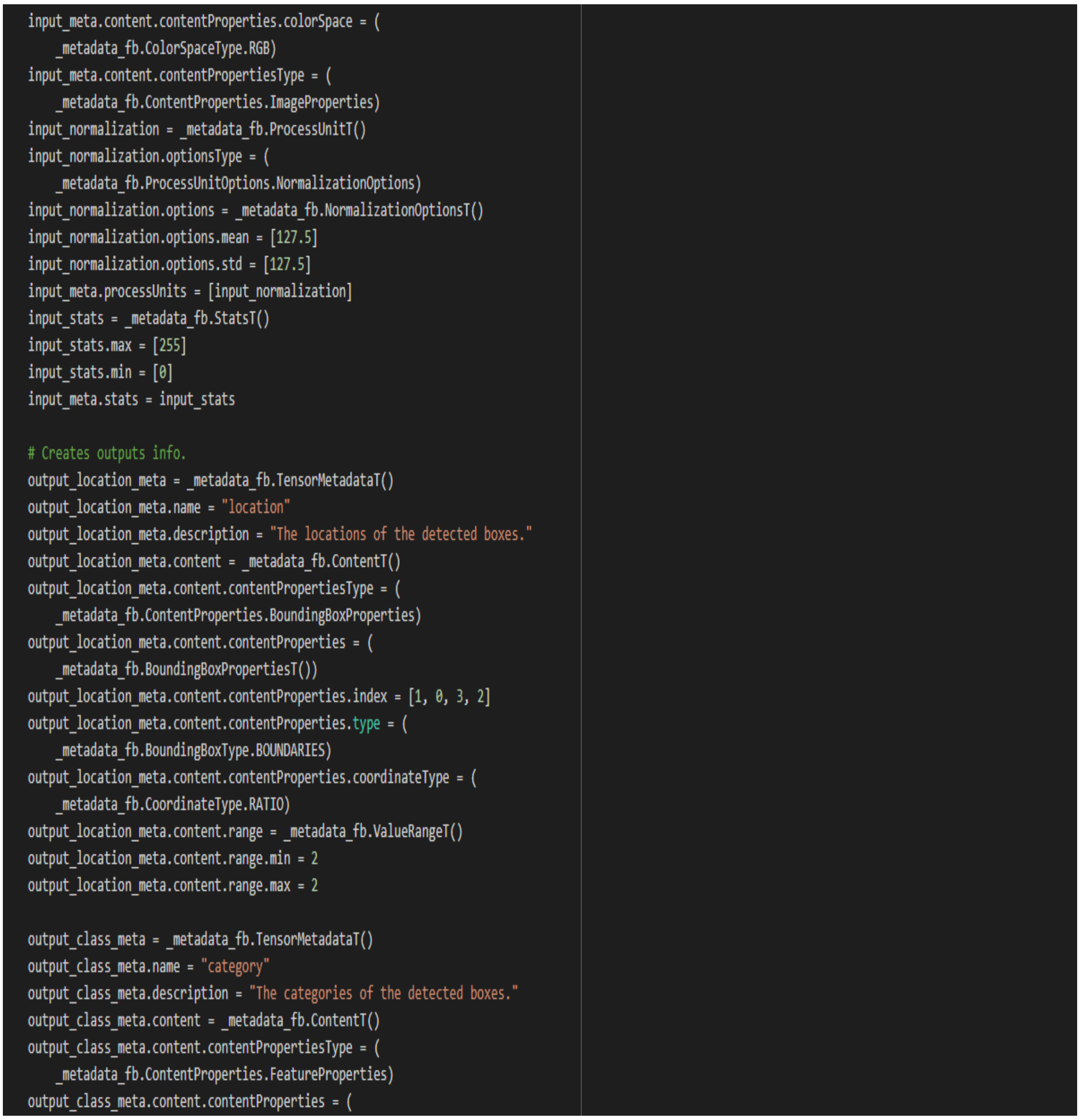
```
pip install tflite_support_nightly
```



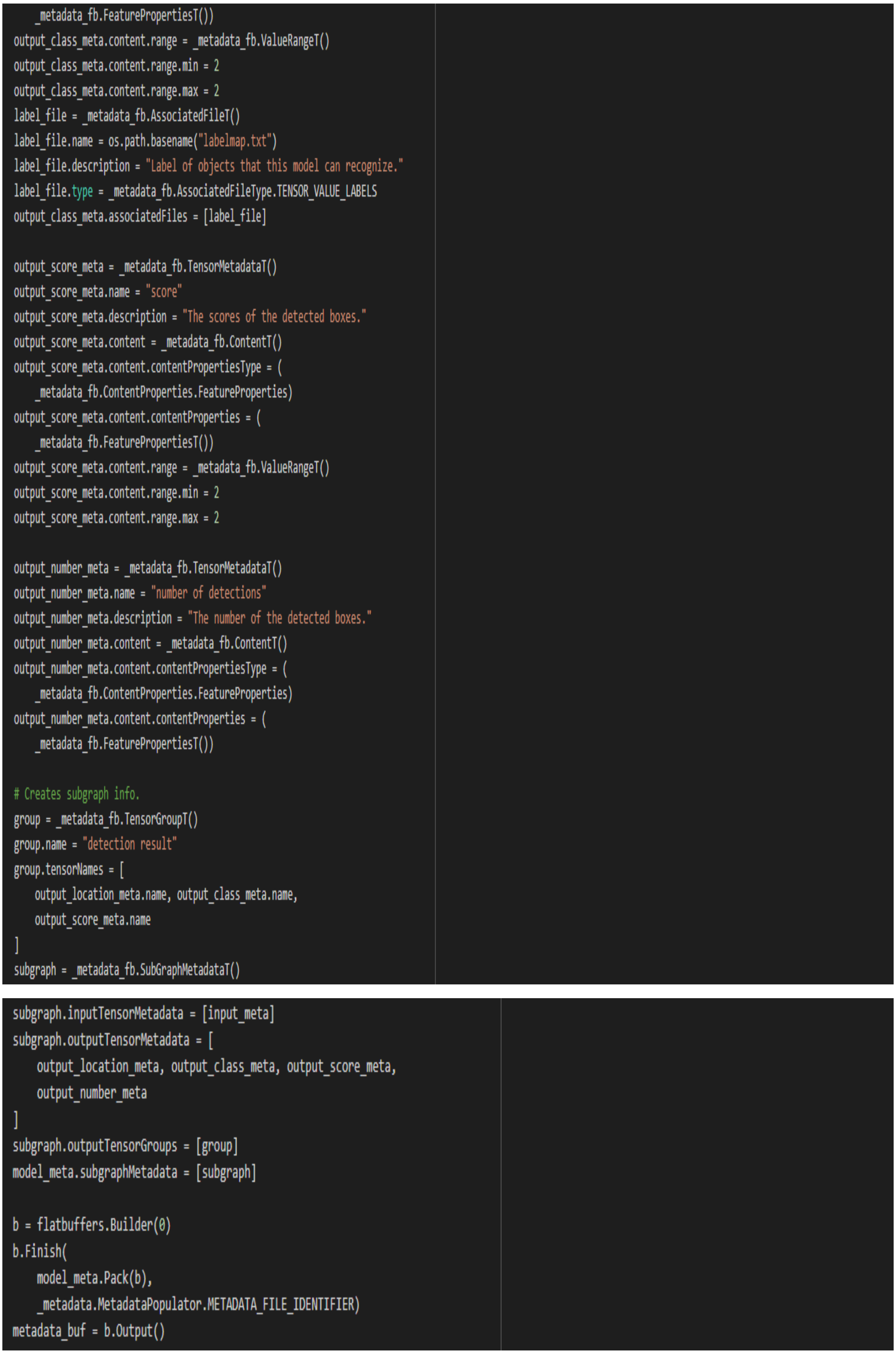
```
%cd /mydrive/customTF2/data/
%cd tflite/
!mkdir tflite_with_metadata
%cd ..
```



```
%cd /mydrive/customTF2/data/

# Attach Metadata to TFLite
from tflite_support.metadata_writers import object_detector
from tflite_support.metadata_writers import writer_utils
from tflite_support import metadata
import flatbuffers
import os
from tensorflow_lite_support.metadata import metadata_schema_py_generated as _metadata_fb
from tensorflow_lite_support.metadata.python import metadata as _metadata
from tensorflow_lite_support.metadata.python.metadata_writers import metadata_info
from tensorflow_lite_support.metadata.python.metadata_writers import metadata_writer
from tensorflow_lite_support.metadata.python.metadata_writers import writer_utils

ObjectDetectorWriter = object_detector.MetadataWriter

_MODEL_PATH = "/mydrive/customTF2/data/tflite/detect.tflite"
_LABEL_FILE = "/mydrive/customTF2/data/labelmap.txt"
_SAVE_TO_PATH = "/mydrive/customTF2/data/tflite/tflite_with_metadata/detect.tflite"

writer = ObjectDetectorWriter.create_for_inference(
    writer_utils.load_file(_MODEL_PATH), [127.5], [127.5], [_LABEL_FILE])
writer_utils.save_file(writer.populate(), _SAVE_TO_PATH)

# Verify the populated metadata and associated files.
displayer = metadata.MetadataDisplayer.with_model_file(_SAVE_TO_PATH)
print("Metadata populated:")
print(displayer.get_metadata_json())
print("Associated file(s) populated:")
print(displayer.get_packed_associated_file_list())

model_meta = _metadata_fb.ModelMetadataT()
model_meta.name = "SSD_Detector"
model_meta.description = (
    "Identify which of a known set of objects might be present and provide "
    "information about their positions within the given image or a video "
    "stream.")

# Creates input info.
input_meta = _metadata_fb.TensorMetadataT()
input_meta.name = "image"
input_meta.content = _metadata_fb.ContentT()
```



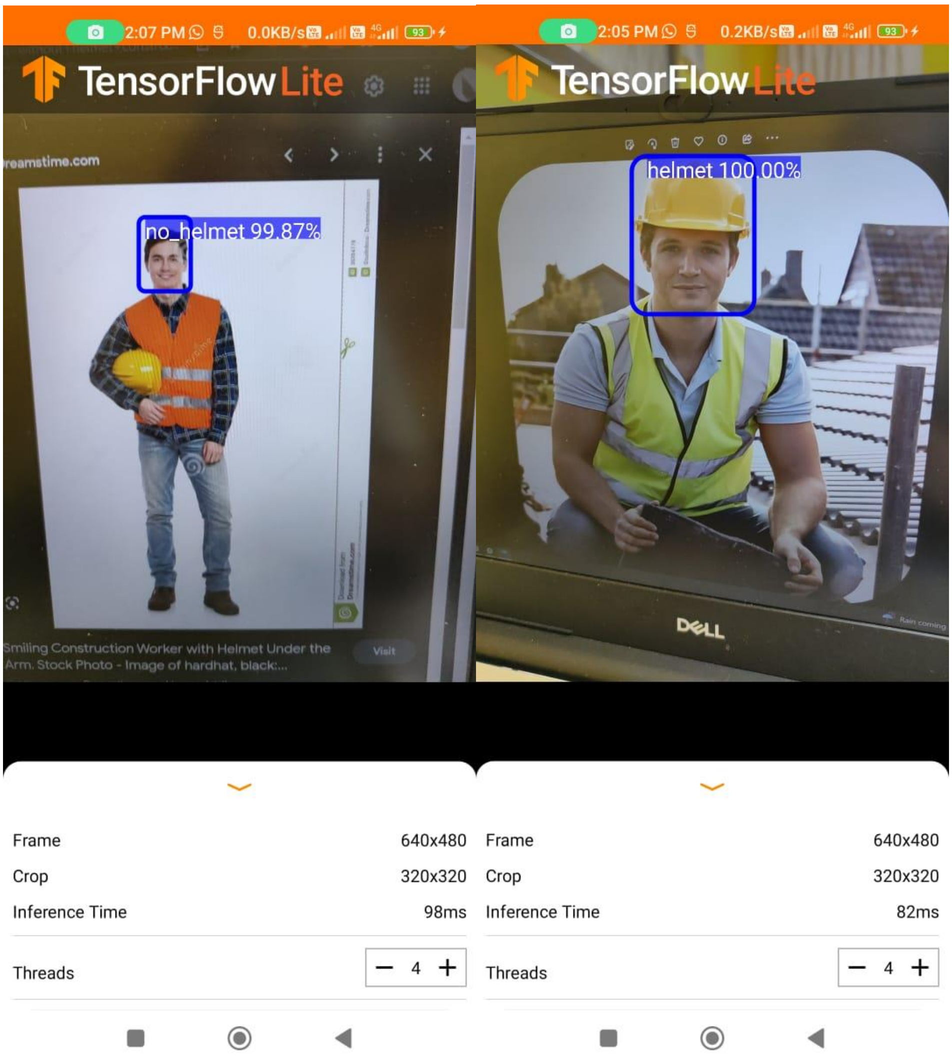
```
input_meta.content.contentProperties.colorSpace = (
    _metadata_fb.ColorSpaceType.RGB)
input_meta.content.contentPropertiesType = (
    _metadata_fb.ContentProperties.ImageProperties)
input_normalization = _metadata_fb.ProcessUnitT()
input_normalization.optionsType = (
    _metadata_fb.ProcessUnitOptions.NormalizationOptions)
input_normalization.options = _metadata_fb.NormalizationOptionsT()
input_normalization.options.mean = [127.5]
input_normalization.options.std = [127.5]
input_meta.processUnits = [input_normalization]
input_stats = _metadata_fb.StatsT()
input_stats.max = [255]
input_stats.min = [0]
input_meta.stats = input_stats

# Creates outputs info.
output_location_meta = _metadata_fb.TensorMetadataT()
output_location_meta.name = "location"
output_location_meta.description = "The locations of the detected boxes."
output_location_meta.content = _metadata_fb.ContentT()
output_location_meta.content.contentPropertiesType = (
    _metadata_fb.ContentProperties.BoundingBoxProperties)
output_location_meta.content.contentProperties = (
    _metadata_fb.BoundingBoxPropertiesT())
output_location_meta.content.contentProperties.index = [1, 0, 3, 2]
output_location_meta.content.contentProperties.type = (
    _metadata_fb.BoundingBoxType.BOUNDARIES)
output_location_meta.content.contentProperties.coordinateType = (
    _metadata_fb.CoordinateType.RATIO)
output_location_meta.content.range = _metadata_fb.ValueRangeT()
output_location_meta.content.range.min = 2
output_location_meta.content.range.max = 2

output_class_meta = _metadata_fb.TensorMetadataT()
output_class_meta.name = "category"
output_class_meta.description = "The categories of the detected boxes."
output_class_meta.content = _metadata_fb.ContentT()
output_class_meta.content.contentPropertiesType = (
    _metadata_fb.ContentProperties.FeatureProperties)
output_class_meta.content.contentProperties = (
```



```
    _metadata_fb.FeaturePropertiesT())
output_class_meta.content.range = _metadata_fb.ValueRangeT()
output_class_meta.content.range.min = 2
output_class_meta.content.range.max = 2
label_file = _metadata_fb.AssociatedFileT()
label_file.name = os.path.basename("labelmap.txt")
label_file.description = "Label of objects that this model can recognize."
label_file.type = _metadata_fb.AssociatedFileType.TENSOR_VALUE_LABELS
output_class_meta.associatedFiles = [label_file]

output_score_meta = _metadata_fb.TensorMetadataT()
output_score_meta.name = "score"
output_score_meta.description = "The scores of the detected boxes."
output_score_meta.content = _metadata_fb.ContentT()
output_score_meta.content.contentPropertiesType = (
    _metadata_fb.ContentProperties.FeatureProperties)
output_score_meta.content.contentProperties = (
    _metadata_fb.FeaturePropertiesT())
output_score_meta.content.range = _metadata_fb.ValueRangeT()
output_score_meta.content.range.min = 2
output_score_meta.content.range.max = 2

output_number_meta = _metadata_fb.TensorMetadataT()
output_number_meta.name = "number of detections"
output_number_meta.description = "The number of the detected boxes."
output_number_meta.content = _metadata_fb.ContentT()
output_number_meta.content.contentPropertiesType = (
    _metadata_fb.ContentProperties.FeatureProperties)
output_number_meta.content.contentProperties = (
    _metadata_fb.FeaturePropertiesT())

# Creates subgraph info.
group = _metadata_fb.TensorGroupT()
group.name = "detection result"
group.tensorNames = [
    output_location_meta.name, output_class_meta.name,
    output_score_meta.name
]
subgraph = _metadata_fb.SubGraphMetadataT()
```

```
subgraph.inputTensorMetadata = [input_meta]
subgraph.outputTensorMetadata = [
    output_location_meta, output_class_meta, output_score_meta,
    output_number_meta
]
subgraph.outputTensorGroups = [group]
model_meta.subgraphMetadata = [subgraph]

b = flatbuffers.Builder(0)
b.Finish(
    model_meta.Pack(b),
    _metadata.MetadataPopulator.METADATA_FILE_IDENTIFIER)
metadata_buf = b.Output()
```

Figure 4.15 Create TFLite metadata



Frame 640x480
Crop 320x320
Inference Time 98ms
Threads − 4 +

Frame 640x480
Crop 320x320
Inference Time 82ms
Threads − 4 +

Figure 4.16 Deploying to TFLite app that detects 2 classes, helmet and no_helmet

4.2.2 Integrated Development Environment (IDE)

The IDEs used in the development of the application are Google Colaboratory, or "Colab" for short, and Android Studio.

Figure 4.17 Google Colaboratory

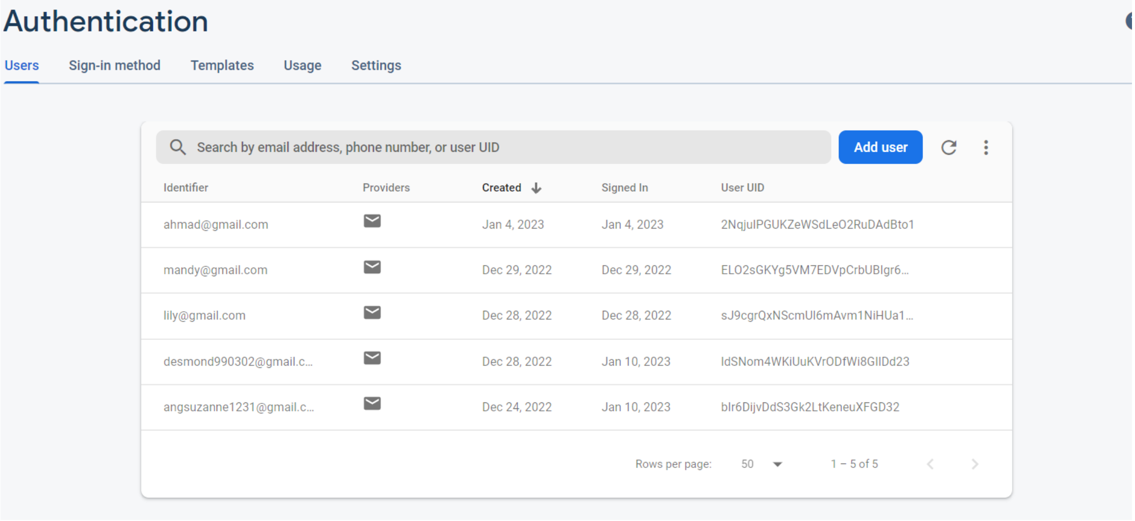
Figure 4.17 shows Google Colab which enables developers to write and run Python code through their browser. It is a hosted Jupyter notebook with a great free edition that provides free access to Google processing resources like GPUs and TPUs and requires no setup.

Figure 4.18 Android Studio

On the other hand, Android Studio, as shown in Figure 4.18 allows us to build android applications and it supports languages like Java and Kotlin for the application development. Java language is used for building the proposed application instead of Kotlin.

Figure 4.19 Firebase Realtime Database

Figure 4.19 is the Firebase Realtime Database which is a cloud-hosted database. The database stores data as JSON and syncs with each connected client in real-time. Firebase Realtime Database stores data and synchronizes with NoSQL cloud database. Data is synced across all clients in real time and is available even when the app is offline. Firebase Realtime Database is used because it uses data synchronization. Whenever the data changes, any connected device will receive the updated data within milliseconds. It provides a collaborative, immersive experience without any network code considerations.



Identifier	Providers	Created ↓	Signed In	User UID
ahmad@gmail.com	✉	Jan 4, 2023	Jan 4, 2023	2NqjuIPGUKZeWSdLeO2RuDAdBto1
mandy@gmail.com	✉	Dec 29, 2022	Dec 29, 2022	ELO2sGKYg5VM7EDVpCrbUBlgr6...
lily@gmail.com	✉	Dec 28, 2022	Dec 28, 2022	sJ9cgrQxNScmUl6mAvm1NiHUa1...
desmond990302@gmail.c...	✉	Dec 28, 2022	Jan 10, 2023	IdSNom4WKiUuKVrODfWi8GIIDd23
angsuzanne1231@gmail.c...	✉	Dec 24, 2022	Jan 10, 2023	blr6DijvDdS3Gk2LtKeneuXFGD32

Figure 4.20 Firebase Realtime Database Authentication page

4.3 System Output and Results

The system is a mobile application which supports Android phones. Figure 4.21 shows the interface for user to log into the application. The application requires the user's email and password to perform further actions on the application. Registering a new account for the manager (admin of the company) and supervisors (normal users of the company) to access the app requires the application admin to create and delete the user accounts.

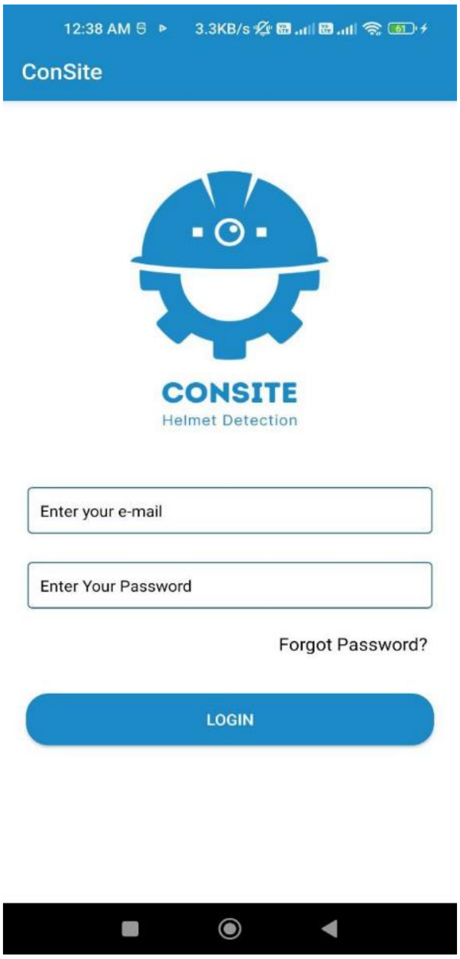
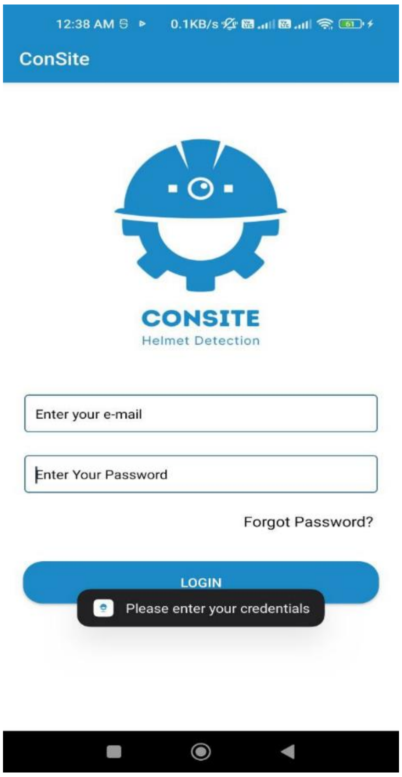


Figure 4.21 Login interface

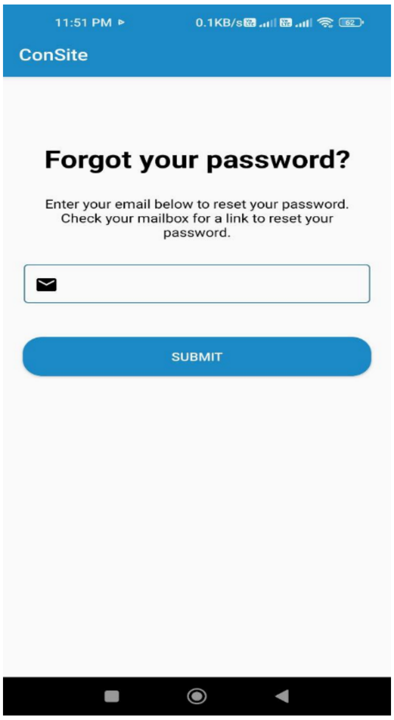
When the user does not enter an email or password or both when logging into the application, an error message will be popped up as shown in Figure 4.22 below to remind the user not to leave the fields blank.



Please enter your credentials

Figure 4.22 Error message for not providing email or password or both during login

Figure 4.23 below describes the interface for the user to reset his account's password by providing the email that is used to access the account.

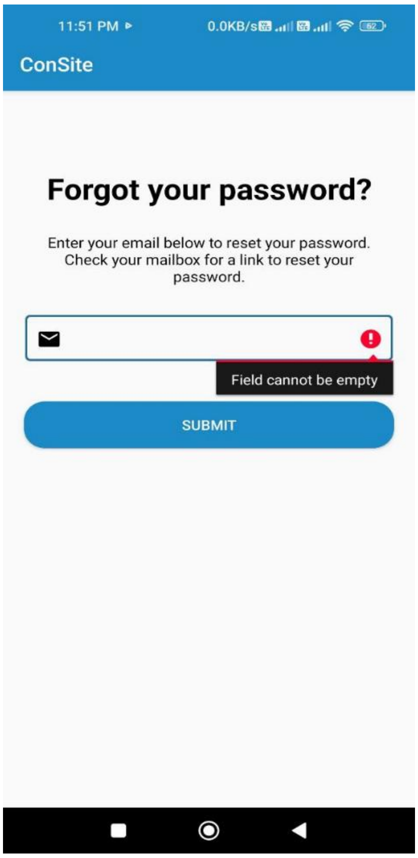


Forgot your password?

Enter your email below to reset your password. Check your mailbox for a link to reset your password.

Figure 4.23 Forgot password interface

When the user does not enter the email for resetting the password, an error message will be popped up as shown in Figure 4.24 below to remind the user not to leave the fields blank.



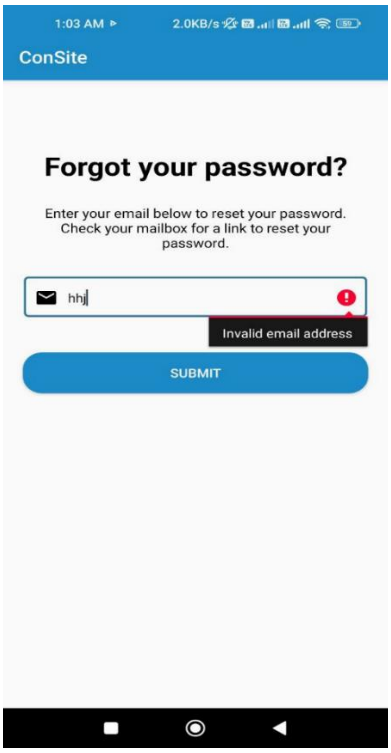
Forgot your password?

Enter your email below to reset your password. Check your mailbox for a link to reset your password.

Field cannot be empty

Figure 4.24 Error message for not providing an email during password reset

If an invalid email is given when resetting the password of an account, an error message as shown in Figure 4.25 below will be popped up to inform the user to provide a proper one to reset the password.



Forgot your password?

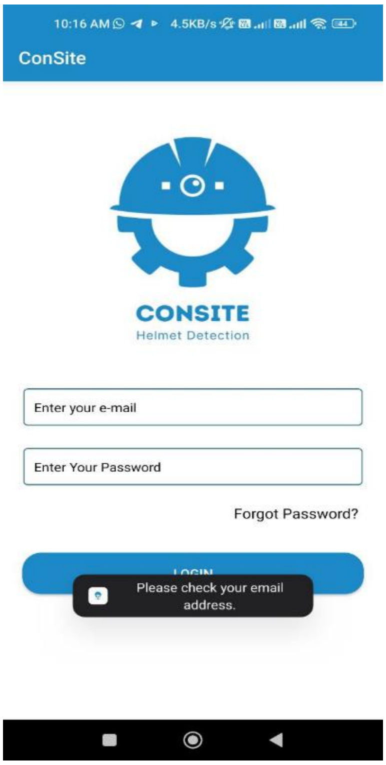
Enter your email below to reset your password. Check your mailbox for a link to reset your password.

hhj

Invalid email address

Figure 4.25 Error message for not providing a proper email during password reset

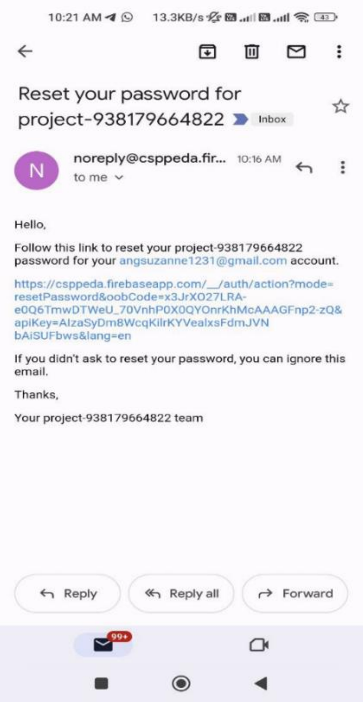
If the reset password link has been successfully sent to the email entered for resetting the password, a pop-up message just as in Figure 4.26 is shown to tell the user to check the email that has been sent to his inbox.



Please check your email address.

Figure 4.26 Pop up message after email account for password reset is valid

Figure 4.27 below shows the link that has been sent to the user's email to reset the password. From here, the user needs to click on the link given.



Reset your password for project-938179664822

noreply@csppeda.fir... 10:16 AM
to me

Hello,

Follow this link to reset your project-938179664822 password for your angsuzanne1231@gmail.com account.

https://csppeda.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=x3JrXO27LRA-e0Q6TmwDTWeU_70VnhP0X0QYOnrKhMcAAAGFnp2-zQ&apiKey=AIzaSyDm8WcqKilrKYVealxsFdmJVNbAiSUFbws&lang=en

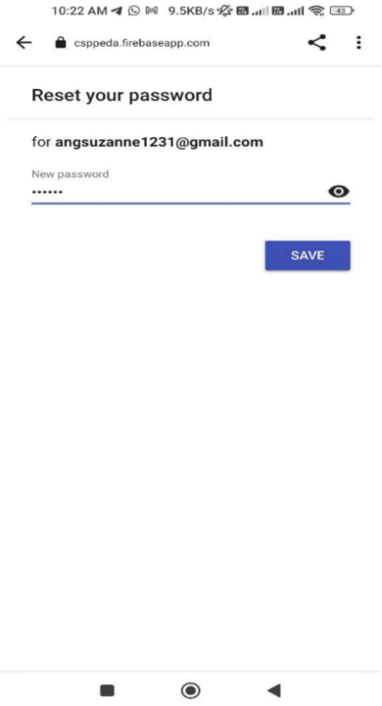
If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-938179664822 team

Figure 4.27 Email containing the password reset link

After the password reset link is clicked, the user is redirected to this page as shown in Figure 4.28. The new password will be entered for the registered email account.



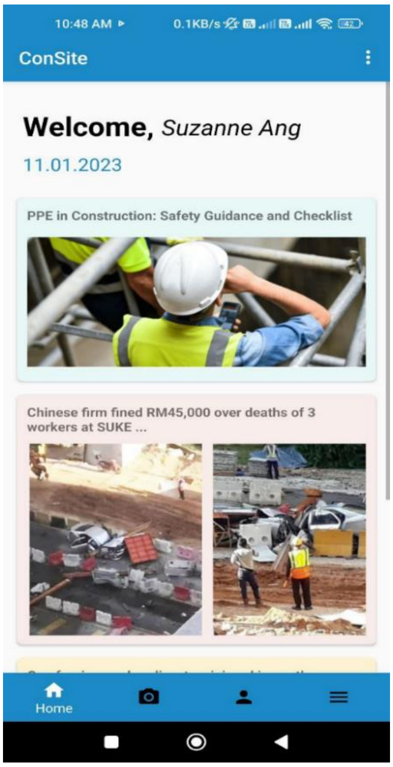
Reset your password

for angsuzanne1231@gmail.com

New password
••••••

Figure 4.28 Text field for entering new password for the email account

After the user is successfully logged into the application, the user will be presented with different interfaces according to the type of user he is. The admin of the company will have a bottom navigation bar with 4 tabs as shown in Figure 4.29, whereas the supervisors will have a bottom navigation bar with 3 tabs as shown in Figure 4.30.



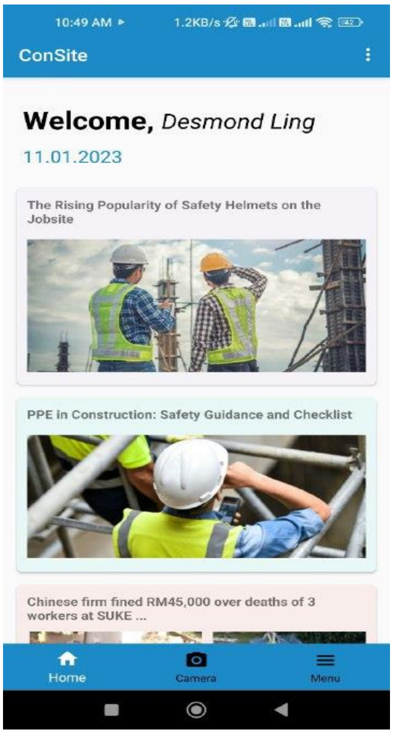
Welcome, *Suzanne Ang*

11.01.2023

PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

Figure 4.29 Admin home interface when admin logs in



Welcome, *Desmond Ling*

11.01.2023

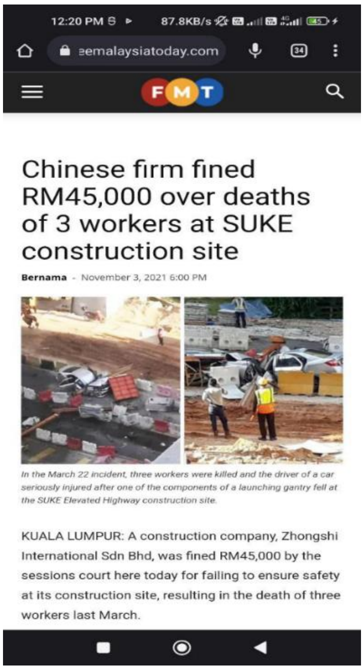
The Rising Popularity of Safety Helmets on the Jobsite

PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

Figure 4.30 Supervisor home interface when supervisor logs in

When any article on the home interface is clicked, the user will be redirected to the article website just as in Figure 4.31 below.



Chinese firm fined RM45,000 over deaths of 3 workers at SUKE construction site

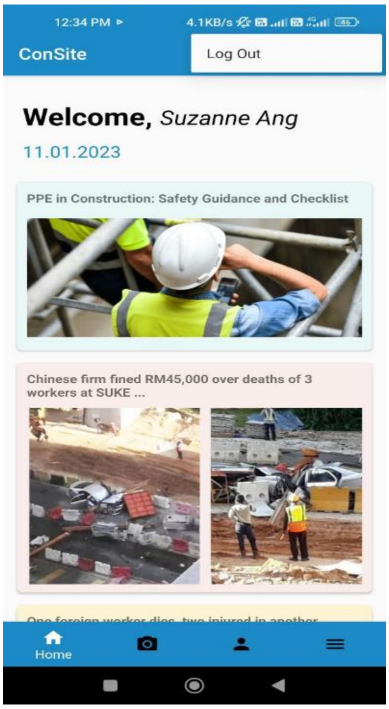
Bernama - November 3, 2021 6:00 PM

In the March 22 incident, three workers were killed and the driver of a car seriously injured after one of the components of a launching gantry fell at the SUKE Elevated Highway construction site.

KUALA LUMPUR: A construction company, Zhongshi International Sdn Bhd, was fined RM45,000 by the sessions court here today for failing to ensure safety at its construction site, resulting in the death of three workers last March.

Figure 4.31 Website loaded after an article is clicked

There is a Logout button at the top bar in both the admin and the supervisors home interface as shown in Figure 4.32. To logout the application, the Log Out button icon at the top bar is clicked.



Welcome, *Suzanne Ang*

11.01.2023

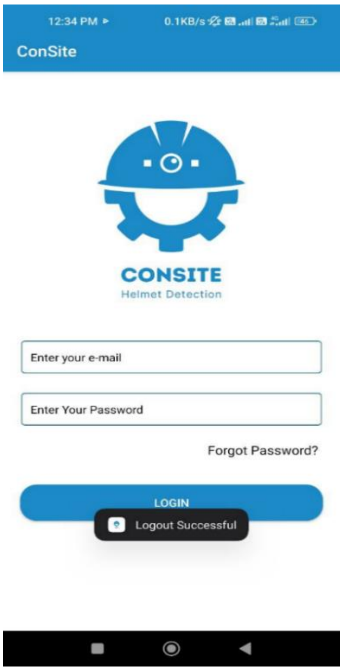
PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

One foreign worker dies, two injured in another

Figure 4.32 Log out button for logging out the application

Once the user presses on the log out button, the user will be redirected back to the login interface and a pop-up message informing he is logged out is displayed as shown in Figure 4.33 below.



12:34 PM ▸ 0.1KB/s

ConSite

CONSITE
Helmet Detection

Enter your e-mail

Enter Your Password

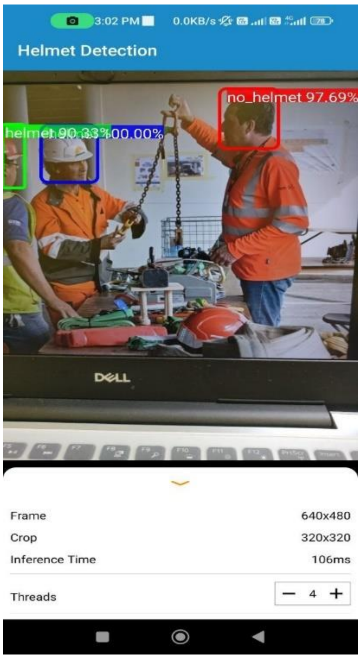
Forgot Password?

LOGIN

Logout Successful

Figure 4.33 Pop-up message of user successfully logged out

The camera feature can be accessed from the application through clicking the camera icon at the bottom navigation bar from the admin and supervisor home page. The camera can detect workers who wear helmet or without helmet as shown in Figure 4.34.



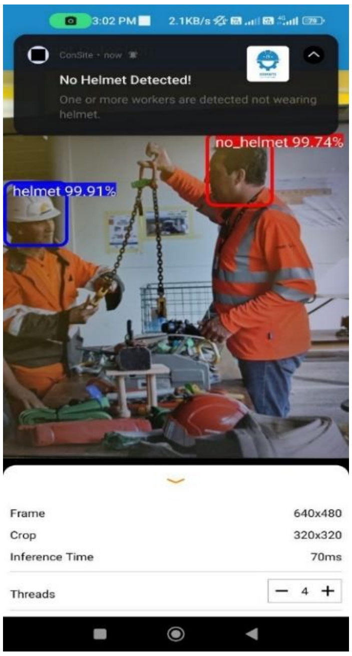
3:02 PM 0.0KB/s

Helmet Detection

Frame	640x480
Crop	320x320
Inference Time	106ms
Threads	− 4 +

Figure 4.34 Camera interface for detecting workers

If a worker is detected without helmet from the camera feature in the application, a notification is triggered as shown in Figure 4.35 and sent to the user's phone to alert him.



ConSite • now

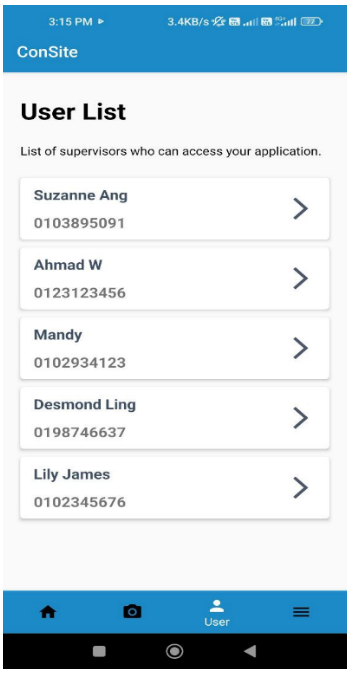
No Helmet Detected!

One or more workers are detected not wearing helmet.

Frame	640x480
Crop	320x320
Inference Time	70ms
Threads	− 4 +

Figure 4.35 Notification when a worker is detected without helmet

Figure 4.36 describes the user list interface that is accessible only by the admin of the company. The admin will be redirected to this page when the user icon is selected from the bottom navigation bar.



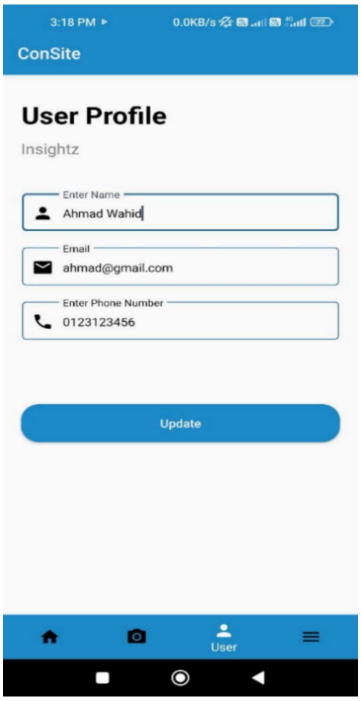
User List

List of supervisors who can access your application.

Suzanne Ang	0103895091
Ahmad W	0123123456
Mandy	0102934123
Desmond Ling	0198746637
Lily James	0102345676

Figure 4.36 User List interface

When a user is selected from the user list, the profile of the user is shown as in Figure 4.37 below where the admin can edit the name and phone number. The email address of the user account is set to be non-editable. The company name is printed out at the top left of the page.



User Profile

Insightz

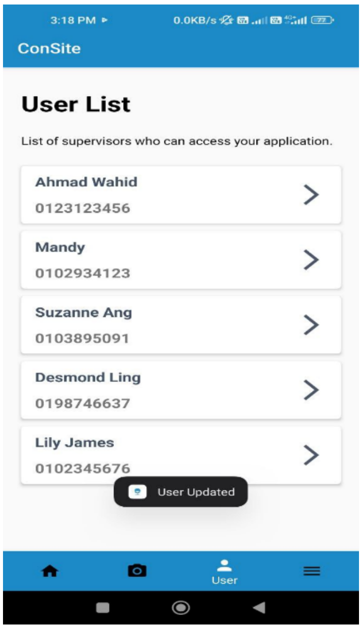
Enter Name: Ahmad Wahid

Email: ahmad@gmail.com

Enter Phone Number: 0123123456

Figure 4.37 Update user information interface

A pop-up message as shown in Figure 4.38 is displayed when the information of a user is successfully updated. The admin will be redirected back to the user list interface.



User List

List of supervisors who can access your application.

Ahmad Wahid	0123123456
Mandy	0102934123
Suzanne Ang	0103895091
Desmond Ling	0198746637
Lily James	0102345676

User Updated

Figure 4.38 Pop-up message after user info is updated

When the menu icon at the bottom navigation bar is clicked, the user is redirected to the menu interface. The functions in the admin menu interface as shown in Figure 4.39 are the same as in the supervisor menu interface as shown in Figure 4.40. The only difference is the admin has the user tab in the bottom navigation bar.

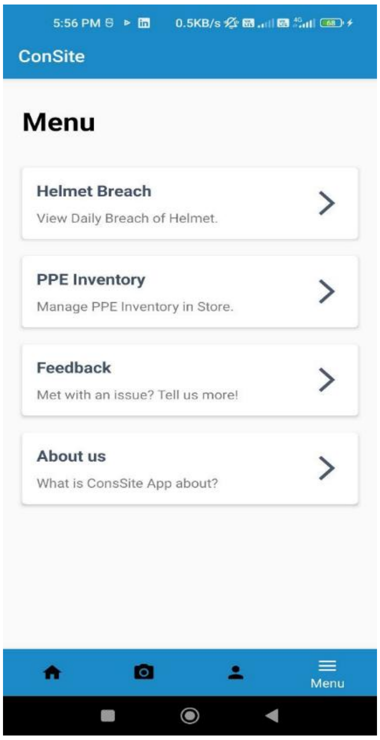

5:56 PM
0.5KB/s
ConSite
Menu
Helmet Breach
View Daily Breach of Helmet.
PPE Inventory
Manage PPE Inventory in Store.
Feedback
Met with an issue? Tell us more!
About us
What is ConsSite App about?
Menu

Figure 4.39 Admin menu interface

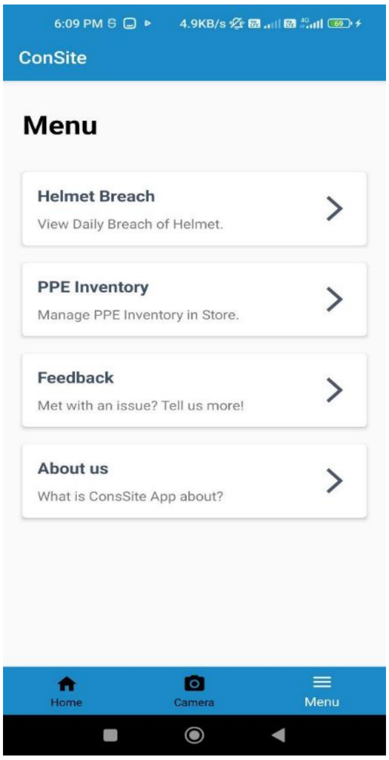
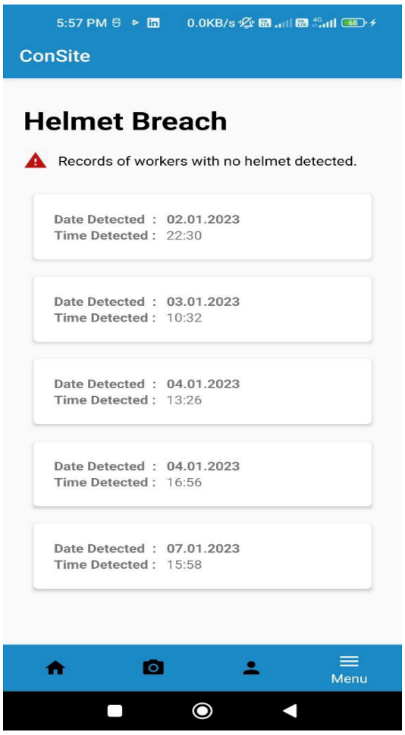

6:09 PM
4.9KB/s
ConSite
Menu
Helmet Breach
View Daily Breach of Helmet.
PPE Inventory
Manage PPE Inventory in Store.
Feedback
Met with an issue? Tell us more!
About us
What is ConsSite App about?
Home
Camera
Menu

Figure 4.40 Supervisor menu interface

When the user selects Helmet Breach from the menu list, the user is redirected to the helmet breach interface as shown in Figure 4.41. Also, if the user clicks on the notification that is sent by the application, the user is redirected to this page. The breaches of workers not wearing helmet are listed out here with the date and time of the workers detected.



Helmet Breach

⚠ Records of workers with no helmet detected.

Date Detected : 02.01.2023
Time Detected : 22:30

Date Detected : 03.01.2023
Time Detected : 10:32

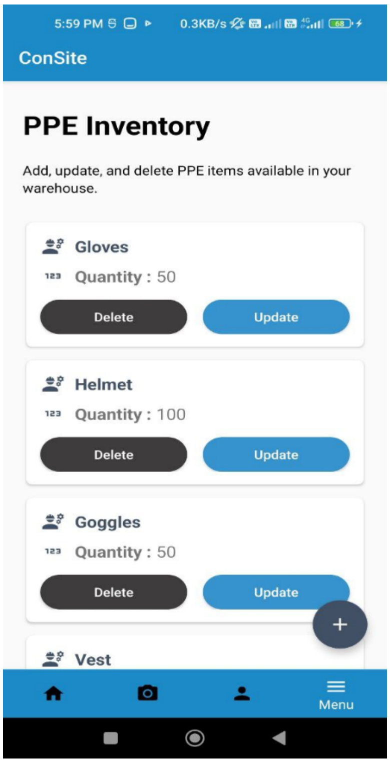
Date Detected : 04.01.2023
Time Detected : 13:26

Date Detected : 04.01.2023
Time Detected : 16:56

Date Detected : 07.01.2023
Time Detected : 15:58

Figure 4.41 Helmet breach interface

When the user selects PPE Inventory from the menu list, the user is redirected to the PPE inventory interface as shown in Figure 4.42. The actions that can be performed are updating the records of the PPE and deleting a record. To add a new PPE, the floating add button can be pressed.



PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 50

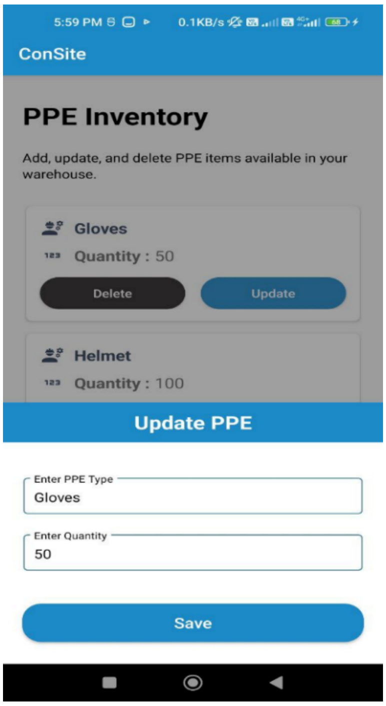
Helmet
Quantity : 100

Goggles
Quantity : 50

Vest

Figure 4.42 PPE inventory interface

When the update button of a PPE item is pressed, a pop-up box is shown on the page just as in Figure 4.43. The details of the item displayed. After pressing on the Save button, a pop-up message is displayed as shown in Figure 4.44 to inform that the item has been successfully updated.



PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 50

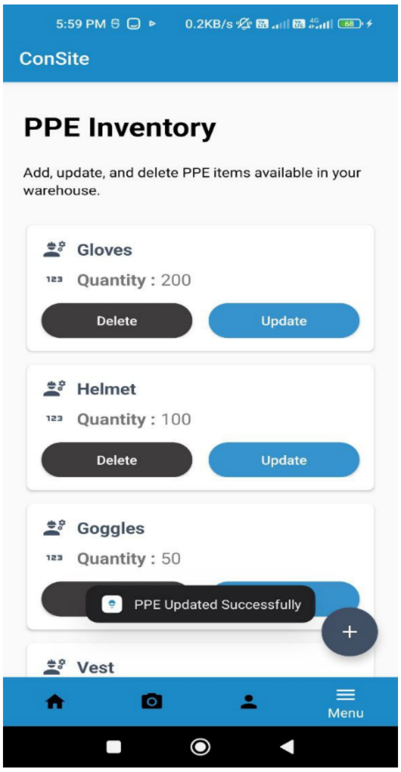
Helmet
Quantity : 100

Update PPE

Enter PPE Type: Gloves

Enter Quantity: 50

Figure 4.43 Update PPE interface



ConSite

PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 200

Helmet
Quantity : 100

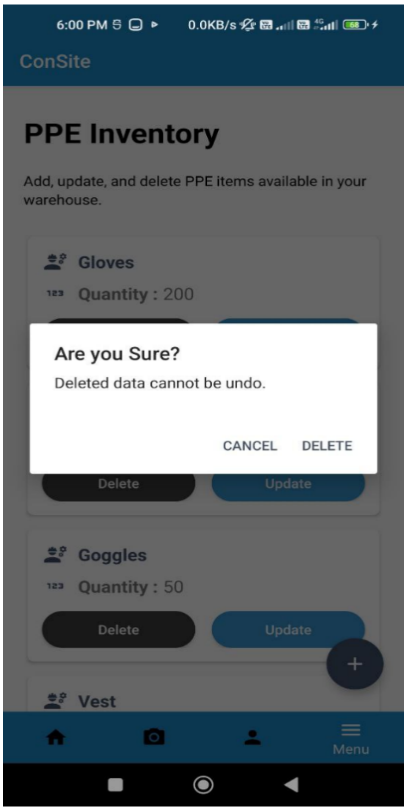
Goggles
Quantity : 50

PPE Updated Successfully

Vest

Figure 4.44 Message after PPE is successfully updated

A confirmation pop-up message as shown in Figure 4.45 below is prompted when the user clicks on the Delete button of a PPE item at the list of PPE.



ConSite

PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 200

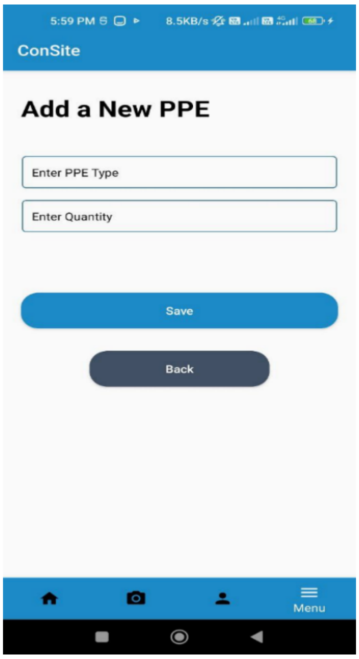
Are you Sure?
Deleted data cannot be undo.

Goggles
Quantity : 50

Vest

Figure 4.45 Confirmation message to delete a PPE

After the user clicks on the floating add button at the PPE inventory interface, the user is redirected to this Add new PPE interface as shown in Figure 4.46 below. The PPE type and the quantity can be added to the PPE list from here.



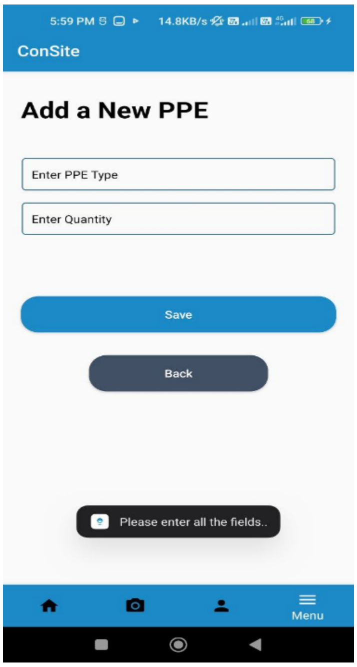
Add a New PPE

Enter PPE Type

Enter Quantity

Figure 4.46 Add new PPE interface

If no input is given in either one of the field or both fields, the application will display a pop-up message as shown in Figure 4.47 below to indicate all the fields must be entered before adding a new PPE into the list.



Add a New PPE

Enter PPE Type

Enter Quantity

Please enter all the fields..

Figure 4.47 Error message when no input is given when adding new PPE

After a PPE is added to the application, a pop-up message is displayed just as in Figure 4.48. The field for adding new PPEs are cleared to allow the user to add another new PPE.

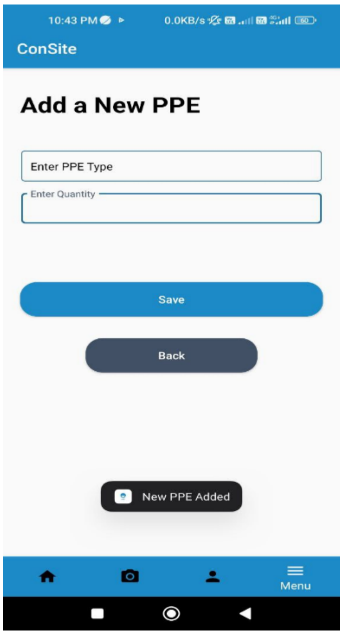


Figure 4.48 Pop-up message that new PPE is successfully added

When Feedback is selected from the menu section by both the admin and supervisor, they will be redirected to this Feedback interface as shown in Figure 4.49. The users can describe their issues with maximum 500 words. The feedback together with the email of the user is saved into the database.

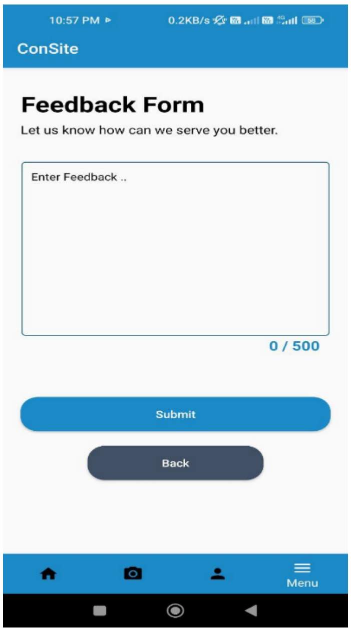
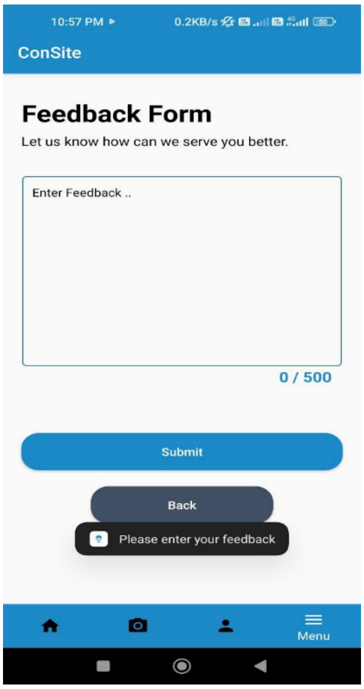


Figure 4.49 Feedback interface

If no input is entered in the field and the user clicks on Save button, the application will prompt an error message just as in Figure 4.50 below to inform the user to fill up the field.


Feedback Form

Let us know how can we serve you better.

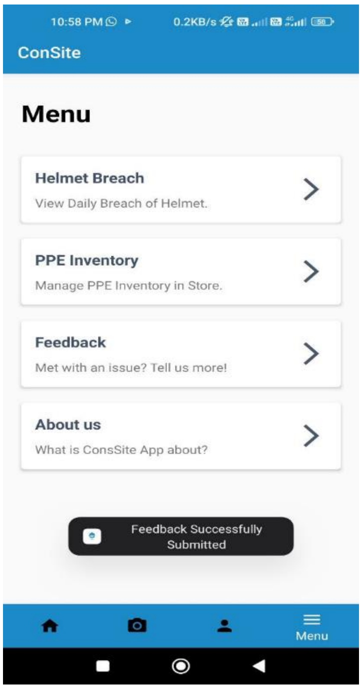
Enter Feedback ..

0 / 500

Please enter your feedback

Figure 4.50 Error message if no field is entered in Feedback interface

After the feedback is successfully submitted, the user is redirected back to the menu section and a successful message is shown just as in Figure 4.51 below.


Menu

Helmet Breach
View Daily Breach of Helmet.

PPE Inventory
Manage PPE Inventory in Store.

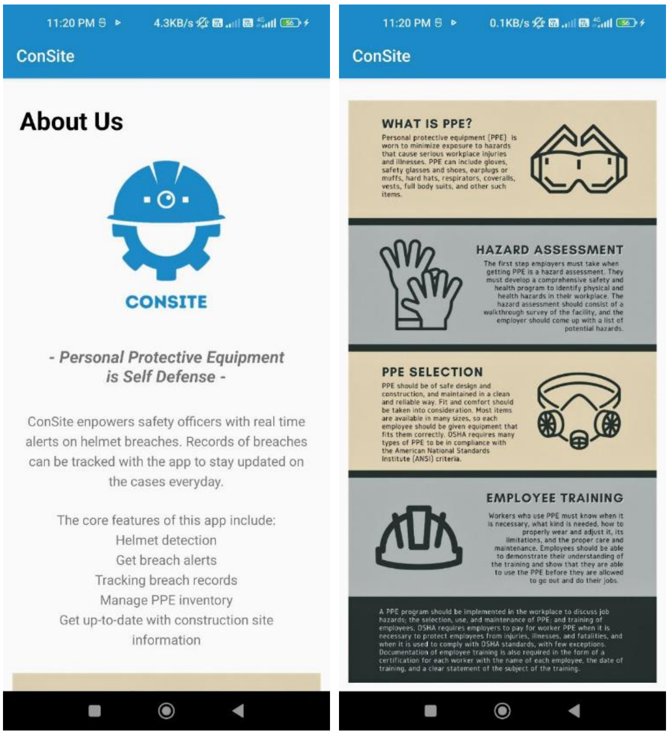
Feedback
Met with an issue? Tell us more!

About us
What is ConsSite App about?

Feedback Successfully Submitted

Figure 4.51 Successful message that feedback is submitted

When About us is selected from the menu section by the admin or the supervisor, the user is redirected to this About us interface as shown in Figure 4.52 below where information about this application can be viewed here.



About Us

CONSITE

- Personal Protective Equipment is Self Defense -

ConSite enpowers safety officers with real time alerts on helmet breaches. Records of breaches can be tracked with the app to stay updated on the cases everyday.

The core features of this app include:
Helmet detection
Get breach alerts
Tracking breach records
Manage PPE inventory
Get up-to-date with construction site information

WHAT IS PPE?

Personal protective equipment (PPE) is worn to minimize exposure to hazards that cause serious workplace injuries and illnesses. PPE can include gloves, safety glasses and shoes, earplugs or muffs, hard hats, respirators, coveralls, vests, full body suits, and other such items.

HAZARD ASSESSMENT

The first step employers must take when getting PPE is a hazard assessment. They must develop a comprehensive safety and health program to identify physical and health hazards in their workplace. The hazard assessment should consist of a walkthrough survey of the facility, and the employer should come up with a list of potential hazards.

PPE SELECTION

PPE should be of safe design and construction, and maintained in a clean and reliable way. Fit and comfort should be taken into consideration. Most items are available in many sizes, so each employee should be given equipment that fits them correctly. OSHA requires many types of PPE to be in compliance with the American National Standards Institute (ANSI) criteria.

EMPLOYEE TRAINING

Workers who use PPE must know when it is necessary, what kind is needed, how to properly wear and adjust it, its limitations, and the proper care and maintenance. Employees should be able to demonstrate their understanding of the training and show that they are able to use the PPE before they are allowed to go out and do their jobs.

A PPE program should be implemented in the workplace to discuss job hazards; the selection, use, and maintenance of PPE; and training of employees. OSHA requires employers to pay for worker PPE when it is necessary to protect employees from injuries, illnesses, and fatalities, and when it is used to comply with OSHA standards, with few exceptions. Documentation of employee training is also required in the form of a certification for each worker with the name of each employee, the date of training, and a clear statement of the subject of the training.

Figure 4.52 About us interface

4.4 Testing Result and Discussion

The application is developed following the Agile methodology, where every sprint contains the testing phase. With this methodology, continuous integration between the development and testing phases is allowed where testing can start at the beginning of the project. The main stages of testing that can be performed during the development of the application are unit testing, integration testing, system testing, and acceptance testing.

4.4.1 Unit Testing

Unit testing will be the initial phase of testing ConSite application. Here, every separate module of the application is tested to check they worked correctly individually before combining. If there is any error found in one of the modules of ConSite, changes and corrections are to be made and the testing phase is repeated till all errors are solved.

4.4.2 Integration Testing

Unit testing will be the initial phase of testing the ConSite application. Here, every separate module of the application is tested to check they worked correctly individually before combining. If there is any error found in one of the modules of ConSite, changes and corrections are to be made and the testing phase is repeated till all errors are solved.

4.4.3 System Testing

This phase of testing will undergo testing on the whole ConSite application, which includes the integration of several modules. The objective is to determine that the application developed can perform the necessary functionalities and meets the standards of the end users.

4.4.4 Acceptance Testing

The prototype of the ConSite application will be used by the end users to get their feedback on the application developed. User Acceptance Test (UAT) will be conducted to ensure the requirements are all met and getting the approval of moving on to the next sprint of the Agile cycle. (Refer Appendix E and Appendix F)

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter discusses on the conclusion of the project that has been developed and completed. The application is an Android mobile app which the main function is for getting alerts on non-compliance workers. The three objectives set at CHAPTER 1 have been accomplished throughout the development of the project. The first objective focuses on identifying the limitation in existing applications of construcition site PPE detection. Three existing applications have been chosen and studied, where all of the three applications have their own pros and cons. The useful features of them are considered and implemented into the proposed system when possible. The second objective is achieved where it is to develop a construction site safety helmet detection mobile application using computer vision-based technology. The application has been developed with the implementation of TensorFlow, a library that hosts computer vision techniques. The last objective involves evaluating the functionalities of the developed application. This objective is achieved through executing the testing phases in CHAPTER 4.

The proposed application is developed through the Agile methodology which helps to produce the product through incremental delivery of small pieces of functionality iteratively. Every iteration involves planning, designing, building, testing, and finally, reviewing the product. The methodology helped to ensure the development of the application are easier to manage since it provides adaptability to change.

The proposed application has some enhancement that can be made for improving the features and functionalities to provide users with a better experience towards using the application. It will be discussed in 5.3.

5.2 Research Constraint

The constraints for the project are:

i. Image is not captured when a worker is detected not wearing the helmet.

The current feature of the application now detects and trigger a notification when a worker is detected not wearing a helmet. However, it does not support capturing the image and saving it into the database for proof of breach. Only the date and time the breach happened is recorded for further review.

ii. The admin cannot add or delete supervisors directly from the application.

Currently the application does not allow the company admin to add or delete a supervisor from the application. The admin should directly contact the application administrator to handle deleting or adding an account. This is because the actions are considered as Firebase security-sensitive actions which cannot be solved through the normal Firebase actions. Lack of time also causes this problem to not be researched further.

iii. The application focueses only on helmet detection.

The application can only detect workers with or without helmet and not the whole PPE attire. This is due to having the whole set of PPE detected requires a large dataset and might take a long time for training the model. Also, the limitation on the amount of RAM caused the training process to take few days if the dataset is very large.

5.3 Future Work

Enhancement and improvement on the application can be made in the future:

i. Adding the feature of capturing images of no helmet detected on workers to allow the management to review which are the workers that are not complying to wearing proper head protection.

ii. Implement Firebase Admin SDK server libraries to allow the company admin to have privileges on handling the adding and deleting of the supervisors’ accounts.

iii. Improve the detection of other PPE types by including datasets of other PPEs and increase the RAM for training the detection model.

iv. Adding an external camera for capturing images from actual construction sites and sends alerts to the management.

REFERENCES

Chi, C. F., Chang, T. C., & Ting, H. I. (2005). Accident patterns and prevention measures for fatal occupational falls in the construction industry. *Applied Ergonomics*, *36*(4), 391–400. https://doi.org/10.1016/J.APERGO.2004.09.011

Choudhry, R. M., & Fang, D. (2008). Why operatives engage in unsafe work behavior: Investigating factors on construction sites. *Safety Science*, *46*(4), 566–584. https://doi.org/10.1016/J.SSCI.2007.06.027

DataVaults. (2021). *Extracting software requirements -the agile way*. https://www.datavaults.eu/extracting-software-requirements-the-agile-way/

Department of Statistics Malaysia Official Portal. (2020). https://www.dosm.gov.my/v1/index.php?r=column%2FcthemeByCat&cat=492&bul_id=czB6elhvaWtoVmgwVktXUGJqRElLZz09&menu_id=WjJGK0Z5bTk1ZElVT09yUW1tRG41Zz09

EasyFlow | PPE Detection. (n.d.). Retrieved June 1, 2022, from https://easyflow.tech/ppe-detection-1/

PPE Detection | V-App. (n.d.). Retrieved June 1, 2022, from https://www.v-app.io/ppe-detection/

Tam, C. M., Zeng, S. X., & Deng, Z. M. (2004). Identifying elements of poor construction safety management in China. *Safety Science*, *42*(7), 569–586. https://doi.org/10.1016/J.SSCI.2003.09.001

The PPE Needed to Work Safely on a Building or Construction Site. (2021). https://www.xamax.co.uk/blog/what-ppe-is-required-for-construction.html

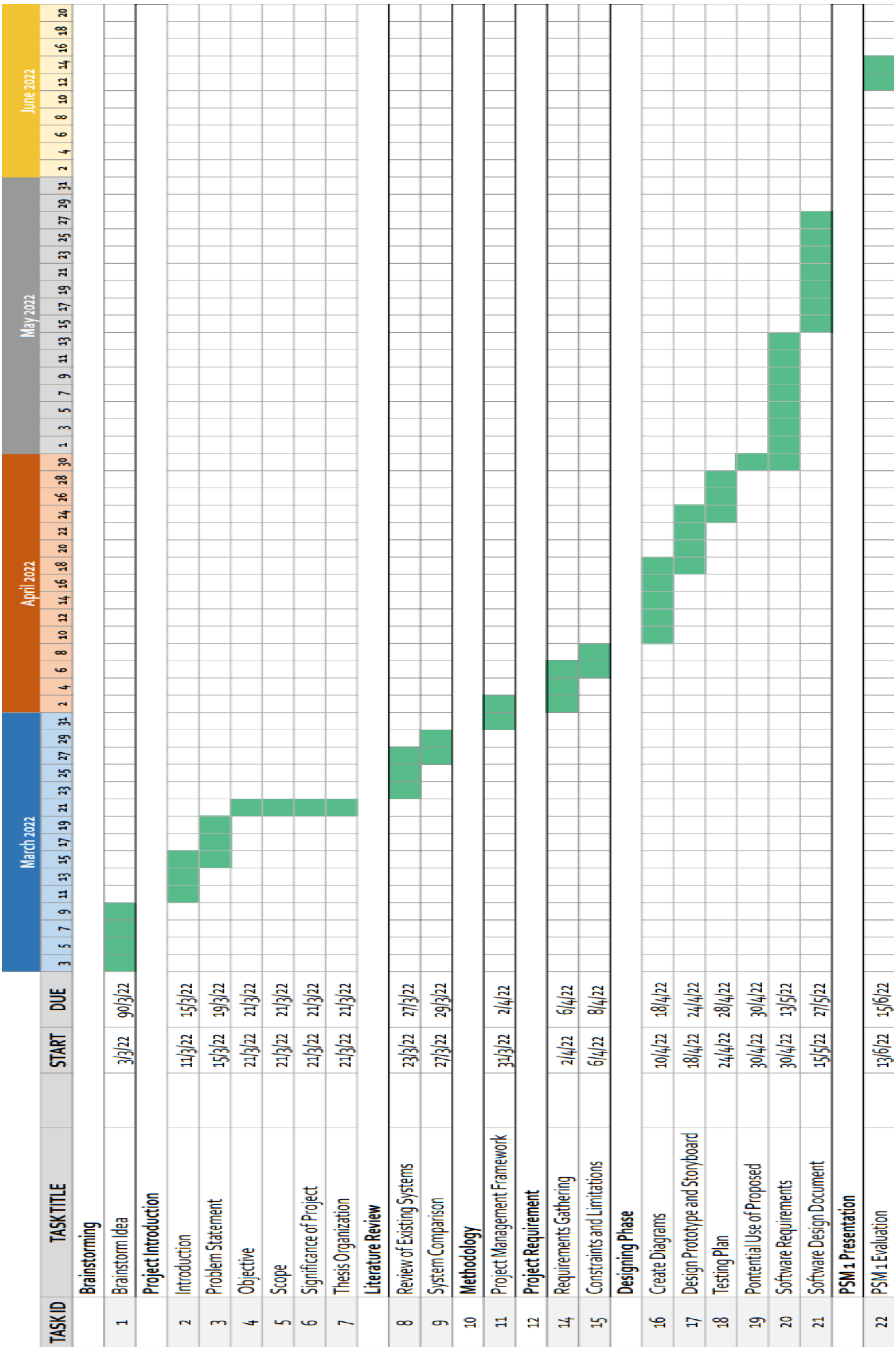
viAct | PPE Detection Solution for Construction Jobsite |Construction Management. (n.d.). Retrieved June 1, 2022, from https://www.viact.ai/ppedetection

Warrier, R. (2019). *Top causes of global construction fatalities, and how to avoid site risks - Construction Week Online*. https://www.constructionweekonline.com/people/training/255830-top-10-causes-of-construction-deaths-and-how-to-prevent-site-accidents

What is a construction site? - LetsBuild. (2019). https://www.letsbuild.com/blog/construction-site

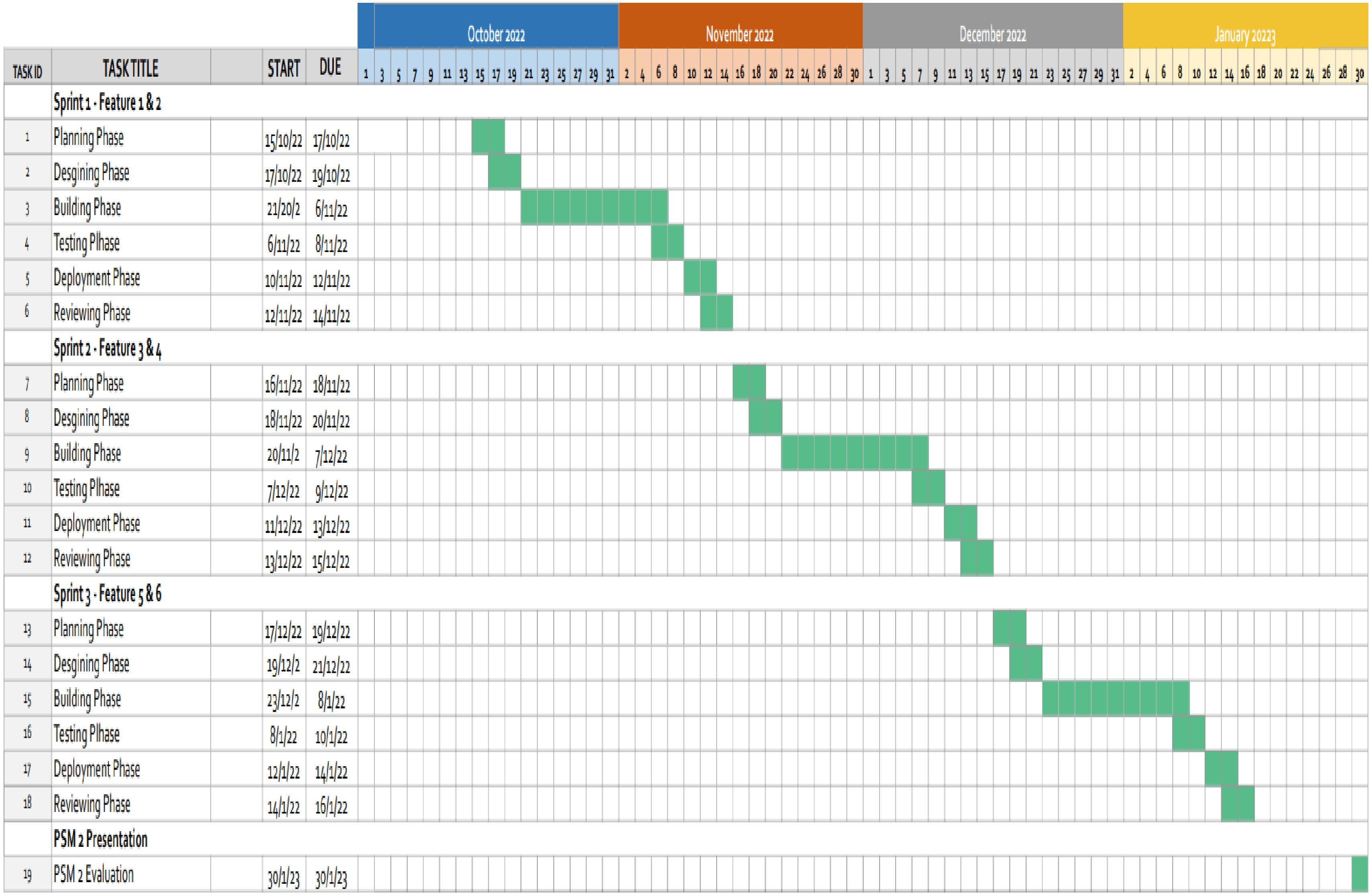
World Statistic. (2011). https://www.ilo.org/moscow/areas-of-work/occupational-safety-and-health/WCMS_249278/lang--en/index.htm

APPENDIX A
GANTT CHART (PSM 1)



TASK ID	TASK TITLE	START	DUE
	Brainstorming		
1	Brainstorm Idea	3/3/22	90/3/22
	Project Introduction		
2	Introduction	11/3/22	15/3/22
3	Problem Statement	15/3/22	19/3/22
4	Objective	21/3/22	21/3/22
5	Scope	21/3/22	21/3/22
6	Significance of Project	21/3/22	21/3/22
7	Thesis Organization	21/3/22	21/3/22
	Literature Review		
8	Review of Existing Systems	23/3/22	27/3/22
9	System Comparison	27/3/22	29/3/22
10	**Methodology**		
11	Project Management Framework	31/3/22	2/4/22
12	**Project Requirement**		
14	Requirements Gathering	2/4/22	6/4/22
15	Constraints and Limitations	6/4/22	8/4/22
	Designing Phase		
16	Create Diagrams	10/4/22	18/4/22
17	Design Prototype and Storyboard	18/4/22	24/4/22
18	Testing Plan	24/4/22	28/4/22
19	Pontential Use of Proposed	30/4/22	30/4/22
20	Software Requirements	30/4/22	13/5/22
21	Software Design Document	15/5/22	27/5/22
	PSM 1 Presentation		
22	PSM 1 Evaluation	13/6/22	15/6/22

APPENDIX B
GANTT CHART (PSM 2)



TASK ID	TASK TITLE	START	DUE
	Sprint 1 - Feature 1 & 2		
1	Planning Phase	15/10/22	17/10/22
2	Desgining Phase	17/10/22	19/10/22
3	Building Phase	21/20/2	6/11/22
4	Testing Plhase	6/11/22	8/11/22
5	Deployment Phase	10/11/22	12/11/22
6	Reviewing Phase	12/11/22	14/11/22
	Sprint 2 - Feature 3 & 4		
7	Planning Phase	16/11/22	18/11/22
8	Desgining Phase	18/11/22	20/11/22
9	Building Phase	20/11/2	7/12/22
10	Testing Plhase	7/12/22	9/12/22
11	Deployment Phase	11/12/22	13/12/22
12	Reviewing Phase	13/12/22	15/12/22
	Sprint 3 - Feature 5 & 6		
13	Planning Phase	17/12/22	19/12/22
14	Desgining Phase	19/12/2	21/12/22
15	Building Phase	23/12/2	8/1/22
16	Testing Plhase	8/1/22	10/1/22
17	Deployment Phase	12/1/22	14/1/22
18	Reviewing Phase	14/1/22	16/1/22
	PSM 2 Presentation		
19	PSM 2 Evaluation	30/1/23	30/1/23

APPENDIX C
SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

.

2020

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

[CONSTRUCTION SITE SAFETY HELMET DETECTION IN MOBILE APPLICATION]

DOCUMENT APPROVAL

	Name	**Date**
Authenticated by: ____________________ Name	ANG SUZANNE	8th February 2023
Approved by: ____________________ Client		

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

Construction Site Safety Helmet Detection Application is a mobile application that uses computer vision-based technology mainly to detect the absence of safety helmet on workers. This application allows for the automated collection and detection of workers' improper dress codes, which in this case, the helmet, on construction sites. The users of the application are the admin and supervisors of the company. There are 6 main modules available in this application, which are Login, Manage User Data, Access Breach Records, Receive Notifications, Manage PPE Inventory, and Add Feedback.

The Login module enables the user to log into the system. The Manage User Data module allows the admin to manage the data of the user which includes updating the user data. The Access Breach Records module allows the admin and supervisor to view the records of helmet breach. The Receive Notifications module allows the admin and supervisor to receive notifications concerning any helmet breach related issues. Manage PPE Inventory enables the users to handle the inventory of PPE, which includes adding, updating, and deleting PPE inventory. The last module, Add Feedback, enables the users to submit feedback on the issues met or improvements to be suggested.

1.2 SYSTEM IDENTIFICATION

This document uses the following naming convention:

System identification number: **CONSITE-SRS-2022-V1**

Table 1. 1 System Identification

CONSITE	Construction Site Safety Helmet Detection Application
SRS	Software Requirements Specification
2022	Year 2022
V1	Version 1

1.3 CONTEXT DIAGRAM

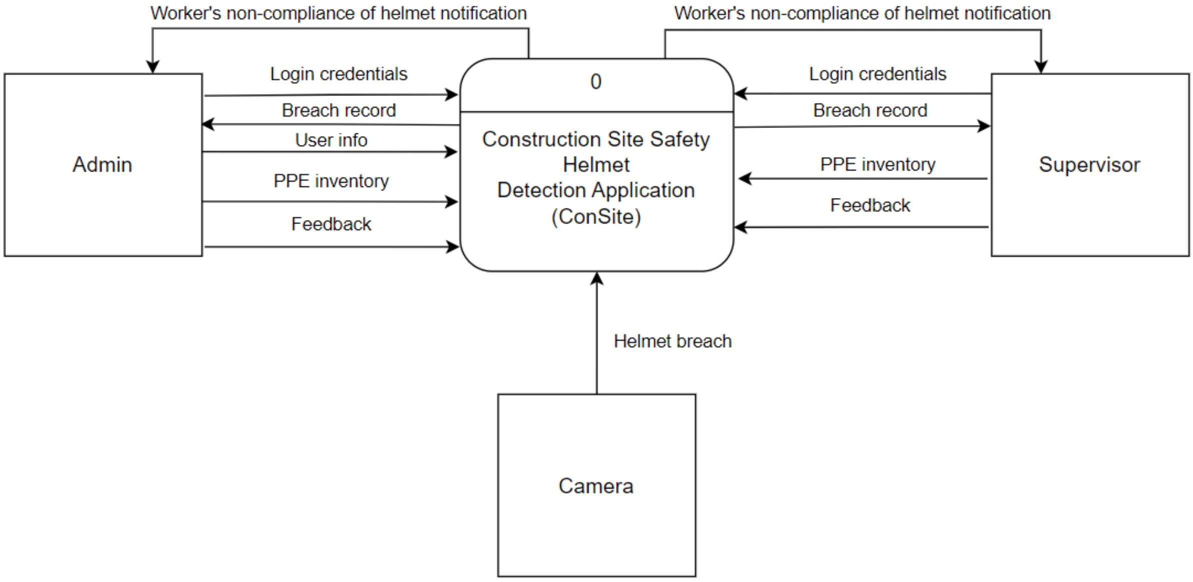

Worker's non-compliance of helmet notification
Worker's non-compliance of helmet notification
Admin
Login credentials
Breach record
User info
PPE inventory
Feedback
0
Construction Site Safety Helmet Detection Application (ConSite)
Login credentials
Breach record
PPE inventory
Feedback
Supervisor
Helmet breach
Camera

Figure 1. 1 Context Diagram

From the context diagram above, the entities involved in the application are the admin, supervisor, and camera. Firstly, login credentials from both the users, in this case the admin and supervisor are needed to log into the application. After logging in, the admin may record and manage users' info who are able to access the application. Both users can manage PPE inventory and send feedback of the application. The application will send notifications of any non-compliance of helmet worn by workers. Breach records of absence of helmet on workers generated from the application can also be accessed by the users. The camera entity provides the helmet breach information to the application.

1.4 DATA FLOW DIAGRAM

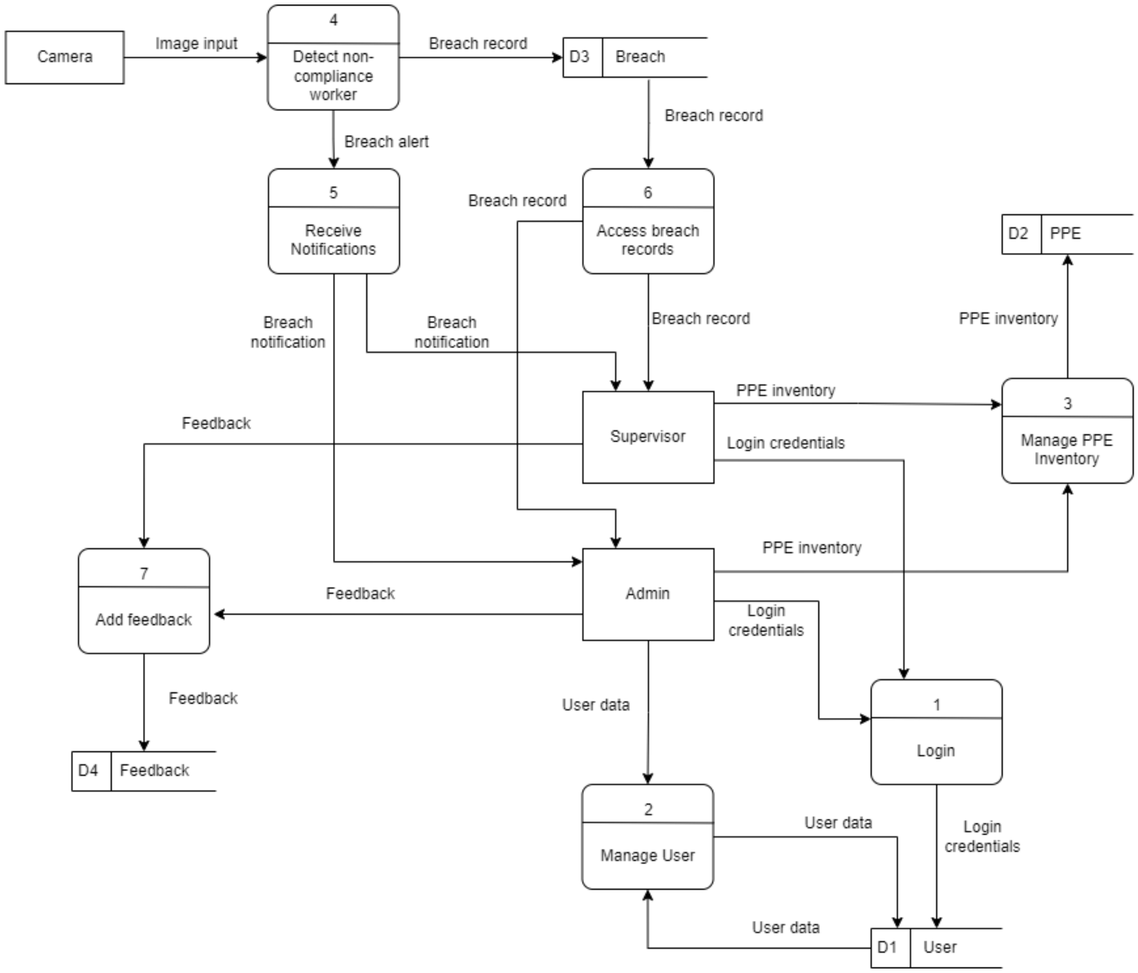


Figure 1. 2 Data Flow Diagram

The figure above describes the data flow of the application. The tables below describe the diagram in detail.

1.4.1 Process

Table 1. 2 Process Description

Process	Description
Login	Process for user to log into the application.
Manage User	Process for admin to manage user’s info who are granted access to the application.
Manage PPE Inventory	Process for user to manage PPE inventory that are available.

Detect non-compliance worker	Process for the camera on detecting inexistence of helmet on workers.
Receive Notifications	Process for user to receive notifications on helmet non-compliance issues.
Access breach records	Process for user to access the records of helmet breach.
Add feedback	Process for user to submit feedback on the issues met or improvements to be suggested.

1.4.2 External Entities

Table 1. 3 External Entities Description

Process	Description
Admin	The admin is the main user of the application, who represents his company to manage the application.
Supervisor	The supervisor is one of the stakeholders involved in the application.
Camera	The camera captures the helmet breach.

1.4.3 Data Store

Table 1. 4 Data Store Description

Process	Description
User	Data store that stores the users' info.
PPE	Data store that stores the PPE inventory info.
Breach	Data store that stores the records of helmet breach.
Feedback	Data store that stores the feedbacks given by the users.

1.4.4 Data Flow

Table 1. 5 Data Flow Description

Process	Description
Login credentials	User enters the login credentials to log into the application.

User data	Admin enters the user data into the application.
PPE inventory	User enters the PPE inventory info into the application for record purpose.
Image input	Camera sends the scene that is captured.
Breach record	Application saves the breach record into the database.
Breach alert	Application sends alert on the helmet breach.
Breach notification	Users receive notifications on the helmet breach.
Feedback	Users submit feedback on the comments about the application.

CHAPTER 2

2.1 USE CASE DIAGRAM AND DESCRIPTION

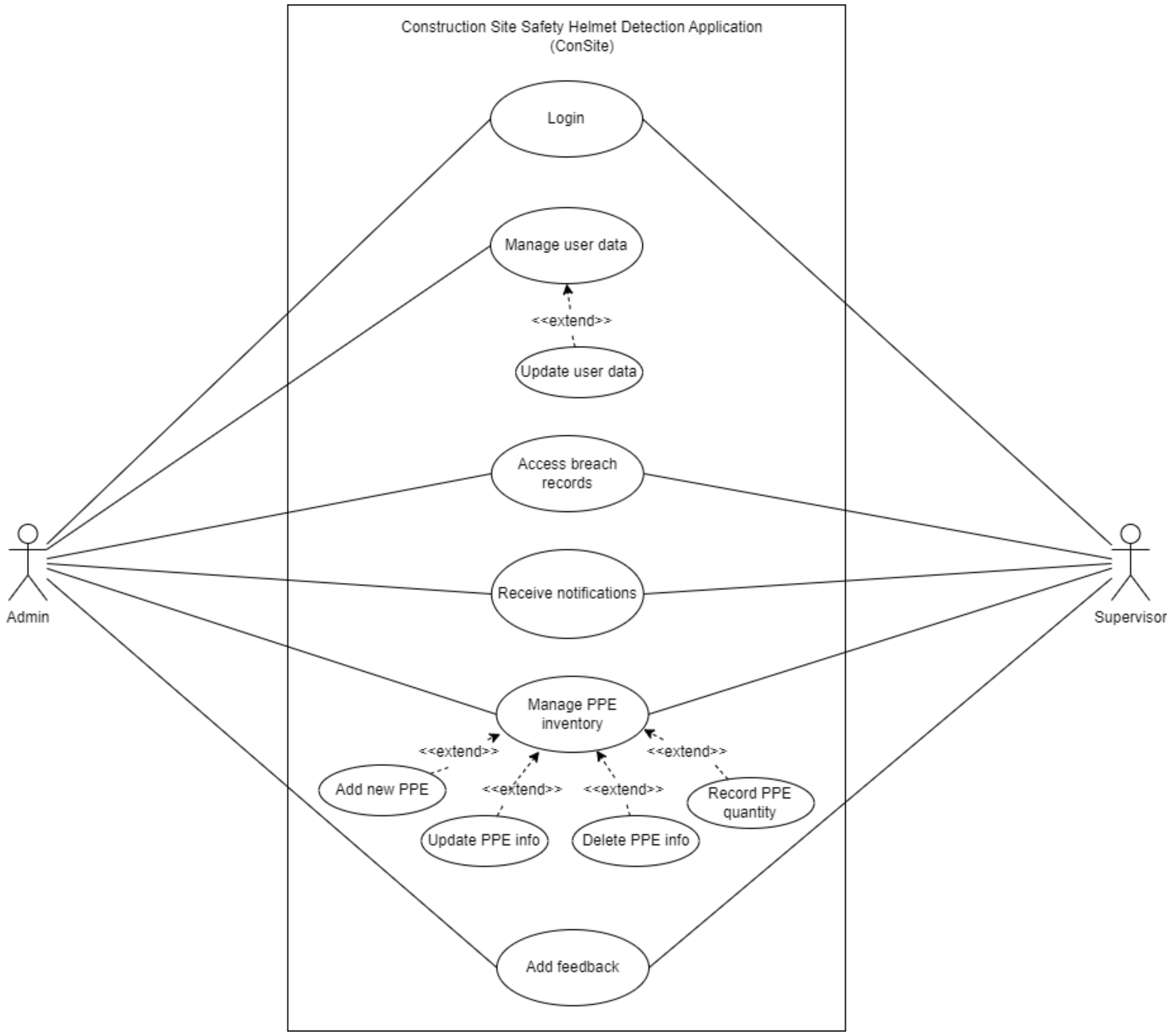


Figure 2. 1 Use Case Diagram

Table 2. 1 Use Case Description

Module	Function
Login	The user is able to log into the system.
Manage user data	The employer can manage other supervisor’s data.

Access breach records	The user can view the records of helmet breach.
Receive notifications	The user can receive notifications on absence of helmet on workers.
Manage PPE inventory	The user can manage the inventory of PPE where adding, updating, and deleting the inventory is allowed.
Add feedback	The user can submit feedback on the issues met or improvements to be suggested.

2.1.1 Login

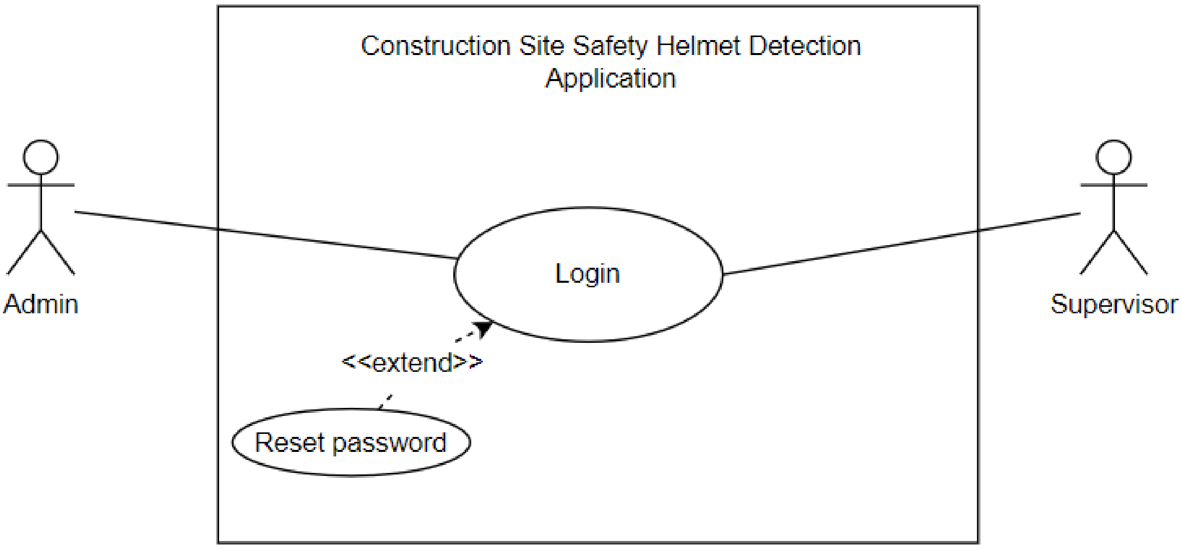


Figure 2. 2 Login Use Case Diagram

Table 2.2 Login Use Case Description

Use Case ID	CONSITE-UC-100
Use Case	Login
Brief Description	This use case describes how the users can log into the application.
Actor	Admin, Supervisor

Pre-Conditions	The application has been downloaded by the user.
Basic Flow	1. The use case begins when user enters the Login interface. 2. The system displays the Login form. 3. User enters the email and password to login. **[A1: Forgot password]** 4. System validates the login details from the database. **[E1: Invalid email or password]** 5. System redirects user to the application main page. 6. The use case ends.
Alternative Flow	**[A1: Forgot password]** 1. The user clicks on <<Forgot Password>> button. 2. The system displays the forgot password interface that requests the user's email. 3. The user enters the email associated with the account. 4. The system sends a password reset link to the user's entered email. 5. The user clicks on the link and set a new password. 6. The system saves the new password into the database. 7. The use case continues at step 2 in basic flow.
Exception Flow	**[E1: Invalid email or password]** 1. The system verifies the email or password does not match those in the database. 2. The system displays an error message. 3. The use case continues at step 3 in basic flow.
Post-Conditions	The user has successfully logged into the system.
Rules	One email address can only be available for one account.
Constraints	-

2.1.2 Manage User Data

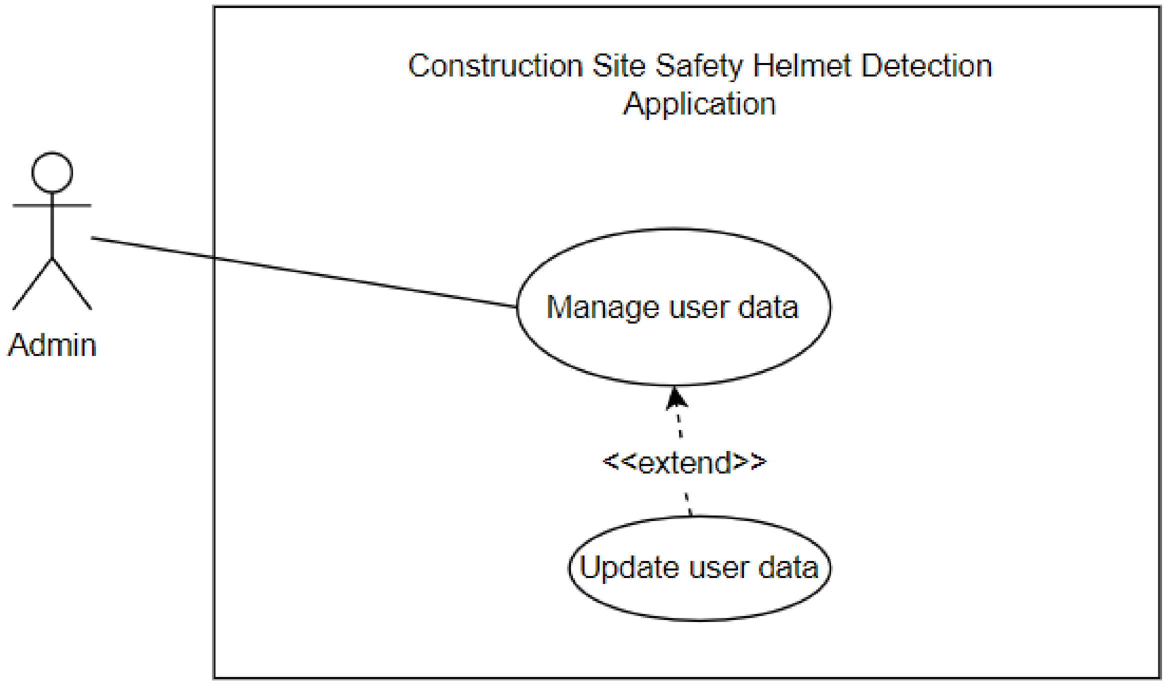


Figure 2. 3 Manage User Data Use Case Diagram

Table 2. 3 Manage User Data Use Case Description

Use Case ID	CONSITE-UC-200
Use Case	Manage User Data
Brief Description	This use case describes how the admin can manage the supervisor's user data.
Actor	Admin
Pre-Conditions	The admin has already logged into the system.
Basic Flow	1. The use case begins when the admin enters the User List interface. 2. The system retrieves the list of users from the database. 3. The system displays the retrieved list onto the page. 4. The admin selects a user from the list. 5. The system retrieves the user's data from the database. 6. The system displays the retrieved data on profile page of the selected user. 7. The admin edits the information of the worker. 8. The employer clicks on <<Update>> button. 9. The system saves the updated info into the database.

	10. The use case ends.
Alternative Flow	-
Exception Flow	-
Post-Conditions	The data of the user is updated in the database.
Rules	-
Constraints	-

2.1.3 Access Breach Records

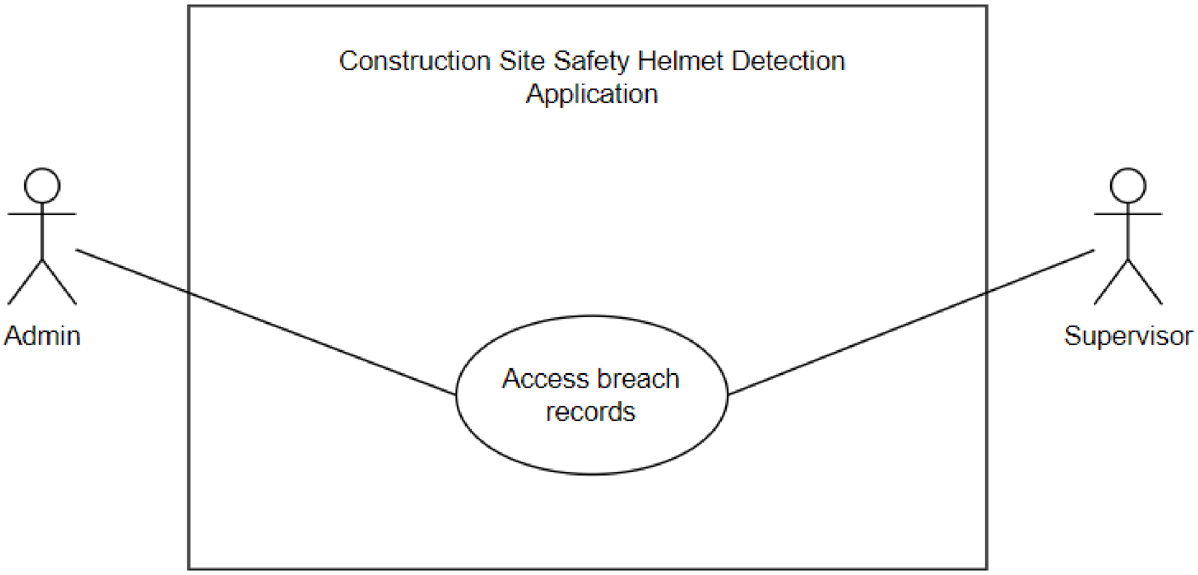


Figure 2. 4 Access Breach Records Use Case Diagram

Table 2. 4 Access Breach Records Use Case Description

Use Case ID	CONSITE-UC-300
Use Case	Access Breach Records
Brief Description	This use case describes how the user can access the breach records list.
Actor	Admin, Supervisor
Pre-Conditions	The user has already logged into the system.
Basic Flow	1. The use case begins when the user enters the Helmet Breach interface. 2. The system retrieves the list of breaches from the database. 3. The system displays the list of breaches.
Alternative Flow	-
Exception Flow	-
Post-Conditions	-
Rules	-
Constraints	-

2.1.4 Receive Notifications

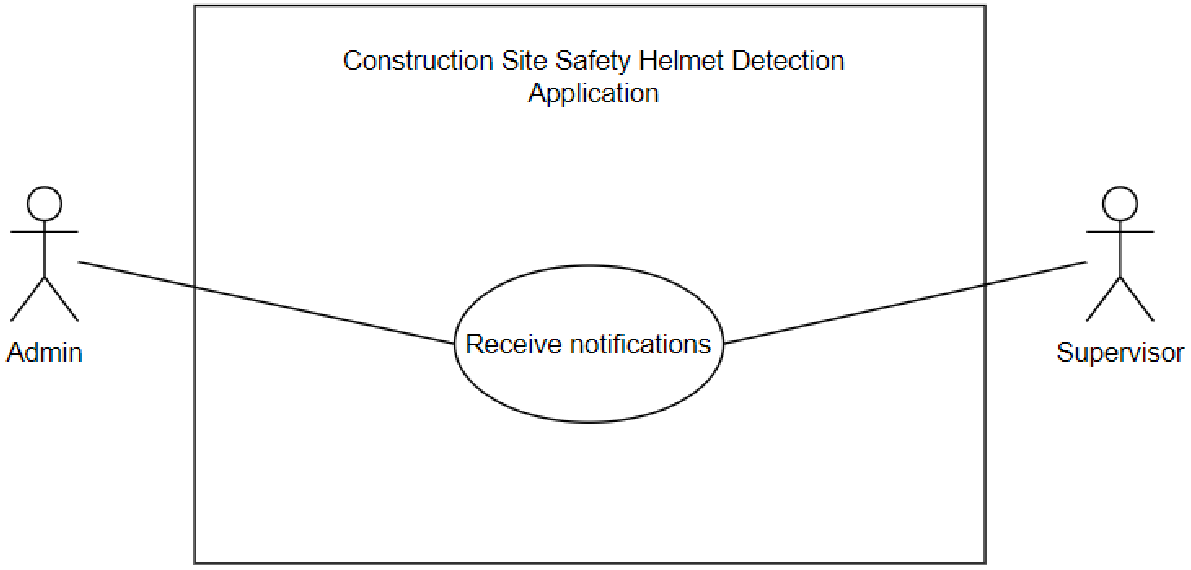


Figure 2. 5 Receive Notifications Use Case Diagram

Table 2. 5 Receive Notifications Use Case Description

Use Case ID	CONSITE-UC-400
Use Case	Receive Notifications
Brief Description	This use case describes how the user receives notifications from the system.
Actor	Admin, Supervisor
Pre-Conditions	The application has been downloaded by the user.
Basic Flow	1. The system detects the absence of helmet on workers. 2. The system saves the helmet breach info into the database. 3. The system generates a push notification to the user. 4. The user clicks into the notification. 5. The system shows the breach records list. 6. The use case ends.
Alternative Flow	-
Exception Flow	-
Post-Conditions	The breach record is saved into the database.
Rules	-
Constraints	-

2.1.5 Manage PPE Inventory

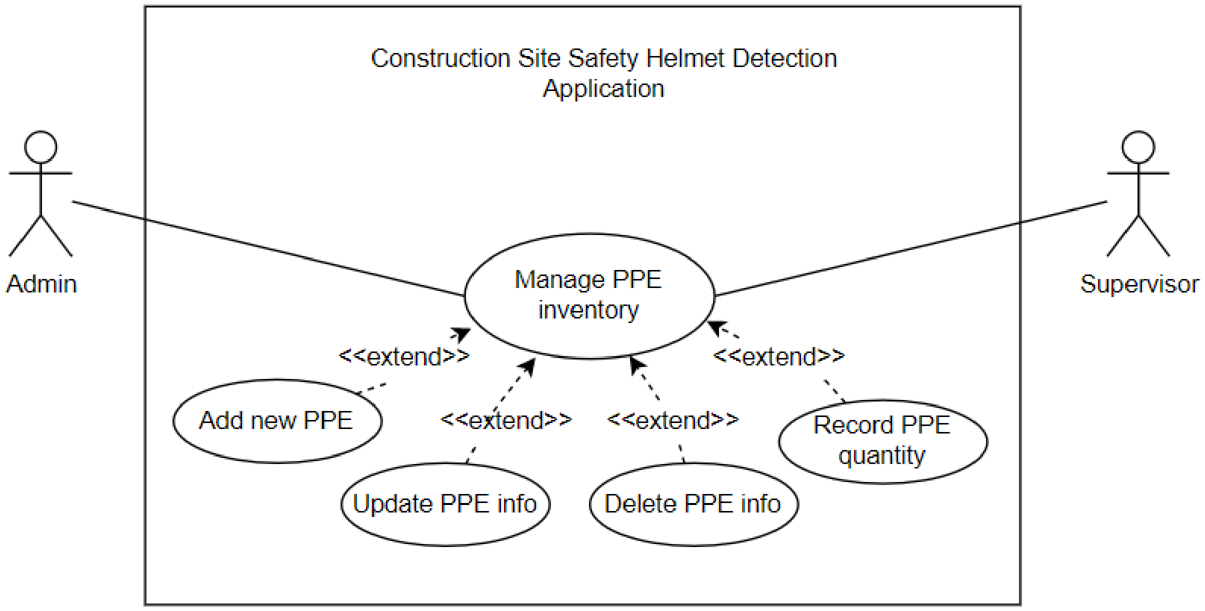


Figure 2. 6 Manage PPE Inventory Use Case Diagram

Table 2. 6 Manage PPE Inventory Use Case Description

Use Case ID	CONSITE-UC-500
Use Case	Manage PPE Inventory
Brief Description	This use case describes how the user can manage the inventory of PPE.
Actor	Admin, Supervisor
Pre-Conditions	The user has logged into the system.
Basic Flow	1. The use case begins when the user enters the PPE Inventory interface. 2. The user clicks on PPE inventory selection. 3. The system retrieves the list of PPE inventory from the database. 4. The system displays the retrieved list onto the interface. **[A1: Add new PPE]** 5. The user selects a PPE from the list. 6. The system retrieves the PPE's data from the database. 7. The system displays the retrieved data on the page of the selected PPE. 8. The user edits the name and quantity of the PPE. **[A2: Delete PPE]**

	9. The user clicks on <<Save>> button. 10. The system saves the updated info into the database. 11. The use case ends.
Alternative Flow	**[A1: Add new PPE]** 1. The user clicks on the <<+>> button. 2. The system displays the Add New PPE interface. 3. The user enters the PPE name and quantity of the PPE. 4. The user clicks on <<Save>> button. 5. The system saves the newly added PPE data into the database. 6. The use case continues at step 3 in basic flow. **[A2: Delete PPE]** 1. The user clicks on the <<Delete>> button. 2. The system prompts a confirmation box to confirm the employer's action. 3. The user clicks on the <<Delete>> button. 4. The system deletes the PPE's data in the database. 5. The use case continues at step 4 in basic flow.
Exception Flow	-
Post-Conditions	The PPE inventory data is updated into the database.
Rules	-
Constraints	-

2.1.6 Add Feedback

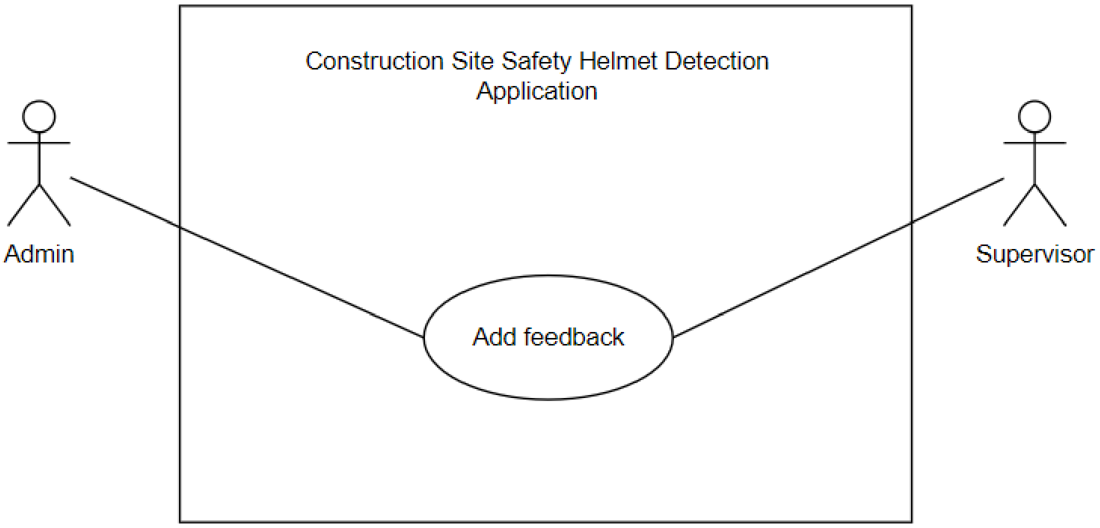


Figure 2. 7 Add Feedback Use Case Diagram

Table 2. 7 Add Feedback Use Case Description

Use Case ID	CONSITE-UC-600
Use Case	Add Feedback
Brief Description	This use case describes how the user can submit feedback on the application.
Actor	Admin, Supervisor
Pre-Conditions	The user has logged into the system.
Basic Flow	1. The use case begins when the user enters the Feedback interface. 2. The user enters the feedback through the text field given. 3. The clicks on <<Submit>> button. 4. The system saves the submitted feedback into the database. 5. The use case ends.
Alternative Flow	-
Exception Flow	-
Post-Conditions	The feedback is saved into the database.
Rules	-
Constraints	-

2.2 SEQUENCE DIAGRAM

2.2.1 Login

Figure 2. 8 describes the basic flow of the Login module. The user will first enter the login interface, then, email and password is entered to log into the system. The email and password entered is sent to the Login controller and User model for verification. Once verified, the controller will display the home page interface to the user.

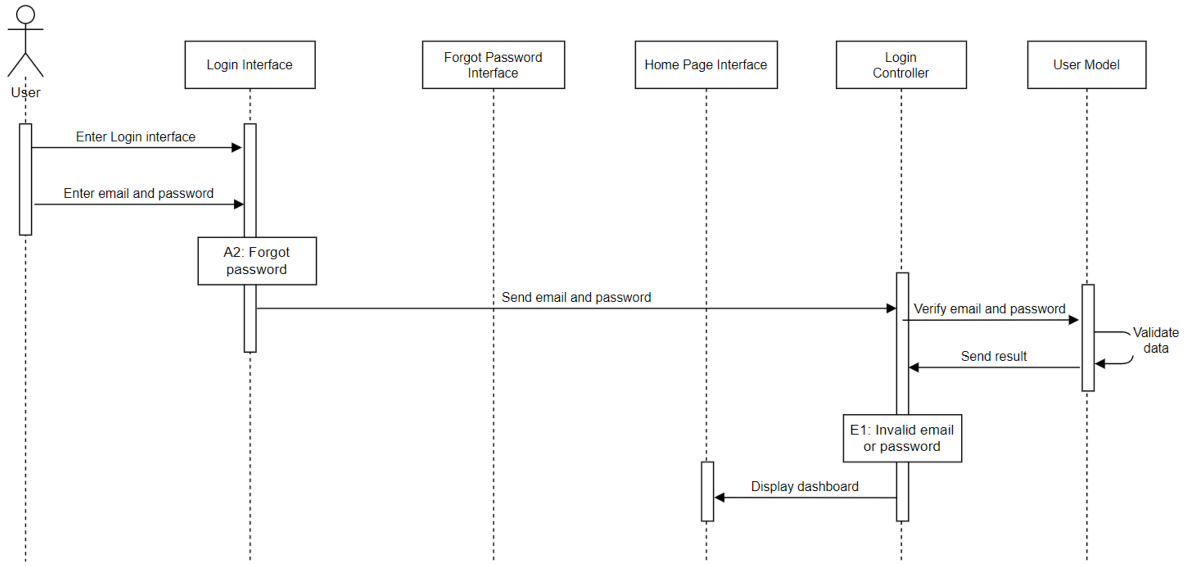


Figure 2. 8 Login Basic Flow

Figure 2. 9 describes the alternative flow in Login module. <<Forgot Password>> button is clicked when user forgot his password and would like to reset it. Once clicked, the request is sent to the controller. Then, the Forgot Password interface is displayed. The user will then have to enter his email associated with the account. The controller will send a password reset link to the user's email. Through the link, the user will set a new password. Then, this password is saved into the model.

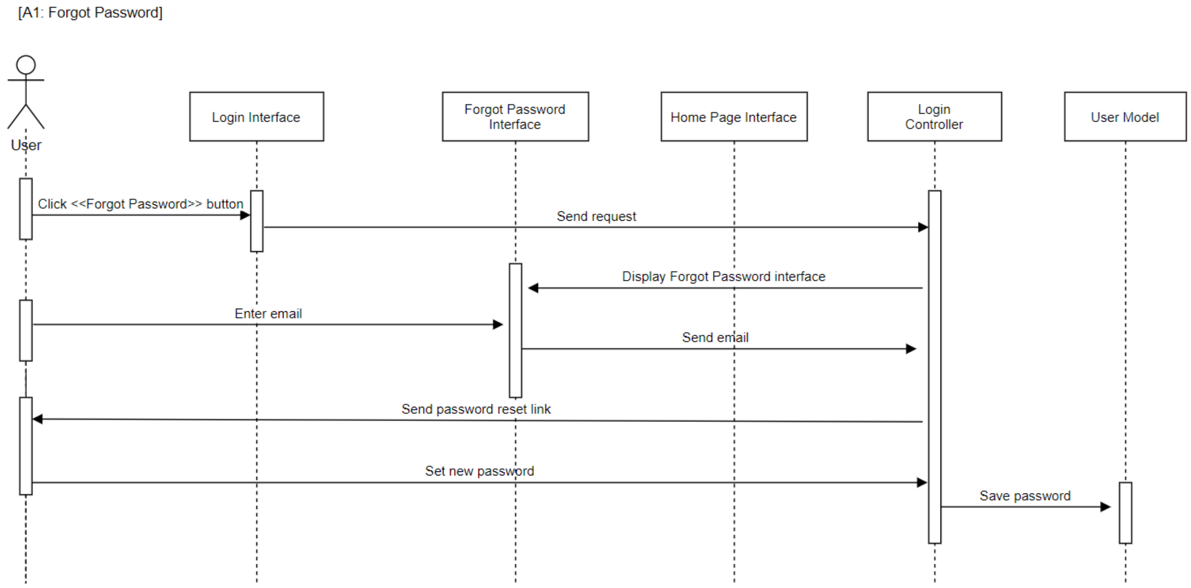


Figure 2. 9 Login [A1: Forgot Password] Alternative Flow

Figure 2. 10 describes the exception flow of the Login module. During the process of verifying user's login credentials, the model detects the credentials do not match with those in the model. The error message will then be sent to notify the user.

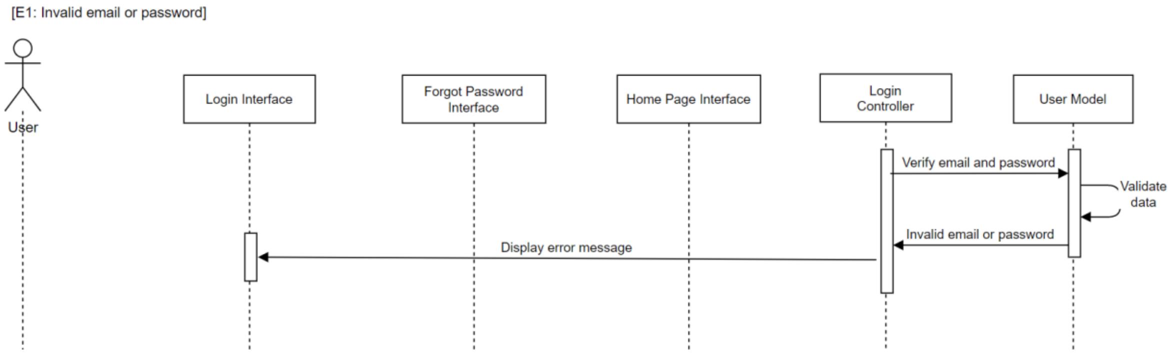

[E1: Invalid email or password]
User
Login Interface
Forgot Password Interface
Home Page Interface
Login Controller
User Model
Verify email and password
Validate data
Invalid email or password
Display error message

Figure 2. 10 Login [E1: Invalid email or password] Exception Flow

2.2.2 Manage User Data

Figure 2. 11 describes the basic flow of Manage User Data module. The admin first enters the user list interface. Then, the controller will retrieve the user list from the model and display it on the interface. From the list, a user will be selected by the admin. Then, the controller retrieves the selected user info from the model. The retrieved info is then displayed onto the user profile interface. The admin can edit the info and click <<Update>> button. This prompts the controller to update the info in the model.

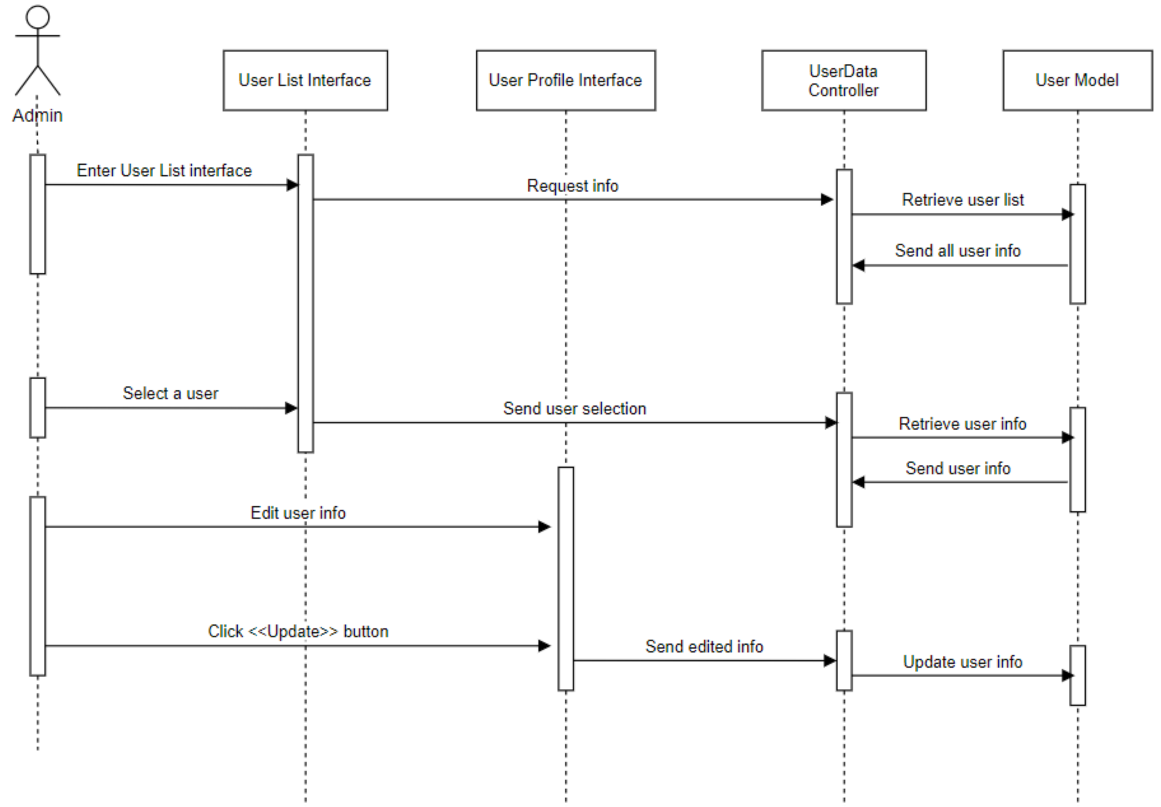


Figure 2. 11 Manage User Data Basic Flow

2.2.3 Access Breach Records

Figure below describes the basic flow of Access Breach Records module. As the user enters the Helmet Breach interface from the menu list, the controller retrieves the breach data and display them onto the interface in a list.

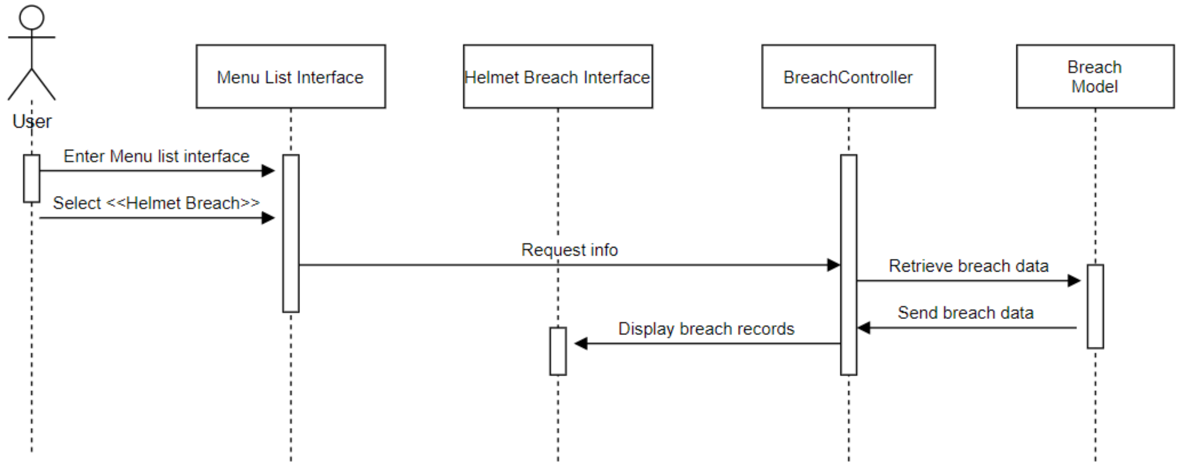


Figure 2. 12 Access Breach Records Basic Flow

2.2.4 Receive Notifications

Figure below describes the basic flow of Receive Notifications module. As the camera detects workers who do not comply with wearing the safety helmet, the controller will save the breach info into the model. Then, a push notification is sent to the user. As the user clicks into the notification, the controller retrieves the breach info and display it onto the Helmet Breach interface.

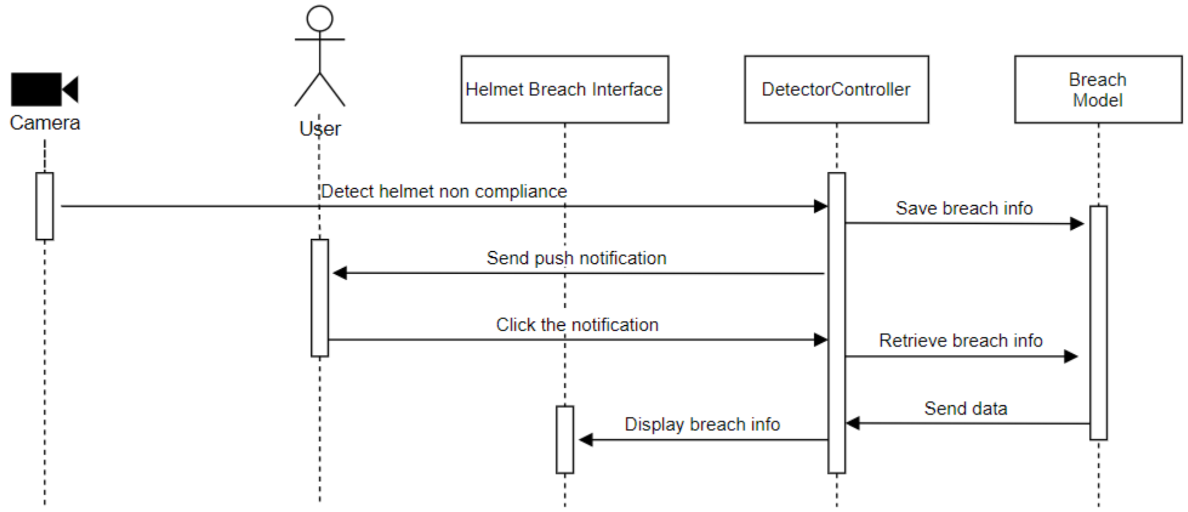


Figure 2. 13 Receive Notifications Basic Flow

2.2.5 Manage PPE Inventory

Figure below shows the basic flow of Manage PPE Inventory module. From the menu list interface, the user can select <<PPE Inventory>>, then the controller retrieves the PPE list from the model and display the list on the PPE inventory interface. Then, as the user selects a PPE from the list, the PPE info is retrieved and display onto the PPE profile interface. Here, user can edit the info of the PPE and click on <<Save>> button. The controller then updates the info in the model.

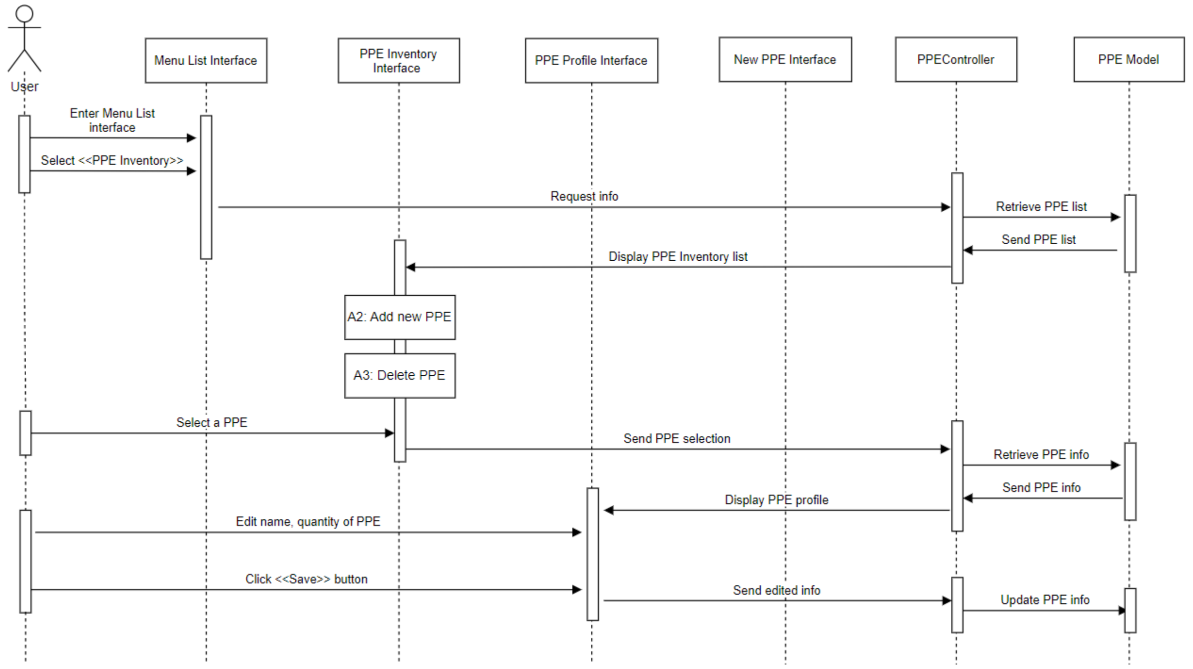


Figure 2. 14 Manage PPE Inventory Basic Flow

Figure below describes the first alternative flow of Manage PPE Inventory module. As the user clicks on the <<Add>> button at the PPE inventory interface, the controller will display the new PPE interface for user to add a new PPE. Then, the details of the PPE is entered and the <<Save>> button is clicked. The controller will then save the newly added PPE into the model.

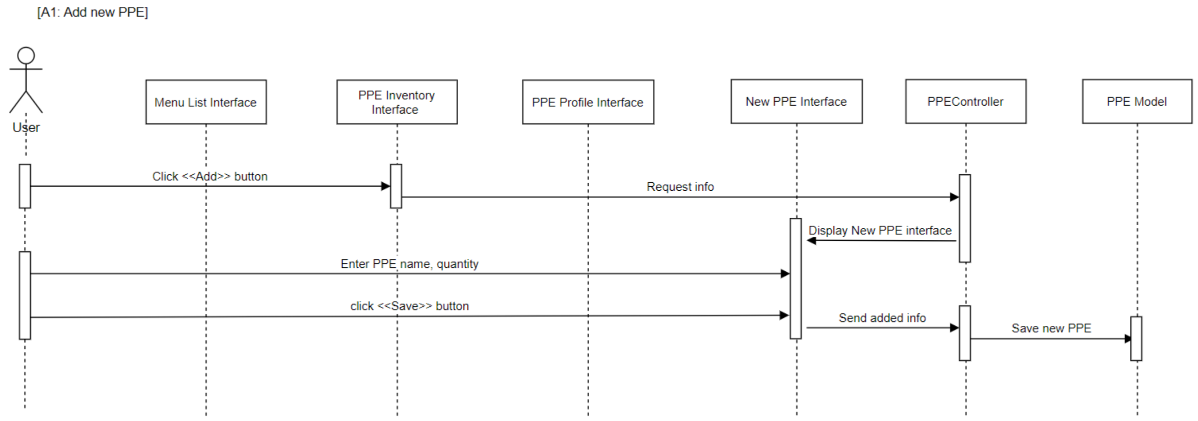


Figure 2. 15 Manage PPE Inventory [A1: Add new PPE] Alternative Flow

Figure below describes the second alternative flow of Manage PPE Inventory module. As user clicks on the <<Delete>> button at a PPE's profile, the decision is sent to the controller and a confirmation window is prompted. The user confirms the decision by clicking on the <<Delete>> button and the PPE info will then be deleted from the model.

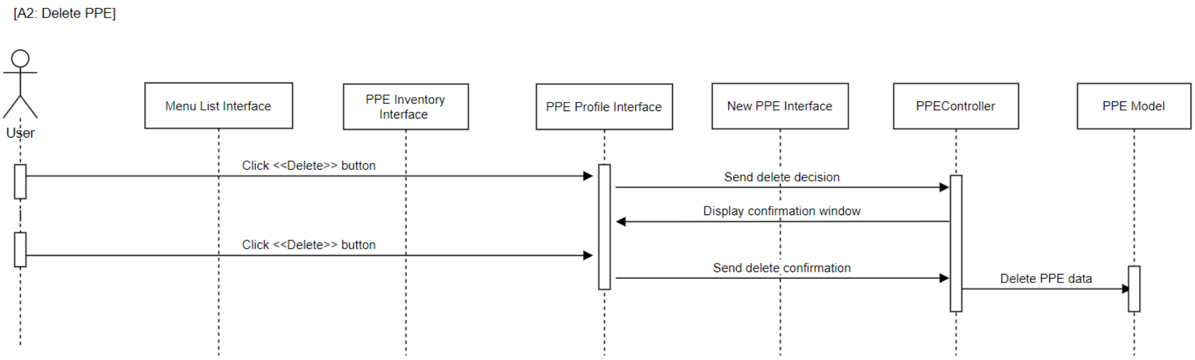


Figure 2. 16 Manage PPE Inventory [A2: Delete PPE] Alternative Flow

2.2.6 Add Feedback

Figure below describes the basic flow of Add Feedback module. As the user selects <<Feedback>> from the menu list, the controller displays the feedback interface. The user then enters the feedback and clicks on <<Submit>> button. The controller saves the feedback added by the user into the database through the model.

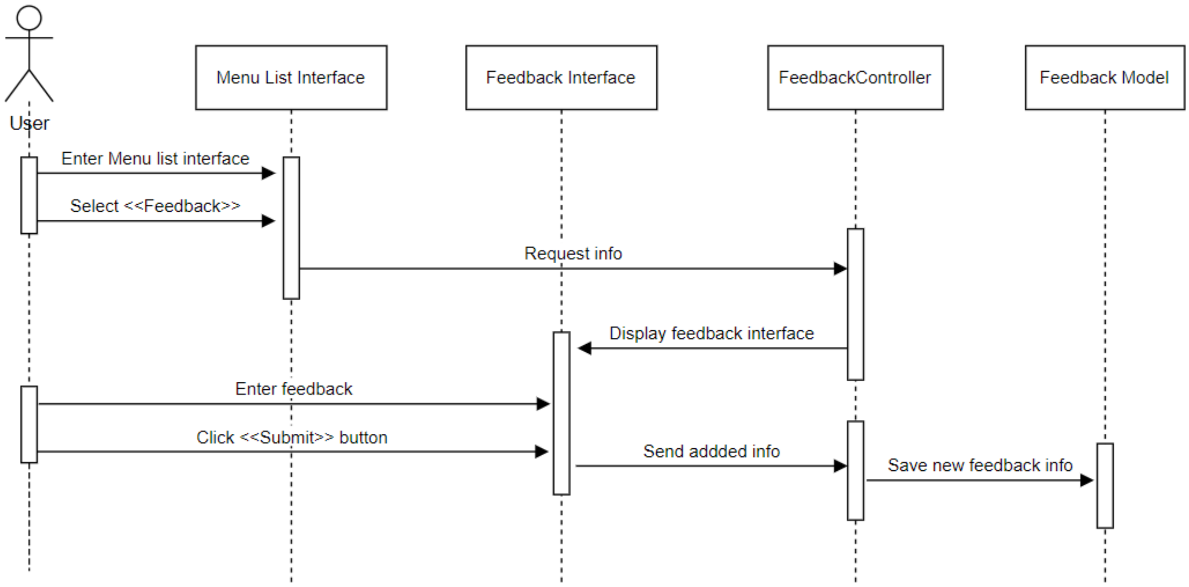


Figure 2. 17 Add Feedback Basic Flow

CHAPTER 3

3.1 INTERFACE DESIGN

Figure 3. 1 shows the interface for user to log into the application. The application requires the user's email and password to perform further actions on the application. Registering a new account for the manager (admin of the company) and supervisors (normal users of the company) to access the app requires the application admin to create and delete the user accounts.

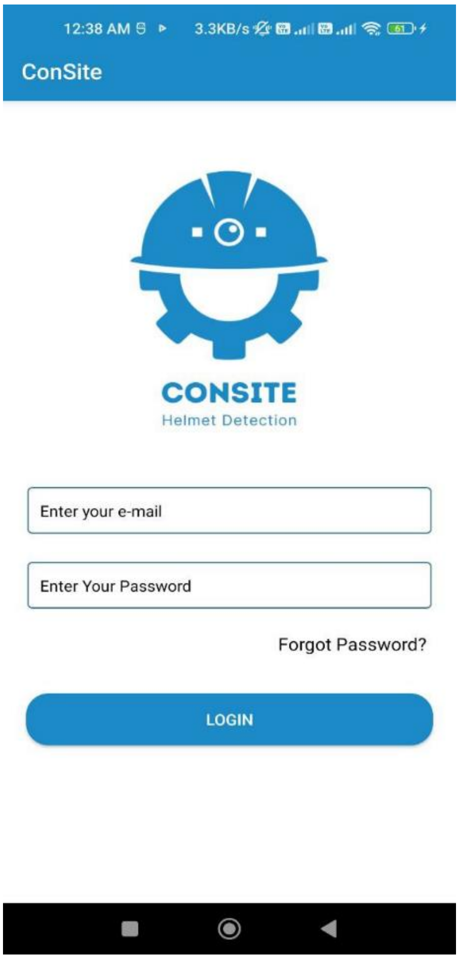
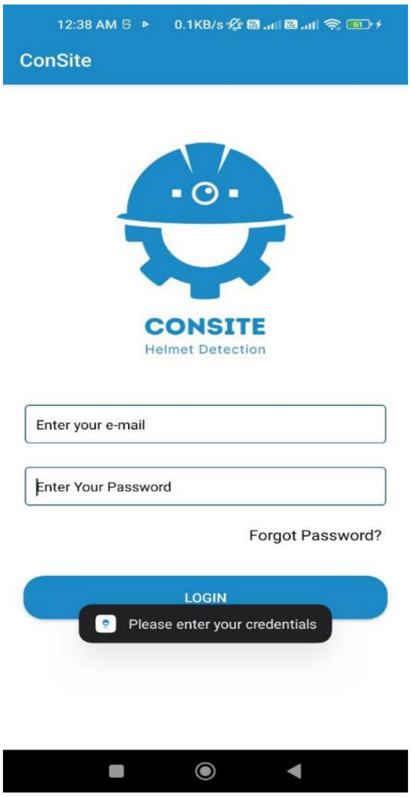


Figure 3. 1 Login interface

When the user does not enter an email or password or both when logging into the application, an error message will be popped up as shown in the figure below to remind the user not to leave the fields blank.



12:38 AM 0.1KB/s

ConSite

CONSITE
Helmet Detection

Enter your e-mail

Enter Your Password

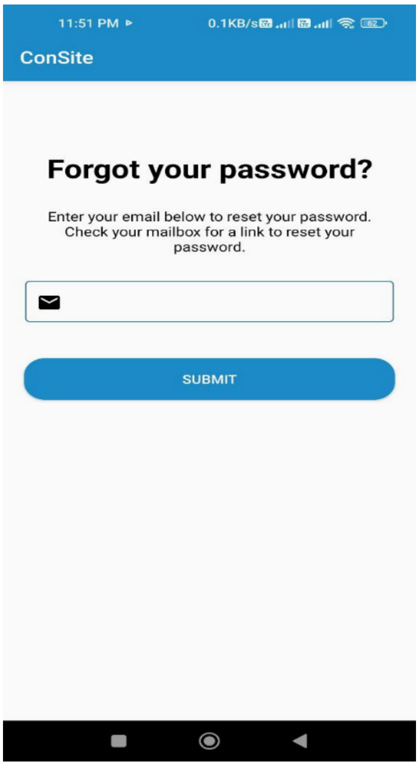
Forgot Password?

LOGIN

Please enter your credentials

Figure 3. 2 Error message for not providing email or password or both during login

Figure 3. 3 below describes the interface for the user to reset his account's password by providing the email that is used to access the account.



11:51 PM 0.1KB/s

ConSite

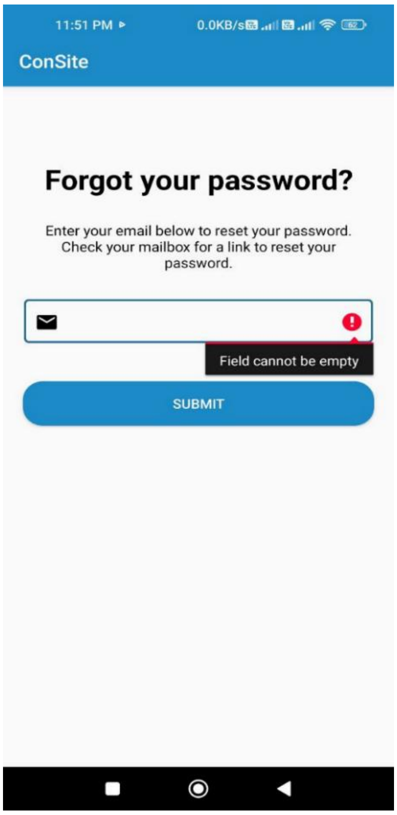
Forgot your password?

Enter your email below to reset your password. Check your mailbox for a link to reset your password.

SUBMIT

Figure 3. 3 Forgot password interface

When the user does not enter the email for resetting the password, an error message will be popped up as shown in Figure 3. 4 below to remind the user not to leave the fields blank.



11:51 PM

ConSite

Forgot your password?

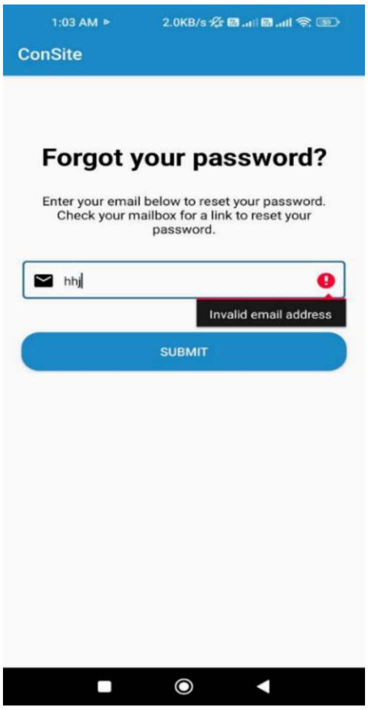
Enter your email below to reset your password. Check your mailbox for a link to reset your password.

Field cannot be empty

SUBMIT

Figure 3. 4 Error message for not providing an email during password reset

If an invalid email is given when resetting the password of an account, an error message as shown in Figure 3. 5 below will be popped up to inform the user to provide a proper one to reset the password.



1:03 AM

ConSite

Forgot your password?

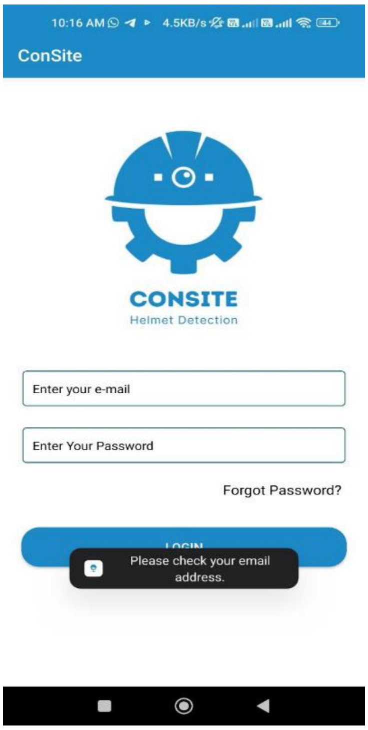
Enter your email below to reset your password. Check your mailbox for a link to reset your password.

hhj

Invalid email address

Figure 3. 5 Error message for not providing a proper email during password reset

If the reset password link has been successfully sent to the email entered for resetting the password, a pop-up message just as in Figure 3. 6 is shown to tell the user to check the email that has been sent to his inbox.



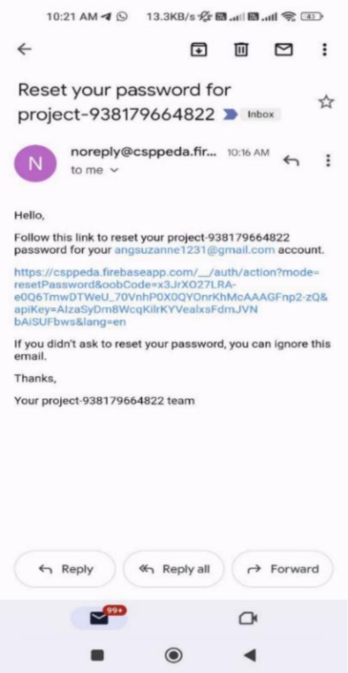
10:16 AM

ConSite

Please check your email address.

Figure 3. 6 Pop up message after email account for password reset is valid

Figure 3. 7 below shows the link that has been sent to the user's email to reset the password. From here, the user needs to click on the link given.



Reset your password for project-938179664822

noreply@csppeda.fir... 10:16 AM
to me

Hello,

Follow this link to reset your project-938179664822 password for your angsuzanne1231@gmail.com account.

https://csppeda.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=x3JrXO27LRA-e0Q6TmwDTWeU_70VnhP0X0QYOnrKhMcAAAGFnp2-zQ&apiKey=AIzaSyDm8WcqKilrKYVealxsFdmJVNbAiSUFbws&lang=en

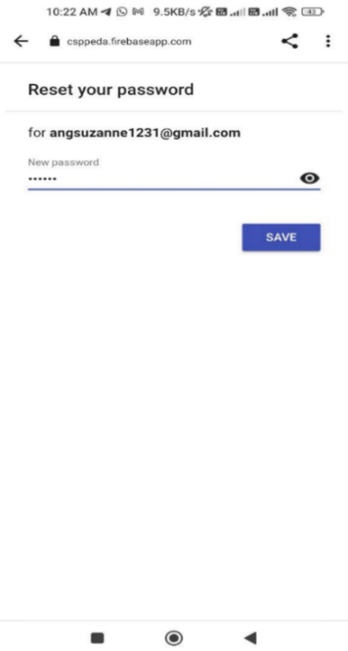
If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-938179664822 team

Figure 3. 7 Email containing the password reset link

After the password reset link is clicked, the user is redirected to this page as shown in Figure 3. 8. The new password will be entered for the registered email account.



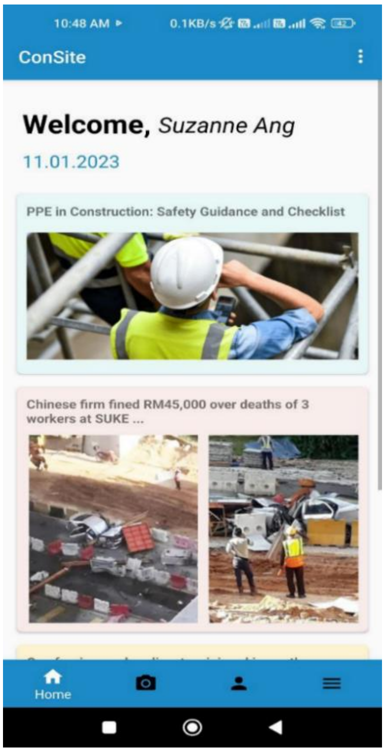
Reset your password

for angsuzanne1231@gmail.com

New password
••••••

Figure 3. 8 Text field for entering new password for the email account

After the user is successfully logged into the application, the user will be presented with different interfaces according to the type of user he is. The admin of the company will have a bottom navigation bar with 4 tabs as shown in Figure 3. 9, whereas the supervisors will have a bottom navigation bar with 3 tabs as shown in Figure 3. 10.



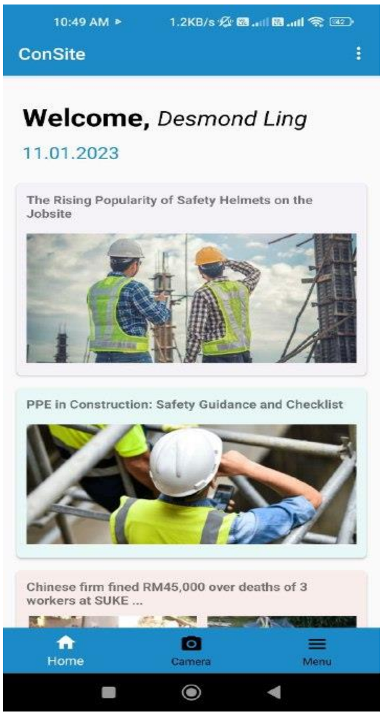
Welcome, *Suzanne Ang*

11.01.2023

PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

Figure 3. 9 Admin home interface when admin logs in



Welcome, *Desmond Ling*

11.01.2023

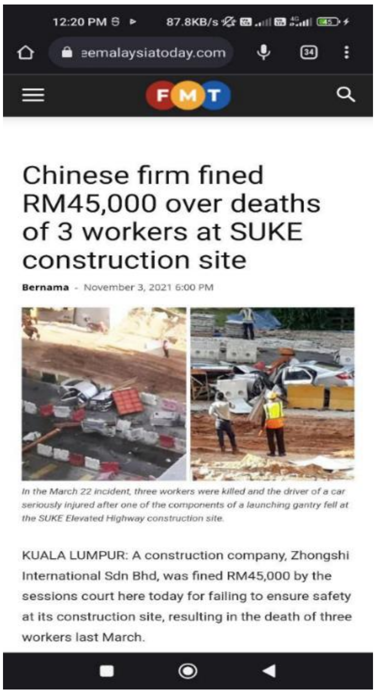
The Rising Popularity of Safety Helmets on the Jobsite

PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

Figure 3. 10 Supervisor home interface when supervisor logs in

When any article on the home interface is clicked, the user will be redirected to the article website just as in Figure 3. 11 below.



Chinese firm fined RM45,000 over deaths of 3 workers at SUKE construction site

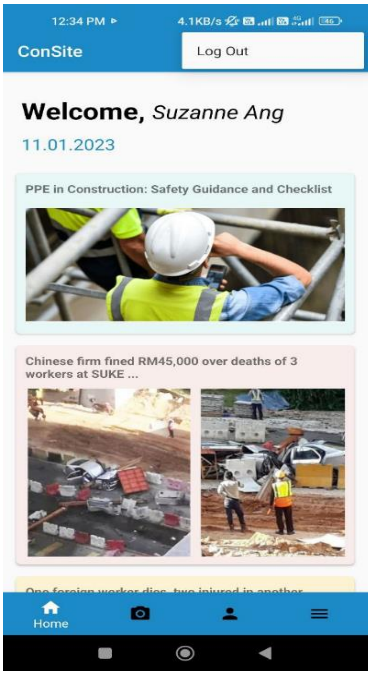
Bernama - November 3, 2021 6:00 PM

In the March 22 incident, three workers were killed and the driver of a car seriously injured after one of the components of a launching gantry fell at the SUKE Elevated Highway construction site.

KUALA LUMPUR: A construction company, Zhongshi International Sdn Bhd, was fined RM45,000 by the sessions court here today for failing to ensure safety at its construction site, resulting in the death of three workers last March.

Figure 3. 11 Website loaded after an article is clicked

There is a Logout button at the top bar in both the admin and the supervisors home interface as shown in Figure 3. 12. To logout the application, the Log Out button icon at the top bar is clicked.



Welcome, *Suzanne Ang*

11.01.2023

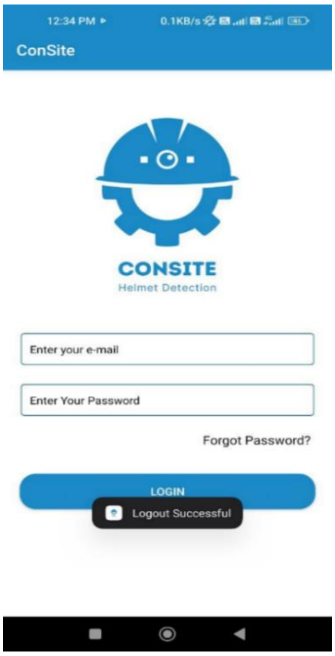
PPE in Construction: Safety Guidance and Checklist

Chinese firm fined RM45,000 over deaths of 3 workers at SUKE ...

One foreign worker dies, two injured in another

Figure 3. 12 Log out button for logging out the application

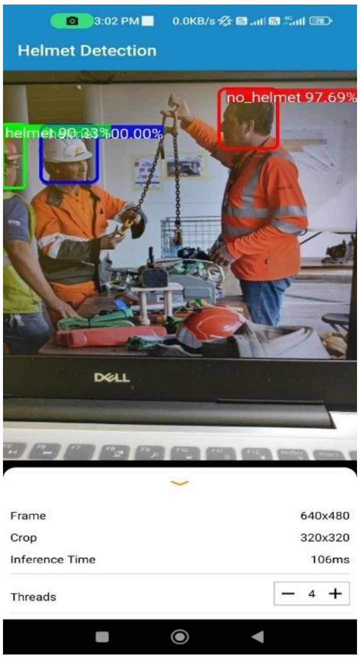
Once the user presses on the log out button, the user will be redirected back to the login interface and a pop-up message informing he is logged out is displayed as shown in the figure below.



Logout Successful

Figure 3. 13 Pop-up message of user successfully logged out

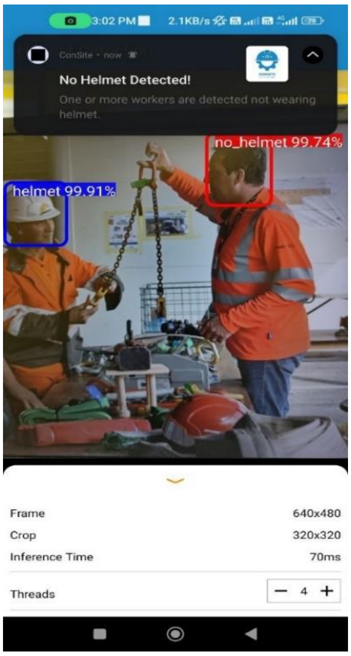
The camera feature can be accessed from the application through clicking the camera icon at the bottom navigation bar from the admin and supervisor home page. The camera can detect workers who wear helmet or without helmet as shown in the figure below.



Frame 640x480
Crop 320x320
Inference Time 106ms
Threads − 4 +

Figure 3. 14 Camera interface for detecting workers

If a worker is detected without helmet from the camera feature in the application, a notification is triggered as shown in the figure below and sent to the user's phone to alert him.



ConSite • now

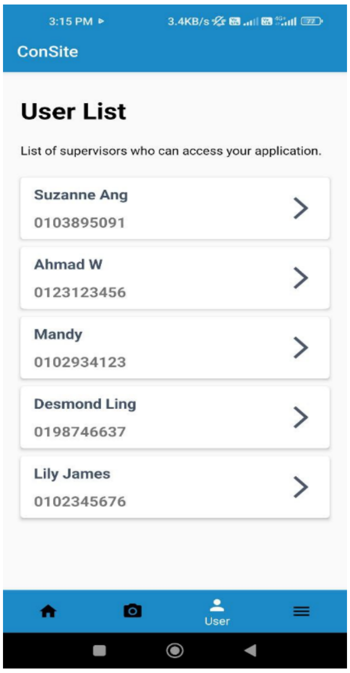
No Helmet Detected!

One or more workers are detected not wearing helmet.

Frame	640x480
Crop	320x320
Inference Time	70ms
Threads	− 4 +

Figure 3. 15 Notification when a worker is detected without helmet

Figure below describes the user list interface that is accessible only by the admin of the company. The admin will be redirected to this page when the user icon is selected from the bottom navigation bar.



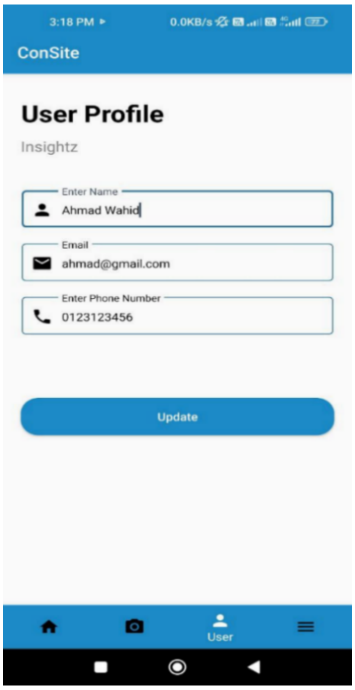
User List

List of supervisors who can access your application.

Suzanne Ang	0103895091
Ahmad W	0123123456
Mandy	0102934123
Desmond Ling	0198746637
Lily James	0102345676

Figure 3. 16 User List interface

When a user is selected from the user list, the profile of the user is shown as in the figure below where the admin can edit the name and phone number. The email address of the user account is set to be non-editable. The company name is printed out at the top left of the page.



User Profile

Insightz

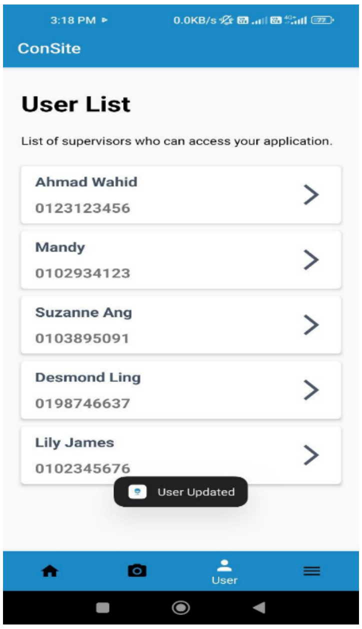
Enter Name: Ahmad Wahid

Email: ahmad@gmail.com

Enter Phone Number: 0123123456

Figure 3. 17 Update user information interface

A pop-up message as shown in the figure below is displayed when the information of a user is successfully updated. The admin will be redirected back to the user list interface.



User List

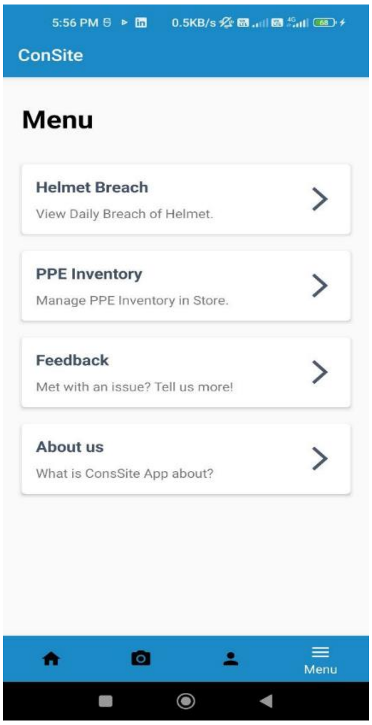
List of supervisors who can access your application.

Ahmad Wahid	0123123456
Mandy	0102934123
Suzanne Ang	0103895091
Desmond Ling	0198746637
Lily James	0102345676

User Updated

Figure 3. 18 Pop-up message after user info is updated

When the menu icon at the bottom navigation bar is clicked, the user is redirected to the menu interface. The functions in the admin menu interface as shown in Figure 3. 19 are the same as in the supervisor menu interface as shown in Figure 3. 20. The only difference is the admin has the user tab in the bottom navigation bar.



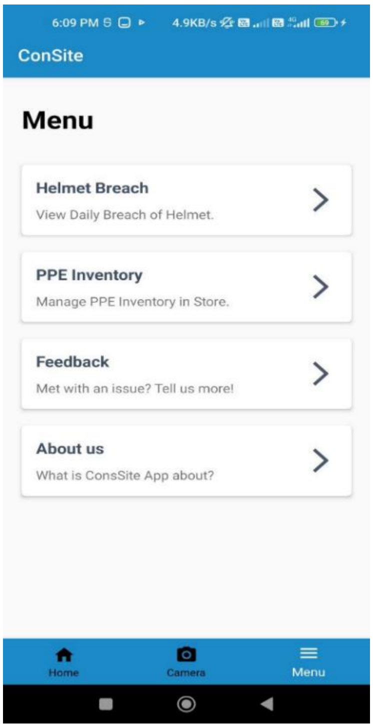
Helmet Breach
View Daily Breach of Helmet.

PPE Inventory
Manage PPE Inventory in Store.

Feedback
Met with an issue? Tell us more!

About us
What is ConsSite App about?

Figure 3. 19 Admin menu interface



Helmet Breach
View Daily Breach of Helmet.

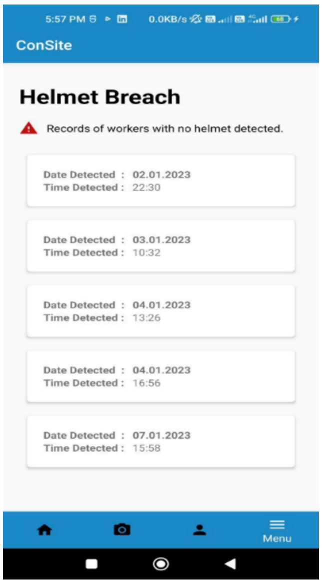
PPE Inventory
Manage PPE Inventory in Store.

Feedback
Met with an issue? Tell us more!

About us
What is ConsSite App about?

Figure 3. 20 Supervisor menu interface

When the user selects Helmet Breach from the menu list, the user is redirected to the helmet breach interface as shown in the figure below. Also, if the user clicks on the notification that is sent by the application, the user is redirected to this page. The breaches of workers not wearing helmet are listed out here with the date and time of the workers detected.



5:57 PM 0.0KB/s

ConSite

Helmet Breach

Records of workers with no helmet detected.

Date Detected : 02.01.2023
Time Detected : 22:30

Date Detected : 03.01.2023
Time Detected : 10:32

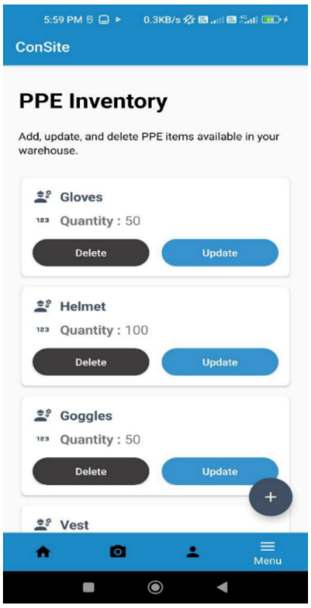
Date Detected : 04.01.2023
Time Detected : 13:26

Date Detected : 04.01.2023
Time Detected : 16:56

Date Detected : 07.01.2023
Time Detected : 15:58

Figure 3. 21 Helmet breach interface

When the user selects PPE Inventory from the menu list, the user is redirected to the PPE inventory interface as shown in Figure 3. 22. The actions that can be performed are updating the records of the PPE and deleting a record. To add a new PPE, the floating add button can be pressed.



5:59 PM 0.3KB/s

ConSite

PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 50
Delete Update

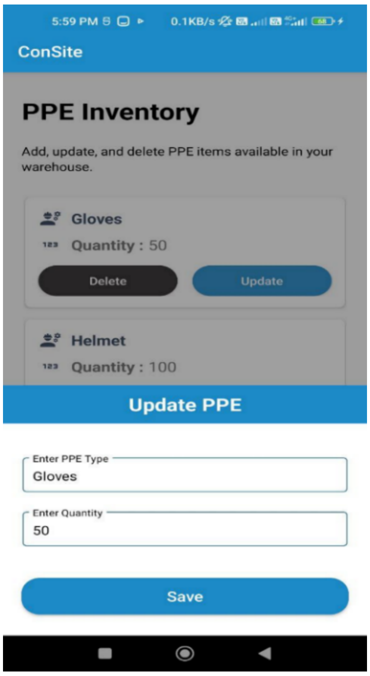
Helmet
Quantity : 100
Delete Update

Goggles
Quantity : 50
Delete Update

Vest

Figure 3. 22 PPE inventory interface

When the update button of a PPE item is pressed, a pop-up box is shown on the page just as in Figure 3. 23. The details of the item displayed. After pressing on the Save button, a pop-up message is displayed as shown in Figure 3. 24 to inform that the item has been successfully updated.



5:59 PM 0.1KB/s

ConSite

PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 50

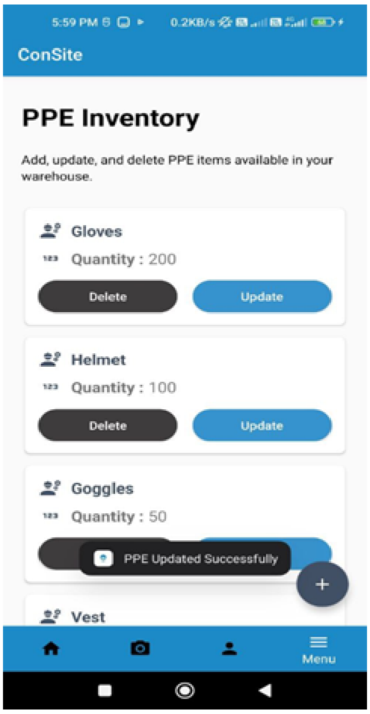
Helmet
Quantity : 100

Update PPE

Enter PPE Type: Gloves

Enter Quantity: 50

Figure 3. 23 Update PPE interface



5:59 PM 0.2KB/s

ConSite

PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves
Quantity : 200

Helmet
Quantity : 100

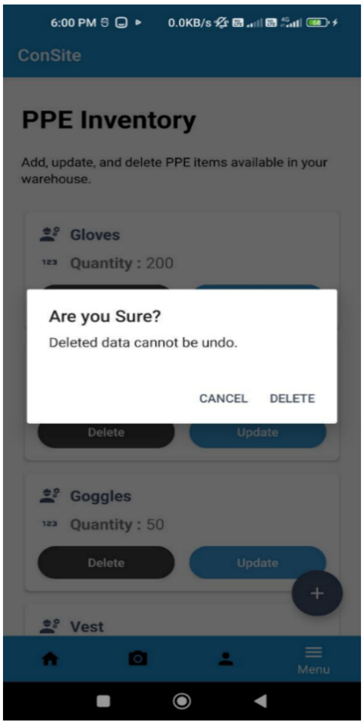
Goggles
Quantity : 50

PPE Updated Successfully

Vest

Figure 3. 24 Message after PPE is successfully updated

A confirmation pop-up message as shown in Figure 3. 25 below is prompted when the user clicks on the Delete button of a PPE item at the list of PPE.



PPE Inventory

Add, update, and delete PPE items available in your warehouse.

Gloves

Quantity : 200

Are you Sure?

Deleted data cannot be undo.

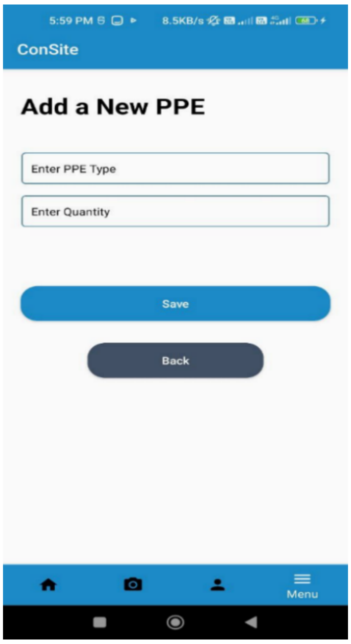
Goggles

Quantity : 50

Vest

Figure 3. 25 Confirmation message to delete a PPE

After the user clicks on the floating add button at the PPE inventory interface, the user is redirected to this Add new PPE interface as shown in the figure below. The PPE type and the quantity can be added to the PPE list from here.



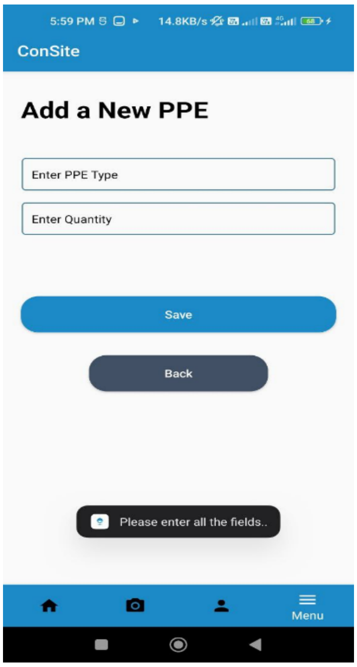
Add a New PPE

Enter PPE Type

Enter Quantity

Figure 3. 26 Add new PPE interface

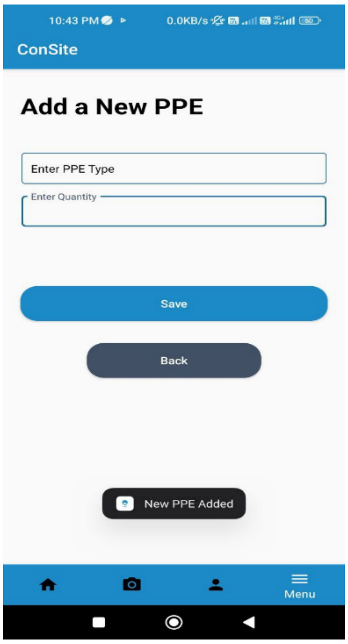
If no input is given in either one of the field or both fields, the application will display a pop-up message as shown in the figure below to indicate all the fields must be entered before adding a new PPE into the list.



Please enter all the fields..

Figure 3. 27 Error message when no input is given when adding new PPE

After a PPE is added to the application, a pop-up message is displayed just as in the figure below. The field for adding new PPEs are cleared to allow the user to add another new PPE.



New PPE Added

Figure 3. 28 Pop-up message that new PPE is successfully added

When Feedback is selected from the menu section by both the admin and supervisor, they will be redirected to this Feedback interface as shown in the figure below. The users can describe their issues with maximum 500 words. The feedback together with the email of the user is saved into the database.

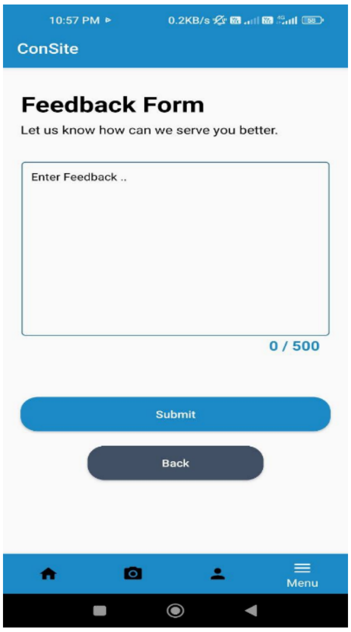


Figure 3. 29 Feedback interface

If no input is entered in the field and the user clicks on Save button, the application will prompt an error message just as in the figure below to inform the user to fill up the field.

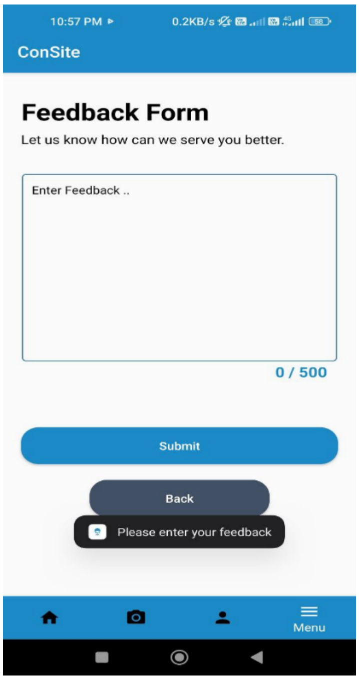
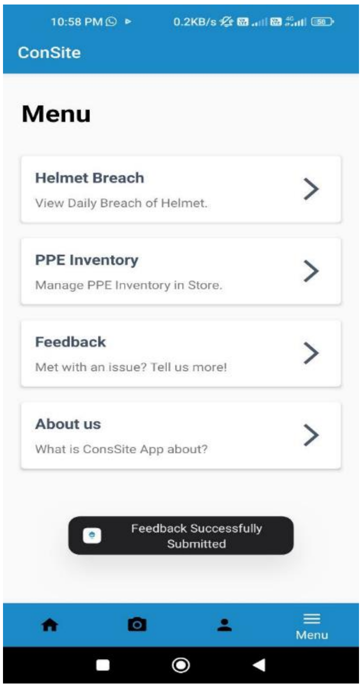


Figure 3. 30 Error message if no field is entered in Feedback interface

After the feedback is successfully submitted, the user is redirected back to the menu section and a successful message is shown just as in the figure below.



Menu

Helmet Breach
View Daily Breach of Helmet.

PPE Inventory
Manage PPE Inventory in Store.

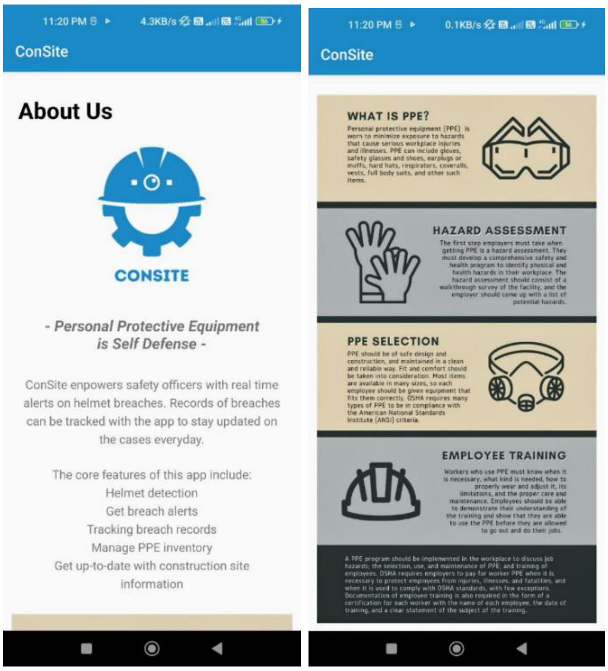
Feedback
Met with an issue? Tell us more!

About us
What is ConsSite App about?

Feedback Successfully Submitted

Figure 3. 31 Successful message that feedback is submitted

When About us is selected from the menu section by the admin or the supervisor, the user is redirected to this About us interface as shown in the figure below where information about this application can be viewed here.



About Us

- Personal Protective Equipment is Self Defense -

ConSite enpowers safety officers with real time alerts on helmet breaches. Records of breaches can be tracked with the app to stay updated on the cases everyday.

The core features of this app include:
Helmet detection
Get breach alerts
Tracking breach records
Manage PPE inventory
Get up-to-date with construction site information

WHAT IS PPE?

Personal protective equipment (PPE) is worn to minimize exposure to hazards that cause serious workplace injuries and illnesses. PPE can include gloves, safety glasses and shoes, earplugs or muffs, hard hats, respirators, coveralls, vests, full body suits, and other such items.

HAZARD ASSESSMENT

The first step employers must take when getting PPE is a hazard assessment. They must develop a comprehensive safety and health program to identify physical and health hazards in their workplace. The hazard assessment should consist of a walkthrough survey of the facility, and the employer should come up with a list of potential hazards.

PPE SELECTION

PPE should be of safe design and construction, and maintained in a clean and reliable way. Fit and comfort should be taken into consideration. Most items are available in many sizes, so each employee should be given equipment that fits them correctly. OSHA requires many types of PPE to be in compliance with the American National Standards Institute (ANSI) criteria.

EMPLOYEE TRAINING

Workers who use PPE must know when it is necessary, what kind is needed, how to properly wear and adjust it, its limitations, and the proper care and maintenance. Employees should be able to demonstrate their understanding of the training and show that they are able to use the PPE before they are allowed to go out and do their jobs.

A PPE program should be implemented in the workplace to discuss job hazards; the selection, use, and maintenance of PPE; and training of employees. OSHA requires employers to pay for worker PPE when it is necessary to protect employees from injuries, illnesses, and fatalities, and when it is used to comply with OSHA standards, with few exceptions. Documentation of employee training is also required in the form of a certification for each worker with the name of each employee, the date of training, and a clear statement of the subject of the training.

Figure 3. 32 About us interface

3.2 HARDWARE AND SOFTWARE SPECIFICATION

Table 3. 1 Hardware Specification

Hardware	Specification	Description
Laptop	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz	Device for development of the application and documentation of the project.
Android mobile phone	Huawei Mi 10T	Device for running the application.

Table 3. 2 Software Specification

Software	Description
Windows 10	Operating system for running all the required software.
Microsoft Word	For system documentation.
Android Studio	For developing the application.
Draw.io	Platform for designing relevant diagrams.
Firebase	Realtime database for application development.

APPENDIX D
SOFTWARE DESIGN DOCUMENT (SDD)

2020

SOFTWARE DESIGN DESCRIPTION (SDD)

[CONSTRUCTION SITE SAFETY HELMET DETECTION IN MOBILE APPLICATION]

DOCUMENT APPROVAL

	Name	**Date**
Authenticated by: Name	ANG SUZANNE	8th February 2023
Approved by: Client		

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

Construction Site Safety Helmet Detection Application is a mobile application that uses computer vision-based technology mainly to detect the absence of safety helmet on workers. This application allows for the automated collection and detection of workers' improper dress codes, which in this case, the helmet, on construction sites. The users of the application are the admin and supervisors of the company. There are 6 main modules available in this application, which are Login, Manage User Data, Access Breach Records, Receive Notifications, Manage PPE Inventory, and Add Feedback.

The Login module enables the user to log into the system. The Manage User Data module allows the admin to manage the data of the user which includes updating the user data. The Access Breach Records module allows the admin and supervisor to view the records of helmet breach. The Receive Notifications module allows the admin and supervisor to receive notifications concerning any helmet breach related issues. Manage PPE Inventory enables the users to handle the inventory of PPE, which includes adding, updating, and deleting PPE inventory. The last module, Add Feedback, enables the users to submit feedback on the issues met or improvements to be suggested.

1.2 SYSTEM IDENTIFICATION

This document uses the following naming convention:

System identification number: **CONSITE-SDD-2022-V1**

Table 1. 1 System Identification

CONSITE	Construction Site Safety Helmet Detection Application
SDD	Software Design Document
2022	Year 2022
V1	Version 1

1.3 ARCHITECTURE / BLUE PRINT

MVC architecture is implemented in the proposed application, in which it consists of 3 layers, Model (M), View (V), and Controller (C). The figure below describes the general architecture of the application in the form of an MVC design architecture,

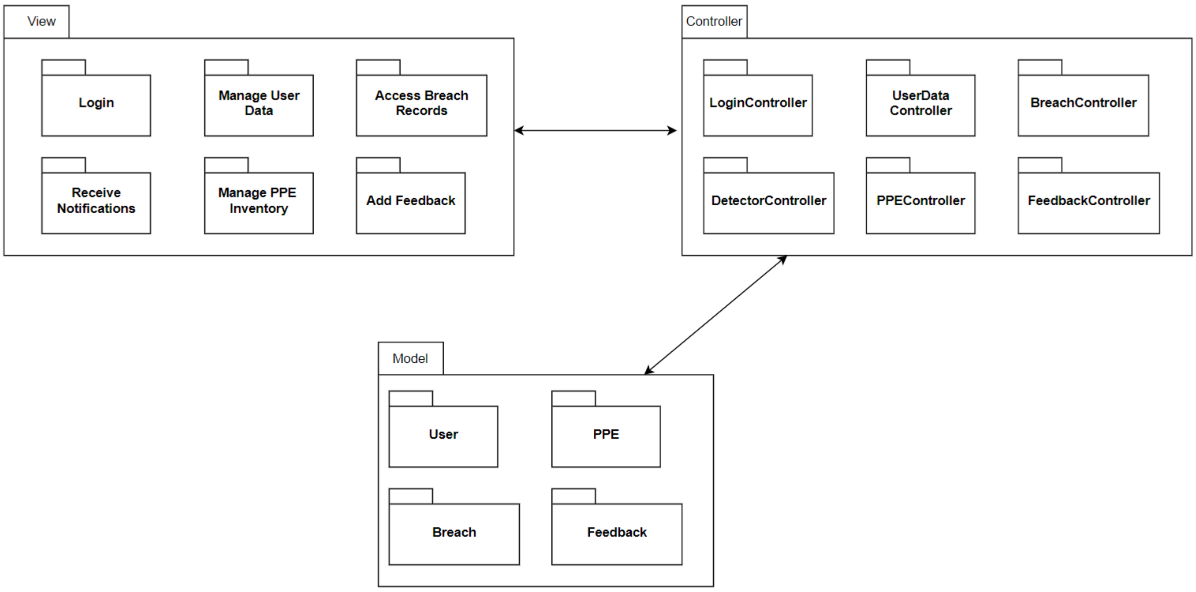


Figure 1. 1 General Architecture

1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

1.4.1 Application Layer

1.4.1.1 Login

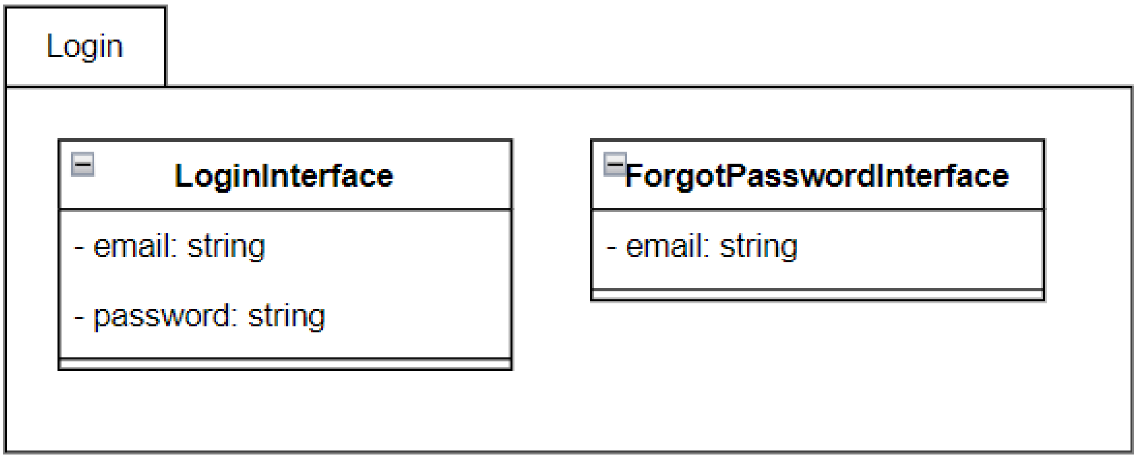


Figure 1. 2 Login Module Application Layer

Table 1. 2 Login Module Application Layer

Class Name	Description
LoginInterface	Interface for users to log into the application once the application is opened.
ForgotPasswordInterface	Interface for users to reset the password.

1.4.1.2 Manage User Data

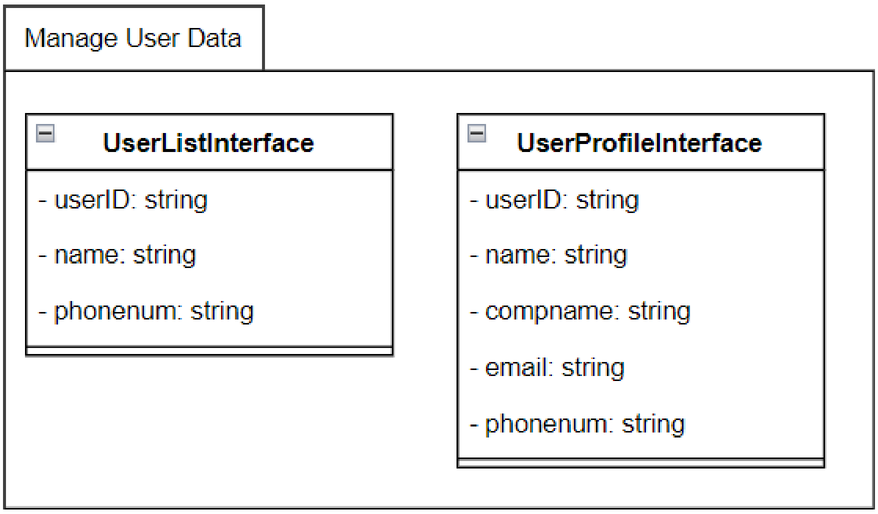


Figure 1. 3 Manage User Data Module Application Layer

Table 1. 3 Manage User Data Module Application Layer

Class Name	Description
UserListInterface	Interface for the admin to view all users' info in a list.
UserProfileInterface	Interface for the admin to view a user's profile and update the user's info.

1.4.1.3 Access Breach Records

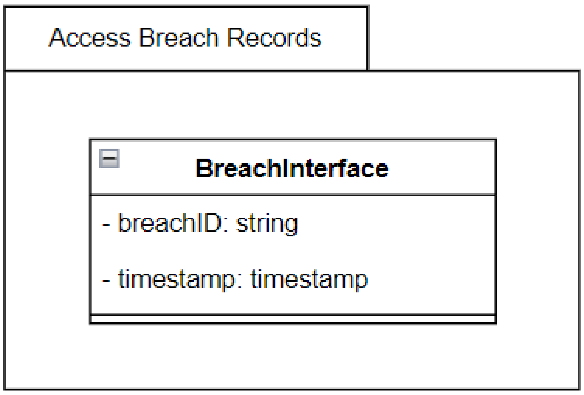


Figure 1. 4 Access Breach Records Module Application Layer

Table 1. 4 Access Breach Records Module Application Layer

Class Name	Description
BreachInterface	Interface for the user to view the breach records in a list.

1.4.1.4 Receive Notifications

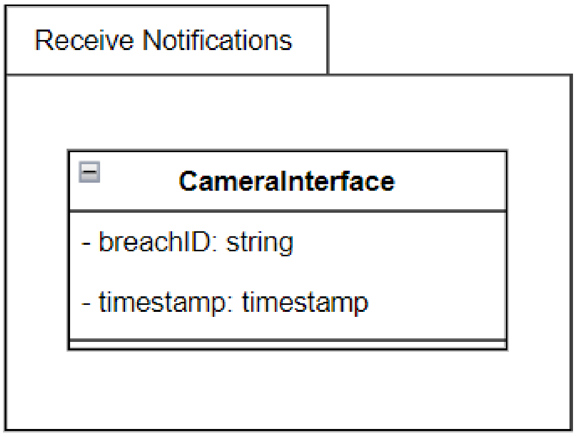


Figure 1. 5 Receive Notifications Module Application Layer

Table 1. 5 Receive Notifications Module Application Layer

Class Name	Description
CameraInterface	Interface for the camera to be activated and detect any person not complying to wearing helmet.

1.4.1.5 Manage PPE Inventory

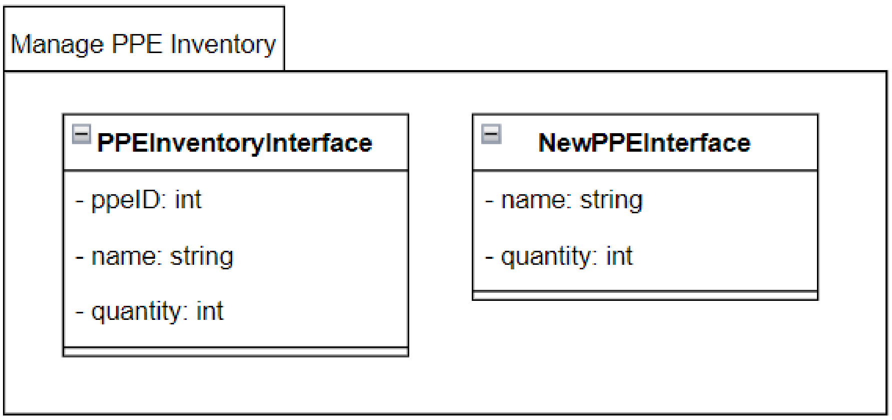


Figure 1. 6 Manage PPE Inventory Module Application Layer

Table 1. 6 Manage PPE Inventory Module Application Layer

Class Name	Description
PPEInventoryInterface	Interface for the user to view all PPE in a list, and update or delete a PPE.
NewPPEInterface	Interface for the user to add a new PPE into the application.

1.4.1.6 Add Feedback

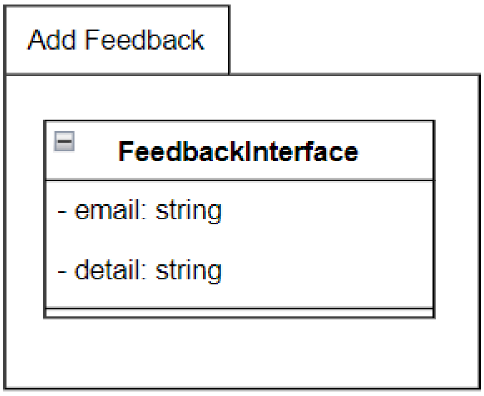


Figure 1. 7 Add Feedback Module Application Layer

Table 1. 7 Add Feedback Module Application Layer

Class Name	Description
FeedbackInterface	Interface for the user to submit feedback on the issue of the application.

1.4.2 Business Services Layer

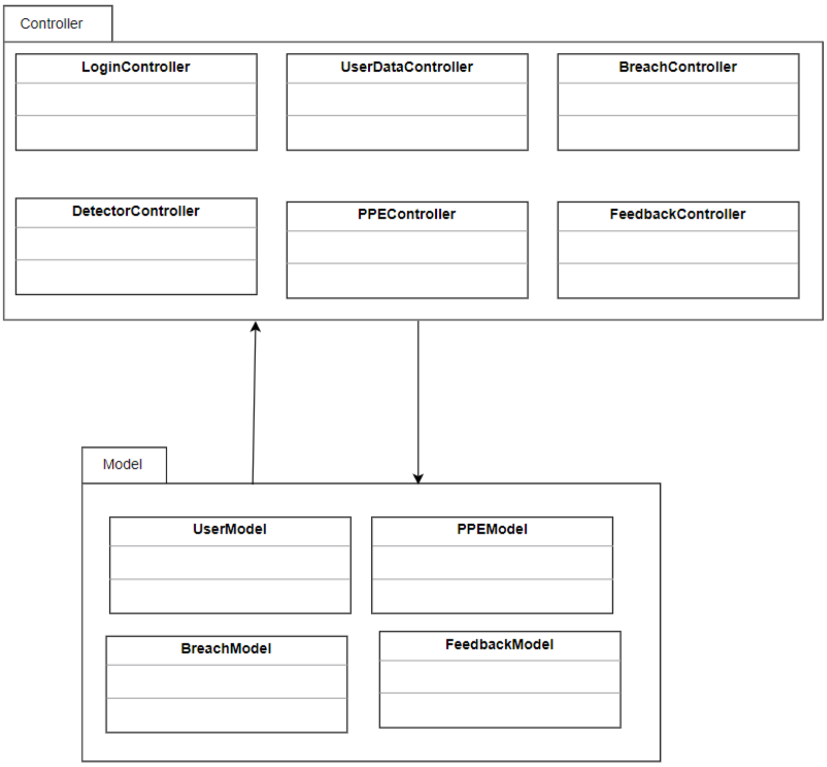


Figure 1. 8 Business Services Layer

1.4.2.1 Controllers

Table 1. 8 Controllers

Class Name	Description
LoginController	This controller communicates and manages the process logic between the interfaces and models for Login module.
UserDataController	This controller communicates and manages the process logic between the interfaces and models for Manage User Data module.

BreachController	This controller communicates and manages the process logic between the interfaces and models for Access Breach Records module.
DetectorController	This controller communicates and manages the process logic between the interfaces and models for Receive Notifications module.
PPEController	This controller communicates and manages the process logic between the interfaces and models for Manage PPE Inventory module.
FeedbackController	This controller communicates and manages the process logic between the interfaces and models for Add Feedback module.

1.4.2.2 Models

Table 1. 9 Models

Class Name	Description
UserModel	This model connects the user table in the database for storing and retrieving data.
PPEModel	This model connects the PPE table in the database for storing and retrieving data.
BreachModel	This model connects the breach table in the database for storing and retrieving data.
FeedbackModel	This model connects the feedback table in the database for storing and retrieving data.

1.4.3 Middleware Layer

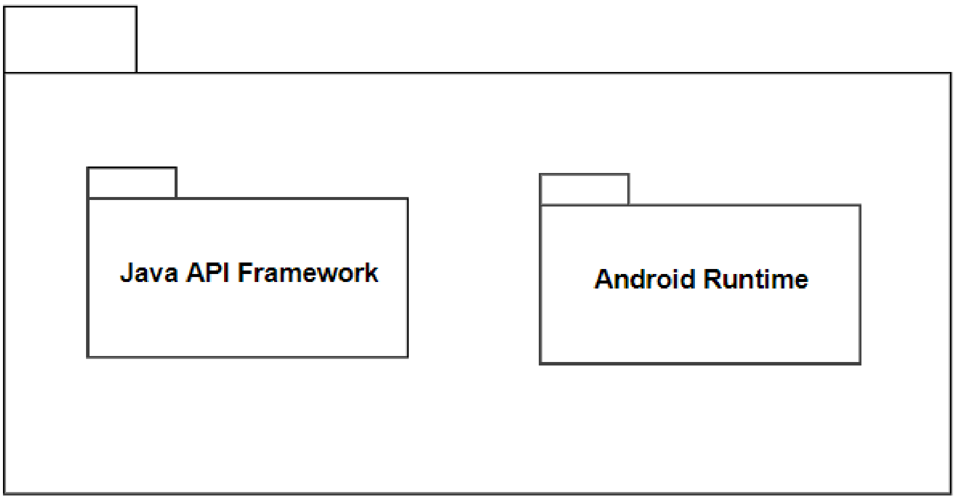


Figure 1. 9 Middleware Layer

Table 1. 10 Middleware Layer

Package Name	Description
Java API Framework	APIs written in the Java language to form the building blocks for creating Android apps by simplifying the reuse of core, modular system components and services.
Android Runtime	Android Runtime (ART) written to run multiple virtual machines on low-memory devices by executing DEX files.

CHAPTER 2

2.1 DETAILED DESCRIPTION

2.1.1 Login

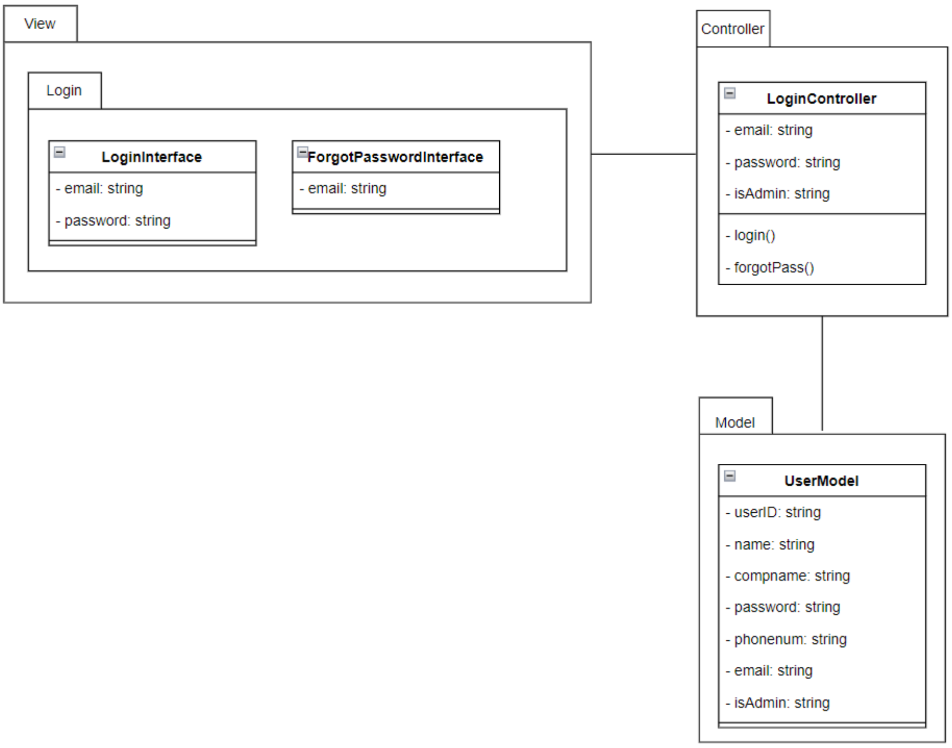


Figure 2. 1 Login Module Package Diagram

2.1.1.1 LoginInterface

Table 2. 1 LoginInterface

Class Type	Boundary class	
Responsibility	An interface to display the login form for the user to log into the application.	
Attributes	Attributes Name	Attributes Type
	email password	String String
Methods	Method Name	Description
	Not Applicable	Not Applicable

Algorithm	Not Applicable	

2.1.1.2 ForgotPasswordInterface

Table 2. 2 ForgotPasswordInterface

Class Type	Boundary class	
Responsibility	An interface to display the forgot password form for users to reset password.	
Attributes	Attributes Name	Attributes Type
	email	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.1.3 LoginController

Table 2. 3 LoginController

Class Type	Boundary class	
Responsibility	A controller to communicate and manage the process logic between the interfaces and models for Login module.	
Attributes	Attributes Name	Attributes Type
	email password isAdmin	String String String
Methods	Method Name	Description

	login( )
	forgotPass( )
		To check user's login credentials.
		To send the password reset link to user's email.

Algorithm	login() BEGIN Get email, password from login form If email and password matches If isAdmin is admin	

	Then redirect admin home page interface Else Redirect user home page interface END forgotPass() BEGIN Get email Send password reset link to email END

2.1.2 Manage User Data

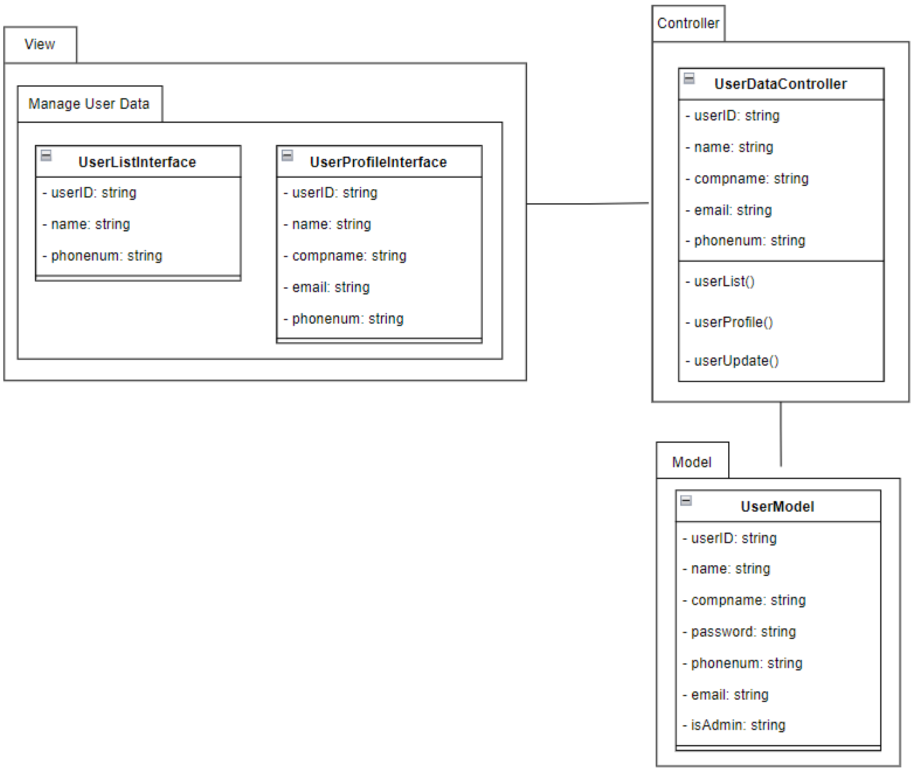


Figure 2. 2 Manage User Data Package Diagram

2.1.2.1 UserListInterface

Table 2. 4 UserListInterface

Class Type	Boundary class

Responsibility	An interface to display the list of users.	
Attributes	Attributes Name	Attributes Type
	userID	String
	name	String
	phonenum	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.2.2 UserProfileInterface

Table 2. 5 UserProfileInterface

Class Type	Boundary class	
Responsibility	An interface to display the selected user’s profile.	
Attributes	Attributes Name	Attributes Type
	userID	String
	name	String
	compname	String
	email	String
	phonenum	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.2.3 UserDataController

Table 2. 6 UserDataController

Class Type	Boundary class	
Responsibility	A controller that communicates and manages the process logic between the interfaces and models for Manage User Data module.	
Attributes	Attributes Name	Attributes Type
	userID	String

	name	String
	compname	String
	email	String
	phonenum	String
Methods	Method Name	Description
	userList()	To retrieve all users and display in a list.
	userProfile()	To display the user’s profile.
	userUpdate()	To update the user’s details.

Algorithm	userList()
	BEGIN
	Retrieve userID, name, phonenum from database
	Display retrieved data on list
	END
	
	userProfile()
	BEGIN
	Retrieve name, compname, email, phonenum from database
	Display retrieved data on profile
	END
	
	userUpdate()
	BEGIN
	Get userID, name, email, phonenum, compname from profile
	Update name, email, phonenum, compname of user’s userID in database
	END	

2.1.3 Access Breach Records

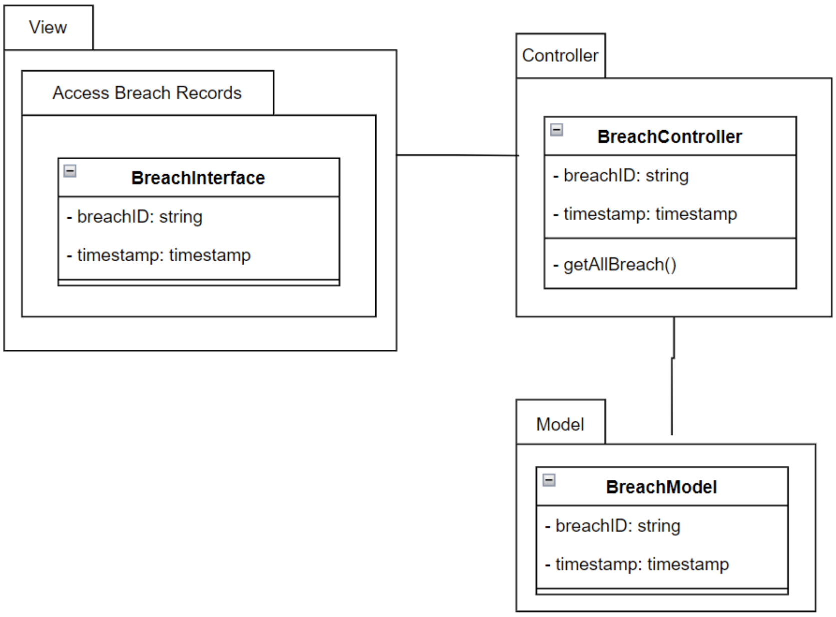


Figure 2. 3 Access Breach Records Package Diagram

2.1.3.1 BreachInterface

Table 2. 7 BreachInterface

Class Type	Boundary class	
Responsibility	An interface to display the list of breach records.	
Attributes	Attributes Name	Attributes Type
	breachID tiemstamp	String Timestamp
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.3.2 BreachController

Table 2. 8 BreachController

Class Type	Boundary class	
Responsibility	A controller that communicates and manages the process logic between the interfaces and models for Access Breach Records module.	
Attributes	Attributes Name	Attributes Type

	breachID tiemstamp	String Timestamp
Methods	Method Name	Description
	getAllBreach()	To retrieve all the breach records and display on a list.

Algorithm	getAllBreach()
	BEGIN
	Retrieve breachID, timestamp from database
	Display retrieved data on list
	END	

2.1.4 Receive Notifications

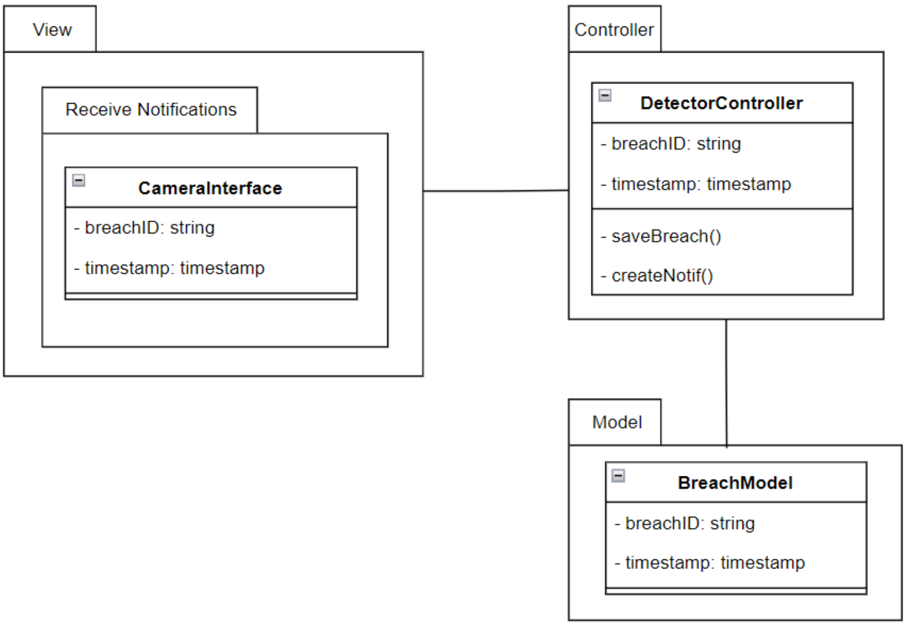


Figure 2. 4 Receive Notifications Package Diagram

2.1.4.1 CameraInterface

Table 2. 9 CameraInterface

Class Type	Boundary class
Responsibility	An interface for the camera to be activated and detect any person not complying to wearing helmet.

Attributes	Attributes Name	Attributes Type
	breachID timestamp	String Timestamp
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.4.2.DetectorController

Table 2. 10 DetectorController

Class Type	Boundary class	
Responsibility	A controller that communicates and manages the process logic between the interfaces and models for Receive Notifications module.	
Attributes	Attributes Name	Attributes Type
	breachID timestamp	String Timestamp
Methods	Method Name	Description

	saveBreach()
	createNotif()
		To save the breach record into the database.
		To send notification to user on the breach.

Algorithm	saveBreach()
	BEGIN
	Detect breach
	Save timestamp into database
	END
	
	sendNotification()
	BEGIN
	Detect breach
	Generate push notification
	END	

2.1.5 Manage PPE Inventory

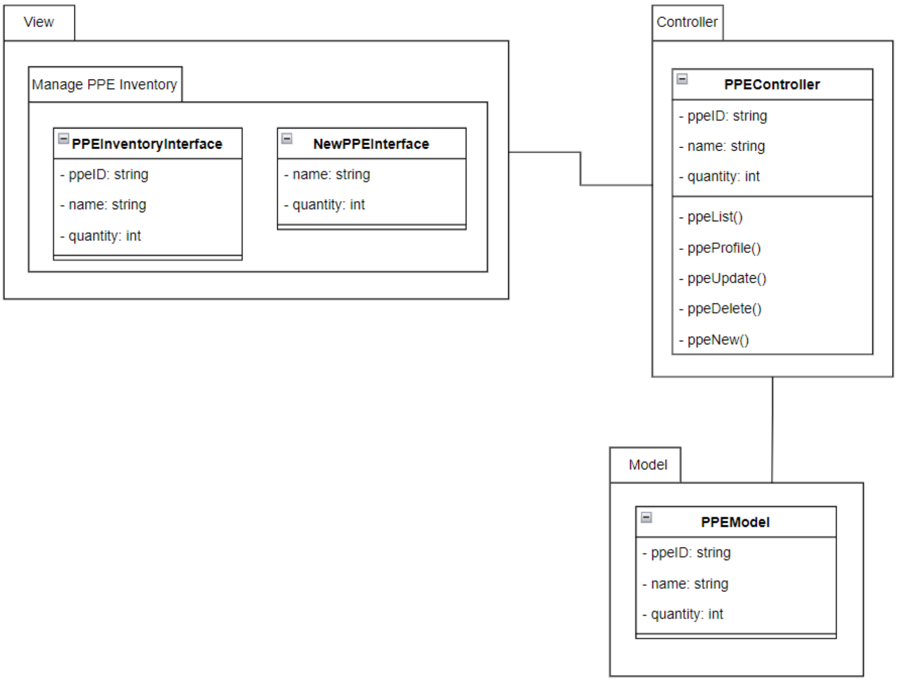


Figure 2. 5 Manage PPE Inventory Package Diagram

2.1.5.1 PPEInventoryInterface

Table 2. 11 PPEInventoryInterface

Class Type	Boundary class	
Responsibility	An interface to display the list of PPE.	
Attributes	Attributes Name	Attributes Type
	ppeID name quantity	String String Int
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.5.2 NewPPEInterface

Table 2. 12 NewPPEInterface

Class Type	Boundary class

Responsibility	An interface to display a form to insert new PPE.	
Attributes	Attributes Name	Attributes Type
	name quantity	String Int
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.5.3 PPEController

Table 2. 13 PPEController

Class Type	Boundary class	
Responsibility	A controller that communicates and manages the process logic between the interfaces and models for Manage PPE Inventory module.	
Attributes	Attributes Name	Attributes Type
	ppeID name quantity	String String Int
Methods	Method Name	Description

	ppeList()
	ppeProfile()
	ppeUpdate()
	ppeDelete()
	ppeNew()
		To display the list of PPE.
		To display the PPE’s profile.
		To update the PPE’s details.
		To delete a PPE.
		To add a new PPE.

Algorithm	ppeList()
	BEGIN
	Retrieve ppeID, name, quantity from database
	Display retrieved data on list
	END
	
	ppeProfile()
	BEGIN	

	Retrieve ppeID, name, quantity from database
	Display retrieved data on profile
	END
	
	ppeUpdate()
	BEGIN
	Get ppeID, name, quantity from profile
	Update name, quantity of PPE’s ppeID in database
	END
	
	ppeDelete()
	BEGIN
	Prompt confirmation window
	If confirm
	  Then remove PPE from database
	END
	
	ppeNew()
	BEGIN
	Get name, quantity from new PPE form
	Insert name, quantity to database
	END

2.1.6 Add Feedback

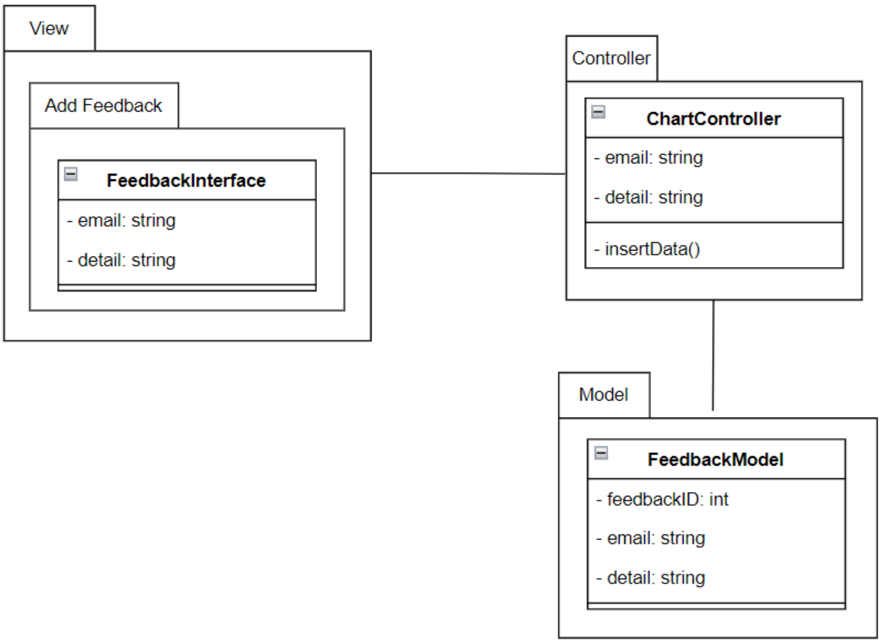


Figure 2. 6 Add Feedback Package Diagram

2.1.6.1 FeedbackInterface

Table 2. 14 FeedbackInterface

Class Type	Boundary class	
Responsibility	An interface to display the feedback form.	
Attributes	Attributes Name	Attributes Type
	email detail	String String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

2.1.6.2 FeedbackController

Table 2. 15 FeedbackController

Class Type	Boundary class

Responsibility	A controller that communicates and manages the process logic between the interfaces and models for Add Feedback module.	
Attributes	Attributes Name	Attributes Type
	feedbackID email detail	String String String
Methods	Method Name	Description
	insertData()	To insert.

Algorithm	BEGIN
	Retrieve email, detail from database
	Save email, detail into database
	END	

2.2 DATA DICTIONARY

Table 2. 16 User

Field Name	Description	Data Type	Constraint
userID	User ID	VARCHAR(255)	PK
name	User’s name	VARCHAR(255)	
compname	Company name	VARCHAR(255)	
email	User’s email	VARCHAR(255)	
phonenum	Phone number	VARCHAR(255)	
password	Password	VARCHAR(255)	
isAdmin	If user is admin	VARCHAR(255)	

Table 2. 17 PPE

Field Name	Description	Data Type	Constraint
ppeID	PPE ID	VARCHAR(255)	PK
name	PPE name	VARCHAR(255)	

quantity	PPE quantity	INT	

Table 2. 18 Breach

Field Name	**Description**	**Data Type**	**Constraint**
breachID	PPE breach ID	VARCHAR(255)	PK
timestamp	Date and time of breach	TIMESTAMP	

Table 2. 19 Feedback

Field Name	**Description**	**Data Type**	**Constraint**
feedbackID	Feedback ID	VARCHAR(255)	PK
detail	Detail of feedback	VARCHAR(255)	
Email	Email of user	VARCHAR(255)	

APPENDIX E
USER ACCEPTANCE TEST (UAT)

2020

USER ACCEPTANCE TEST (UAT)

[CONSTRUCTION SITE PPE DETECTION IN MOBILE APPLICATION]

DOCUMENT APPROVAL

	Name	Date
Authenticated by: ________________________ Name	ANG SUZANNE	
Approved by: ________________________ Client		

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF TABLES

TEST REPORT

User acceptance testing (UAT), also known as application testing or end-user testing, is a stage of the software development process when the application is tested in the real world. UAT is implemented here in ConSite application to confirm that software can handle practical activities and execute in accordance with development requirements.

The test cases are derived based on the functionalities and modules of ConSite application. The inputs, expected outcomes and actual outcomes are outlined here.

Table 1.1 Login Function Testing

Module	Login				
Objective	To test the login function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-01-01	Login without any input	Blank input	Display error message	Display error message	Pass
TC-01-02	Login with unregistered email	Unregistered email	Display error message	Display error message	Pass
TC-01-03	Login with wrong email or password	“sam@gmail.com” “123”	Display error message	Display error message	Pass
TC-01-04	Login with correct email and password	Registered email and password	Redirect to home page	Redirect to home page	Pass
TC-01-05	Reset password without	Blank input	Display error message	Display error message	Pass

	entering email				
TC-01-06	Reset password with invalid email	“123@”	Display error message	Display error message	Pass
TC-01-07	Reset password with unregistered email	Unregistered email	Display error message	Display error message	Pass
TC-01-08	Reset password with correct email	Registered email	Receive reset password link in email	Receive reset password link in email	Pass

Table 1.2 Manage User Data Function Testing

Module	Manage User Data				
Objective	To test the manage user data function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-02-01	Access user list page	Click on “User” icon at menu bar	Display user list	Display user list	Pass
TC-02-02	View a user profile	Click a user profile	Display correct user profile	Display correct user profile	Pass

TC-02-03	Update user info	New phone number	Display successful message	Display successful message	Pass

Table 1.3 Access Breach Records Function Testing

Module	Access Breach Records				
Objective	To test the access breach records function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-03-01	Access breach record page	Click on "Helmet Breach" at menu list	Display list of breach records	Display list of breach records	Pass

Table 1.4 Receive Notifications Function Testing

Module	Receive Notifications				
Objective	To test the receive notification function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-04-01	Receive non-compliance workers notification	Scan camera at person without helmet	Prompt notification alert	Prompt notification alert	Pass
TC-04-02	View breach details	Click on the notification alert	Display breach record list	Display breach record list	Pass

Table 1.5 Manage PPE Inventory Function Testing

Module	Manage PPE Inventory				
Objective	To test the manage PPE inventory function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-05-01	Access PPE inventory list page	Click on "PPE Inventory" at menu list	Display PPE list	Display PPE list	Pass
TC-05-02	Add a new PPE inventory	Click on "+" button	Display add new PPE page	Display add new PPE page	Pass
TC-05-03	Add a new PPE inventory	Blank input	Display error message	Display error message	Pass
TC-05-04	Add a new PPE inventory	Enter all required fileds	Display successful message	Display successful message	Pass
TC-05-05	Edit a PPE information	Click on "Update" button	Display pop-up bottom box	Display pop-up bottom box	Pass
TC-05-06	Edit a PPE information	Blank input	Display error message	Display error message	Pass
TC-05-07	Edit a PPE information	Insert all required fields	Display successful message	Display successful message	Pass
TC-05-08	Delete a PPE inventory	Click on "Delete" button	Display pop-up	Display pop-up	Pass

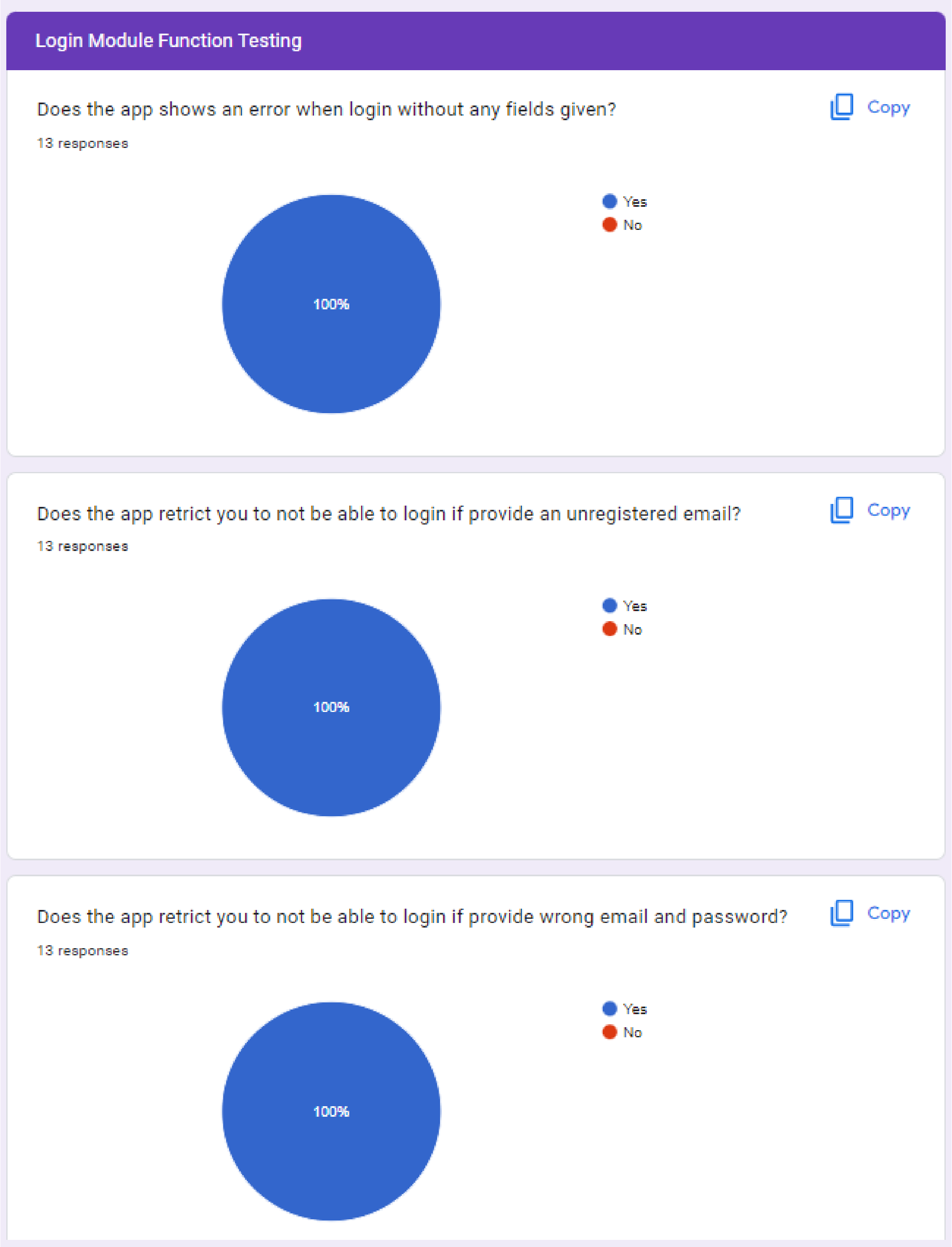
			confirmation message	confirmation message	
TC-05-09	Delete a PPE inventory	Click on “Cancel” selection	Confirmation message dismisses without deleting data	Confirmation message dismisses without deleting data	Pass
TC-05-10	Delete a PPE inventory	Click on “Delete” button	Display successful message and data deleted	Display successful message and data deleted	Pass

Table 1.6 Add Feedback Function Testing

Module	Add Feedback				
Objective	To test the add feedback function of the application.				
Test ID	**Test Cases**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Pass / Fail**
TC-06-01	Access feedback page	Click on “Feedback” at menu list	Display feedback page	Display feedback page	Pass
TC-06-02	Submit feedback	Blank input	Display error message	Display error message	Pass
TC-06-03	Submit feedback	Words exceed 500 words	Disable typing words that are over 500	Disable typing words that are over 500	Pass

TC-06-04	Submit feedback	Insert less than 500 words	Display successful message	Display successful message	Pass

APPENDIX F
USER ACCEPTANCE TEST (UAT) GOOGLE FORM



Login Module Function Testing

Does the app shows an error when login without any fields given?

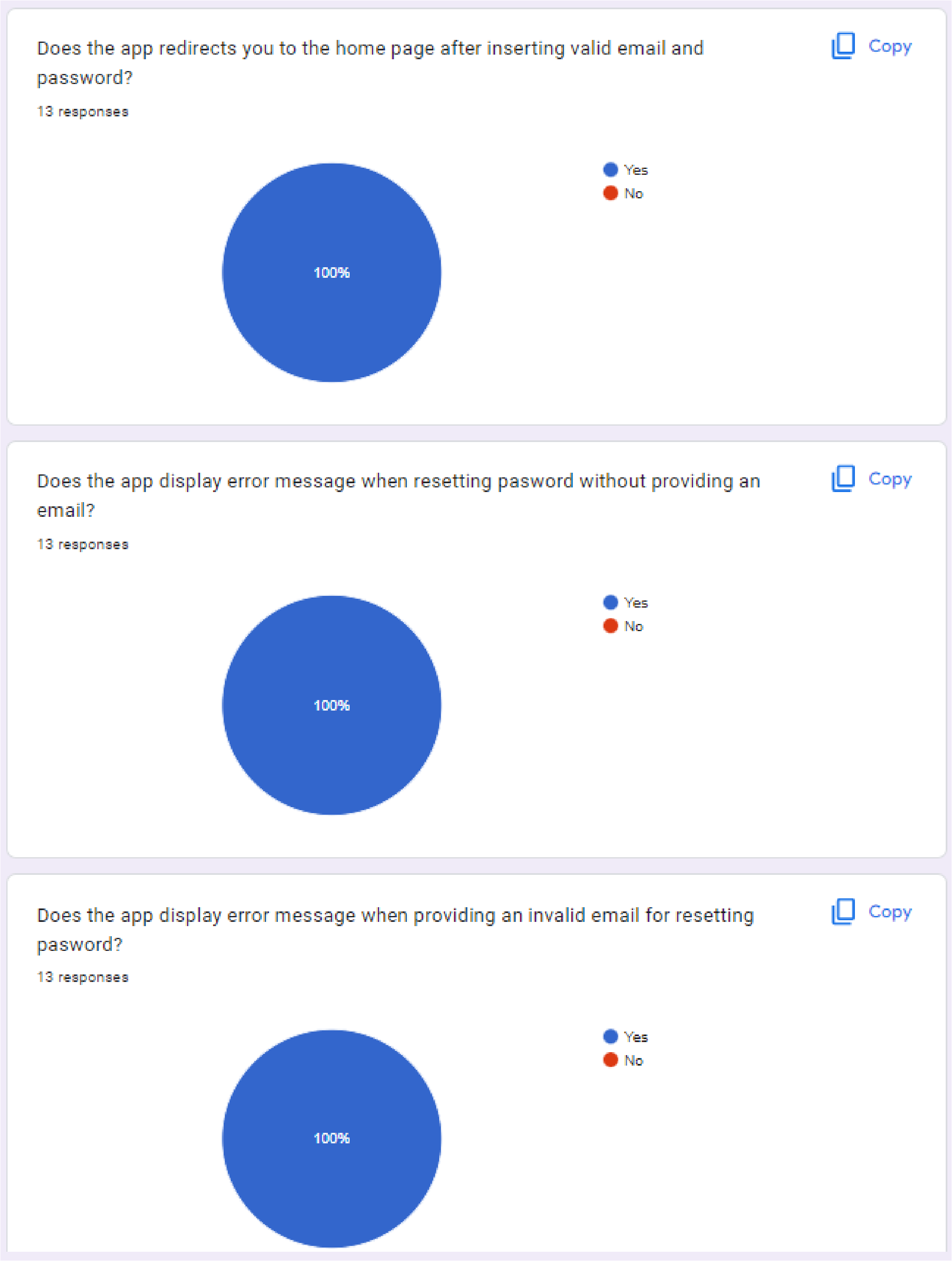
13 responses

Does the app retrict you to not be able to login if provide an unregistered email?

13 responses

Does the app retrict you to not be able to login if provide wrong email and password?

13 responses



Does the app redirects you to the home page after inserting valid email and password?

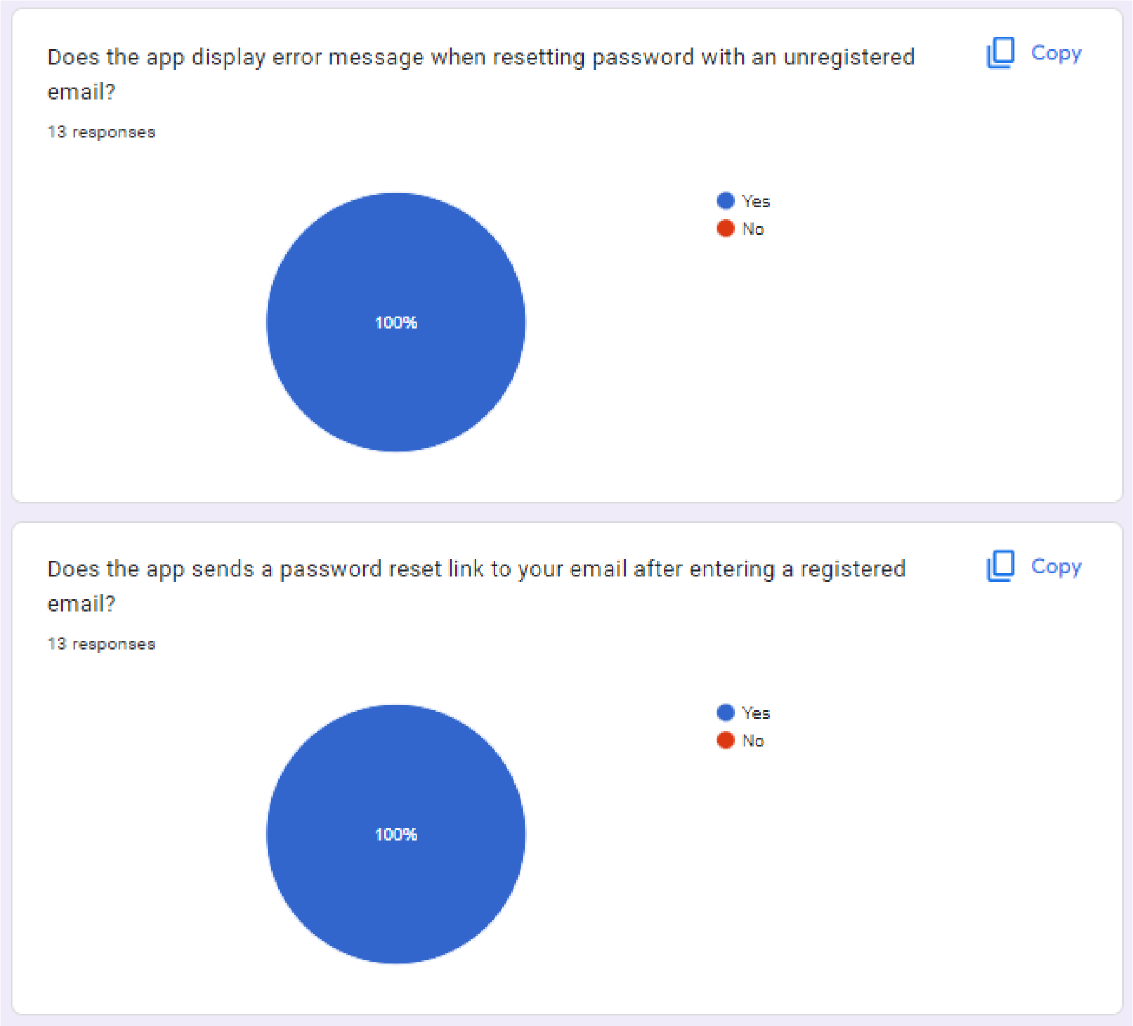
13 responses

Does the app display error message when resetting pasword without providing an email?

13 responses

Does the app display error message when providing an invalid email for resetting pasword?

13 responses

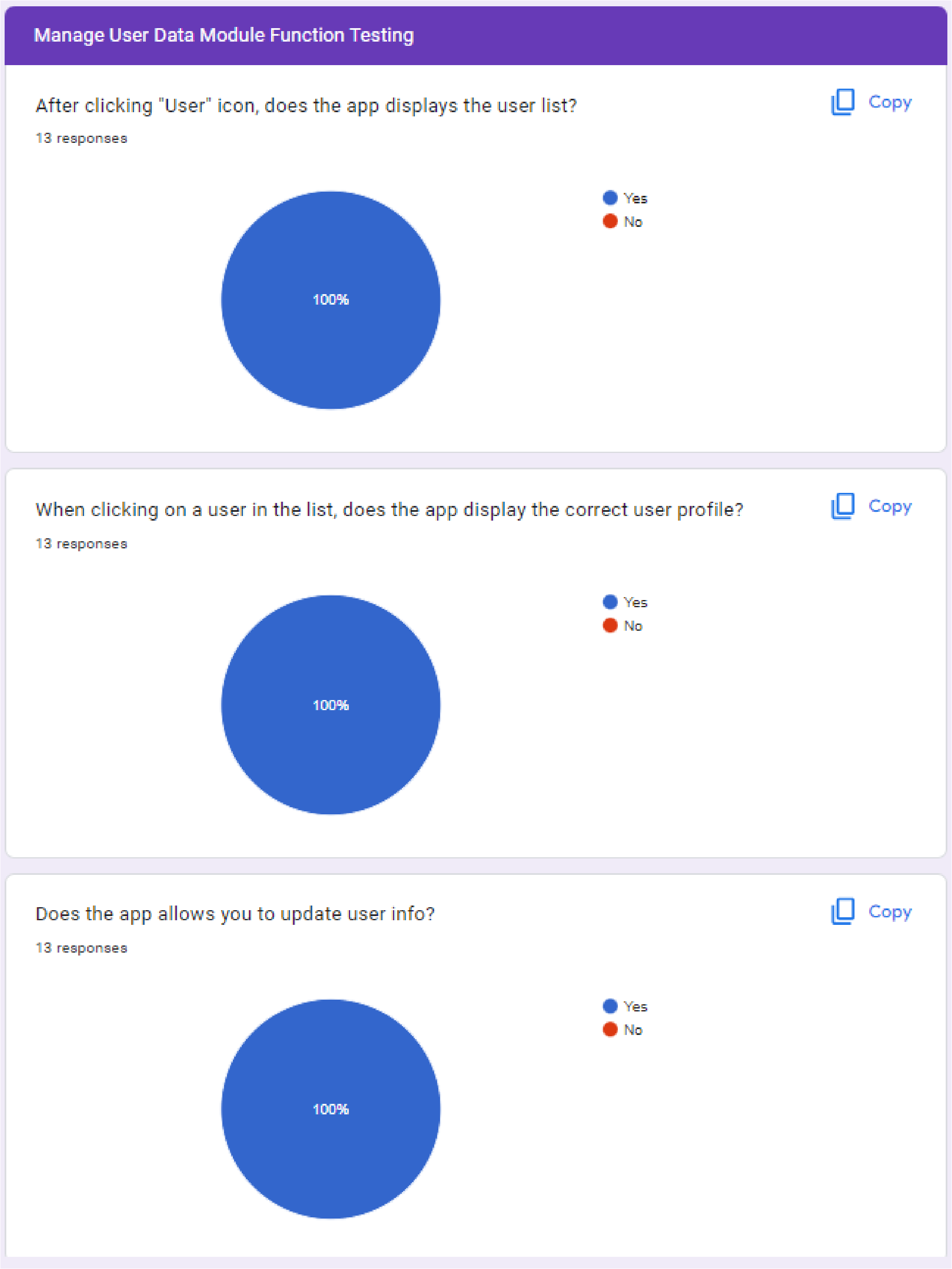


Does the app display error message when resetting password with an unregistered email?

13 responses

Does the app sends a password reset link to your email after entering a registered email?

13 responses



Manage User Data Module Function Testing

After clicking "User" icon, does the app displays the user list?

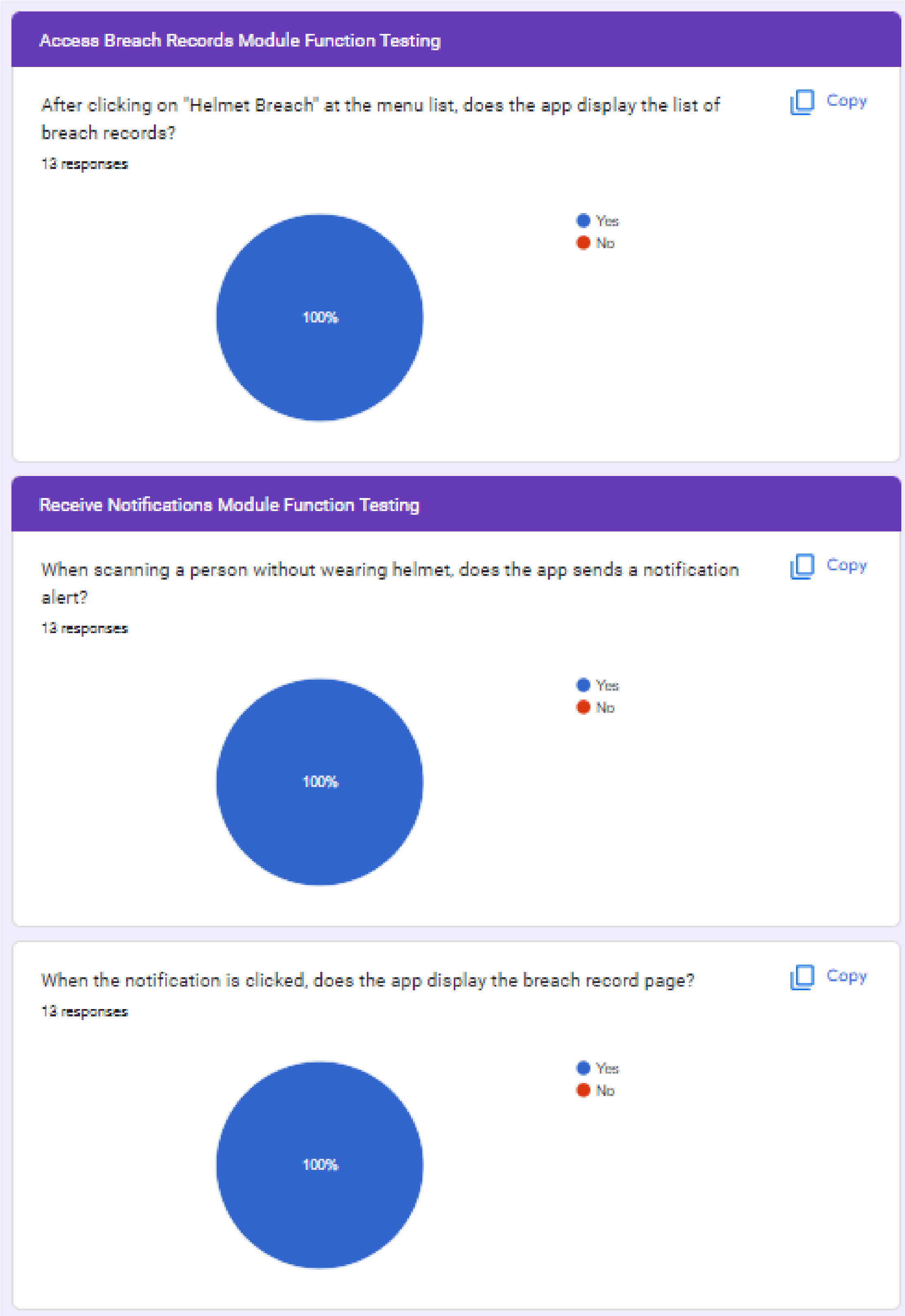
13 responses

When clicking on a user in the list, does the app display the correct user profile?

13 responses

Does the app allows you to update user info?

13 responses



Access Breach Records Module Function Testing

After clicking on "Helmet Breach" at the menu list, does the app display the list of breach records?

13 responses

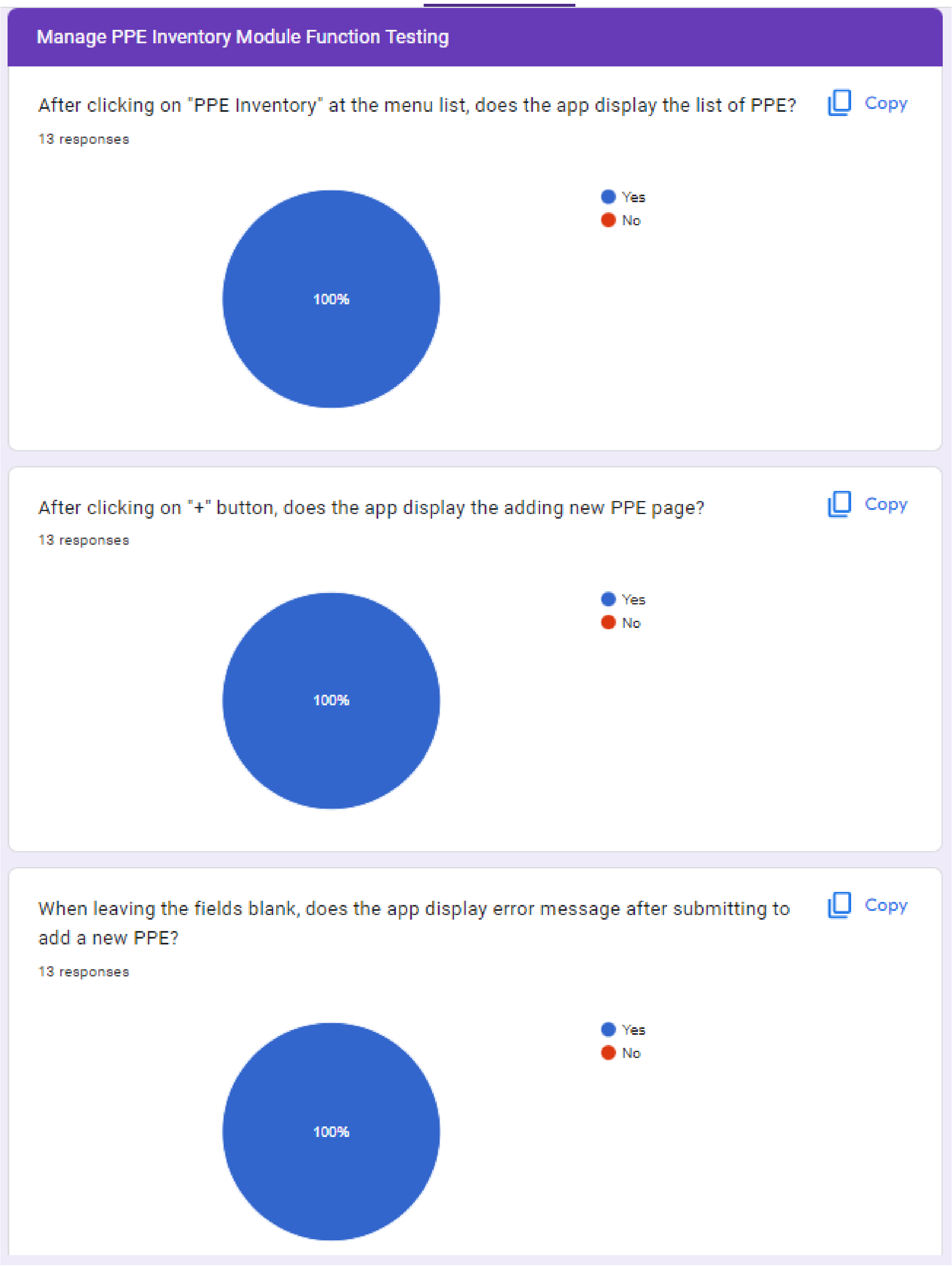
Receive Notifications Module Function Testing

When scanning a person without wearing helmet, does the app sends a notification alert?

13 responses

When the notification is clicked, does the app display the breach record page?

13 responses



Manage PPE Inventory Module Function Testing

After clicking on "PPE Inventory" at the menu list, does the app display the list of PPE?

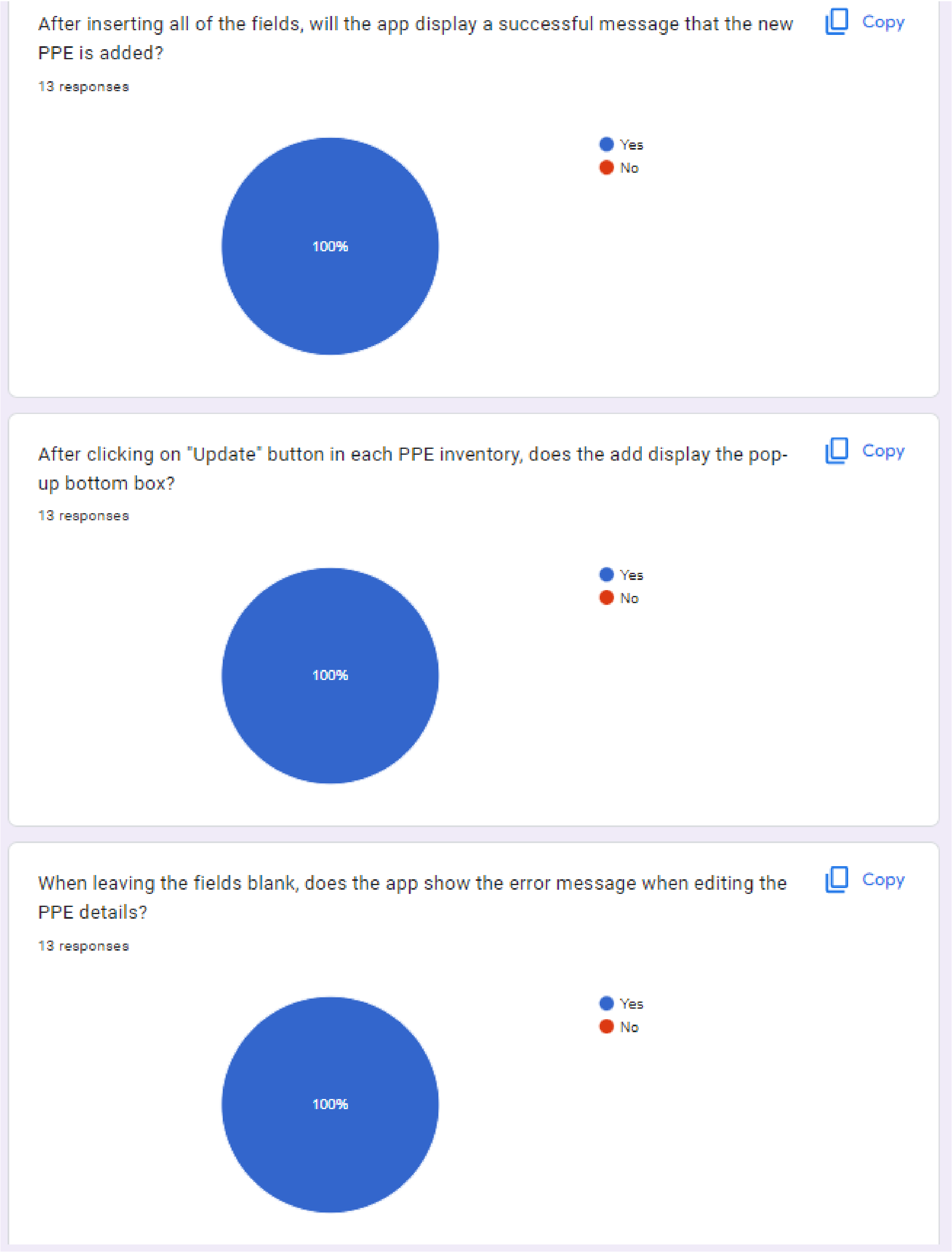
13 responses

After clicking on "+" button, does the app display the adding new PPE page?

13 responses

When leaving the fields blank, does the app display error message after submitting to add a new PPE?

13 responses



After inserting all of the fields, will the app display a successful message that the new PPE is added?

13 responses

● Yes
● No

100%

After clicking on "Update" button in each PPE inventory, does the add display the pop-up bottom box?

13 responses

● Yes
● No

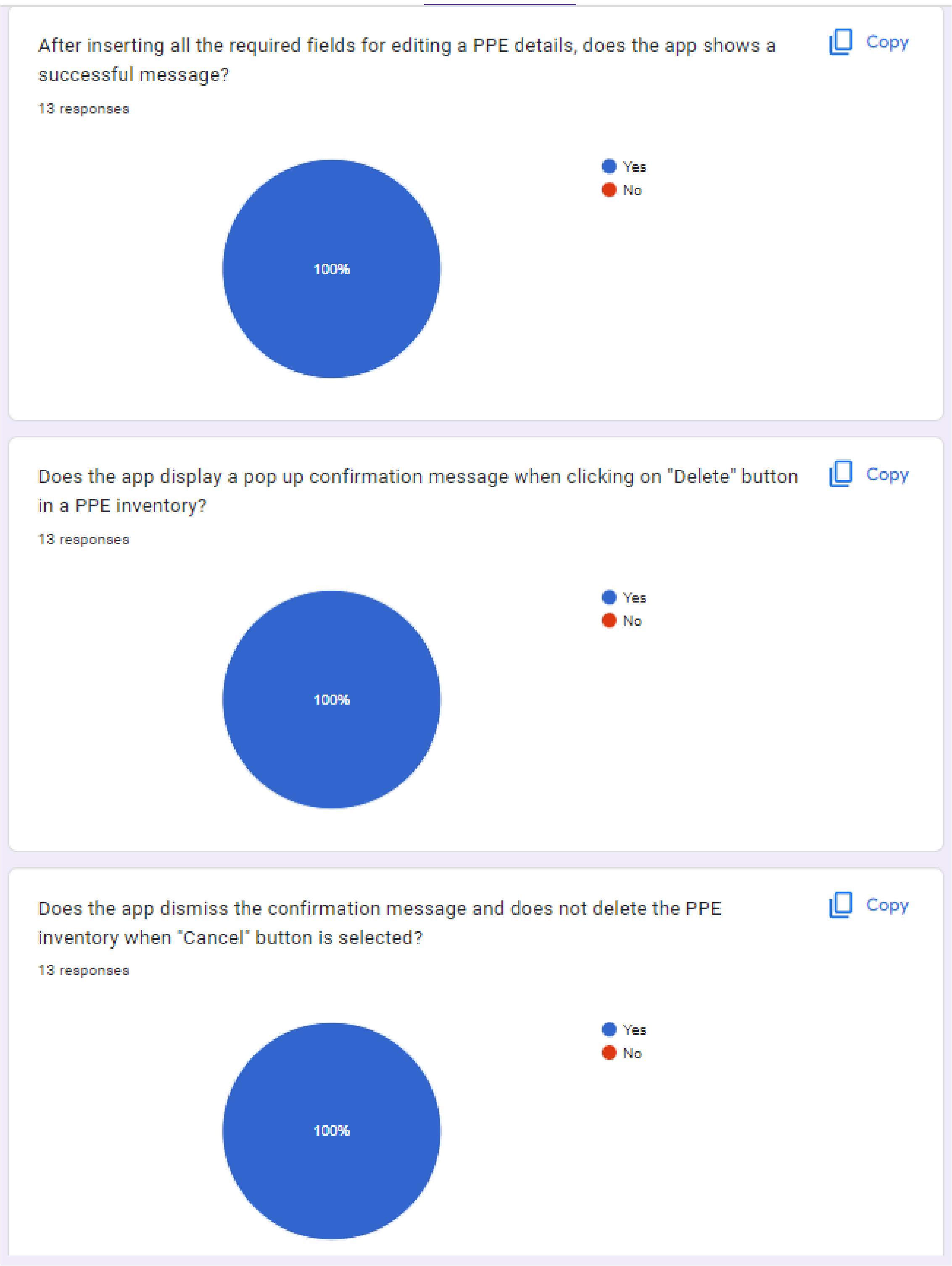
100%

When leaving the fields blank, does the app show the error message when editing the PPE details?

13 responses

● Yes
● No

100%



After inserting all the required fields for editing a PPE details, does the app shows a successful message?

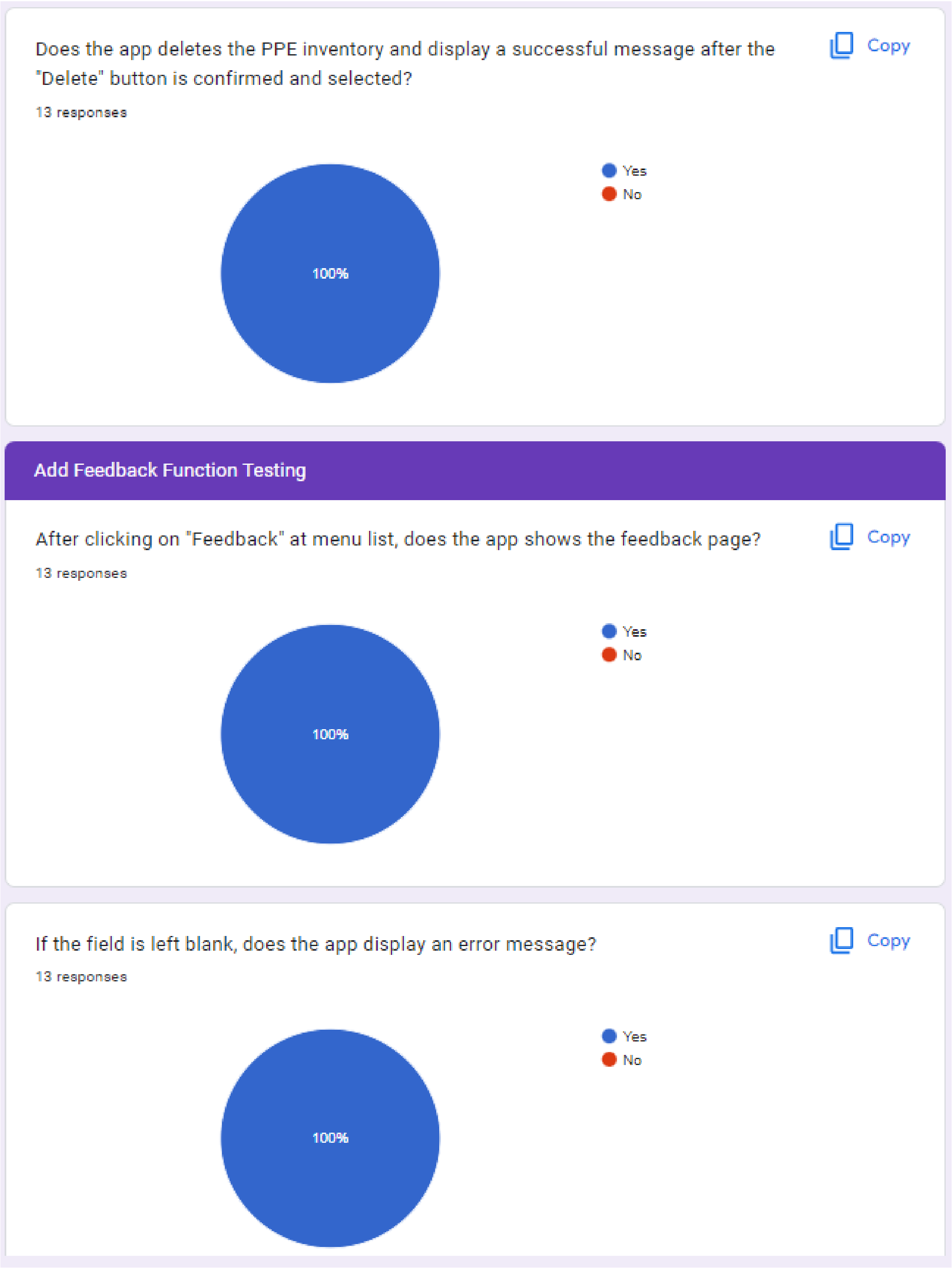
13 responses

Does the app display a pop up confirmation message when clicking on "Delete" button in a PPE inventory?

13 responses

Does the app dismiss the confirmation message and does not delete the PPE inventory when "Cancel" button is selected?

13 responses



Does the app deletes the PPE inventory and display a successful message after the "Delete" button is confirmed and selected?

13 responses

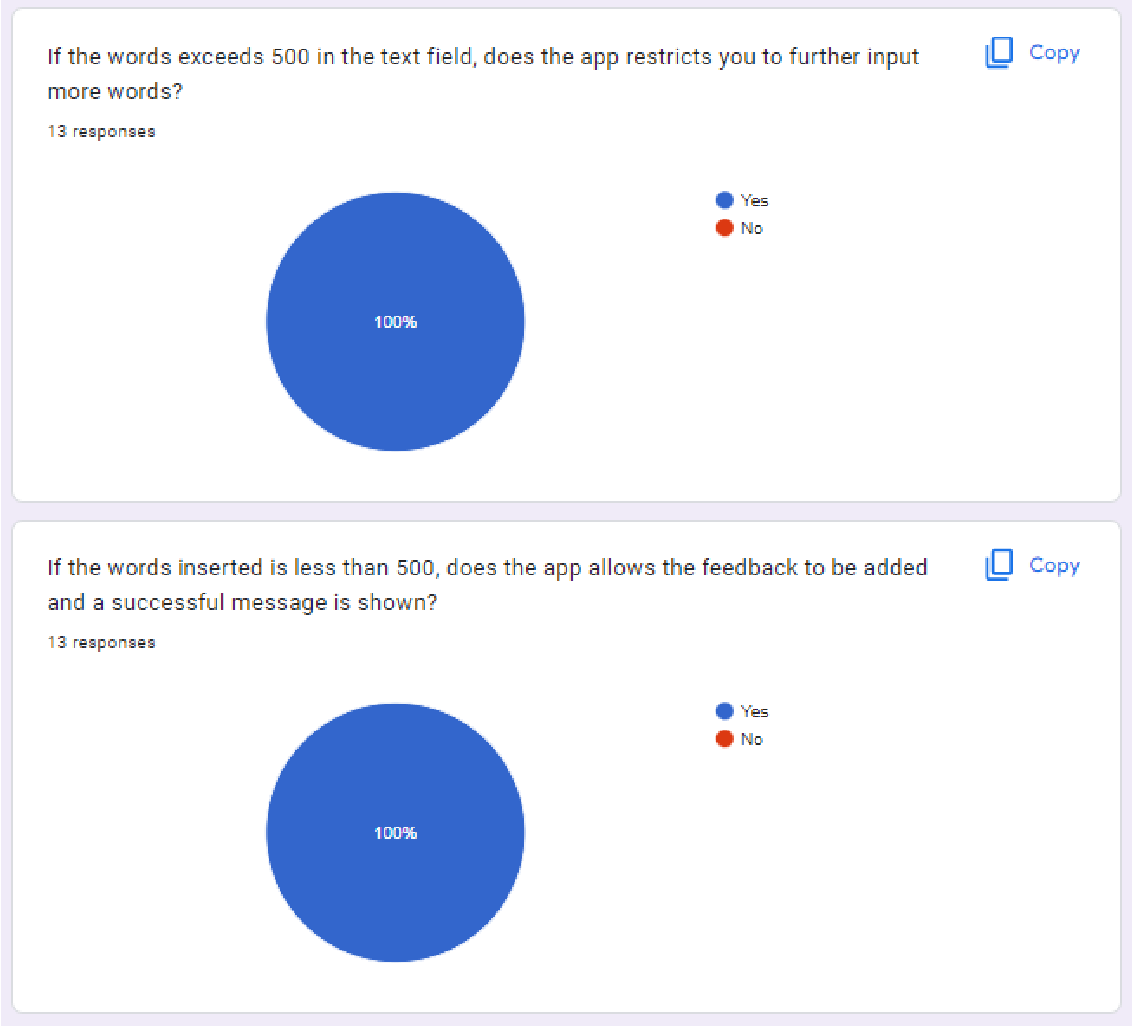
Add Feedback Function Testing

After clicking on "Feedback" at menu list, does the app shows the feedback page?

13 responses

If the field is left blank, does the app display an error message?

13 responses



If the words exceeds 500 in the text field, does the app restricts you to further input more words?

13 responses

If the words inserted is less than 500, does the app allows the feedback to be added and a successful message is shown?

13 responses