STEELGRADE APP: MOBILE APP FOR STEEL PRODUCT QUALITY ASSESSMENT

YEOH WEI BIN

UNIVERSITI TUNKU ABDUL RAHMAN

STEELGRADE APP: MOBILE APP FOR STEEL PRODUCT QUALITY ASSESSMENT

YEOH WEI BIN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : YEOH WEI BIN

ID No. : 2002200

Date : 20 April 2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"STEELGRADE APP: MOBILE APP FOR STEEL PRODUCT QUALITY ASSESSMENT"** was prepared by **YEOH WEI BIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : MOHAMMAD BABRDEL BOANB

Date : 16/05/2024

Signature :

Co-Supervisor :

Date :

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Mohammad Babdrel Bonab for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement during the completion of this project.

ABSTRACT

The leading steel manufacturer companies in Malaysia named CSC Steel Holding Berhad faces challenges in their existing steel product Quality Assessment System in a way that the steel product evaluation process is carried out manually by the domain experts. This manual evaluation process somehow is tedious, time consuming, and experts dependent. Thus, this project aims to revolutionise the manual evaluation process by leveraging the AI-based solution. Development of a mobile-based Quality Assessment System with CNN algorithm is proposed and implemented to tackle the issue. Steel grading processing via the help of trained CNN model to automate the experts evaluation process, and easy-to-interpret dashboard for the data analysis are the two core functionalities in this proposed mobile application. Rapid Application Development (RAD) phased development methodology is adopted to break down the whole complex application into smaller parts for development. Lastly, testing phases are also conducted to ensure the application is error-free, high user satisfaction and great usability. In conclusion, this project has achieved success, fulfilled all the project objectives, and the application is developed successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

QAS	Quality Assessment System
CSCS	CSC Steel Holdings Berhad
CNN	Convolutional Neural Network
1D	1 Dimension
API	Application Programming Interface
UI	User Interface
RAD	Rapid Application Development
CRUD	Create, Read, Update, Delete
AI	Artificial Intelligence
IoT	Internet of Things
IDE	Integrated Development Environment
ML	Machine Learning
DL	Deep Learning
ASTM	American Society for Testing and Materials
SAE	Society of Automotive Engineers
ANN	Artificial Neural Network
SDLC	Software Development Life Cycle
WBS	Work Breakdown Structure
ERD	Entity Relationship Diagram
UAT	User Acceptance Testing
FR	Functional Requirement
NFR	Non-Functional Requirement
UML	Unified Modeling Language
DFD	Data-Flow Diagram
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
SUS	System Usability Scale

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

In this new era, all industries aim to sustain their business and become competitive by attracting as many customers as possible using various kinds of strategies no matter in terms of product quality assurance, customer satisfaction or even price war. Worth, 2020, pp. 233-291 cited in Drozd and Wolniak, 2021 emphasized that "The problems of contemporary industry lie in strict competition". When such competition takes place, it is undeniable that quality of the product becomes the crucial parameter to win this competition. According to Rothaermel, 2021 cited in Drozd and Wolniak, 2021, most of the customers are willing to buy the high enough quality product with an appropriate price. At this point, a quality assessment system comes into the picture.

Quality Assessment System (QAS) is a framework or algorithm that exploits the knowledge of data collection and analysis to evaluate the degree of conformity to predetermined standards and criteria. In other words, it is a systematic method and standard to assess a product based on various criteria. This kind of system mostly involves industry fields with the final goal that ensures the product or deliverable is satisfying. To some extent, the QAS can be defined as a workflow or procedure that the products must go through before they can be released as the final deliverables. All in all, the QAS is responsible for checking and assuring the product quality which can meet customers' requirements and expectations. It also contributes to the industry in the aspects of have an overview about all product quality, maintaining the product quality and product quality improvement.

This project is aimed to analyse the issues regarding the QAS faced by the CSC Steel Holdings Berhad (CSCS) and propose optimal solutions to solve the issues. This chapter here will be discussing the project introduction which are the problem background, problem statements, project objectives, proposed solution and the project scope.

1.2 Problem Background

CSC Steel Holding Berhad (CSCS) is one of the leading steel manufacturing companies located at Ayer Keroh, Melaka, Malaysia. It mainly focuses on the manufacturing and trading of steel products worldwide.

There are various key activities that are carried out by the CSCS which are beneficial to their stakeholders. As a steel company, steel manufacturing and steel processing cannot be run away from the daily routine. CSCS produces different kinds of steel products including Hot Rolled & Pickled Oiled Steel (PO), Cold Rolled Steel (CR), Hot Dipped Galvanized Steel (GI), Pre-painted Galvanized Steel (PPGI) and Medium & High Carbon Steel. These steel products play an important role in a wide range of industries such as construction, furniture, automotive and so on (CSC Steel Holdings Berhad, 2022). CSCS is also engaged in the steel processing which provides the service to customize the steel products depending on the customer requirements.

Since CSCS service is international, the company also emphasizes the quality of the steel product to assure that the quality always meets the industry standards and specifications. Gallimore, 2023 claimed that quality evaluation can bring the organization and business myriad of benefits and businesses should intelligently invest in quality evaluation or assessment to accomplish their visions and become more competitive in today's fast-paced world. Therefore, a QAS is vital to a business or industry to secure the position in the market.

However, the QAS practised by the CSCS is not optimal in some way. Figure 1.1 shows the workflow of the QAS within CSCS.

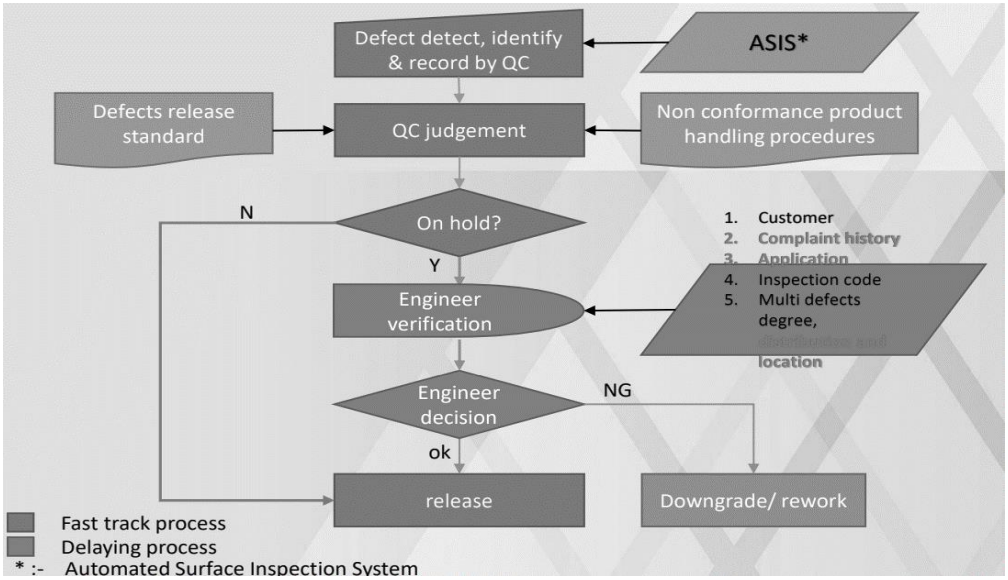


Figure 1.1: Current On Line QC Judging Workflow (Lai, 2023).

This QAS implemented by CSCS can be separated into a 2-layered process. Once the inspection system detects and reports the steel products in the sense that having surface defects, the defective products will undergo the first round of the quality evaluation which involves a quality control judgement using the defect release standard and non-conformance product handling procedures to determine whether the products can be released or need put on-hold. If the defective products are in the on-hold state, a second round of evaluation will be carried out by the product engineer to make a decision for each of the products one-by-one, either moving them into the release state or downgrading them to rework state. This kind of QAS or workflow is somehow not the optimal practical way the CSCS wants as it might bring some drawbacks when time comes.

1.3 Problem Statement

The current problem that arises in the CSCS is mainly about their QAS. Thus, the problem statements listed below are focusing on the issues in their current QAS. These include the aspects of standard, efficiency as well as the information acquisition.

1.3.1 Lack of Standardized Quality Evaluation

Standardization is a very vital foundation in Industry 4.0 as it changes the nature of the industrial process from the traditional manual or semi-manual form towards an automation model. By embracing standardization, the industry can improve performance and ensure product quality (Velasquez Villagran et al., 2019; Casallo et al., 2021). In this context, standardization of work is the tool that assures the steel product quality in the steel industry. With a standardized quality evaluation system, this can preserve the expertise knowledge of steel quality evaluation even though the absence of experienced employees (Satyendra, 2014). However, the CSCS lacks a consistent and standardized approach for evaluating the quality of steel products. This inconsistency often leads to ambiguity in the steel grading and makes it challenging for the employees to make informed decisions when the experienced employee is not around. Therefore, the absence of this reliable and standardized system for steel quality assessment necessitates the development of an efficient and easy-to-use mobile application integrated with AI algorithms that can provide accurate and standardized grading based on the industry standards and specifications.

1.3.2 Manual and Time-Consuming Quality Grading Process

In today's fast-paced era, all industries pursue the rapidness, effectiveness and efficiencies in their working system. In order to achieve that, more and more industries take the initiative to automate their business operations and workflow as far as possible. This is supported by the Federal Steel Supply (2023) summarizing that robotics and automation will be the biggest trends that rapidly improve the steel industry. This can definitely secure the quality and the productivity. Unfortunately, the current process of grading steel products implemented in CSCS involves manual calculations and highly depends on the experienced employees where the subjective judgement comes in. As a result, this leads to variations in grading results and sometimes consuming valuable time and resources because the employee cannot work 24/7. This manual approach often results in errors and delays, affecting the overall efficiency of the steel supply chain. A mobile application that automates the steel grading process will significantly streamline the operations, boost productivity in the

CSCS and ensure the steel is evaluated in the same way using the same algorithm or logic instead of human subjective judgement.

1.3.3 Limited Access to Quality Grading Information

Currently, the product quality assessment process in CSCS is executed by the particular employees which are the P410 Product Engineer. In short, they will focus on the steel product grading based on the surface defect (Lai, 2023). Nevertheless, the problem occurs when the product engineer is missing in the process, leading to order delivery delays and massive downgraded loss. This is because except for the product engineers, the employees in the CSCS may not have easy access to comprehensive information about the steel quality grading system. This limited access restricts their ability to make well-informed decisions regarding the steel grading, steel manufacture, steel quality improvement and etc. The only solution for them is just waiting for the evaluation from the product engineer. Hence, the development of a user-friendly mobile application that offers comprehensive quality grading information and educational resources will empower the other employees with the knowledge to make confident and correct decisions and improve the overall quality standards within the CSCS.

1.4 Project Objectives

The main objectives to be achieved throughout this project are:

1. Conduct a comprehensive review of existing quality assessment applications and the CNN algorithms to inform the development of the mobile application by 30 July 2023.
2. Implement and validate the CNN algorithm to automate the steel product grading process, ensuring the mobile application’s accuracy and reliability by April 2024.
3. Integrate dashboard functionality into the mobile application, enabling informed business decision-making within the CSCS before April 2024.
4. Evaluate the mobile application’s effectiveness in streamlining steel product quality assessment and its impact on decision-making within the CSCS through user feedback and user acceptance testing by achieving a 95% passing rate before April 2024.

1.5 Proposed Solution

To overcome the problems aforementioned in **Problem Statement**, development of a mobile-based Quality Assessment System with CNN algorithm which can automate the steel product grading process is proposed. The main or target user and the functionality of this application will be discussed in **Scope of Project**.

1.5.1 System Overview

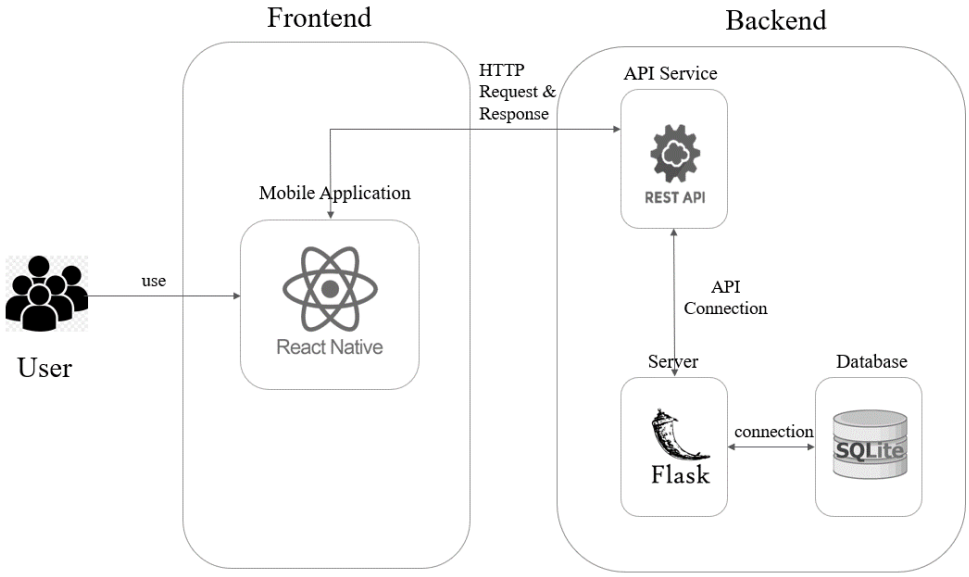


Figure 1.2: System Overview.

Figure 1.2 illustrates the system overview for this proposed project. It shows the high-level concept of how this mobile application will be developed. For the frontend part, the famous framework for mobile application development which is React-Native will be adopted. It will be adopted to develop the user interface which will be rendered on the screen for users. As for the backend development, the server will be adopted is the web framework written in python which is Flask and the database that will be connected to adopt is SQLite. To facilitate the connection between frontend and backend, the API Service – REST API will be adopted. Once the user makes http requests and there are any actions that will trigger some updates on the frontend UI, relevant requests or data will be sent

to the server via the REST API. Upon receiving these requests, the server will process the data and return the necessary responses and updates to the frontend UI. By adopting the RESTful architecture, the user can enjoy a responsive user experience.

1.6 Proposed Approach

The proposed system development methodology for this project is the Phased Development Methodology which is derived from the Rapid Application Development (RAD). Figure 1.3 shows the system development phases and the overview of the Phased Development Methodology.

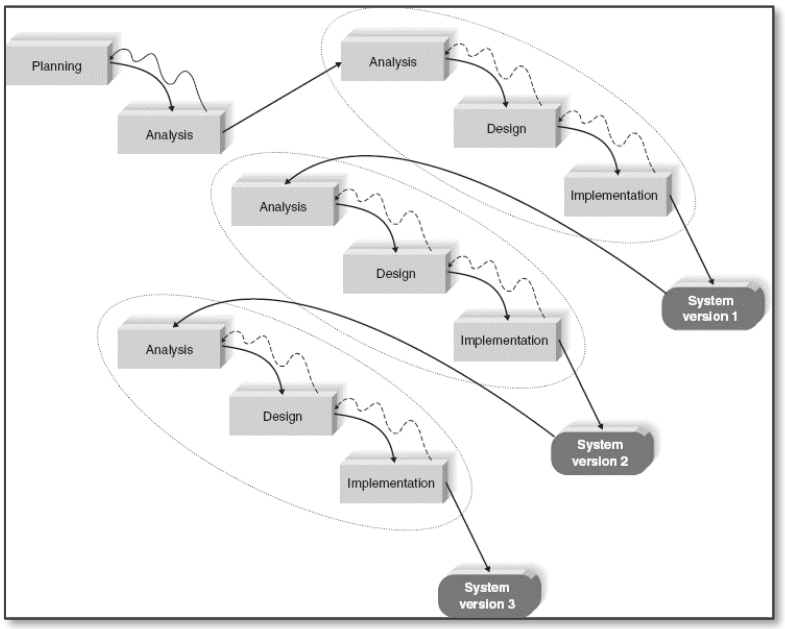


Figure 1.3: Overview of Phased Development Methodology (Dennis, Wixom, and Tegarden, 2015).

RAD is one of the prevalent methodologies used to quickly develop a functional system more effectively and efficiently. Using this RAD, the system developed is closer to the end-users and high-level coverage of user requirements specifications is achieved. For phased development methodology that extends from the RAD, it divides the whole complex system into different versions of subsystems and these subsystems are then developed and implemented sequentially. Once the project is initiated and after the planning phase, the overall system concepts, functionalities, requirements and so on are

analysed at the analysis phase and the requirements are divided and categorized into several versions. For the system version 1, the most important and fundamental requirements are implemented through the phase of analysis, design and implementation. After the system version 1 has been implemented and met the user's satisfaction, another round which is system version 2 of analysis, design and implementation are carried out immediately with the additional analysis performed as well as the new ideas based on the system version 1. This iteration process continues until a functional system is completely developed (Dennis, Wixom, and Tegarden, 2016).

For this project adopted phased development methodology, the overall QAS can be divided into three versions in which the analysis, design and implementation phase have to be done 3 times. The most critical first system version will work on the data which are datasets preparation and the database development. The deliverables of the system version 1 will be brought to system version 2 which is CNN model development. Lastly, to come up with a functional system, the system version 3 will focus on the mobile application development such as user interface, final integration, CRUD operations and etc.

1.7 Scope of Project

This section discusses the development platform, target user for this proposed application, features covered, features not covered and the development tools adopted.

1.7.1 Platform

The project aims to develop a better version of QAS for CSCS on a mobile platform. It focuses on implementing advanced AI algorithms to automate the steel product grading process, giving decisive quality grading information and etc.

1.7.2 Target User

The target user or direct user will be the employees who are working in the CSCS. They can utilize this mobile application easily into their QAS workflow. Instead of depending on the experienced employee – P410 Product Engineer to assess and grade the steel product, whoever working in the CSCS can just use

this application to complete that job. Thus, this proposed project is beneficial for all CSCS employees.

1.7.3 Feature Covered

a) Sign Up

Since the application is used in the working environment, the user must register as a valid employee in the application to have a profile account first before using the features in the application. The registration requires the user's details such as employee ID, full name, email address, username and password.

b) Login

Once the user has an account in the application, the registered user can log in to the application by using the employee ID and password to utilize the application's features.

c) Edit Profile

The user can edit and change the personal information in the profile section if there is any wrong information inputted during the sign up section or the information is wrongly captured by the system.

d) Steel Product Grading Processing

This main feature in the application allows users to assess the grade of the steel product using the CNN algorithms in a few seconds. In this section, the user just needs to enter manually the parameter inputs of the features for the steel product with defected surface in the corresponding defect area, then the output which is the steel product grade will be generated and displayed shortly. In other words, the application will receive the parameter inputs from user inputs. Then, the CNN algorithm at the backend server will predict and display the grade of the steel product based on the parameter inputs received. This main function automates the steel grading process and improves the

QAS. Also, the information about that steel product such as the features (input parameter) and the quality (output class label) will be stored into the backend database and then updated on the current dashboard.

e) Summary

The summary part will give an overview to the logged in user in the sense that how many steel product grading processing has been done by all of the application user with some description such as time, the grade of the examined steel, the name of user operates the grading processing and so on.

f) Dashboard

The user can view and explore the dashboard where the information of steel product quality is displayed. The dashboard showcases the distribution of the different grades of steel products and other important information about the steel product. This dashboard also integrates with different types of chart to improve the data visualization. The user can use this dashboard feature to do data analysis which will be helpful in business decision making. In addition, this dashboard will be updated every time after the steel grade processing functionality is carried out so that the user can always get the latest insights.

g) Information Area

This part in the application provides the comprehensive quality grading information to the user. It gives the information about how this application will be used, what this application offers, some application information and etc. With all this comprehensive information, all the employees within CSCS can be involved in the QAS workflow but not always rely on the particular employee.

1.7.4 Features Not Covered

There are some additional features that are not going to be covered in this project as they are considered less important and not the main focus for this project. First and foremost is the application security part. For example, if the user logins with the incorrect password over three times, the application will still remain opened for the user. Nothing about account locks will happen in this application. There is also no way to find or retrieve back the user's password if the user forgets about that. As this kind of security involves another field of study, this project will just ignore and will not implement it.

Furthermore, this project will also not cover the exportation feature. For instance, if the user wishes to export the dashboard content or the quality grading information into a pdf or external file, there is no such option in this application.

Last but not least, the IoT device development which can act as the alternative approach to detect and capture the parameter inputs for the steel product with defected surface, then send it to the mobile application, is not covered in this proposed project as this development is considered as another domain knowledge and require reasonable development time. Hence, for this proposed project, the user is required to key in those necessary parameters inputs himself / herself to obtain the steel product quality which is the outcome from the backend algorithm.

1.7.5 Development Tools

The development tools adopted for this proposed project can be categorized into 5 parts which are frontend development tool, backend development tool, Integrated Development Environment (IDE), prototyping tool and version control tool.

For frontend development, the tool adopted is React-Native framework whereas for the for the backend development, the tools adopted are Flask framework, SQLite Database and Pusher API Service. As for the IDE adopted, they are Visual Studio Code and Android Studio. Lastly, the prototyping tool adopted in this project is Figma whereas the version control tool adopted is GitHub. Specific description of these development tools and the reason for choosing them will be discussed in the Chapter 3 later.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Developing a steel product QAS or grading system that seamlessly integrates with the CNN algorithm can be considered as a complicated and sophisticated task. This complexity arises from the fusion of the advanced technologies like Artificial Intelligence (AI) or Machine Learning (ML) or Deep Learning (DL). Furthermore, it is quite a novel project in the steel product grading system market as there are limited information and resources on the domain knowledge. Given the relatively limited information available on the similar applications, conducting a comprehensive literature review becomes paramount to better understand the domain areas that are useful to the proposed idea of this project. Hence, this chapter focuses on:

1. Understand the steel grading system
2. Explore today's trend of product grading system.
3. Review the basic concept of CNN algorithm.
4. Investigate how to design a dashboard inside mobile application.

2.2 Steel Grading System

2.2.1 How Steel Is Graded

Steel grading system or process involves a comprehensive analysis of various factors, including the chemical composition, mechanical properties, structural behaviour and surface defect level. These kinds of factors normally allow the steel to be classified and graded based on its performance characteristics. Reliance Foundry (n.d.) claimed that the steel grading process starts with examining its chemical composition, treatment process and mechanical attributes. Besides, the microstructure of the steel can significantly affect the steel's mechanical qualities and at the end upgrade or downgrade a steel product. Reliance Foundry (n.d.) also highlighted that steel is graded and placed in specific categories based on several factors including internal soundness, chemical composition and uniformity, degree of surface imperfections, extent

of testing during manufacturing, the number, size, and distribution of inclusions as well as the hardenability. This kind of steel grading factor is further supported by Beskopylny et al. (2020) clarify that the assessment of structural steel's mechanical properties, especially their strength and deformation characteristics, holds immense significance in classifying its grade. The evolving properties of steel under varying loads, deformations, and temperatures also will affect its mechanical behaviour and contribute to the steel grade.

Hence, there is a fact that steel grading requires a holistic approach that understands the complex relationship between composition, microstructure, mechanical attributes and degree of surface imperfections. By analysing and examining these variables or parameters, the engineers and manufactures can easily classify steel based on its performance capabilities. This classification not only helps in determining the suitable steel grade for specific applications but also ensures the durability and reliability of the structures.

2.2.2 ASTM Steel Grading System

The American Society for Testing and Materials (ASTM) is a standardized system for categorizing different grades of steel based on their distinct properties and characteristics. It gives a consensus for engineers, manufacturers, and various industries to communicate and ensure the quality of the steel products. This ASTM grading system uses a combination of letter prefixes and sequentially assigned numbers to denote and grade different types of steel products. The letter prefix is corresponding to the overall steel category whereas sequentially assigned numbers provide the insights into the composition and characteristics of the material or in other word the metal unique qualities (Anderson, 2021). For example, on ASTM A53, the 'A' indicates a ferrous metal and '53' is the number assigned to galvanized carbon steel. For further grading, the steel or metal with the same ASTM number can have different grades (Precise Pipe, 2022). Figure 2.1 shows the ASTM A106 with Grade A and Grade B as well as their chemical composition. The difference between the grades of the same ASTM number is their chemical composition which can lead to the mechanical or physical properties. In this example, A106 Grade A has lower carbon content so that it is softer steel and easier to bend as compared to A106 Grade B.

What about the composition of A106 Grade A and B pipes?

Grade	A	B
Copper	0.4	0.4
Vanadium	0.08	0.08
Chromium	0.4	0.4
Sulfur	0.035	0.035
Nickel	0.4	0.4
Silicon	0.10	0.10
Molybdenum	0.15	0.15
Phosphorus	0.035	0.035
Manganese	0.27 - 0.93	0.29 - 1.06
Carbon	0.25	0.30

Figure 2.1: Chemical composition of A106 Grade A and Grade B (Precise Pipe, 2022).

2.2.3 SAE Grading System

Society of Automotive Engineers (SAE) Grading System is another standardized system which contributes to the classification of steel by utilizing a four-digit numbering scheme. The first two digits of the number represent the type of steel and its alloying element concentration, while the last two digits provide information about the carbon concentration inside the metal. The Hastings Machine Company (n.d.) also further clarified that the second digit of the 4-digit number indicates the concentration of the major element in percentiles where 1 equals 1% and the last two digits indicate the carbon concentration to 0.01%. For example, SAE 2530 represents a nickel steel alloy, containing 5% of nickel and 0.30 of carbon. Table 2.1 shows the example of the SAE steel grading system.

Table 2.1: The SAE/AISI steel numbering designation system (Hastings Machine Company, n.d.)

Carbon steels	10XX	Plain carbon, Mn 1.00% max
	11XX	Resulfurized free machining
	12XX	Resulfurized/rephosphorized free machining
	15XX	Plain carbon, Mn 1.00-1.65%
Manganese steels	13XX	Mn 1.75%

Nickel steels	23XX	Ni 3.50%
	25XX	Ni 5.00%
Nickel-chromium steels	31XX	Ni 1.25%, Cr 0.65-0.80%
	32XX	Ni 1.75%, Cr 1.07%
	33XX	Ni 3.50%, Cr 1.50-1.57%
	34XX	Ni 3.00%, Cr 0.77%
Molybdenum steels	40XX	Mo 0.20-0.25%
	44XX	Mo 0.40-0.52%
Chromium-molybdenum steels	41XX	Cr 0.50-0.95%, Mo 0.12-0.30%
Nickel-chromium-molybdenum steels	43XX	Ni 1.82%, Cr 0.50-0.80%, Mo 0.25%
	47XX	Ni 1.05%, Cr 0.45%, Mo 0.20-0.35%
Nickel-molybdenum steels	46XX	Ni 0.85-1.82%, Mo 0.20-0.25%
	48XX	Ni 3.50%, Mo 0.25%
Chromium steels	50XX	Cr 0.27-0.65%
	51XX	Cr 0.80-1.05%
	50XXX	Cr 0.50%, C 1.00% min
	51XXX	Cr 1.02%, C 1.00% min
	52XXX	Cr 1.45%, C 1.00% min
Chromium-vanadium steels	61XX	Cr 0.60-0.95%, V 0.10-0.015%
Tungsten-chromium steels	72XX	W 1.75%, Cr 0.75%
Nickel-chromium-molybdenum steels	81XX	Ni 0.30%, Cr 0.40%, Mo 0.12%
	86XX	Ni 0.55%, Cr 0.50%, Mo 0.20%
	87XX	Ni 0.55%, Cr 0.50%, Mo 0.25%
	88XX	Ni 0.55%, Cr 0.50%, Mo 0.35%
Silicon-manganese steels	92XX	Si 1.40-2.00%, Mn 0.65-0.85%, Cr 0-0.65%
Nickel-chromium-molybdenum steels	93XX	Ni 3.25%, Cr 1.20%, Mo 0.12%
	94XX	Ni 0.45%, Cr 0.40%, Mo 0.12%
	97XX	Ni 0.55%, Cr 0.20%, Mo 0.20%
	98XX	Ni 1.00%, Cr 0.80%, Mo 0.25%

2.2.4 Significances and Limitations

The typical steel grading system which involves some sort of certifying the quality of steel via the assessment of their chemical composition, physical properties and so on, holds immense significance in some way. This approach not only ensures the consistency and reliability of steels but also gives a streamlined and efficient method for evaluating the steel. For example, the experts or engineers can easily and swiftly determine the steel grade by visually inspecting the steel and understanding its properties as the reference is already there which is the grading system (ASTM / SAE). (Anderson, 2021) also argued that the use of standardized ASTM grading system facilitates this process by providing a very clear guideline and benchmark for different steel grades. This can somehow enhance the speed of grading and determine the appropriate steel for specific needs. Therefore, the typical or traditional steel grading system such as ASTM, SAE and other certificate-based system is considered as a vital tool in achieving consensus among various industries.

However, the aforementioned “traditional” steel grading system, particularly based on sophisticated chemical compositions, gives some limitations that need to be taken into consideration. One notable limitation is the inherent complexity of this method. Using this grading system, the involvement of experts that are well-versed in metallurgy and material science is very important. This is because they are the ones who have the domain knowledge and are familiar with the technical terms and language. In this case, they can easily grade the steel by using the “traditional” grading system (ASTM/SAE) as reference. Also, the necessities of experts’ involvement in the grading process somehow is a time-intensive and slow process. Other than those experts, the individual who lacks knowledge and experience cannot participate in this process. This also highlighted by Lai (2023) that the steel judging or grading process in his working company cannot be carried out smoothly once the domain knowledge engineer is not involved in the process. At the end, this results in process flow delays, order delivery delays, high working load and boring routine for specific engineer, high quantity of steel products on holds as cannot be evaluated quickly as well as other significant business impacts.

In short, the typical steel grading system offers significant advantages and conveniences in ensuring consistency and efficiency in steel evaluation across industries, but its limitations also cannot be neglected, especially the situation where experts' involvement is restricted comes in.

2.3 Today's Trend of Product Grading System

In the present era, the trend of the product grading system has shifted decidedly towards automation and robotics, representing a transformation from the previous mostly manual process. The manual grading system and process, marked by inaccuracy, inconsistency, subjectively and labour-intensive, have proven to be impractical in most of the cases such as large-scale production, short process time and so on. The contemporary solution which utilizes the advanced technology like AI and machine learning always ensures and guarantees high grading accuracy, swift processing time and cost-effectiveness. These automated AI systems exhibits the potential to revolutionize most of the industries, enhancing productivity and delivering more standardized results (Ali and Thai, 2017; Beskopylny et al., 2020; Varghese et al., 2021; Tu et al., 2022; Yusianto et al., 2022). Table 2.2 summarizes some of the contemporary trends for the proposed product grading system.

Table 2.2: Summarization of Contemporary Trend for Proposed Product Grading System (Ali and Thai, 2017; Beskopylny et al., 2020; Varghese et al., 2021; Tu et al., 2022; Yusianto et al., 2022).

Type of Grading System	**Technology Used**	**Motivation of System Proposal**	**Benefits**	**Classification Features**
Steel	ANN algorithm	Industries need frequent determination of steel grade used in structure	Successful grading with average accuracy (>95%) sufficient for practical,	Strength characteristics (Brinell hardness)

			engineer's expertise	
Fruit and Vegetable	Machine vision and CNN	Traditional grading system is impractical due to large-scale production	Able to grade and predict shelf life for various fruits and vegetables in real-time	Not mentioned
Potato	Digital image processing and fuzzy logic	Manual grading leads to increased time, cost and variability	Grading process can be automated with 86% accuracy	Color, shape and size
Fruit	Automatic visual inspection	Necessity of high-quality fruit to consumers and the impractical manual grading process	Enhance fruit quality, reduce production costs, and replace inconsistence and labor-intensive manual grading process	Color, size, shape, texture, surface defect and decay
Scrap steel	Computer vision, deep CNN	Manual grading process is subjective and inaccurate	Can detect scrap steel precisely and evaluating its grade accurately	Not mentioned

2.4 CNN Concept

Convolutional Neural Network (CNN) is one type of supervised learning algorithm which mimics and is inspired by the structure of the neurons in human's brain. It is designed to automatically identify and recognize the relevant features from the input data like images without the need for human oversight. CNN excels at identifying complex patterns in data by utilizing the convolutional layers, which imitate the process of visual recognition in the brain. This makes the CNN an effective classification tool to correctly classify the class label of an unseen new data based on the model training using given labelled training data. Typically, CNN consists of a number of layers in its architecture.

2.4.1 Convolutional Layer

Convolutional Layer is a core component or layer in the CNN. It is used by CNN to find out the relevant and important patterns automatically and effectively in the data given. It generates the feature maps that highlight these patterns using small filters called kernels that slide over the input data. Each kernel has its weights that it learns during the training process to identify the important and meaningful features (Alzubaidi et al., 2021; Bhatt et al., 2021).

2.4.2 Pooling Layer

In order to facilitate the sub-sampling of feature maps produced by the earlier convolutional operations, the pooling layer is crucial to be included in the CNN. This layer is responsible for downsizing the larger size of feature maps into the smaller ones while retaining and keeping the critical information and features presented. For different applications, there are different types of pooling methods such as max pooling, min pooling, global average pooling and so on. The primary goal of this pooling layer is aimed to reduce the computational demands and aiding in the extraction of significant features that are invariant to position and rotation. In short, this pooling layer contributes to the training efficiency, computational workload reduction, dimensionality reduction and prevention of overfitting in the CNN framework (Alzubaidi et al., 2021; Bhatt et al., 2021).

2.4.3 Activation Layer

The activation layer or activation function is essential for adding the non-linearity to the network's computations. After the Convolutional and Fully Connected Layers with weights, this activation function comes into the picture. Generally, its main purpose is to transform the output of the previous layers, which is typically a weighted sum of inputs and biases, into a format that allows for complex learning and feature extraction. The activation function within the layer determines whether a neuron should 'fire' or activate based on the input. The output produced because of this firing choice is then subsequently passed to the next layers. This nonlinearity activation function enables the CNN to capture the more complicated and sophisticated relationships within the data, learn more complex information from the input data (Alzubaidi et al., 2021; Bhatt et al., 2021).

2.4.4 Fully Connected Layer

The fully connected layer which consists of the complex hidden layer and output layer typically located at the end part of the CNN. In simple words, it is the point that previously extracted and refined features are brought together to generate the final prediction and classification. Each neuron in this layer links to all the neurons in the preceding layer, allowing it to capture the complicated relationship between different features. This layer receives the flattened feature vectors as input which is the vital characteristics detected earlier. Through the weighted summation, these input values are combined, with each neuron having its set of weights that regulate the contribution of each feature. Then, the activation function comes in. The output produced from the activation function serves as the final prediction and classification. In a classification task, each neuron in the output layer represents a specific class, and the neuron with the highest activation value signifies the model's prediction (Alzubaidi et al., 2021; Bhatt et al., 2021). For example, in a multi-class classification problem, the softmax activation function is used to further transform the final output of the fully connected layer into the value between 0 to 1, yielding a probability distribution that can be used to indicate the class label for a new unseen data needed to be predicted.

2.5 How to Design Dashboard in Mobile Application

Dashboard is a very useful visual display that provide a brief overview of complex data sets. It can be used in various domains, from business, health care, to manufacturing industries. There is no fixed definition of a dashboard, but they generally share some common traits and characteristics. The most important is that a good dashboard should not overwhelm users, avoid the visual clutter and be aligned with the existing workflows. Bach et al. (2023) in the research highlighted that it is very vital to strike the balance between maximizing the information and maintaining a user-friendly experience within the area of a small screen when developing an effective mobile application dashboard. He also discussed some considerations including design principles, guidelines, dashboard patterns and tradeoff, in order to create an effective and user-friendly dashboard for mobile applications. For an informative dashboard, the developers or creators should relentlessly prioritize the data elements that directly contribute to user goals. Every element on the screen should serve a clear purpose and avoid overwhelming the users with a lot of useless information. Simplicity is the key to developing an effective dashboard. Also, it is necessary to understand the primary tasks users perform within the application so that the important information for them to make informed decisions or complete their tasks quickly is known and can be fully leveraged. Furthermore, it is necessary to ensure that the responsive design which means that ensure the dashboard always adapts well to different screen size and orientations. Visual hierarchy concept should also be adopted to emphasize the important information and guide users' attention. To ensure the consistency of the dashboard, some design elements such as colors, typography, layout and etc throughout the dashboard must be taken into consideration. Besides, Bach et al. (2023) introduced some common dashboards types like static dashboard, analytic dashboard, magazine dashboard, infographic dashboard, repository dashboard and embedded mini dashboard which might be used for different scenarios. Lastly, some design tradeoffs are needed to consider while developing the dashboard. For example, information abstraction versus details. It is a must to balance the level of detail in visualization to avoid overwhelming users while providing sufficient information. Screenspace utilization which maximizing the use of screenspace without cluttering the interface is also a

tradeoff of designing the dashboard. Moreover, it is a good practice to provide the interactive elements for exploration and navigation while maintaining the simplicity and usability.

2.6 Conclusion

All in all, this chapter covered some literature review on the domain area of this project to get better understanding on those fields and integrate the knowledge learnt into this proposed project. In this chapter, the method of grading a steel was reviewed, the typical or traditional steel grading systems had been explored as well as their significance and limitations in the industry. In addition, some product grading systems across various industries were studied to thoroughly find out the ongoing trend in the grading system field nowadays. This chapter also reviewed the basic concept of the CNN framework or algorithm so that this concept can be adopted in the proposed system. Lastly, the investigation on the development of an effective dashboard had been covered to assure the appropriate dashboard integration in the mobile application.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter will be discussing the software development methodology adopted for this proposed project with detailed phases. Apart from that, the project plan which includes the Work Breakdown Structure (WBS) and Gantt Chart for the project scheduling will be created and developed. Last but not least, the development tools selected for this project will also be discussed and analysed.

3.2 RAD Phased Development Methodology

The RAD Phased Development Methodology is selected to adopt as the software development methodology for this proposed project. The philosophy of this phased development methodology is that it divides and breaks down the whole sophisticated system or software into several small manageable versions, which each handles different aspects of the development. The most important and critical backbone with highest priority of this proposed system will be developed first and proceed with the second important part in the subsequent version and so forth. The final products or deliverables of the first development will be brought to the next development. In other words, the successor version will only start to be developed if all the phases and tasks in the predecessor version have been completed successfully. For this proposed project and system, this methodology consists of five phases in total which are planning, analysis, design, implementation and testing. Different from the traditional architecture of the phased development methodology, the structure of the phased development methodology adopted in this proposed project will be modified as illustrated in the Figure 3.1 to cater the nature of this proposed project.

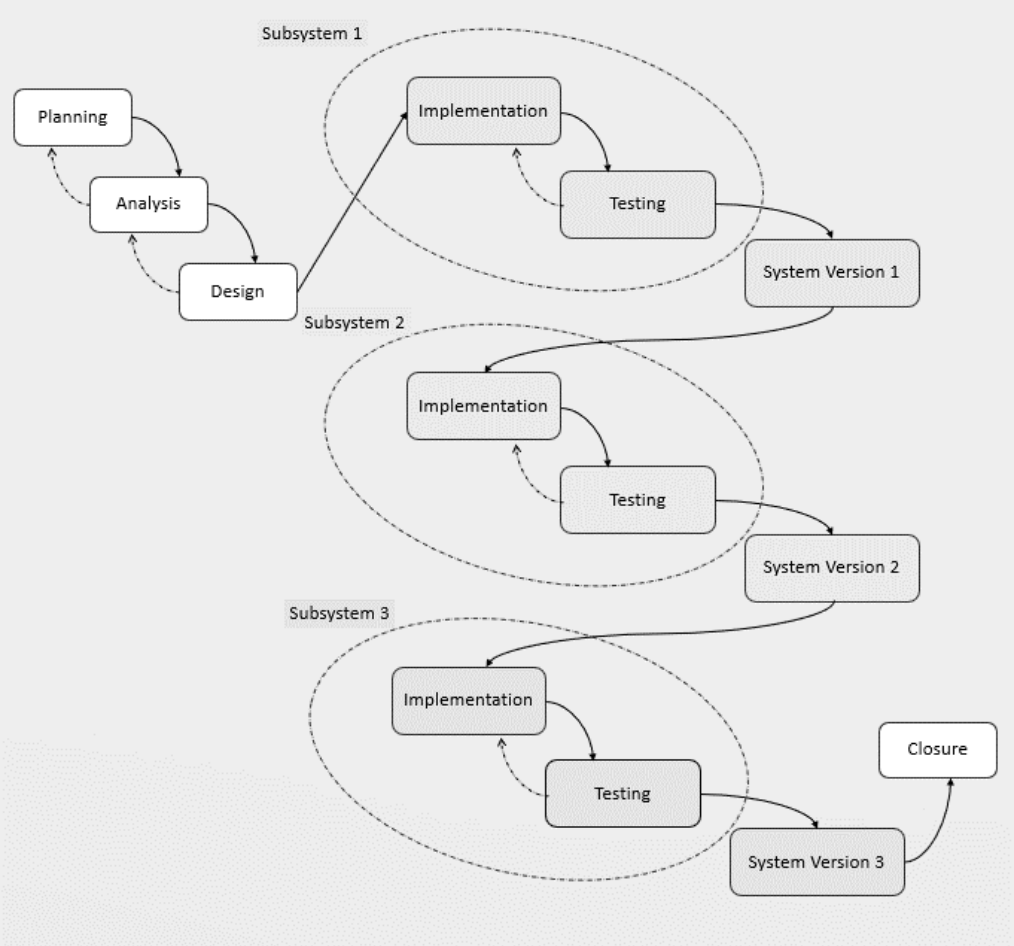


Figure 3.1: Phased Development Methodology For Proposed Project.

3.2.1 Planning and Analysis

Using the phased development methodology, the proposed project starts with the planning and analysis phases to look deeply into and gain better understanding about the project. The planning phase involves the comprehensive studies for the project which are identification of the problem, project background, project objective, possible solution and scope. For the analysis phase, activities such as requirement gathering and elicitation, literature review of related domain knowledge, analysis of appropriate software development methodologies and selection of suitable development tools will be carried out. After this planning and analysis phase, the key deliverable which is the project plan includes WBS and Gantt Chart will be developed and created to effectively oversee the progress of the project and assure its completion within the scope and deadlines.

3.2.2 Design

After the planning and analysis have been completed, some diagrams and architectures will be created in this design phase to illustrate how this proposed system will be built. First and foremost, the higher-level of system architecture will be developed to show the structure of this proposed system in terms of both frontend and backend. Since this project integrates the CNN algorithm into the backend development, the CNN architecture model also will be developed to obtain a basic idea on how the CNN works and how to build this algorithm technically. Furthermore, to model the system behaviour and indicate the interaction between the system and external agents or direct users, a use case diagram needs to be created as part of the design in this project. Along with the use case diagram, a set of use case descriptions also will be created to outline the workflow for each user requirement with details. An Entity Relationship Diagram (ERD) is also necessary in this phase to model the relationship between different entities of the system so that the implementation of the database for the proposed system can be done quickly. Last but not least, some simple screen prototypes are developed to illustrate the user interface design and provide a basic understanding of the system. All these designs are aimed to ease the implementation phase in the later stage.

3.2.3 Implementation and Testing

After all the necessary design, the implementation and testing phase comes into picture. In the implementation phase, the proposed system will start to be built technically where the execution and coding part take place. All the implementation is supported and facilitated by the diagrams and architectures that have developed in the design phase. Upon implementation, unit testing will participate to make sure all the features and functions developed can work and behave as expected. For this proposed project, the implementation and testing phase will be carried out three times as the whole complex system is decomposed into three subsystems with each subsystem responsible for a particular part of the system. Upon development and implementation of the three subsystems, several testings such as unit testing, integration testing, UAT and so on will be done to secure the functionality of the final system.

3.2.4 Subsystem 1

In subsystem 1, the dataset preparation will be carried out. After receiving the Excel file consists of the steel dataset prepared by the CSCS, some data preprocessing such as data cleaning, data transformation, data normalization and so on will be done to ensure the dataset is ready and perfect enough that can be used to train the CNN model. The database which is used to store the training dataset (used to train CNN model) and also the data within the frontend mobile application (user profile information and etc) will also be set up in this subsystem in which all related tables will be created with some dummy data insertion for the convenient of the testing in the later development. After the development and implementation of the dataset preparation and database setup, some testing will be taken place to ensure each configuration and setup is correct without any issues.

3.2.5 Subsystem 2

In this subsystem, the CNN model will be trained and developed using the deliverables from subsystem 1 which are the prepared dataset and database that has been set up. After the development of the CNN model, testings are carried out to check if the model can work accurately and as expected using the training dataset.

3.2.6 Subsystem 3

Upon the development of the CNN model, the subsystem 3 starts with the frontend part which is the mobile application development such as the user interfaces, information exhibition, CRUD operations and so on. Some testings like unit testing, integration testing and monkey testing will also be carried out in this final subsystem to make sure all the features can perform and behave correctly before moving into the closure phase of the project.

3.3 Project Plan

The entire project plan for this proposed project will be scheduled with detailed tasks and resources so that this proposed project can be completed within the

scope and timeframe. In this case, the WBS and Gantt Chart will be created for the project scheduling.

3.3.1 WBS List

1.0 Planning

- 1.1 Analyse problems
  - 1.1.1 Identify problem background
  - 1.1.2 Identify existing problems
- 1.2 Define objectives using SMART metric
- 1.3 Propose solution
  - 1.3.1 Propose system overview
  - 1.3.2 Propose 1D CNN model architecture
- 1.4 Propose approach
- 1.5 Define scope
  - 1.5.1 Identify development platform
  - 1.5.2 Identify target user
  - 1.5.3 Identify features covered
  - 1.5.4 Identify features not covered
  - 1.5.5 Identify development tools

2.0 Analysis

- 2.1 Literature review
  - 2.1.1 Review typical steel grading systems
    - 2.1.1.1 Study on how steel is graded
    - 2.1.1.2 Explore the existing steel grading systems
    - 2.1.1.3 Conclude significances and limitations
  - 2.1.2 Summarize different product grading systems
  - 2.1.3 Grasp CNN basic concept
  - 2.1.4 Investigate the development of effective dashboard
- 2.2 Adoption of software development methodology
  - 2.2.1 Choose adopted methodology
  - 2.2.2 Separate workflow based on methodology chosen
- 2.3 Selection of software development tools
  - 2.3.1 Choose adopted development tools

2.3.2 Justify the chosen development tools

2.4 Requirements Elicitation

2.4.1 Identify requirements discovery and gathering method

2.4.2 Develop system requirements

2.4.2.1 Develop functional requirements

2.4.2.2 Develop non-functional requirements

2.5 Create project plan

2.5.1 Create WBS

2.5.2 Create Gantt Chart

3.0 Design

3.1 Design system architecture

3.2 Design 1D CNN model architecture

3.3 Design use case diagram

3.4 Design use case description

3.5 Design ERD

3.6 Design system prototypes

4.0 Implementation and Testing 1.0

4.1 Dataset preprocessing

4.1.1 Data cleaning

4.1.2 Data reduction

4.1.3 Data transformation

4.1.4 Data normalization

4.1.5 Data sampling

4.2 Database development

4.2.1 Create tables

4.2.2 Insert necessary data

4.3 Unit Testing

5.0 Implementation and Testing 2.0

5.1 CNN model development

5.1.1 Split dataset into training and testing data

5.1.2 Fit training data into model

5.1.3 Evaluate model accuracy

5.1.3.1 Prediction on testing data using trained model

3.3.2 WBS Diagram

WBS diagram can be referred in the **Appendix A.**

3.3.3 Gantt Chart

Gantt Chart of this proposed project is attached in the **Appendix B**.

3.4 Development Tools

This section will be discussing the development tools that were adopted to develop the proposed mobile-based system. They can be categorized into 5 areas which are frontend part, backend part, prototype tool, version control system and IDE as illustrated in the Figure 3.2.

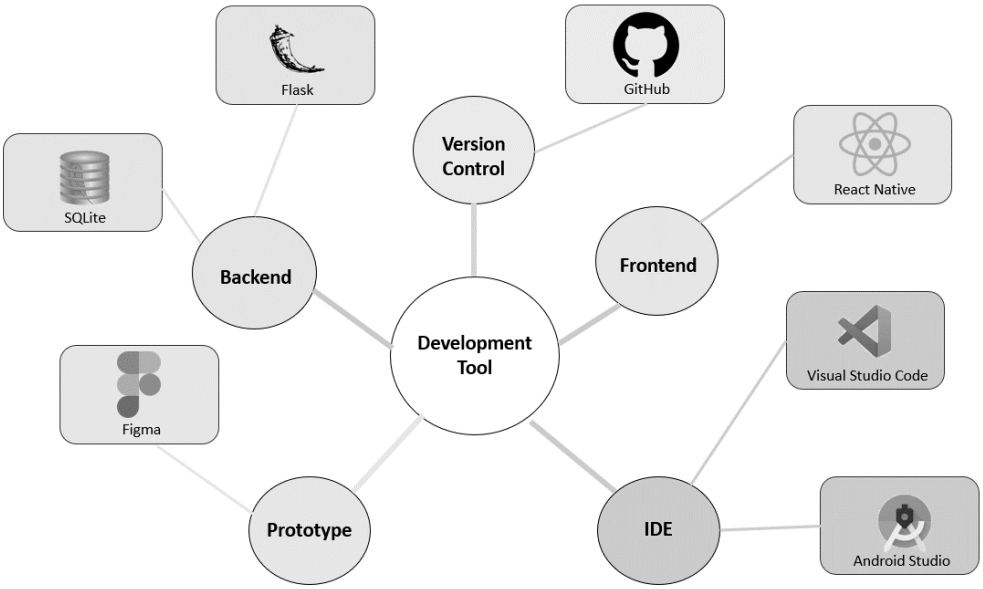


Figure 3.2: Development Tools Adopted.

3.4.1 React Native

React Native is an open-source JavaScript Framework for creating and building mobile applications. It is a popular framework used in mobile applications because of its ability to create cross-platform apps with a single codebase, cutting development time and costs. React Native also provides high performance user interfaces that closely resemble native app experiences by utilizing native components. Its ‘hot reload’ feature speeds up the development,

and a vibrant community provides a variety of tools and libraries. Hence, React Native is the top choice for this proposed mobile application.

3.4.2 Flask

Flask is a lightweight and flexible micro web framework built in Python. For this proposed project, Flask serves as a powerful backend server where the CNN and database are developed. Flask as a backend server in this proposed project, is able to handle the HTTP requests and responses. This makes it an ideal bridge between the backend database, CNN model and the React Native frontend. By creating API endpoints in Flask, the frontend mobile application can send requests, and Flask can then process these requests and return the responses back to the frontend mobile application. The simplicity, robustness and integration capabilities of the Flask ensures the smooth interaction between the backend part and frontend part in this proposed project.

3.4.3 SQLite

SQLite is a self-contained and serverless relational database management system. It is not a client-server database engine, instead it is embedded in the end program. SQLite's lightweight nature is ideal for mobile application development, delivering excellent performance while consuming little resources. For this proposed project, SQLite plays a vital role in storing and managing the data needed for the mobile-based system. It is designed specifically for the local storage, offers an way to store various form of data, ranging from the data within the frontend mobile application to the datasets needed for training the CNN. By including SQLite into the proposed project, the data storage, manipulation, and retrieval across both frontend and backend will become effective and efficient.

3.4.4 Visual Studio Code

Visual Studio Code is a powerful IDE that aids in the development process for this proposed project. It provides an ideal environment for the coding, testing and managing the entire application. Its broad collection of extensions such as Python, JavaScript, Prettier and so on boosts productivity by offering the capabilities like code autocompletion, debugging and error reporting.

Furthermore, the user-friendly design of Visual Studio Code makes it very easy to operate. It also supports various type of programming languages including JavaScript and Python, and provides smooth collaboration between frontend and backend for this proposed project. In short, Visual Studio Code promotes efficient coding practices, streamlines the debugging process and allows the seamless integration of the Flask framework and React Native components.

3.4.5 Android Studio

In this proposed project, Android Studio is mainly used to launch the emulator for the mobile application testing purpose. Having Android Studio, the emulator provides a targeted solution for testing and enhancing the mobile application. Besides, the emulator gives a simulated environment in which the application can be executed and evaluated without the need for a physical device. This functionality helps to validate the integration of the backend part and frontend part within the application to make sure the developed mobile application is functioning optimally and able to provide smooth user interface and user experience.

3.4.6 Figma

Figma is a collaborative design and prototyping tool which is critical in enhancing the user interface and user experience aspects. In other words, it is an intuitive platform to help to design and develop the low and high-fidelity prototypes efficiently with the final goal that the application can be designed better in terms of user interface and user experience. Figma also provides interactive prototyping features which allow to visualize the interactions within the application. For this proposed project, Figma is used to create the prototypes prior to the actual implementation so that the real mobile application will be more realistic and closer to the end users.

3.4.7 GitHub

GitHub is a web server for distributed software version control system based on Git. In this proposed project, it acts as the remote code repository hosting platform for the version control. It can be used to store and control the different

versions of the system periodically. Specifically, the revert benefit provided by the GitHub can be utilized if the latest version of the system uploaded to the GitHub encounters integration issues or any errors. Its pull request feature also can assure the code quality before the real implementation or deployment.

3.5 Conclusion

In conclusion, this chapter had discussed and explored the software development methodology adopted in this proposed project which is the RAD Phased Development Methodology with clarification of the tasks and deliverables in each phase and subsystem. Next, WBS and Gantt Chart for this proposed project were developed to map each necessary task to the project timeline so that this proposed project can be completed within the scope and deadline as expected. Also, all of the development tools including the frontend, backend, IDE, version control system and prototyping tool adopted to develop the proposed system were discussed, analysed and justified.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter will be discussing the requirement discovery, requirement specifications of the proposed project, use case diagram, use case description, ERD as well as some simple prototypes for the proposed application.

4.2 Requirement Discovery

The strategy or method that is used to perform the requirements elicitation is user-based. After understanding the existing problems faced by the CSCS, the problems are studied deeply and some objectives as well as the scope that must be covered are clarified further. In this case, the needs of the CSCS are clearly identified and this proposed project is initiated. With clear-cut problem statements, objectives, solutions, scopes and so on, a set of related requirement specifications will be developed to consolidate the proposed application's needs and avoid running out of the main topic and scope.

4.3 Requirement Specification

The system requirement specifications can be categorized into two sections which are the functional requirements and non-functional requirements.

4.3.1 Functional Requirements

Functional requirements (FR) are the characteristics and features that a software application and system must fulfil. These requirements specify what the software is expected to do and how it should behave. They provide a clear outline of the development of the software and act as the basis for design, development, implementation and testing. The functional requirements of the application for this proposed project are created as shown in Table 4.1.

Table 4.1: Functional Requirements

ID	Functional Requirements
FR01	The application shall allow users to sign up a new profile account so that they can utilize the application.
FR02	The application shall allow users to log in their profile account before they can make use of the application.
FR03	The application shall allow users to edit their account information to manage their profile.
FR04	The application shall be able to show some brief information about the concepts of this application to the new users.
FR05	The application shall be able to receive and capture the surface parameter inputs of defected steel product from the interaction with users then predict the grade of the steel product.
FR06	The application shall be able to showcase the dashboard related to the steel product grade information in a informative way.
FR07	The application shall be able to show the summary of the steel grade processing records to the users.
FR08	The application shall allow users to log out their account after they had completed their tasks using the application.

4.3.2 Non-Functional Requirements

Non-Functional Requirements (NFR) are used to describe how a software system should behave and operate rather than what it should accomplish. They prioritize the characteristics such as speed, reliability, availability, security and so on. For this proposed project, the NFRs of the application are shown as in Table 4.2.

Table 4.2: Non-Functional Requirements

ID	Non-Functional Requirements	Category of NFR
NFR01	The application should load each of the functionalities in less than 1 minute.	Performance

NFR02	The downtime of the application should less than 5 minutes per day.	Availability
NFR03	The application should provide 24/7 services.	Availability
NFR04	The application should run and operate successfully in Android and iOS platform.	Interoperability
NFR05	The application should prevent the unauthorized access.	Security
NFR06	The application should authenticate the valid user using the correct username and password.	Security
NFR07	The application should provide simple user interface for easy learning and operating purpose.	Usability
NFR08	The application should update the data-related interfaces using the newest data.	Reliability

4.4 Use Case

Use case acts as a concise and simple description of how a user or actor should interact with a system to accomplish a specific goal. It outlines the stage and interactions involved in a certain event. The purpose of the use case is to capture and express the system functional requirements from the view of the end users. It can be used to help in understanding user requirements, guiding system design and ensuring the system satisfies and meets requirements.

4.4.1 Use Case Diagram

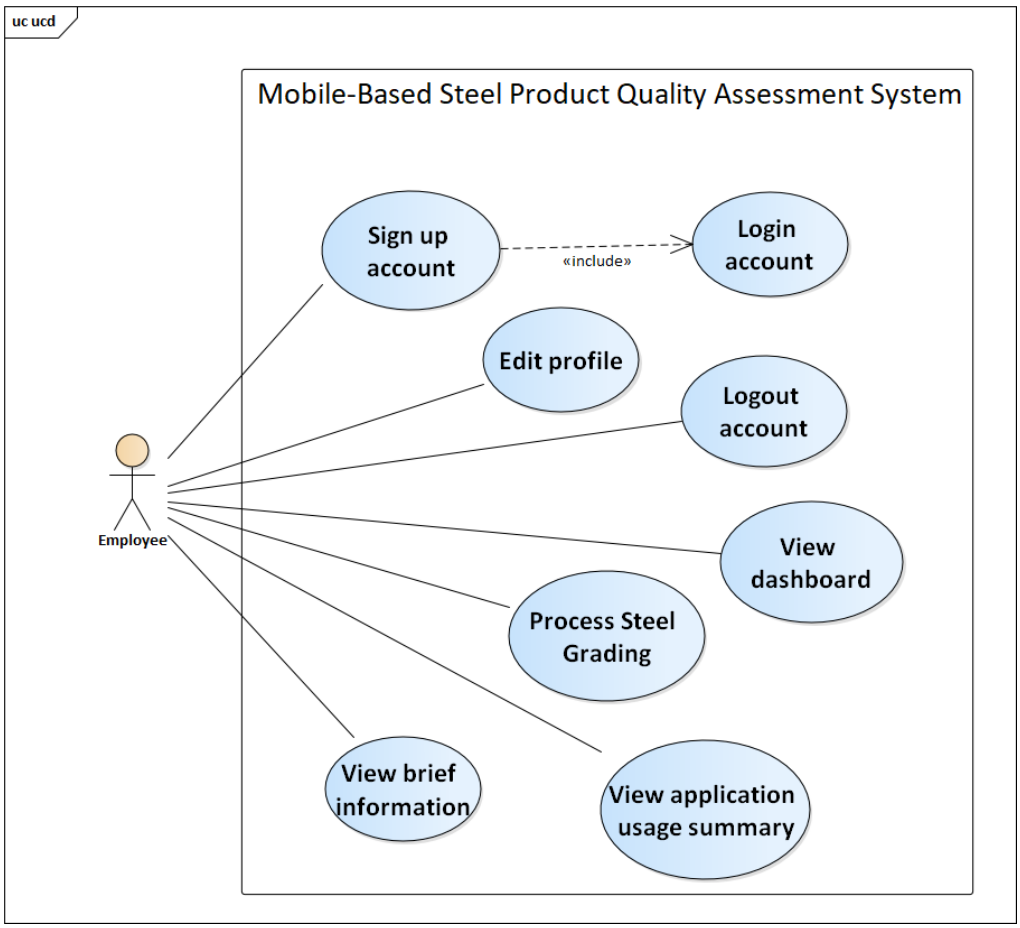


Figure 4.1: Use Case Diagram.

This use case diagram showcases how the end-user, which is the employee from the CSCS can interact with the mobile-based system. First and foremost, the employee can utilize the most important functionality of the system which is the 'process steel grading'. Using the mobile-based system, the employee can know the grade of the steel product predicted by the AI model which has been trained at the backend site. Once the employee wants to process the steel grading within the system, they just need to manually enter the surface parameter inputs of the defected steel product in the corresponding text fields. After a few seconds, the result which is the steel grade will be predicted and generated by the backend AI model. All the parameter inputs and the predicted steel grade will be stored to update the dashboard. In addition, the employee also can perform the second most core functionality in this system which is the 'view dashboard'. This feature allows the employee to have a look and make relevant analysis of the

steel grade related information in real-time. In other words, the dashboard will keep updating using the newest data whenever the ‘process steel grading’ is performed. To robust the system, some complementary functionalities are added in. For example, the ‘sign up account’, ‘login account’, ‘edit profile’, ‘logout account’ and ‘view application usage summary’. The ‘view application usage summary’ is mainly used to track the history activity within this application such as who performs the ‘steel grade processing’, when he/she does it, what kind of grade output he/she gets and so on.

4.4.2 Use Case Description

Table 4.3: Use Case Description of Sign Up Account.

Use Case Name**: Sign Up Account**	ID**: 01**	Importance Level**: High**
Primary Actor**: Employee**	Use Case Type**: Detail, Real**	
Stakeholders and Interests**:** **Employee – wants to sign up as a employee user profile.**		
Brief Description**:** **This use case describes how employee sign up a employee user profile in the system.**		

Trigger**: Employee wants to use the CSCS Quality Assessment System to conduct the steel grading process.**
Relationships: Association**: Employee** Include**: Login Account** Extend**: N/A** Generalization**: N/A**
Normal Flow of Events**:** 1. Employee chooses to sign up a new user profile on the welcome screen. 2. Employee enters employee ID, username, full name, email and password. 3. Employee clicks the Sign Up button to sign up a new profile. 4. The system presents sign up successfully and customer profile is created. 5. The system will prompt the login session to the employee, sub-flows 5.1 and 5.2 will be performed.
Sub-flows**:** 5.1. The system will prompt the employee to enter the employee ID and password. 5.2. If the employee ID and password match, employee logins to the system successfully, else re-enter.
Alternate/Exceptional Flows: 2.a. If any each of employee ID, username, full name, email or password is entered in an invalid format, the system will prompt the employee to re-enter. 2.b. If any each of employee ID, username or email is already existed in the system, the system will prompt the employee to re-enter**.**

Table 4.4: Use Case Description of Login Account.

Use Case Name: **Login Account**	ID: **02**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to login to the system as employee**		
Brief Description: **This use case describes how employee can login to the system in order to utilize the features within the system.**		
Trigger: **Employee wants to use the CSCS Quality Assessment System to conduct the steel grading process.**		
Relationships: Association: **Employee** Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. The employee chooses to login to the system on the welcome screen. 2. The system prompts the employee to enter the corresponding employee ID and password. 3. The employee clicks on the login button to login to the system.		

Sub-flows: N/A
Alternate/Exceptional Flows: 2.a. If the employee ID is in an invalid format, the system prompts user to re-enter. 2.b. If the employee ID and password entered is not matched or incorrect, the system prompts user to re-enter.

Table 4.5: Use Case Description of Edit Profile.

Use Case Name: **Edit Profile**	ID: **03**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to edit his /her profile to amend some personal information**		
Brief Description: **This use case describes how employee can manage his / her profile by editing and making changes for his / her personal information.**		
Trigger: **Employee wants to modify his / her personal information in the profile to do some updates or changes.**		

Relationships:

Association: **Employee**

Include: N/A

Extend: N/A

Generalization: N/A

Normal Flow of Events:

1. The employee wishes to edit his / her profile information.
2. The employee opens the drawer navigator aside the system and clicks on the ‘Profile’ section.
3. The system navigates the employee to the Profile screen.
4. In the Profile screen, the employee taps on the ‘pencil’ icon.
5. The system displays a ‘edit profile’ screen for employee to do the editing and updating.
6. The employee starts to edit his / her desired sections and sub-flows 6.1 – 6.4 will be performed.
7. The employee clicks on the ‘tick’ button at the bottom to save all his / her changes.
8. The system will update all the changes made into the backend database.

Sub-flows:

6.1. If the employee wants to change his / her profile avatar, he / she clicks on the avatar and the system will prompt a screen for the avatar selection.

6.2 If the employee wants to edit his / her username, he / she deletes the contents inside username input area and re-enter the new value.

6.3. If the employee wants to edit his / her full name, he / she deletes the contents inside full name input area and re-enter the new value.

6.4. If the employee wants to edit his / her email, he / she deletes the contents inside email input area and re-enter the new value.

Alternate/Exceptional Flows:

6.a. If any each of username, full name or email is entered in an invalid format, the system will prompt the employee to re-enter.

6.b. If the email or username is already existed in the system, the system will prompt the employee to re-enter**.**

Table 4.6: Use Case Description of Process Steel Grade.

Use Case Name: **Process Steel Grade**	ID: **04**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to evaluate the grade of the steel product by automation via the help of AI model**		
Brief Description: **This use case describes how employee can evaluate the grade of the steel product automatically.**		
Trigger: **Employee wants to check on the grade of the steel product in which the input parameters are entered and selected and grade prediction result will be displayed.**		
Relationships: Association: **Employee** Include: N/A Extend: N/A Generalization: N/A		

Normal Flow of Events: 1. The employee wishes to check on the grade of the steel product in which the input parameters are entered and selected and the grade prediction result will be displayed on the screen. 2. The employee opens the drawer navigator aside the system and clicks on the 'Steel Grade Processing' section. 3. The system navigates the employee to the Steel Grading Processing screen. 4. The employee enters and selects the necessary input parameters of the steel product with defected surface. 5. After getting all the input parameters, the system will send, predict and generate the grade of the steel product using the backend trained AI model. 6. The grade predicted will be stored by the system into the backend database and sub-flow 6.1 will be performed. 7. Step 4 – 6 will be repeated until there are no any more inputs from the user.
Sub-flows: 6.1. The system will use the newly data in the database to update the dashboard.
Alternate/Exceptional Flows: 4.a. If any of the input parameter text field is entered in an invalid format, the system will prompt the employee to re-enter.

Table 4.7: Use Case Description of View Dashboard.

Use Case Name: **View Dashboard**	ID: **05**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	

Stakeholders and Interests:

Employee - wants to view the dashboard to understand the steel grading related information for further analysis

Brief Description:

This use case describes how employee can view the dashboard to gain insight on the useful information.

Trigger:

Employee wants to view the dashboard that reflect the steel grading processing and steel grade information.

Relationships:

Association: **Employee**

Include: N/A

Extend: N/A

Generalization: N/A

Normal Flow of Events:

1. The employee wishes to view the dashboard to further gain insights into the steel grading processing and steel grade information.
2. The employee opens the drawer navigator aside the system and clicks on the ‘Dashboard’ section.
3. The system navigates the employee to the Dashboard screen.
4. The employee takes a brief overview about the steel grade processing information and sub-flows 4.1 to 4.3 is performed.

Sub-flows: 4.1 If the employee wants to further explore the ‘Grade Distribution’, he / she clicks on the corresponding button and the system will direct him / her to the relevant chart or graph. 4.2 If the employee wants to further investigate the ‘Grade Trend over Time’, he / she clicks on the corresponding button and the system will direct him / her to the relevant chart or graph. 4.3 If the employee wants to view the detailed ‘Total Grade Prediction By AI’, he / she clicks on the corresponding button and the system will direct him / her to the relevant chart or graph.
Alternate/Exceptional Flows: N / A

Table 4.8: Use Case Description of View Application Usage Summary.

Use Case Name: **View Application Usage Summary**	ID: **06**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to view the summary of the records on the steel grading processing.**		
Brief Description: **This use case describes how employee can view the summary of the records on the steel grading processing.**		
Trigger: **Employee wants to view the summary of the records on the steel grading processing.**		

Relationships:

Association: **Employee**

Include: N/A

Extend: N/A

Generalization: N/A

Normal Flow of Events:

1. The employee wishes to view the summary of the records on the steel grading processing such as who has used this system for the steel grading, when the employee does the processing and so on.
2. The employee opens the drawer navigator aside the system and clicks on the ‘Summary’ section.
3. The system navigates the employee to the Summary screen.
4. The employee views the records summary and sub-flow 4.1 is performed.

Sub-flows:

4.1 If the employee wants to sort the displayed records summary, he / she clicks on the ‘Sort By’ button and choose the sorting type such as ‘by processing time’, ‘by employee ID’, ‘by grade’ and ‘by processing ID’.

Alternate/Exceptional Flows: N/A

Table 4.9: Use Case Description of View Brief Information

Use Case Name: **View Brief Information**	ID: **07**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to view the brief information about this system**		
Brief Description: **This use case describes how employee can view some brief information regarding the system in order to get familiar with the system as soon as possible.**		
Trigger: **Employee wants to view and obtain the brief information and concept related to the system.**		
Relationships: Association: **Employee** Include: N/A Extend: N/A Generalization: N/A		

Normal Flow of Events: 1. The employee wishes to view some brief important information related to this system so that the employee can get started with the system quickly. 2. The employee opens the drawer navigator aside the system and clicks on the 'Get Help' section. 3. The system navigates the employee to the Information screen. 4. The employee views the brief information provided and understand what and how the system operates.
Sub-flows: N/A
Alternate/Exceptional Flows: N/A

Table 4.10: Use Case Description of Logout Account

Use Case Name: **Logout Account**	ID: **08**	Importance Level: **High**
Primary Actor: **Employee**	Use Case Type: **Detail, Real**	
Stakeholders and Interests: **Employee - wants to logout his/her account after completed tasks using the application**		
Brief Description: **This use case describes how employee can logout his/her account successfully after completed the tasks using the application.**		
Trigger: **Employee wants to logout the account after completed tasks using the application**		

Relationships: Association: **Employee** Include: N/A Extend: N/A Generalization: N/A
Normal Flow of Events: 1. The employee wishes to logout the application after he / she had completed his / her tasks using the application. 2. The employee opens the drawer navigator aside the system and clicks on the 'Logout' section. 3. The system navigates the employee to the Logout screen. 4. The employee clicks the 'Logout' button at the middle bottom of the screen. 5. The employee has successfully logout and the sub-flow 5.1 is performed.
Sub-flows: 5.1 The system navigates the employee to the welcome screen again in case the employee wants to login their account to do anything.
Alternate/Exceptional Flows: N/A

4.5 Prototype

Once the end user which is employee starts up the mobile application, the application will display the welcome screen and prompt the employee to sign up and / or login the application to utilize the features as shown in Figure 4.2, Figure 4.3 and Figure 4.4.

Figure 4.2: Welcome Screen.

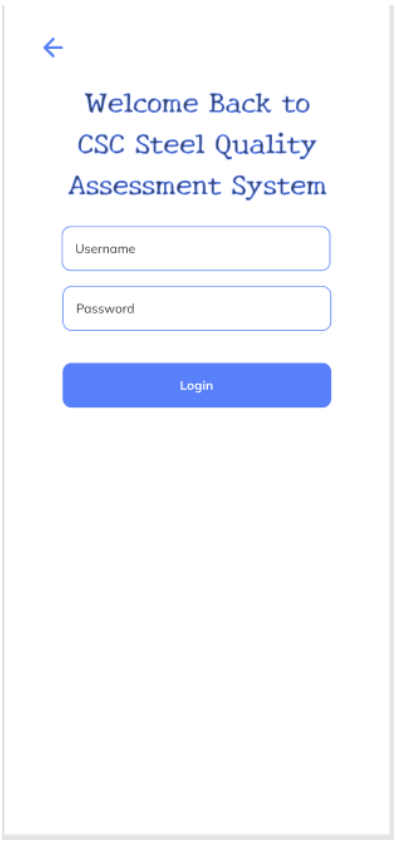


Figure 4.3: Login Screen.

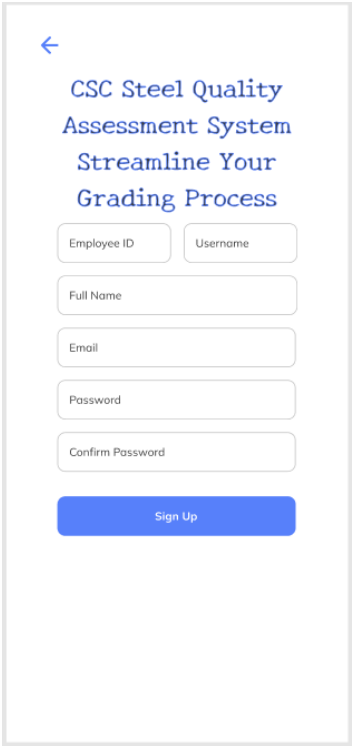


Figure 4.4: Sign Up Screen.

After that, the employee can manage his/her profile under the Profile screen by navigating to the screen using the drawer navigator as illustrated in Figure 4.5, Figure 4.6 and Figure 4.7.

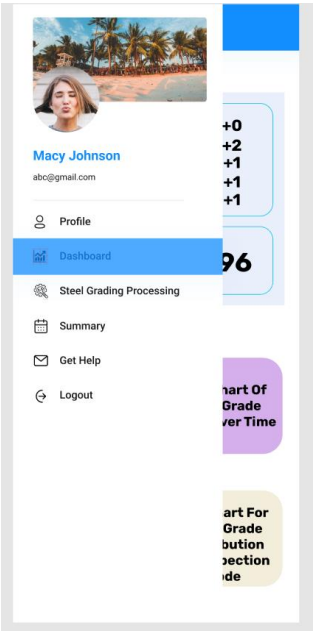


Figure 4.5: Drawer Navigator For Navigating To Other Screen.

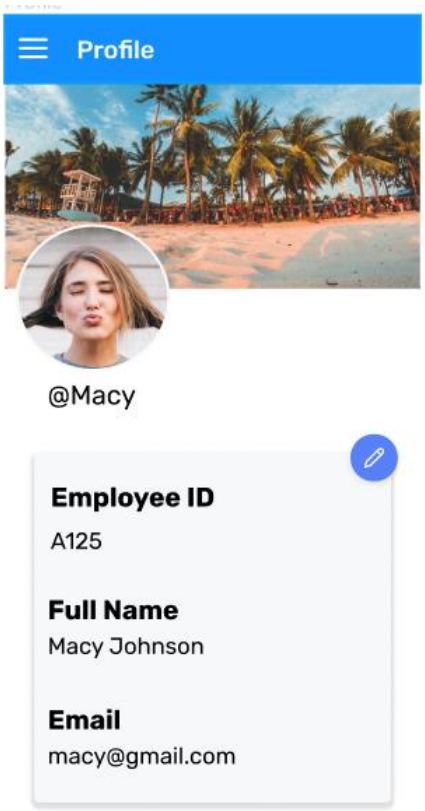


Figure 4.6: User Profile Screen.

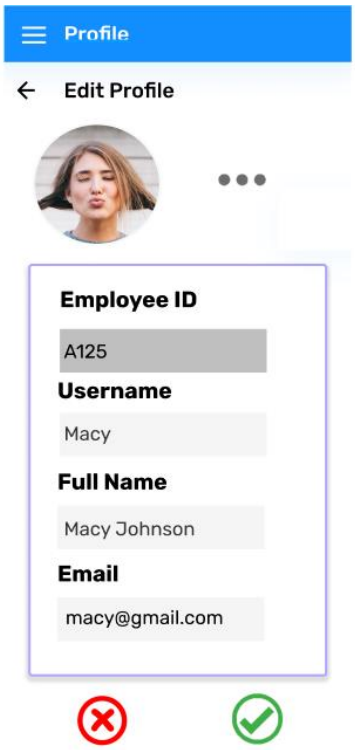


Figure 4.7: Edit Profile Screen.

For the main and most important features in this application which are the ‘steel grade processing’ using AI model and the dashboard visualization in real time, the employee can navigate to using the Drawer Navigator as well. Figure 4.8 and Figure 4.9 below show the main functionalities of the application.

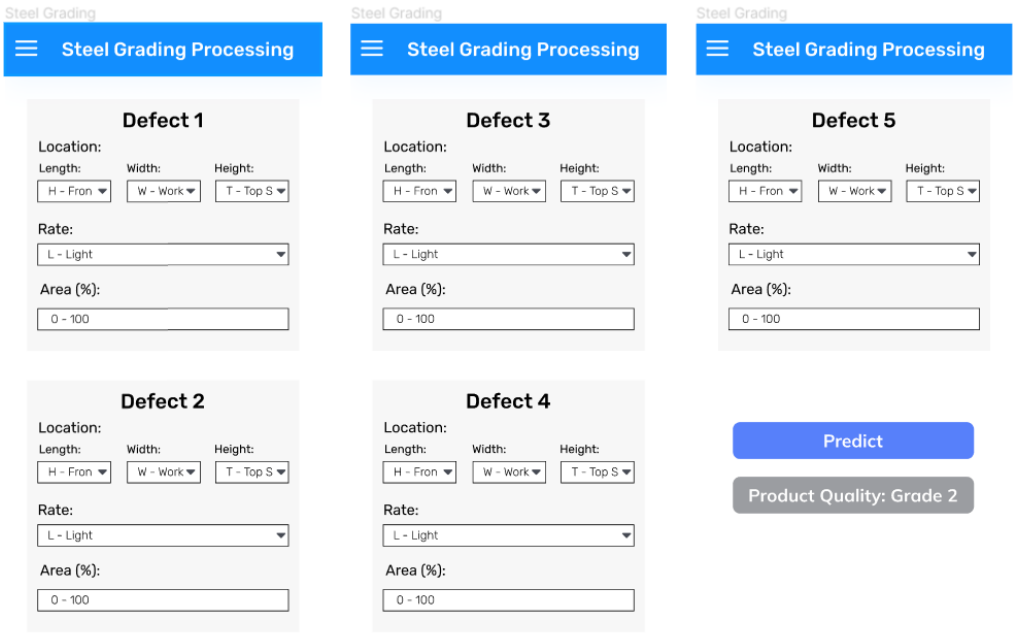


Figure 4.8: Steel Grade Processing Screen.

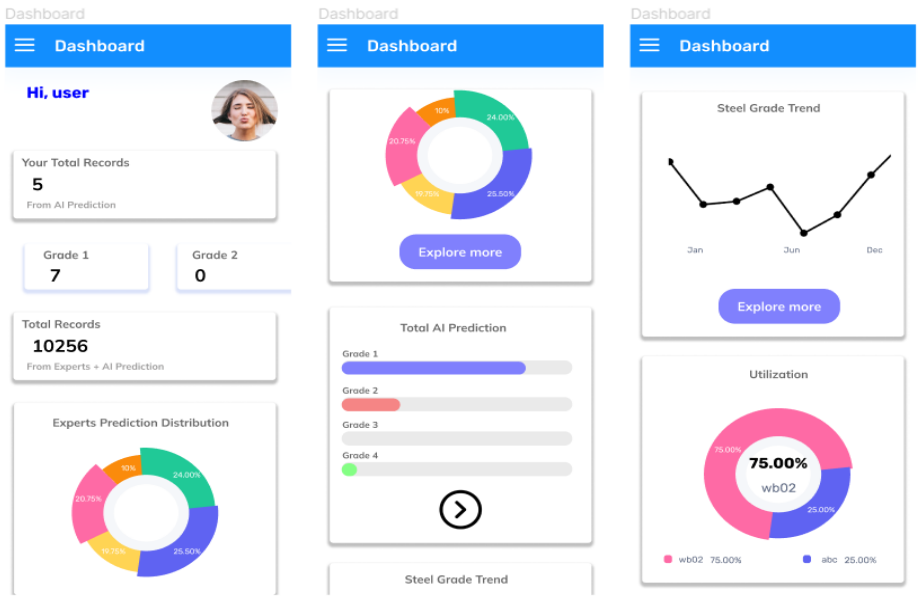


Figure 4.9: Dashboard Visualization Screen.

In the Dashboard Visualization screen, the employee can choose on the desired graph or chart to do further analysis and exploration by just clicking on the button. Figure 4.10, Figure 4.11, and Figure 4.12 displays some of the examples of the graphs or charts.

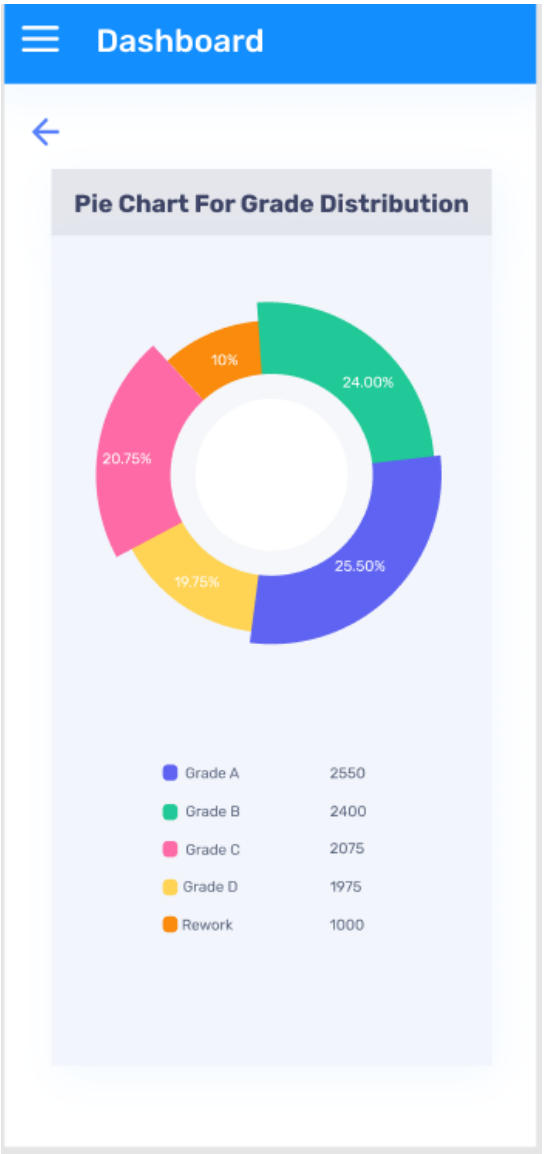


Figure 4.10: Grade Distribution.

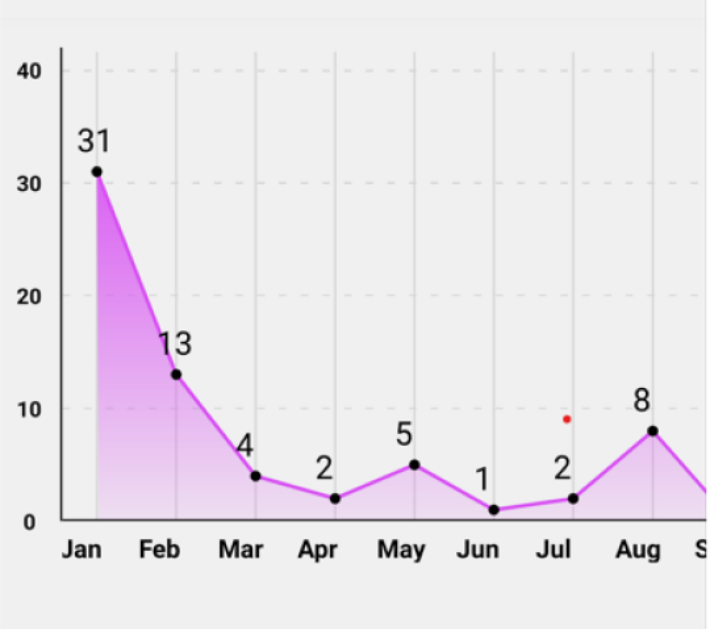


Figure 4.11: Grade Trend Over Time.

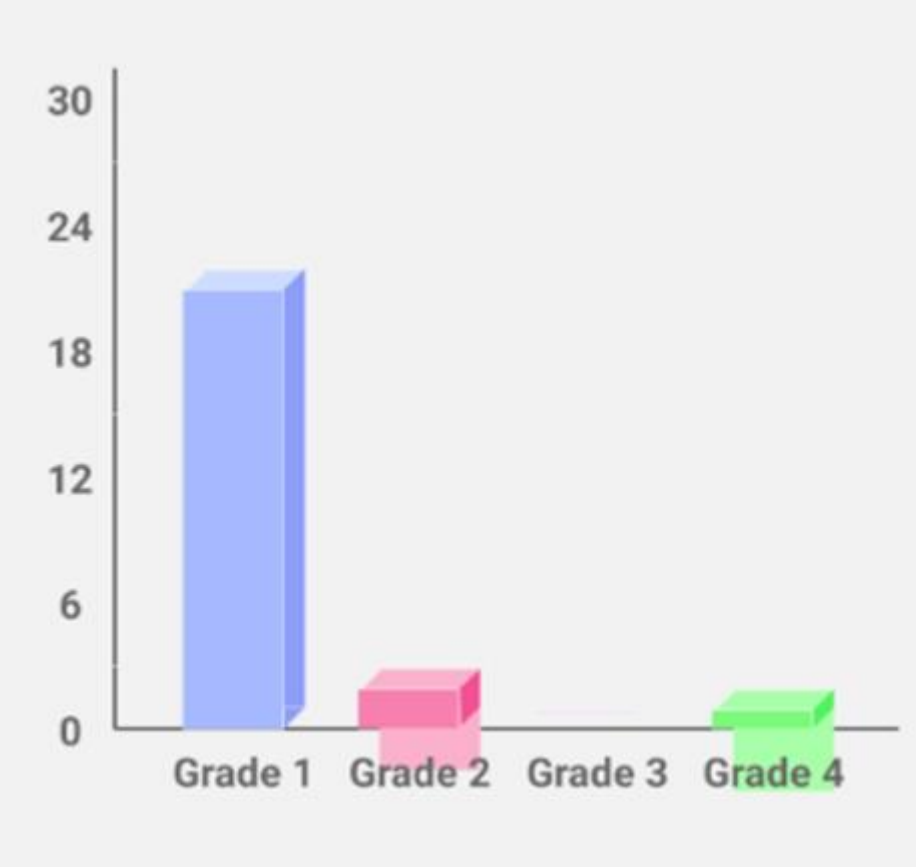


Figure 4.12: Total Grade Prediction By AI.

Apart from that, this application also implements some supplementary features for the employee to consolidate a more user-friendly environment. For example, the application offers one 'get help' screen to provide some information regarding this application so that the new user can be familiar with this application as soon as possible.

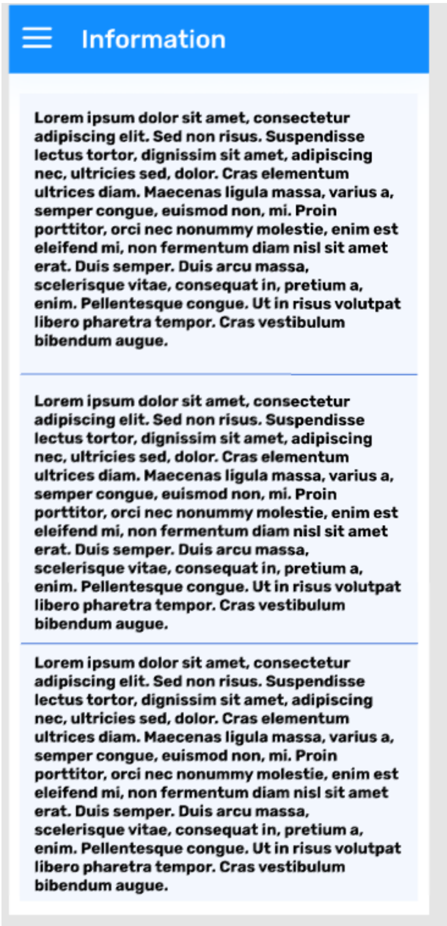


Figure 4.13: Example Of The Information Screen.

Last but not least, the summary screen will also be implemented to give the employee a quick overview on who has used this application on what time and so on. This feature somehow can be used as the tracking purpose as well. Figure 4.14, Figure 4.15 and Figure 4.16 show the summary screen with the sorting function and the screen if there are not any records yet.

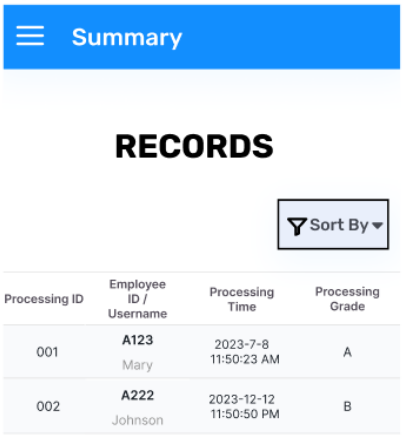


Figure 4.14: Summary Screen.

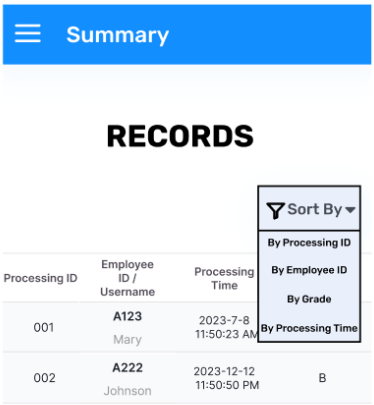


Figure 4.15: Summary Screen With Sorting.

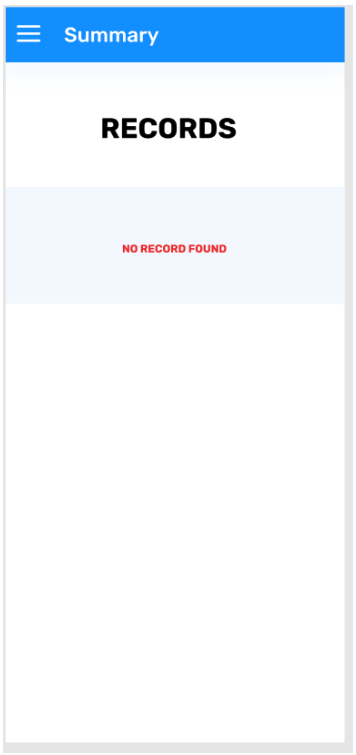


Figure 4.16: Summary Screen If There Are No Any Records Yet.

4.6 Summary

In a nutshell, this chapter had discussed the process to gather and elicit the requirements. Then, the functional and non-functional requirements were created afterwards. To have a better understanding of the functional requirements and how the system interacts with the end-user, employee, use case diagram and use case description were developed. Lastly, some prototypes were produced to show the UI and design of the proposed system so that the later development and implementation can become easier.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

In this chapter, the detailed system designs including the mobile application architecture and the CNN model framework will be discussed. These architecture designs can consolidate the development of the application and model so that the development will maintain consistency throughout the project. It gives the overall direction of how the application and model should be developed. Besides, the Entity Relationship Diagram (ERD) and Data Dictionary will also be designed to improve the data organization, reduce errors in the later stage and streamline the database development process. Unified Modelling Language (UML) diagrams such as the Data Flow Diagram (DFD) and Activity Diagram are emphasized in this chapter as well to illustrate the structure, behaviour and interaction within the mobile application. For example, the DFDs represent the flow of data within application whereas the Activity Diagram model the workflow and sequence of activities within the processes in the application.

5.2 System Architecture Design – Mobile Application

The system architecture design for the mobile application in this proposed project follows the three-tier client-server architecture design. It separates the application into distinct layers which include the frontend (mobile app), backend server (business logic) and also the database (data storage).

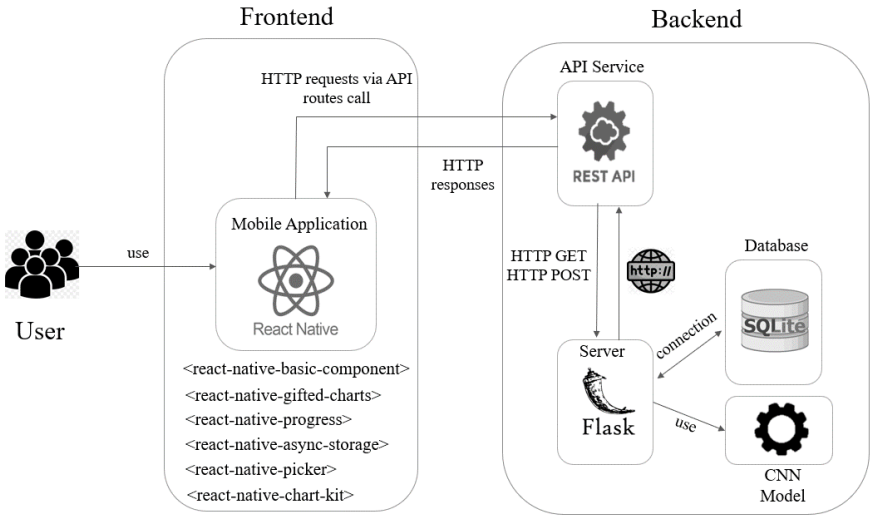


Figure 5.1: System Architecture Design – Mobile Application.

Figure 5.1 above depicts the system architecture design that will be adopted for the mobile application. As the frontend part that focuses to display the UI for interaction between application and users, the open-source UI software framework which is React Native is applied. Some appropriate React Native libraries are also imported to create a responsive, visual-appealing and user-friendly UI frontend. For example, the basic components such as View, Image, ActivityIndicator, TouchableOpacity and so on are used to build up the skeleton of the mobile application UI. Moreover, few third-party components like picker and progress bar are utilized to further enhance the frontend UI. Regarding the chart parts in the dashboard functionality, the react-native-gifted-chart and react-native-chart-kit will be adopted to promote the creation of the effective charts and dashboard. For the backend part, a Flask-based server is adopted to handle and process the business logic. For example, if there are any requests from the frontend or require any response data from backend, the Flask server is the one that responsible for that. To robust the backend Flask server, the database storage, SQLite is integrated into it. The SQLite mainly focuses on data storage and data retrieval. After the integration of SQLite, the backend Flask server can easily and quickly process the business logic that requires some data by just querying it in the database. Since the core functionality in this mobile application is the steel grading prediction by AI model, the trained CNN model is also necessary to be developed and used by the Flask server. In this

way, the Flask server is able to make the grade prediction by utilizing the CNN model. To facilitate the communication between frontend and backend, the REST API is adopted as the ‘middleman’ or intermediary. Once there are any HTTP requests made from the frontend part, those requests will be sent via the REST API to the backend Flask server for processing. Similarly, after the Flask server processes the requests, the response data will be returned to the user screen via the REST API. In another word, the Flask server should set up a lot of necessary API routes in which each API route is corresponding to different functions. If there are any HTTP requests or HTTP responses existed between the frontend and backend, programmatically, the API routes will be called along with the necessary data to do the data exchange between both parties and enable the communication.

5.3 System Architecture Design – CNN Model

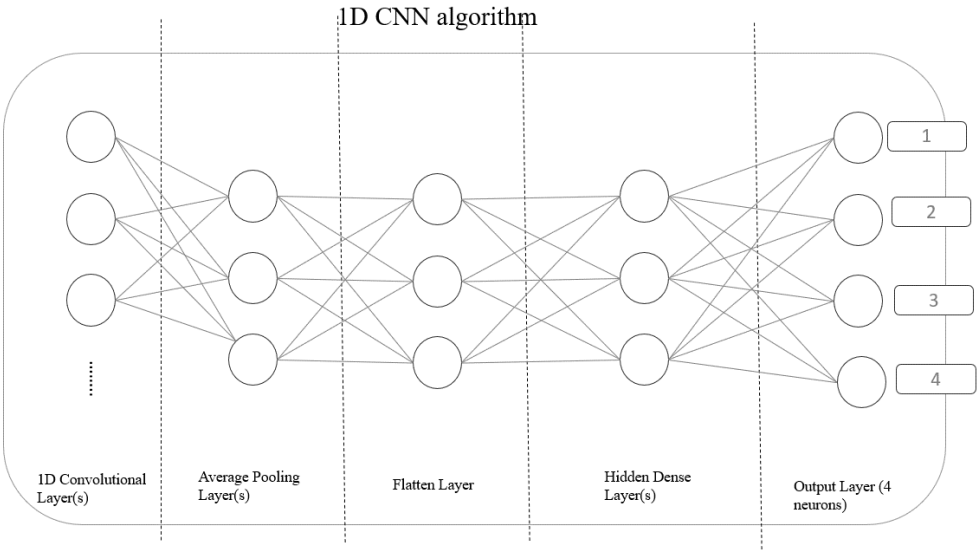


Figure 5.2: ID CNN Model Architecture.

Figure 5.2 demonstrates the architecture of the 1D CNN model that will be adopted in the application to automate the steel product grading process. The model can be classified into a few layers which are, 1D Convolutional Layer, Average Pooling Layer, Flatten Layer, Hidden Dense Layer and the Output Layer. The Convolutional Layer processes the input sequence to capture and extract the important patterns and features from the input sequence data. There

can be multiple Convolutional Layers in the model depending on the complexity of the data and patterns. In this case, 1 Convolutional Layer is adopted first and gradually increases the complexity or the number of Convolutional Layers if needed. After that, the output from the Convolutional Layer acts as the input of the Averaging Pooling Layer to reduce the spatial dimensions of the feature maps from the previous convolutional layer by taking the average value within each window. After that, the Flatten Layer is applied to convert the output into a flat 1-dimensional vector for reshaping the output into a format that the Hidden Dense Layers can understand and process. In other words, the Flatten Layer serves as a bridge between the Convolutional Layer and the Hidden Dense Layers. In the Hidden Dense Layer, the flattened vector will be used to learn the complex relationships. In this context, 2 hidden layers will be implemented and gradually increase if needed to get better accuracy and performance CNN model. Lastly, the final prediction is made through the Output Layer (4 neurons as having 4 target class labels) using the probability carried by each neuron, which provides the classification result to determine whether the surface defect steel product belongs to grade 1, 2, 3, or 4. The rule of thumb is that the neuron or class with the highest predicted probability is the one that the model considers most likely for the given input data.

5.4 Entity Relationship Diagram (ERD)

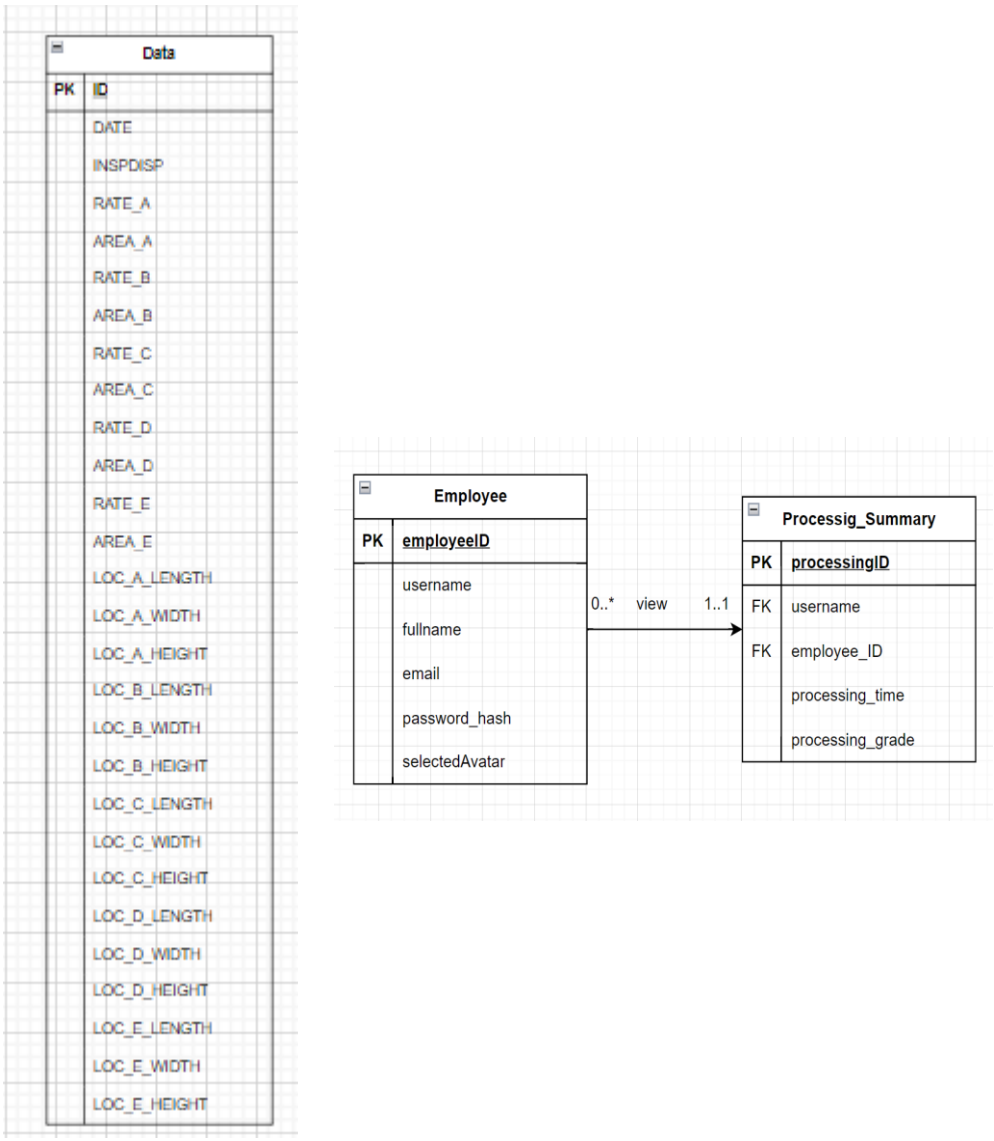


Figure 5.3: Entity Relationship Diagram.

This ERD represents the tables or entities that will be created in the database for the proposed mobile-based system. In the ERD shown, it implies that the 'Employee' and 'Processing_Summary' entities have a relationship in which the employee can view the processing summary by creating the relationship using the primary key and foreign key. As for the 'Data' entity, it has no relationship with other entities in the database. It is just created to store the excel file data (the experts predictions / evaluations data used to develop the CNN model) that is needed to create the dashboard. Also, whenever there is new data captured for the steel grade processing, these data will be stored in the 'Processing_Summary' entity and be extracted as well as queried to update the dashboard for the users.

5.5 Data Dictionary

Data Dictionary is a structured repository or collection of metadata that provides detailed information about the data within a database. It serves as the central reference for understanding the meaning, structure, and usage of the attributes within the database. Table below shows the data dictionary for the SQLite database that will be adopted in the mobile application.

Table Name: employee

Table 5.1: Data Dictionary for 'employee' Table.

Field Name	Description	Data Type	Unique	A.I.	PK	FK
employeeID	User's employee ID	TEXT	✓		✓	
username	User's username in application	TEXT	✓			
fullname	User's full name	TEXT				
email	User's email	TEXT	✓			
password_hash	User's application password in SHA256 hash	TEXT				
selectedAvatar	User's profile avatar picture	TEXT				

Table Name: processing_summary

Table 5.2: Data Dictionary for 'processing_summary' Table.

Field Name	Description	Data Type	Unique	A.I.	PK	FK
processingID	Unique identifier of a processing	TEXT	✓	✓	✓	
processing_time	The time that the processing is carried out in	DATETIME				

	YYYY-MM-DD HH:MM:SS format					
processing_grade	The grade of the steel product after the AI prediction	TEXT				
employeeID	User's employee ID	TEXT	✓			✓
username	User's username in application	TEXT	✓			✓

Table Name: data

Table 5.3: Data Dictionary for 'data' Table.

Field Name	Description	Data Type	Unique	A.I.	PK	FK
ID	Unique identifier of a data	INTEGER	✓	✓	✓	
DATE	The date that the steel product is evaluated by the experts	DATE				
INSPDISP	The steel product grade	TEXT				
RATE_A	The surface defect degree in location A (Light / Medium / Heavy)	TEXT				
AREA_A	The area of surface defect in location A	REAL				
RATE_B	The surface defect degree in location B (Light / Medium / Heavy)	TEXT				

AREA_B	The area of surface defect in location B	REAL				
RATE_C	The surface defect degree in location C (Light / Medium / Heavy)	TEXT				
AREA_C	The area of surface defect in location C	REAL				
RATE_D	The surface defect degree in location D (Light / Medium / Heavy)	TEXT				
AREA_D	The area of surface defect in location D	REAL				
RATE_E	The surface defect degree in location E (Light / Medium / Heavy)	TEXT				
AREA_E	The area of surface defect in location E	REAL				
LOC_A_LENGTH	Length of location A	TEXT				
LOC_A_WIDTH	Width of location A	TEXT				
LOC_A_HEIGHT	Height of location A	TEXT				
LOC_B_LENGTH	Length of location B	TEXT				
LOC_B_WIDTH	Width of location B	TEXT				
LOC_B_HEIGHT	Height of location B	TEXT				

LOC_C_LENGTH	Length of location C	TEXT				
LOC_C_WIDTH	Width of location C	TEXT				
LOC_C_HEIGHT	Height of location C	TEXT				
LOC_D_LENGTH	Length of location D	TEXT				
LOC_D_WIDTH	Width of location D	TEXT				
LOC_D_HEIGHT	Height of location D	TEXT				
LOC_E_LENGTH	Length of location E	TEXT				
LOC_E_WIDTH	Width of location E	TEXT				
LOC_E_HEIGHT	Height of location E	TEXT				

5.6 UML Diagram – Data Flow Diagram (DFD)

DFD illustrates the flow of data within the mobile application and depicts how the data moves through various processes, data stores and external entities in the application. In this section, context diagram and DFD Level-0 are drawn to show the data flow within the mobile application for this proposed project.

5.6.1 Context Diagram

The context diagram below depicts the high-level overview of the data flow across the entire mobile application by showing the application as a single process, external entities and the interaction between them.

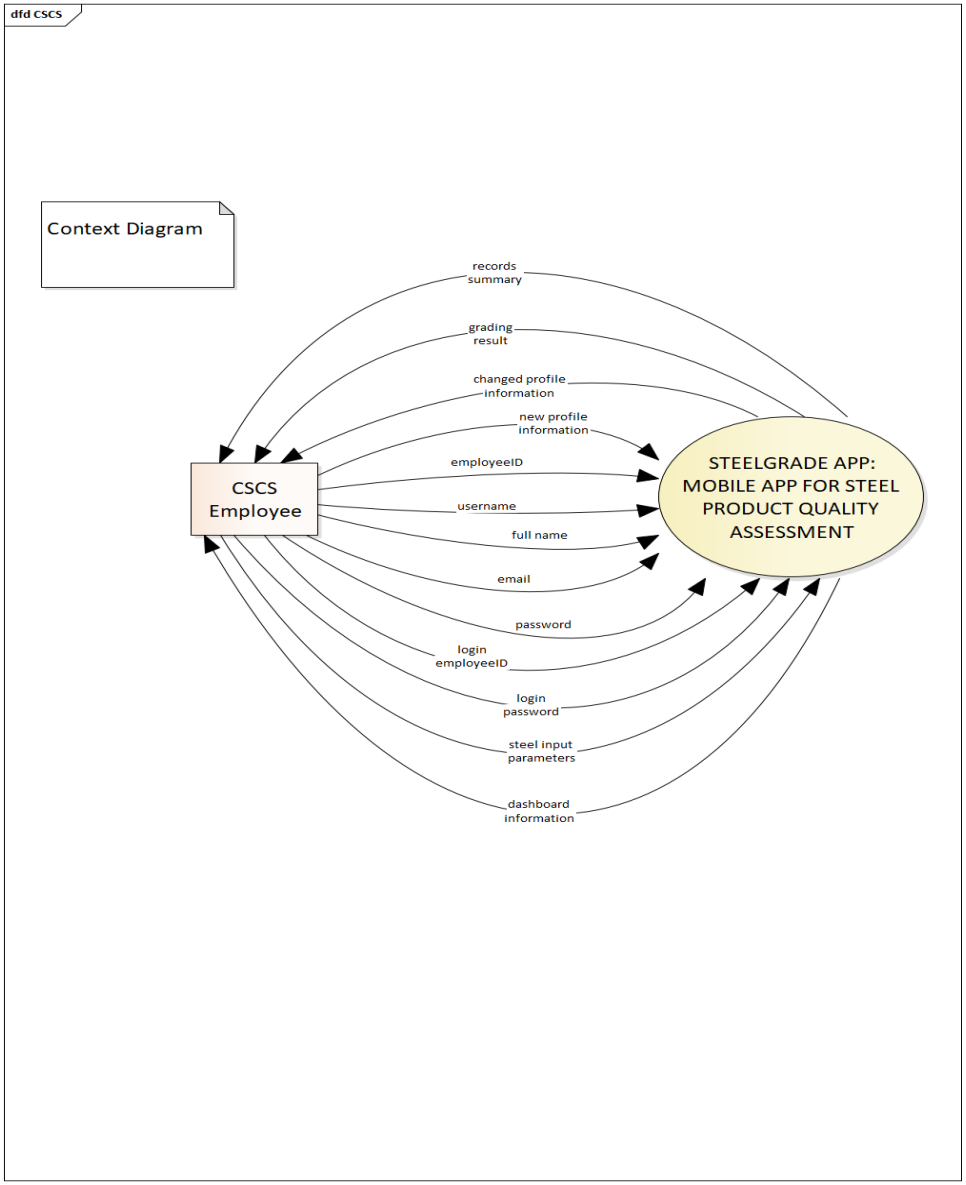


Figure 5.4: Context Diagram.

5.6.2 DFD Level-0

Following the Context Diagram, the DFD Level-0 can further break down the single process in the Context Diagram into a few main subprocesses. It shows the major processes or functionalities within the application, along with the data flows and any external entities or data stores involved.

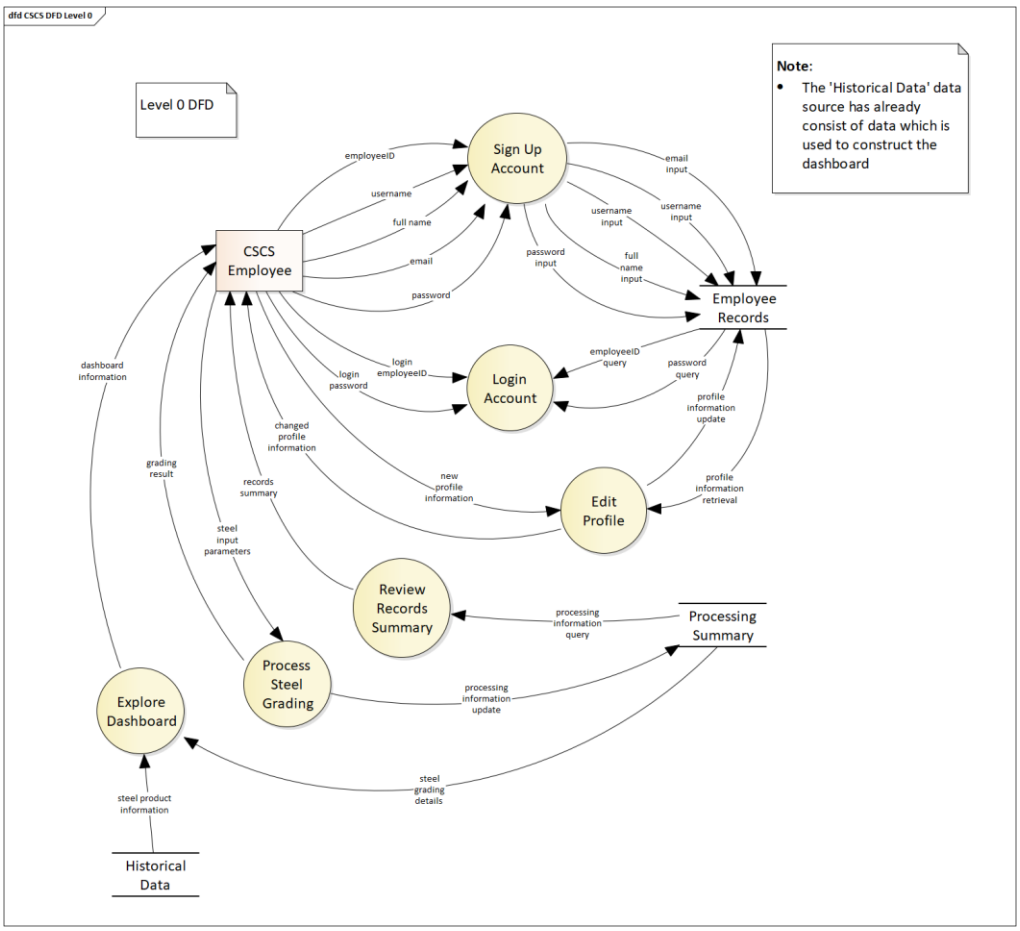


Figure 5.5: DFD Level-0.

5.7 UML Diagram – Activity Diagram

Activity Diagram is used to model the flow of activities or actions within the application. It provides the visual representation of the sequential and parallel activities that occur in the application. Using Activity Diagram, the steps to perform the functionalities within the application can be visualized easily.

5.7.1 Sign Up Account

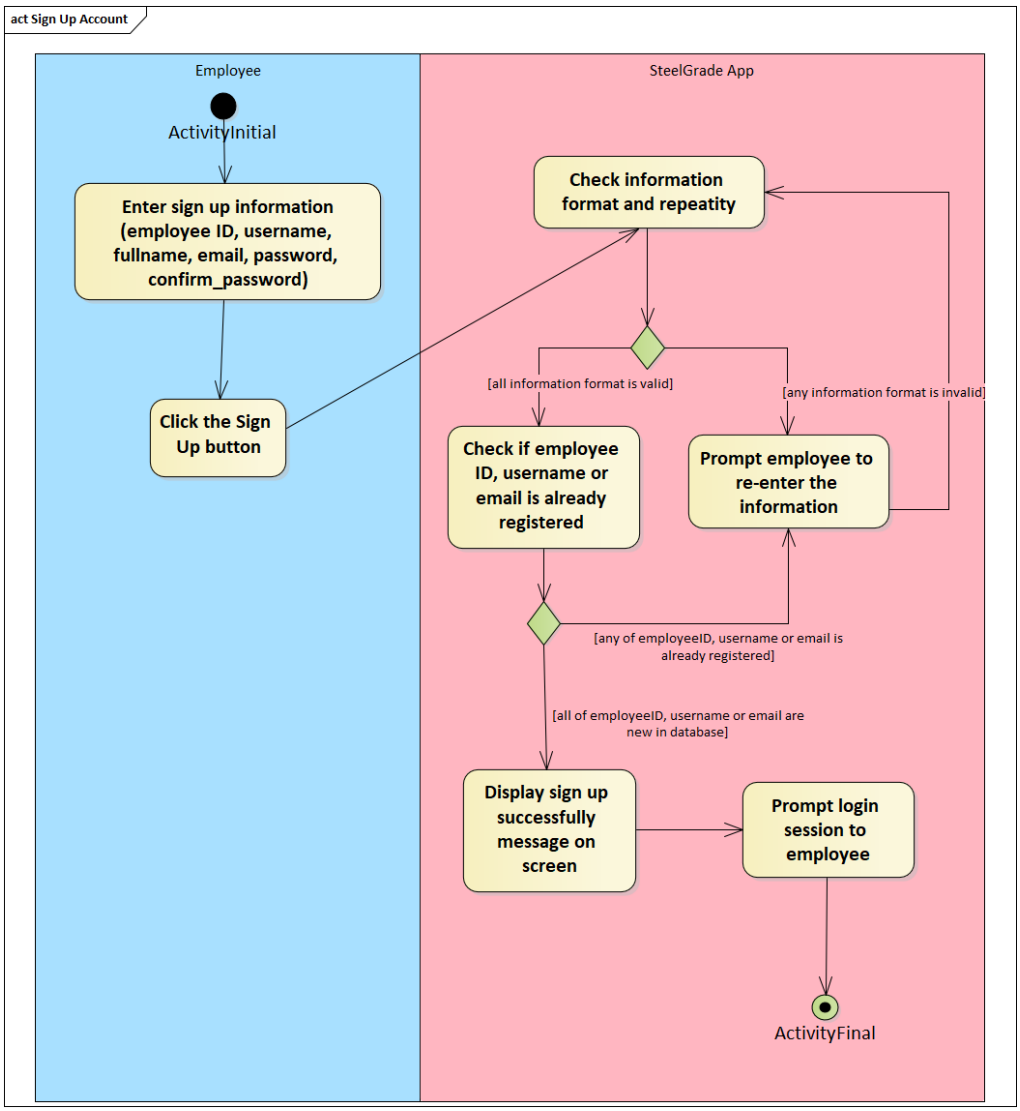


Figure 5.6: Activity Diagram for Sign Up Account.

5.7.2 Login Account

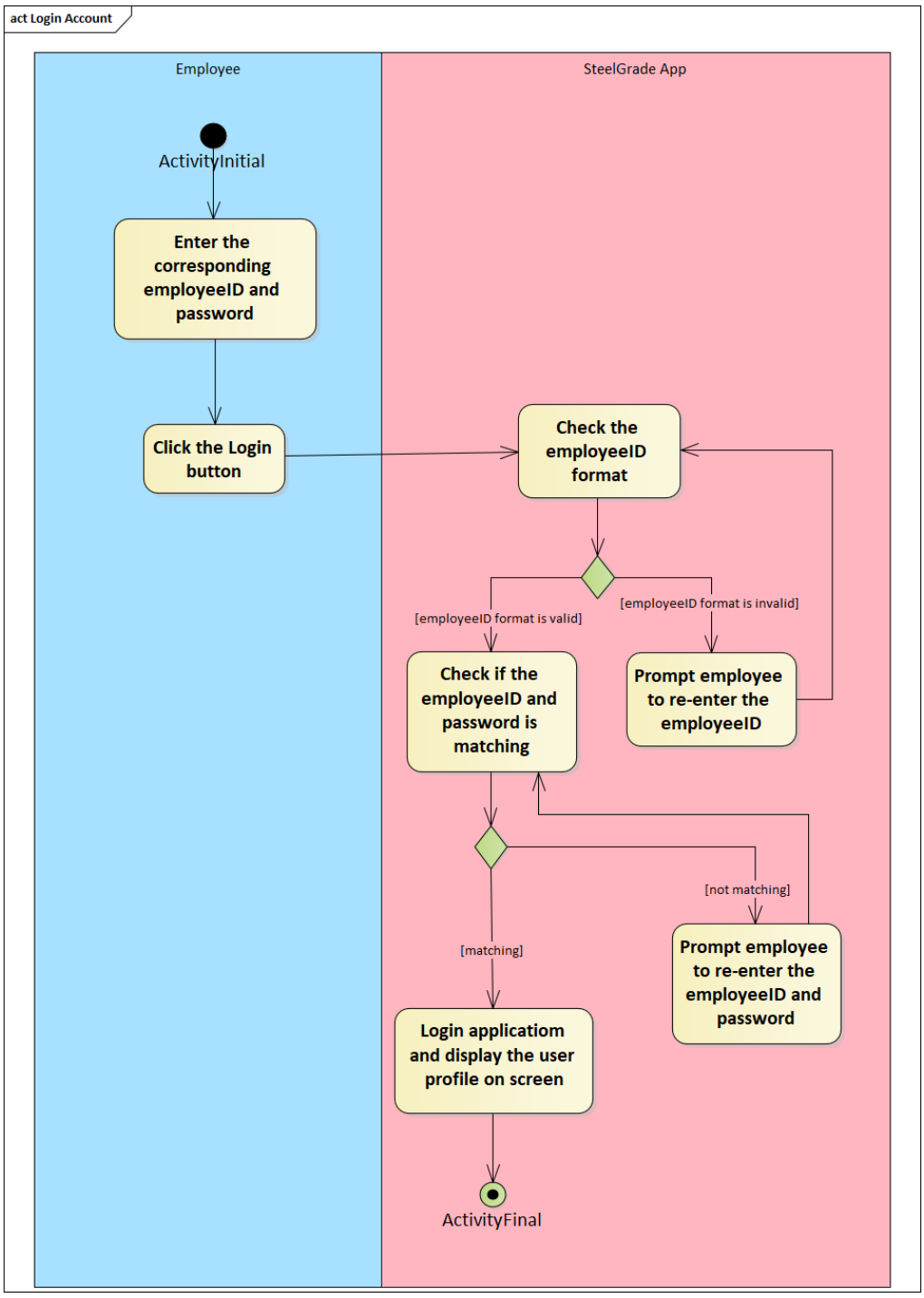


Figure 5.7: Activity Diagram for Login Account.

5.7.3 Edit Profile

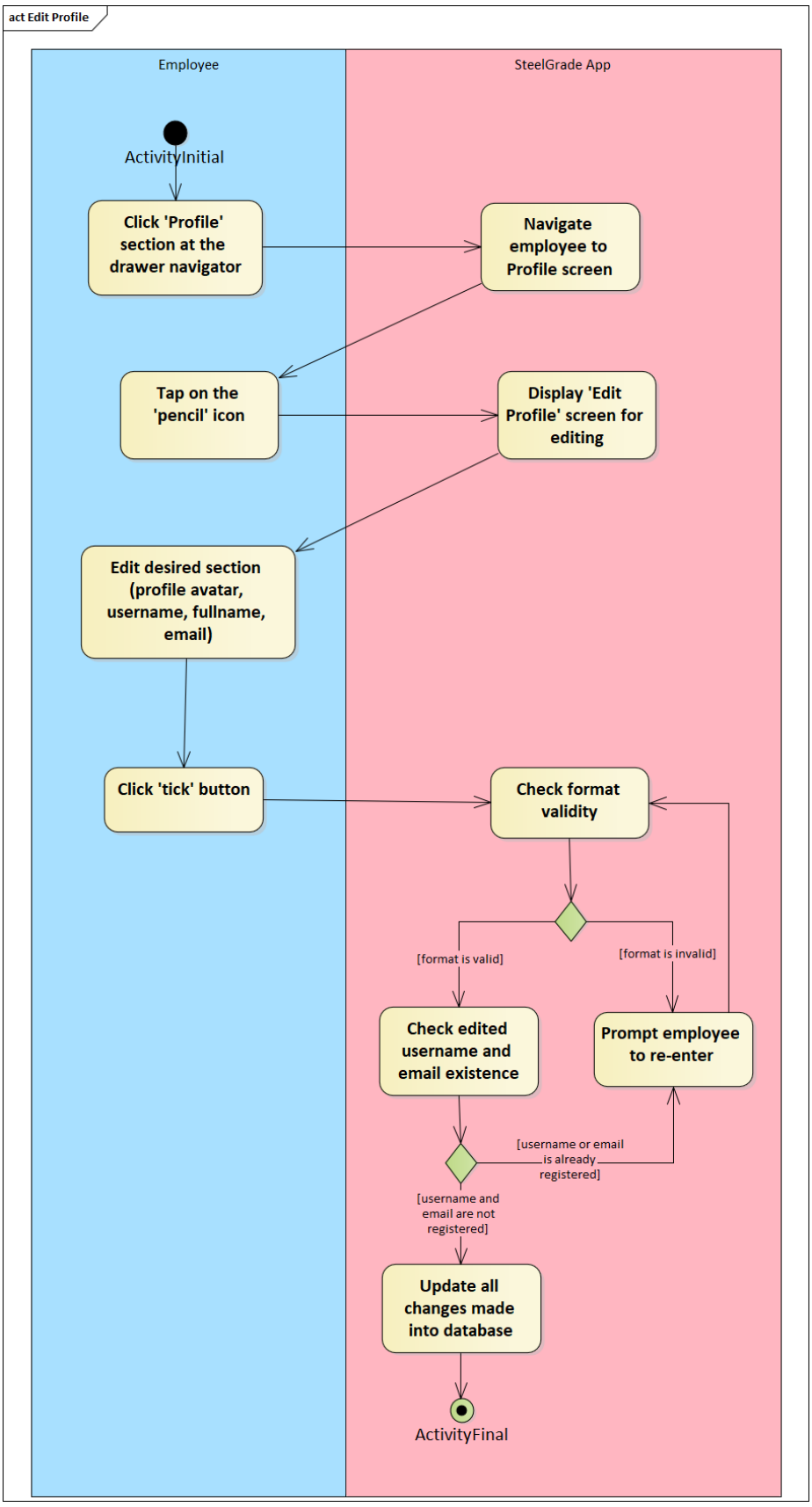


Figure 5.8: Activity Diagram for Edit Profile.

5.7.4 Process Steel Grade

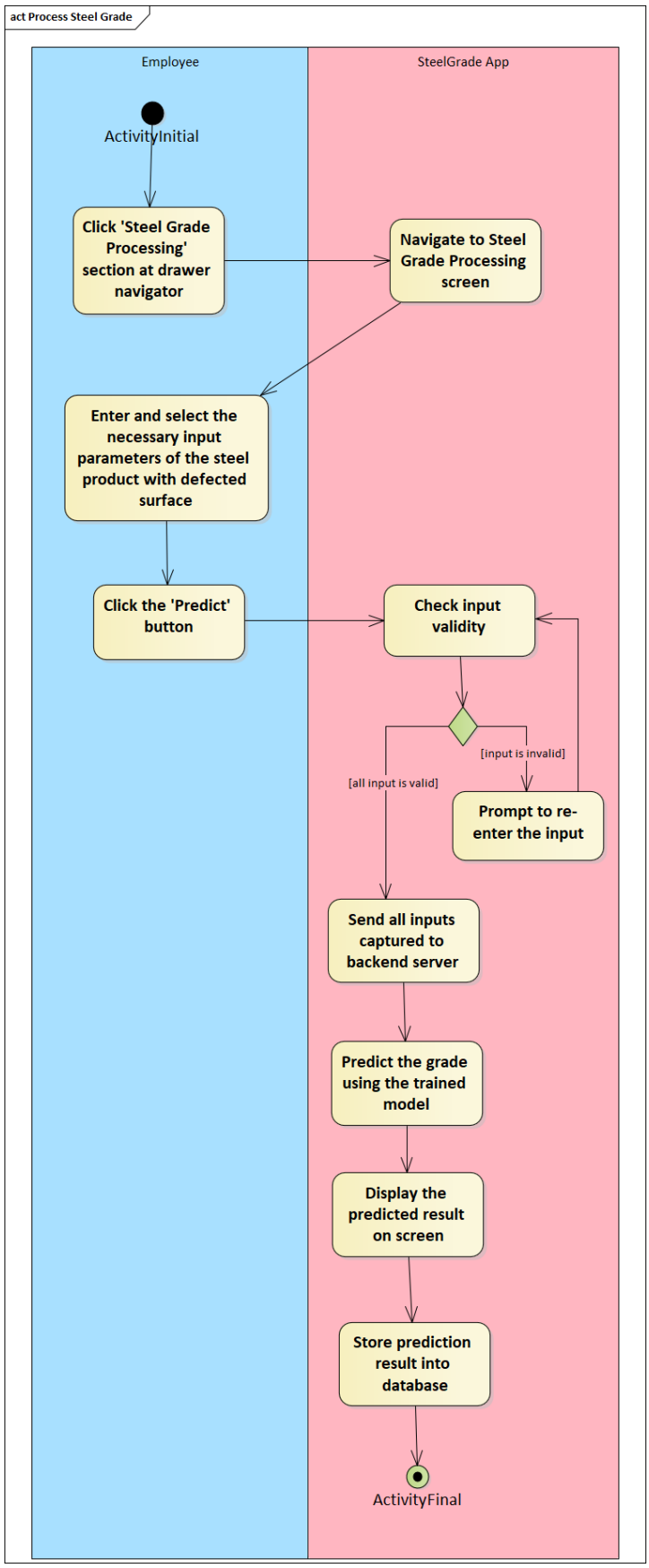


Figure 5.9: Activity Diagram for Process Steel Grade.

5.7.5 View Dashboard

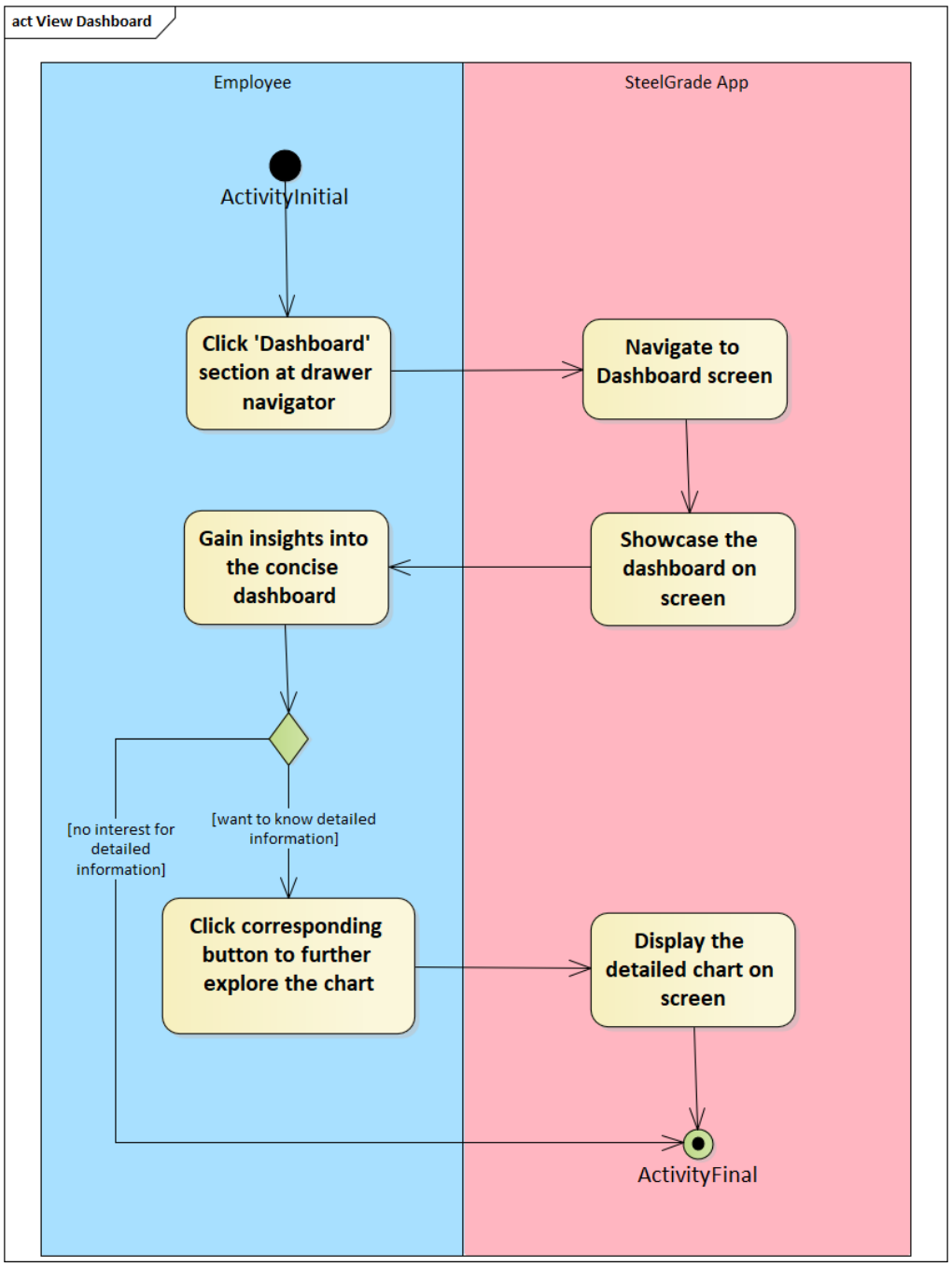


Figure 5.10: Activity Diagram for View Dashboard.

5.7.6 View Application Usage Summary

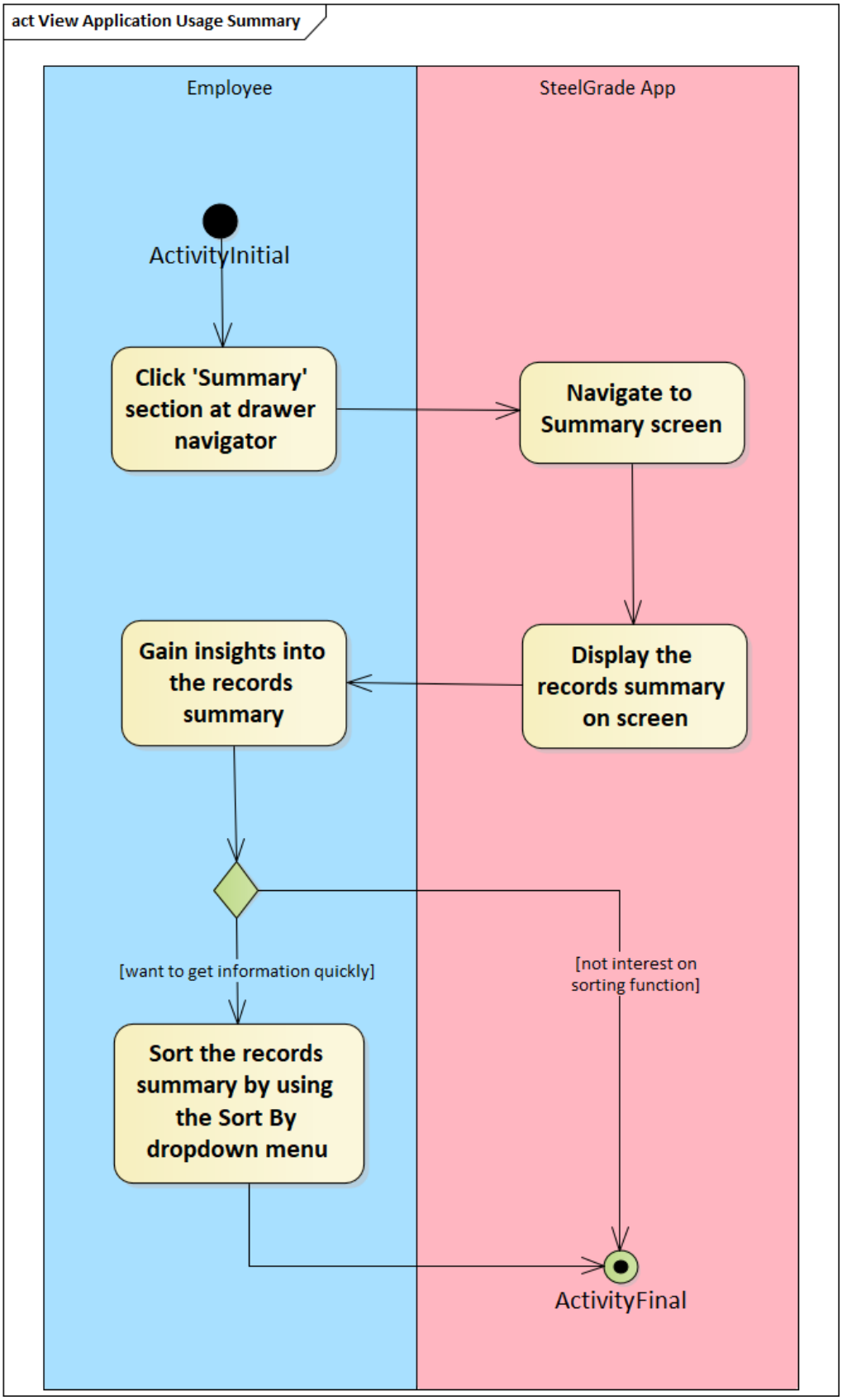


Figure 5.11: Activity Diagram for View Application Usage Summary.

5.7.7 View Brief Summary

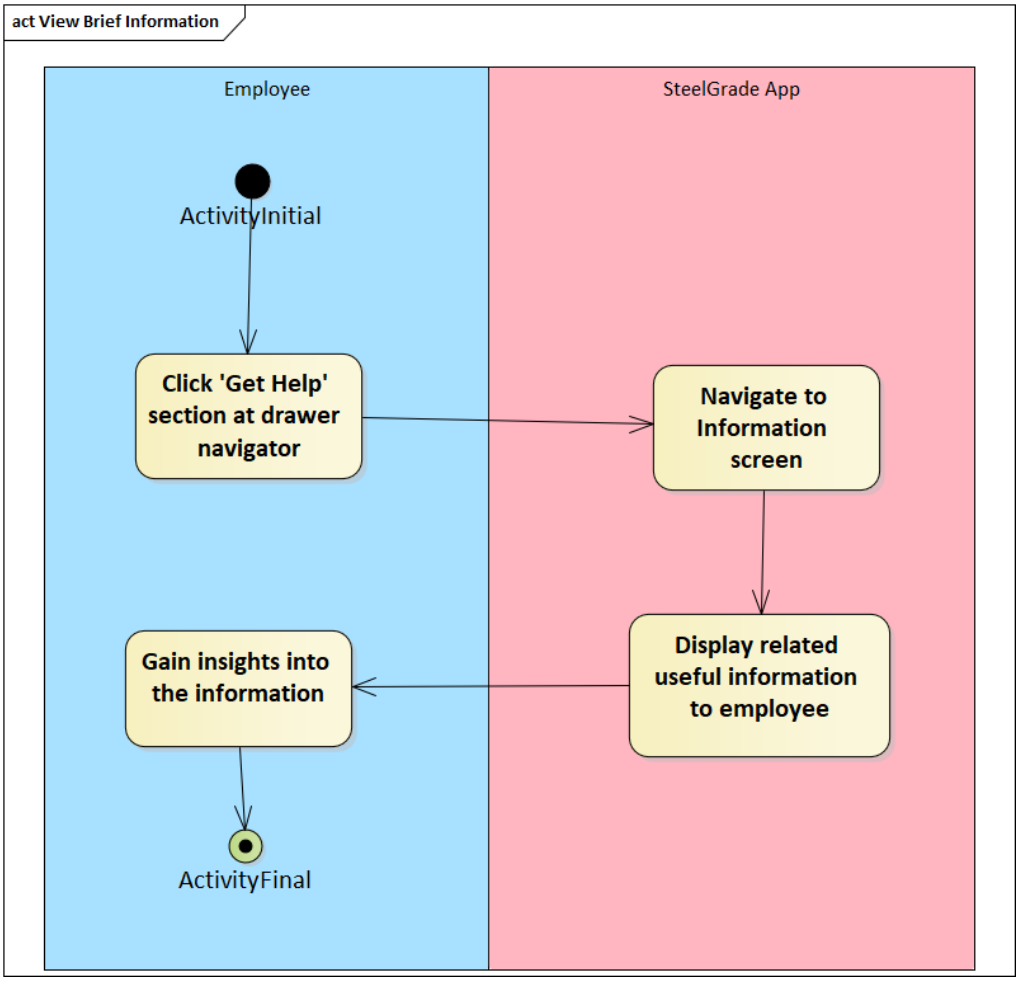


Figure 5.12: Activity Diagram for View Brief Information.

5.7.8 Logout Account

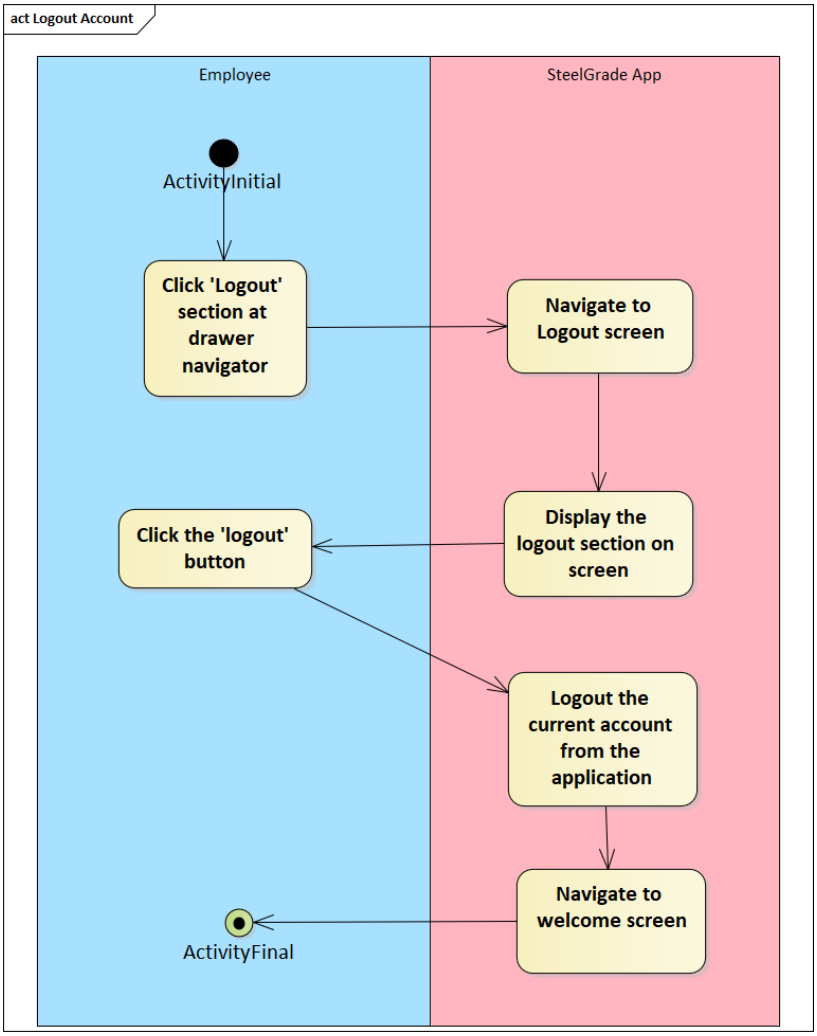


Figure 5.13: Activity Diagram for Logout Account.

5.8 Summary

In short, this chapter had discussed and highlighted the system architecture designs adopted for both mobile application and backend CNN model which is used for the autonomous grade evaluation. ERD and Data Dictionary had also been designed for the purpose to ease the development of the database in the later stage. Lastly, some useful UML diagrams such as DFDs and Activity Diagram had been drawn and made to visually represent the data flow, and detailed steps for processes within the application so that the development phases can be carried out smoothly and as expected in the project specifications.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

In this chapter, the comprehensive implementation steps will be highlighted and touched to promote the successful development of the complete mobile application from zero. In fact, this section will mainly emphasize detailly about the development of the CNN model, the build up of the backend server, the SQLite database preparation, the layout of the frontend UI, and the integration and communication between the frontend and backend.

6.2 CNN Model Development

This section discusses the development of the CNN model, from the raw data set to the trained model which can be used anywhere to do the classification task.

6.2.1 Data Preprocessing

After getting the raw data from the Excel file, it is necessary to understand and examine the data thoroughly. For those data which are not in interest or not in proper format, data preprocessing is the crucial step to carry out before the data can be fed into the model and promote the model training and development. Figure 6.1 shows the raw data from the Excel file before data preprocessing.

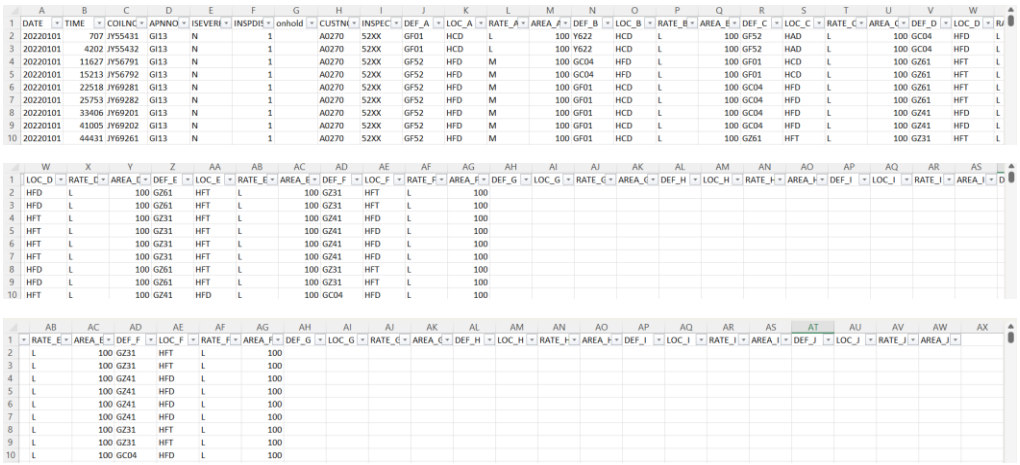


Figure 6.1: Original Dataset.

In the data preprocessing step, those unnecessary columns that are not going to be used and fitted in the model are dropped. Some data columns that are compact (contain many attributes) are also splitted into individual data columns. Furthermore, rows with unexpected or null value are dropped and removed to avoid any errors in the model building and training. Changing the unwanted data value to expected data value is also done in this stage. Also, the features scaling using the MinMaxScaler and data transformations using the OneHotEncoder and OrdinalEncoder are participated to make the data more appropriate and suitable in the CNN model development. Lastly, the processed data set is splitted into input (independent features) and target (dependent feature / class label) for the model training, model prediction and model evaluation task. Figure below illustrates some snippets of code in the data preprocessing step.

```
# Drop unnecessary columns
data.drop(["DATE", "TIME", "COILNO", "APNNO", "ISEVERHOLD",
           "onhold", "CUSTNO", "INSPECTIONCODE",
           "DEF_A", "DEF_B", "DEF_C", "DEF_D", "DEF_E",
           "DEF_F", "LOC_F", "RATE_F", "AREA_F",
          "DEF_G", "LOC_G", "RATE_G", "AREA_G",
          "DEF_H", "LOC_H", "RATE_H", "AREA_H",
          "DEF_I", "LOC_I", "RATE_I", "AREA_I",
          "DEF_J", "LOC_J", "RATE_J", "AREA_J",], axis=1, inplace=True)
```

Figure 6.2: Snippet Code of Drop Unnecessary Columns.

```
# Split LOCATION into LENGTH, WIDTH, and HEIGHT
data[['LOC_A_LENGTH', 'LOC_A_WIDTH', 'LOC_A_HEIGHT']] = data['LOC_A'].str.split("", expand=True).drop([0,4], axis=1)
data.drop("LOC_A", axis=1, inplace=True)
data[['LOC_B_LENGTH', 'LOC_B_WIDTH', 'LOC_B_HEIGHT']] = data['LOC_B'].str.split("", expand=True).drop([0,4], axis=1)
data.drop("LOC_B", axis=1, inplace=True)
```

Figure 6.3: Snippet Code of Splitting the Data Columns into Targeted Column.

```
# Remove rows with unexpected value
data.drop(data[
    (data['LOC_A_LENGTH'] != "H") &
    (data['LOC_A_LENGTH'] != "U") &
    (data['LOC_A_LENGTH'] != "M") &
    (data['LOC_A_LENGTH'] != "V") &
    (data['LOC_A_LENGTH'] != "T") &
    (data['LOC_A_WIDTH'] != "W") &
    (data['LOC_A_WIDTH'] != "X") &
    (data['LOC_A_WIDTH'] != "C") &
    (data['LOC_A_WIDTH'] != "Y") &
    (data['LOC_A_WIDTH'] != "D") &
    (data['LOC_A_WIDTH'] != "A") &
    (data['LOC_A_WIDTH'] != "F") &
    (data['LOC_A_HEIGHT'] != "T") &
    (data['LOC_A_HEIGHT'] != "B") &
    (data['LOC_A_HEIGHT'] != "D") &
    (data['LOC_B_LENGTH'] != "H") &
    (data['LOC_B_LENGTH'] != "U") &
```

Figure 6.4: Snippet Code of Drop Unexpected Value.

```
# drop row that contains null value
data = data.dropna(axis=0)
```

Figure 6.5: Snippet code of Drop Null Value.

```
# Preprocess label column (R --> 4)
data.loc[data['INSPDISP'] == 'R', 'INSPDISP'] = 4
```

Figure 6.6: Snippet Code of Change Label Column Value.

```
# Preprocess RATE Column L, M, H --> 0, 1, 2
data = OrdinalEncoder(cols=['RATE_A','RATE_B','RATE_C','RATE_D','RATE_E']).fit(data).transform(data)
```

Figure 6.7: Snippet Code of Data Transformation - OrdinalEncoder.

```
# Preprocess LOC columns (One-Hot Encoder)
onehot_X = OneHotEncoder(
    cols=['LOC_A_LENGTH', 'LOC_B_LENGTH', 'LOC_C_LENGTH', 'LOC_D_LENGTH', 'LOC_E_LENGTH',
          'LOC_A_WIDTH', 'LOC_B_WIDTH', 'LOC_C_WIDTH', 'LOC_D_WIDTH', 'LOC_E_WIDTH',
          'LOC_A_HEIGHT', 'LOC_B_HEIGHT', 'LOC_C_HEIGHT', 'LOC_D_HEIGHT', 'LOC_E_HEIGHT'],
    use_cat_names=True).fit(X_scaled).transform(X_scaled)
```

Figure 6.8: Snippet Code of Data Transformation – OneHotEncoder.

```
# Feature Scaling (MinMaxScaler)
cols = ['RATE_A', 'AREA_A','RATE_B', 'AREA_B','RATE_C', 'AREA_C','RATE_D', 'AREA_D','RATE_E', 'AREA_E']
mmSc = MinMaxScaler()
X[cols] = mmSc.fit_transform(X[cols])
X_scaled = X
```

Figure 6.9: Snippet Code of Feature Scaling– MinMaxScaler.

```
# Split into input and target
X = data.drop("INSPDISP", axis="columns")
y = pd.DataFrame(data.INSPDISP)
```

Figure 6.10: Snippet Code of features splitting into X and Y.

6.2.2 Train-Test Split

After processing the dataset into the appropriate format that is suitable for supervised machine learning task, the whole dataset is splitted into train set and test set for evaluating the performance of the CNN model in the later stage. By doing this, it also can assess the generalization capabilities of the CNN model, prevent overfitting and make informed decisions about the model selection and parameters tuning.

```
# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(onehot_X, y, test_size=0.33, random_state=1)
```

Figure 6.11: Snippet Code of Train-Test Split.

Figure above illustrates that in the CNN model development, the whole dataset is splitted into two parts, train set and test set, with 0.33 test size indicating that 33% of the data is used for testing whereas the remaining 67% is used for training. The parameter random_state is set to 1 for the reproducibility. By setting the random_state, it can ensure that each time executing the model, the data will be split into the same way, so the results and model developed are always consistent.

6.2.3 Model Building

To build up the CNN model, first, the CNN model is defined using the Sequential API. Then, the convolutional layers with ‘tanh’ activation function is added to extract the important features and patterns, followed by the averaging pooling layers to reduce the dimensionality of the input data by computing the average value within non-overlapping windows of size 2. After adding 3 convolutional layers and 2 averaging pooling layers, one flattening layer is added to convert the 3D features map to 1D so that the input data can be fed into the hidden layers / fully connected dense layers. Next, the artificial neural network part comes into the picture. The fully connected dense layers are added with the ‘tanh’ activation function to learn the complex relationships in the data. This activation function is a hyperbolic tangent activation function, which squashes the output values to the range of -1 to 1. Finally, the final output layer is added for the multi-class classification. Since the possible class label in the

data is 4, the neurons or units for the output layer are set to 4 to classify the input data into one of the 4 classes. This output layer also uses the 'softmax' activation function, which converts the output into a probability distribution over the 4 classes, and the class with the highest probability is chosen as the predicted class. Lastly, the model is compiled using the sparse categorical cross-entropy loss and Adam optimizer to optimize the model performance and minimize the classification errors.

```
# Instantiate the model
model = Sequential()
model.add(Conv1D(filters=6, kernel_size=5, activation='tanh', input_shape=(85,1)))
model.add(AveragePooling1D(pool_size=2, strides=2))

model.add(Conv1D(filters=16, kernel_size=5, activation='tanh'))
model.add(AveragePooling1D(pool_size=2, strides=2))

model.add(Conv1D(filters=120, kernel_size=5, activation='tanh'))

model.add(Flatten())

model.add(Dense(units=84, activation='tanh'))
model.add(Dense(units=4, activation='softmax'))

model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
```

Figure 6.12: Snippet Code of CNN Model Building.

6.2.4 Model Training

After configuring and compiling the CNN model, the next step is to use the data that had been processed and splitted to train the model. Thus, the train set is used to train the model with a specific batch size and number of epochs. The batch_size indicates that for the whole train set, the number of samples processed by the model in each iteration is 16. Then, the epochs implies that the model will iterate over all batches of the training data for 20 times. Combining the batch size and epoch can effectively compute the loss, updates the parameters or weights, and then optimize the model performance and accuracy.

```
# Fit the model
model.fit(X_train, y_train, batch_size=16,epochs=20, verbose=1)
```

Figure 6.13: Snippet Code of Model Training.

6.2.5 Model Serialization

Lastly, the trained model is saved using the pickle file so that it can be loaded and used in the backend Flask server without any errors.

```
# Make pickle file of our model
pickle.dump(model, open("model.pkl", "wb"))
```

Figure 6.14: Snippet Code of Model Serialization.

After executing the model file by running **python model.py** command, the pickle file is generated and saved in the current folder as figure below so that the backend server can load the pickle file and use the trained model.

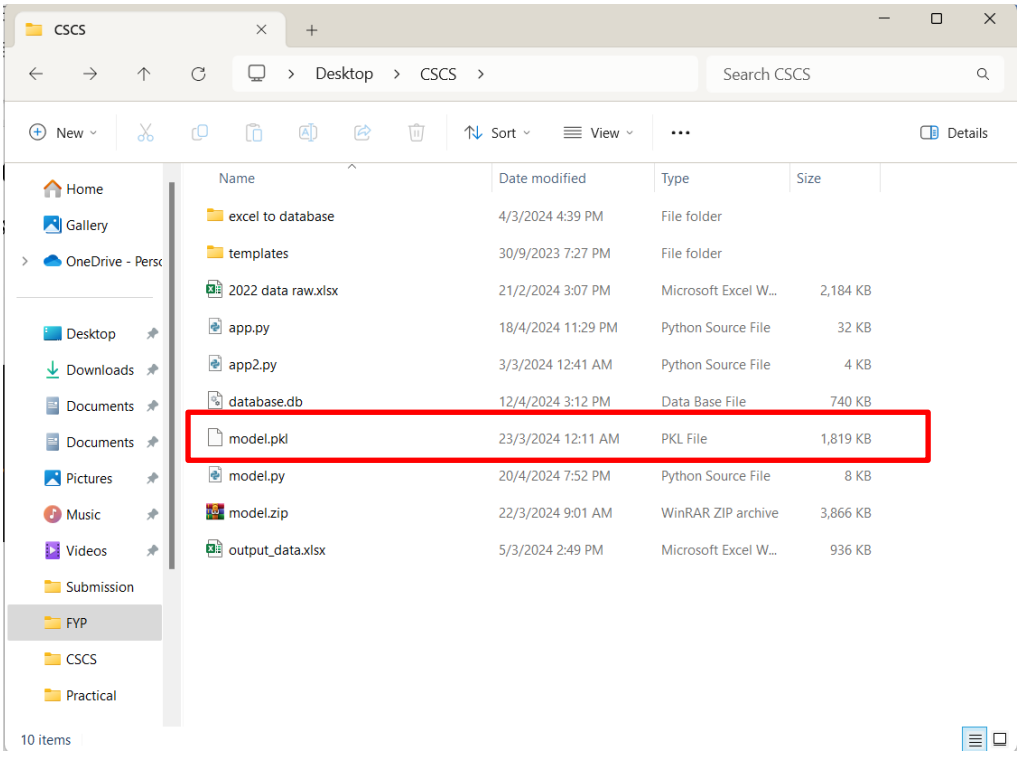


Figure 6.15: Generated model.pkl.

6.2.6 Model Evaluation

Model evaluation is also one of the important steps in the CNN model development to assess its performance and ensure the model is reliable as well

as achieve certain level of accuracy. First of all, the evaluate model is used to evaluate and calculate the model accuracy regarding on the unseen test data.

```
# model evaluation
test_loss, test_accuracy = model.evaluate(X_test, y_test, verbose=1)
print(f"Test Accuracy: {test_accuracy*100:.2f}%")
```

Figure 6.16: Model Evaluation Using evaluate Method.

From the result, the test accuracy achieves 95.30% indicating that the CNN model has high accuracy on the test dataset or unseen data. This suggests that the CNN model has learned to effectively extract features from the input data and make reliable predictions. Therefore, overally, the model has strong performance in the classification tasks with the test accuracy of 95.30%.

```
119/119 [==============================] - 0s 2ms/step - loss: 0.1253 - accuracy: 0.9530
Test Accuracy: 95.30%
```

Figure 6.17: Model Evaluation Result – Test Accuracy.

Even though the accuracy of the CNN model is calculated and assessed, evaluate some additional performance metrics such as precision, recall and F1-score can provide additional insights into the model's performance, particularly its ability to correctly classify instances across different classes. To assess the precision, recall and F1-score, the classification_report function is used to generate the comprehensive classification report with a more detailed assessment of model's performance beyond just accuracy.

```
# Predict the test set labels
y_pred = np.argmax(model.predict(X_test), axis=-1)

# Generate classification report
class_names = ['Class 1', 'Class 2', 'Class 3', 'Class 4']
print(classification_report(y_test, y_pred, target_names=class_names))
```

Figure 6.18: Model Evaluation Using Classification Report.

Figure below demonstrates the detailed CNN model's performance using the classification report.

```
119/119 [==============================] - 0s 2ms/step
              precision    recall  f1-score   support

     Class 1       0.97      0.99      0.98      3550
     Class 2       0.40      0.35      0.38        17
     Class 3       0.43      0.11      0.17        28
     Class 4       0.71      0.54      0.61       213

    accuracy                           0.95      3808
   macro avg       0.63      0.50      0.53      3808
weighted avg       0.95      0.95      0.95      3808
```

Figure 6.19: Model Evaluation Result – Classification Report.

From the classification report, the precision represents the ratio of correctly predicted positive observations to the total predicted positives. For example, class 1 has 0.97 precision indicates that 97% of the instances predicted as Class 1 were actually Class 1; class 2 has 0.40 precision indicates that 40% of the instances predicted as Class 2 were actually Class 2; so on and so forth. Likewise, the recall (or sensitivity) is the ratio of correctly predicted positive observations to all the observations in actual class. For instances, class 1 has 0.99 recall indicates that 99% of the actual Class 1 instances were correctly predicted; class 2 has 0.35 recall indicates that 35% of the actual Class 2 instances were correctly predicted; so on and so forth. From the support which indicates the number of actual occurrences of the class in the dataset, it is obvious that there is an imbalance nature as the support of Class 2 and Class 3 are quite low. Thus, the F1-score is useful to measure the test's accuracy in this imbalanced scenario. Based on the interpretation, even though the CNN model has a 95.30% accuracy overally, the model performs well for Class 1 (0.98 f1-score), reasonably for Class 4 (0.61 fl-score), but slightly poorly for Class 2 (0.38 f1-score) and Class 3 (0.17 fl-score), likely due to the low support and imbalanced nature factors.

Furthermore, the confusion matrix is constructed to show the tabular representation of the model's predictions compared to the actual labels across different classes. Using the confusion matrix, it is easy to evaluate the model's

performance, visualize the predictions and identify the class imbalance. Figure below shows the model evaluation using the confusion matrix and its result.

```
# Compute confusion matrix
conf_mat = confusion_matrix(y_test, y_pred)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

Figure 6.20: Model Evaluation Using Confusion Matrix.

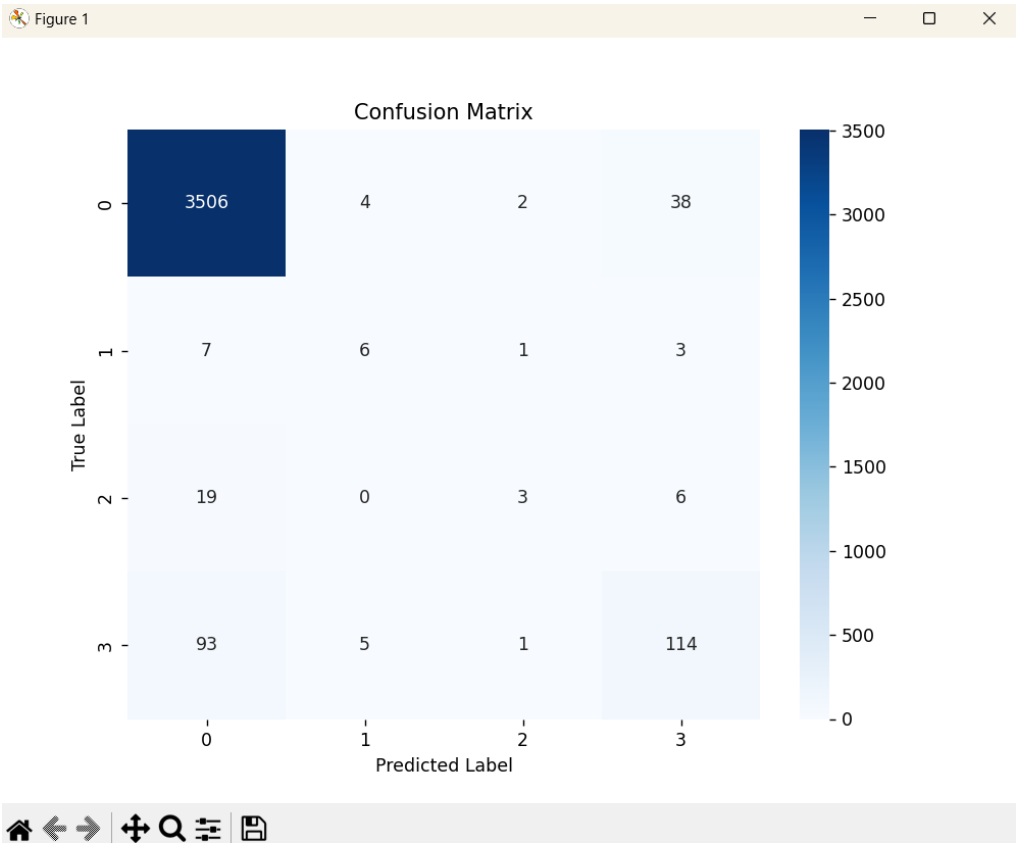


Figure 6.21: Model Evaluation Result – Confusion Matrix

6.3 Database Preparation

To store all the data and information that needed within the mobile application, the database system, SQLite is adopted. It serves as the data storage part, integrating with the Flask server, promoting the robust backend system for the mobile system. First of all, the SQLite database is initialized at the backend Flask server so that it can store the necessary data required in creating the mobile application.

```
def initialize_database():
    conn = sqlite3.connect(DB_PATH)
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS employee (
            employeeID TEXT PRIMARY KEY,
            username TEXT NOT NULL,
            fullname TEXT NOT NULL,
            email TEXT NOT NULL UNIQUE,
            password_hash TEXT NOT NULL,
            selectedAvatar TEXT DEFAULT 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRKSEuKKwqIqbJH-NRiDHlu(
        )
    ''')

    c.execute('''
        CREATE TABLE IF NOT EXISTS processing_summary (
            processing_id TEXT PRIMARY KEY,
            processing_time DATETIME NOT NULL,
            processing_grade TEXT NOT NULL,
            employeeID TEXT NOT NULL,
            FOREIGN KEY (employeeID) REFERENCES employee(employeeID)
        )
    ''')
```

Figure 6.22: Initialise Database Using SQL Command.

The figure above shows that the two necessary tables (employee, processing_summary) are initialised and created with respective table columns using the SQL command. In this point, the data within the mobile application can be successfully stored inside the table respectively.


```
{
  "employees": [
    {
      "email": "wb@gmail.com",
      "employeeID": "EMP777",
      "fullname": "WEI BIN",
      "selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSgOEdgA9z6JfuAGPbEOaBQeF6QShIIqFTsBw&usqp=CAU",
      "username": "wb02"
    },
    {
      "email": "wb@gmail.comm",
      "employeeID": "EMP001",
      "fullname": "Carly",
      "selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTun6bB9mPzGhmCIuMhMv00hc9qgDQClQ1YI4SVbmG5fly1B8tZBMZ6y5xwrHXPvIfDOcs&usqp=CAU",
      "username": "carly123"
    },
    {
      "email": "testacc@gmail.com",
      "employeeID": "EMP002",
      "fullname": "TEST ACC",
      "selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRksES88c6suZ8F_YP0erB8tMAhAvpsBKHRIQ&usqp=CAU",
      "username": "wbtest"
    }
  ]
}
```


Figure 6.23: Example of Records in ‘employee’ Table.

```
{
  "processing_summary": [
    {
      "employeeID": "EMP777",
      "processing_grade": "1",
      "processing_id": "P001",
      "processing_time": "2024-03-02 21:55:45",
      "username": "wb02"
    },
    {
      "employeeID": "EMP777",
      "processing_grade": "1",
      "processing_id": "P002",
      "processing_time": "2024-03-02 23:50:25",
      "username": "wb02"
    },
    {
      "employeeID": "EMP777",
      "processing_grade": "1",
      "processing_id": "P003",
      "processing_time": "2024-03-02 23:54:27",
      "username": "wb02"
    },
    {
      "employeeID": "EMP777",
      "processing_grade": "1",
      "processing_id": "P004",
      "processing_time": "2024-03-03 00:04:00",
      "username": "wb02"
    },
```


Figure 6.24: Example of Records in ‘processing_summary’ Table.

Another table named ‘data’ required in the mobile application is constructed by converting the existing Excel file into the SQLite table. Figure below shows the snippet code of converting the Excel file into SQLite table.

```
def convert_excel_to_sqlite(DB_PATH, excel_info_table_name):
    # Read the Excel file into a pandas DataFrame
    data = process_data()

    # Connect to the SQLite database
    conn = sqlite3.connect(DB_PATH)

    # Convert the DataFrame to an SQLite table
    data.to_sql(excel_info_table_name, conn, if_exists='replace', index=False)

    # Close the connection
    conn.close()
```

Figure 6.25: Snippet Code of Converting Excel File into SQLite Table.

After converting, all the data columns and rows inside the Excel file will be stored into the ‘data’ table in terms of the table columns and records. Each record will have the table columns such as AREA_A, AREA_B, DATE, INSPDISP, RATE_A, LOC_A_HEIGHT and so on.


```
{
  "data": [
    {
      "AREA_A": 100.0,
      "AREA_B": 100.0,
      "AREA_C": 100.0,
      "AREA_D": 100.0,
      "AREA_E": 100.0,
      "DATE": 20220101,
      "INSPDISP": 1,
      "LOC_A_HEIGHT": "D",
      "LOC_A_LENGTH": "H",
      "LOC_A_WIDTH": "C",
      "LOC_B_HEIGHT": "D",
      "LOC_B_LENGTH": "H",
      "LOC_B_WIDTH": "C",
      "LOC_C_HEIGHT": "D",
      "LOC_C_LENGTH": "H",
      "LOC_C_WIDTH": "A",
      "LOC_D_HEIGHT": "D",
      "LOC_D_LENGTH": "H",
      "LOC_D_WIDTH": "F",
      "LOC_E_HEIGHT": "T",
      "LOC_E_LENGTH": "H",
      "LOC_E_WIDTH": "F",
      "RATE_A": "L",
      "RATE_B": "L",
      "RATE_C": "L",
      "RATE_D": "L",
      "RATE_E": "L"
    },
    {
      "AREA_A": 100.0,
      "AREA_B": 100.0,
      "AREA_C": 100.0,
      "AREA_D": 100.0,
      "AREA_E": 100.0,
      "DATE": 20220101,
      "INSPDISP": 1,
      "LOC_A_HEIGHT": "D",
      "LOC_A_LENGTH": "H",
      "LOC_A_WIDTH": "C",
      "LOC_B_HEIGHT": "D",
      "LOC_B_LENGTH": "H"
```


Figure 6.26: Example of Records in ‘data’ Table.

6.4 Backend Flask Server Development

To manage the business logic and assure the functionality within the mobile application running smoothly, a robust backend server, Flask application is built up. In short, this Flask server handles the use of the trained CNN model for the prediction task, define functions required to handle the business logic and define the API routes that the frontend can call to do some data and logic processing.

6.4.1 Trained CNN Model Adoption

To make use of the trained model in the Flask server, the model.pkl file generated in the earlier stage is loaded in and the predict function is called for the prediction task. In this way, the trained CNN model is utilized and successfully integrated into the Flask server.

```
# Load the pickle model
model = pickle.load(open("model.pkl", "rb"))
```

Figure 6.27: Load the Pickle File into the Flask Server.

```
prediction = model.predict(features)
```

Figure 6.28: Make the Prediction Using the Trained Model.

6.4.2 Functions for Business Logic

Some functions are defined and created inside the Flask server to handle the business logic part. These functions or definitions will receive some parameters as input or / and return some values as output if necessary.

Table 6.1: Functions Defined in Flask server.

Function	Descriptions	Input Parameter(s)	Value(s) Returned
initialize_database()	Create the necessary database tables if do not exist when the mobile application is launched	N/A	N/A
initialize()	Called the initialize_database() function and return the message if created successful	N/A	{‘message: Database Initialized’}
get_employees()	Get all the employees’ (application user) information	N/A	employeeID username fullname email selectedAvatar
insert_user()	Insert user data into database	employeeID username fullname email password	N/A

employeeID_exists()	Check if employeeID has already registered in ‘employee’ table	employeeID	exists
email_exists()	Check if email has already registered in ‘employee’ table	email	exists
username_exists()	Check if username has already registered in ‘employee’ table	username	exists
email_exists_ when_updating()	Check if email has already registered in ‘employee’ table when updating the user profile	employeeID email	exists
username_exists_ when_updating()	Check if username has already registered in ‘employee’ table when updating the user profile	employeeID username	exists
signup()	Help the user to sign up a new profile account	N/A	{‘message’: ‘Account Created Successfully’}
verify_credentials()	Verify the user identity when user login the application	employeeID password	N/A
login()	Help the user to login to the application	N/A	{‘message’: ‘Login Successfully’} OR

			{‘message’: ‘Invalid EmployeeID or Password’}
get_employee_info()	Obtain the specific employee’s information	employeeID	employeeID username fullname email selectedAvatar OR {‘message’: ‘Employee Not Found’}
update_profile()	Help the user to update the user profile	N/A	{‘message’: ‘Profile Updated Successfully’}
insert_processing _summary()	Insert a processing record into the ‘processing_summary’ table	processing_id employeeID processing_time processing_grade	N/A
predict()	Help user to predict the steel grade	N/A	processing_grade

get_processing_summary()	Get the all data in the ‘processing summary’ table	N/A	processing_summary _list
get_dashboard_processing _summary()	Get the ‘processing_grade’ data from the ‘processing summary’ table, then calculate the total processing for each grade and the total sum number of processing for all grade	N/A	grade_count total_processing
process_data()	Carried out the data preprocessing on the Excel file data (ease the conversion from Excel file data to SQLite table)	N/A	processed_data
convert_excel_to_sqlite()	Convert the Excel file data into SQLite table	database_path table_name	N/A
convert_data()	Start the conversion process	N/A	N/A
get_total_records()	Get the total records in the ‘data’ table	N/A	total_count
get_total_by_grade()	Get the total records for each grade in the ‘data’ table	N/A	total_count_1 total_count_2 total_count_3 total_count_4
get_grade_count_ by_month()	Get the total number of grade count for each grade in each month	N/A	each_grade_ count_per_month

get_AIPrediction_ by_employee()	Get the ‘processing_grade’ data from the ‘processing summary’ table, then calculate the total processing for each grade and the total sum number of processing for all grade for specific employee	employeeID	grade_count total_processing
get_total_processing _by_employee	Get the number of total processing processed by each employee	N/A	total_processing username
total_processing _by_employee	Execute the function to compare the total number of processing by each of the employee in the database	N/A	total_processing username

6.4.3 API Endpoint Definition

For ensuring the frontend application can utilize the functions defined at the backend Flask server to process the business logic, the API routes are defined to act as the intermediary in which the functions will be executed whenever the API routes are called.

Table 6.2: API Routes Definition.

API Route	Method	Description	Required Parameters	Request Data	Response Data
/initialize	GET	Initialize the database to create the necessary tables if the table does not exist	-	-	{‘message’: ‘Database Initialized’}
/employees	GET	Get all the employees’ information in the ‘employee’ table	-	-	{‘employees’: ‘employees_data’}
/signup	POST	Let the user to sign up a new profile account	-	employeeID username fullname email password	{‘ message’: ‘Account Created Successfully’}
/login	POST	Let the user to login to the application	-	employeeID password	{‘ message’: ‘Login Successfully’} OR

					{‘ message’: ‘Invalid employeeID or password’}
/employee/<employeeID>	GET	Obtain the specific employee’s information	employeeID	-	{‘message’: ‘Employee not found’}
/update_profile	PUT	Let the user to update their user profile	-	employeeID username fullname email selectedAvatar	{‘message’: ‘Profile Updated Successfully’}
/predict	POST	Let user to predict the steel grade by utilizing the trained CNN model	-	defects employeeID	{‘prediction’: ‘prediction_y_serializable’}
/processing_summary	GET	Get the all data in the ‘processing summary’ table	-	-	{‘processing_summary’: ‘processing_summary_list’}
/dashboard_processing _summary	GET	Get the total processing for each grade and the total sum number of processing for all grade in the ‘processing_summary’ data	-	-	{‘total_processing’: total_processing, ‘grade_counts’: grade_counts}

/convert_data	GET	Convert the Excel file data into SQLite table	-	-	{‘message’: ‘Data Converted Successfully’}
/total_records	GET	Get the total records in the ‘data’ table	-	-	{‘total_count’: total_count}
/total_by_grade_in _experts_prediction	GET	Get the total records for each grade in the ‘data’ table	-	-	total_by_grade_dict
/grade_count_by_month	GET	Get the total number of grade count for each grade in each month	-	-	grade_count_dict
/AIPredictionByEmployee/ <employeeID>	GET	Get the total processing for each grade and the total sum number of processing for all grade in the ‘processing_summary’ table for specific employee	employeeID	-	{‘total_processing’: total_processing, ‘grade_counts’: grade_counts}
/total_processing_ for_each_employee	GET	Get the number of total processing processed by each employee	-	-	{‘total_processing _by_employee’: result}

6.5 Communication Between Frontend and Backend

After the successful implementation and development of the backend system, it is vital to define a method that facilitates the communication between the frontend part and backend part. To connect the backend part, the API endpoints constructed in the earlier stage are called in the frontend part. In other words, the fetch function is utilized at the frontend part to summon the particular API routes along with some request data if necessary, and the API routes will execute the corresponding functions to handle the specific business logic, then send back the response generated to the frontend again.

```
fetch('http://192.168.0.16:5000/predict',{
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({...data, employeeID: employeeID}),
  })
.then(response => {
    if (!response.ok) {
        throw new Error('Network response faces problems. Please check the server.');
    }
    return response.json();
})
.then(data => {
    const predictionResult = data.prediction;
    this.setState({predictionResult})

    this.setState({
        defects: [
```

Figure 6.29: fetch Function To Call the API Endpoint.

In this case, the frontend JavaScript code is making an asynchronous HTTP POST request to the backend Flask server endpoint to get the prediction result. The ‘fetch’ function is used to initialise the request to the ‘/predict’ API endpoint. Since it is a POST request, it has a request body for sending the frontend data (input from user) to the backend for processing. After the Flask server has processed the request, the response data generated will be sent back to the frontend by using the ‘then’ method. Now, the response from the server will exist in the frontend part. Therefore, this is the way that the frontend part can always communicate with the backend server via the API routes, fetch function and also the promise chaining (.then) method.

6.6 Frontend Development

The frontend UI part is built up using a combination of the React Native components and the JavaScript functions or methods. React Native provides a set of pre-built UI components that are similar to HTML elements. In React Native, these components such as View, Button, Text, TextInput and so on are utilized to define the structure and layout of the mobile application UI. The state management and lifecycle components in React Native are also used to manage the dynamic behaviour and lifecycle events of the application. Furthermore, the JavaScript functions are used to define the logic and behaviour of the mobile application in frontend. These functions are used to handle the user interactions, perform data fetching, control the application flow and etc. Following the prototype designed in Figma, the frontend UI part can be developed and implemented to provide a seamless user experience to the user.

6.6.1 Welcome Screen

This screen is displayed once the user starts the mobile application. It lets the user sign up a new profile account or sign in the application.

Figure 6.30: Welcome Screen.

6.6.2 Sign Up Screen

This screen lets the user sign up a new profile account for the application. If the text input is empty or the information entered by the user is invalid, the application prompts the user to re-enter again. Also, if the employeeID, username or email has already registered, the application prompts the user to re-enter again. Once the user fills in all the information correctly, the application displays the message indicating the sign up process has completed.

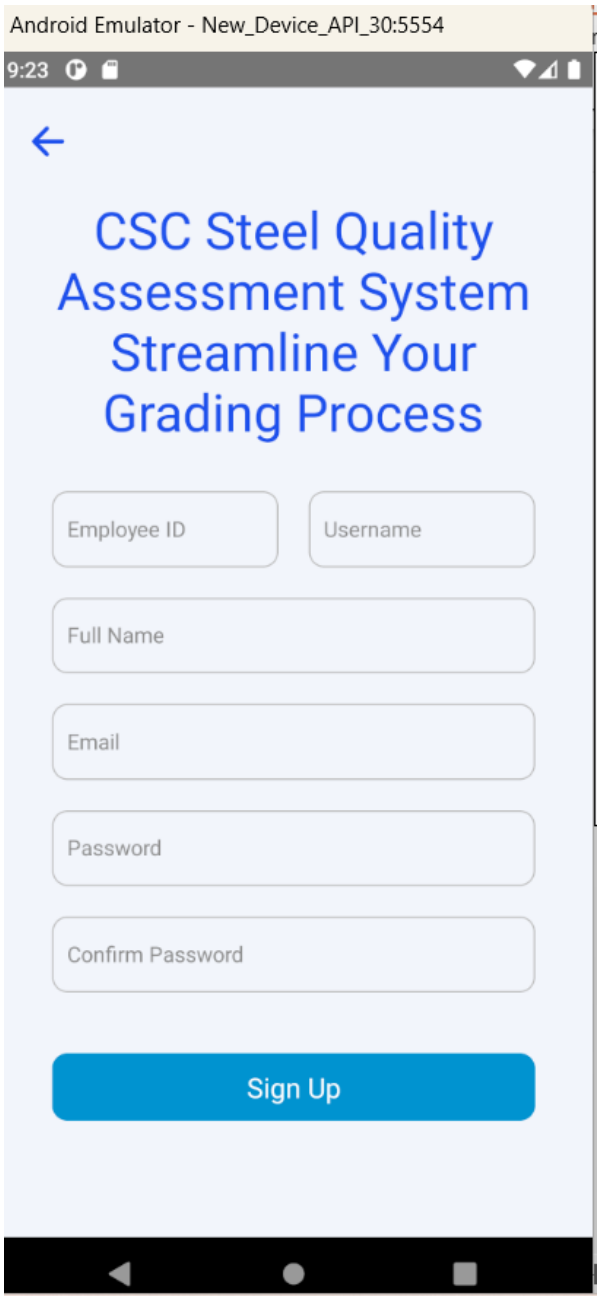


Figure 6.31: Sign Up Screen.

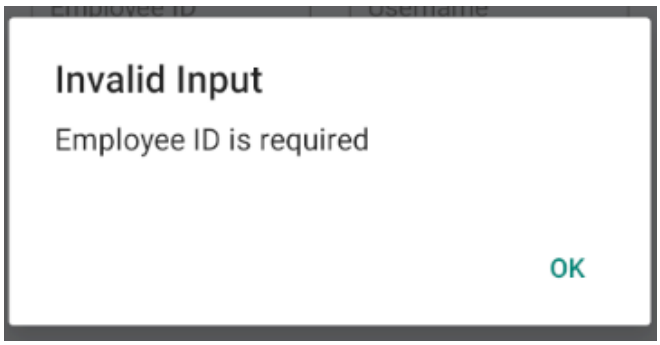


Figure 6.32: Example Of Error Message – Empty Input.

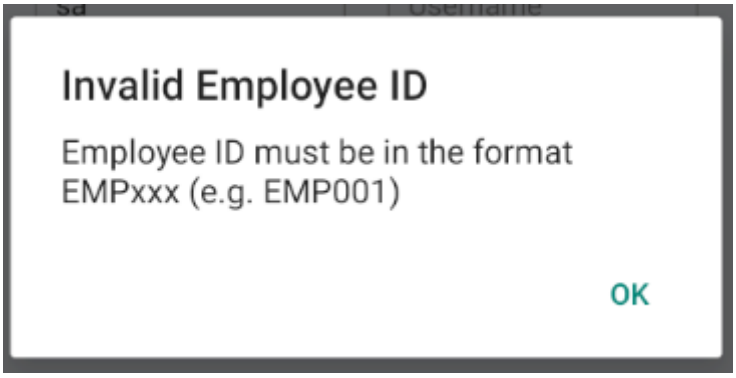


Figure 6.33: Example Of Error Message – Invalid Input.

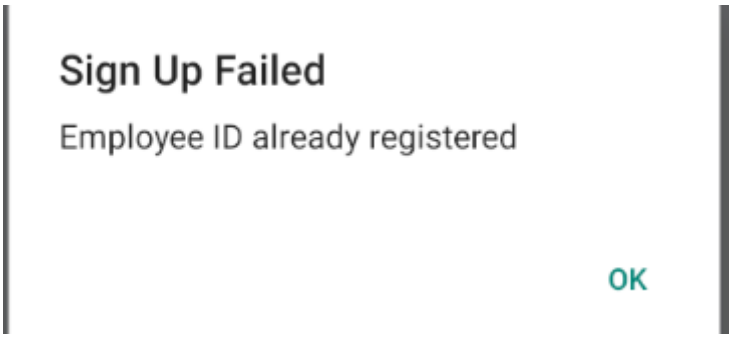


Figure 6.34: Example Of Error Message – Record Existed.

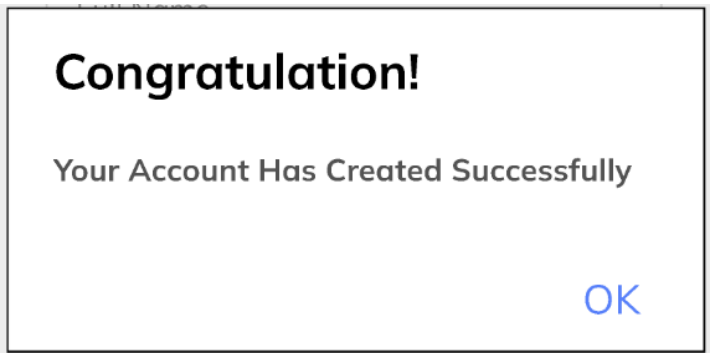


Figure 6.35: Successful Sign Up.

6.6.3 Sign In Screen

This screen lets the user sign into the application using their registered profile account. Similarly, if the text input is empty or the employeeID entered by the user is invalid, the application prompts the user to re-enter again. If the employeeID and password are not matching, the application blocks the user from successful login and prompts the user to re-enter.

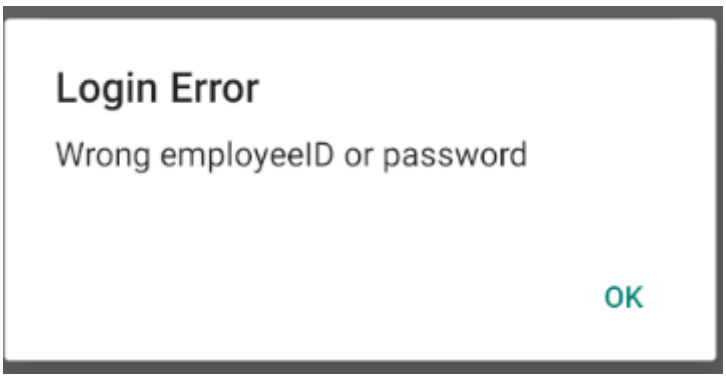


Figure 6.36: Failed Login.

6.6.4 Profile Screen

This screen displays the user information and consists of an edit button at the top right corner to let the user edit the profile information if necessary.

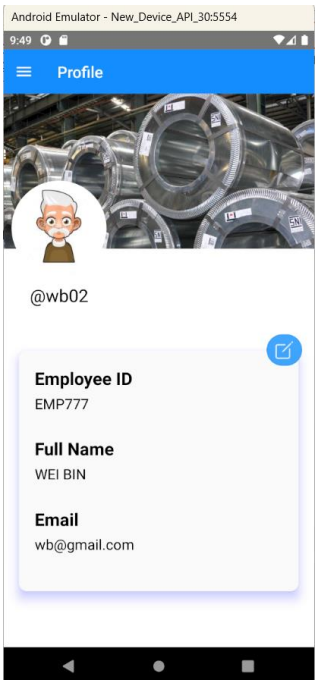


Figure 6.37: Profile Screen.

6.6.5 Navigation Drawer

This drawer serves as the intermediary to navigate the user to different screens for carrying out corresponding functionalities in the application.

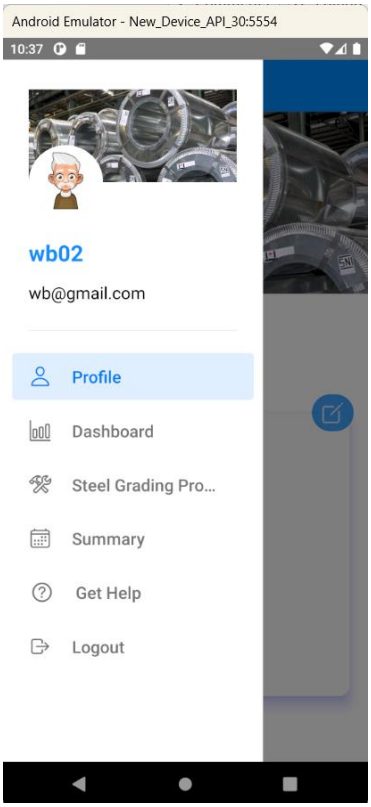


Figure 6.38: Navigation Drawer.

6.6.6 Edit Profile Screen

This edit profile screen lets the user edit the profile information if necessary. By clicking the three small dots beside the avatar picture, the application navigates the user to choose the desired avatar picture. If the text input is empty or the information entered by the user is invalid, the application prompts the user to re-enter again. Also, if the username or email has already registered, the application prompts the user to re-enter again.

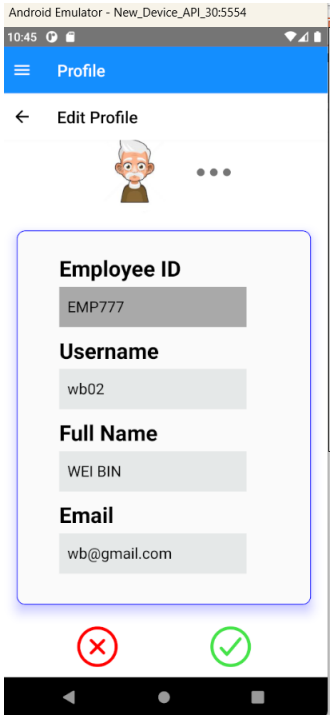


Figure 6.39: Edit Profile Screen.

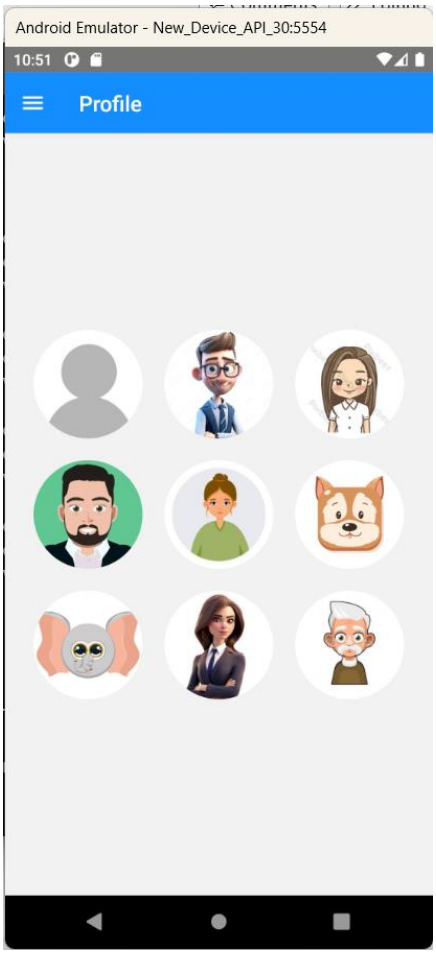


Figure 6.40: Avatar Picture Selection Screen.

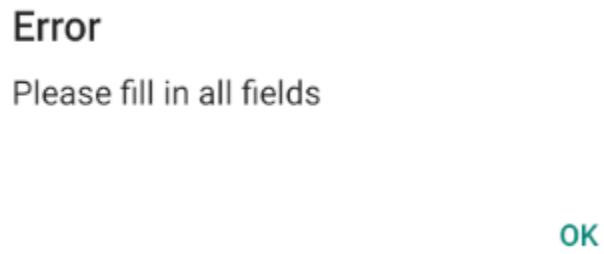


Figure 6.41: Error Message When Updating – Empty Field.

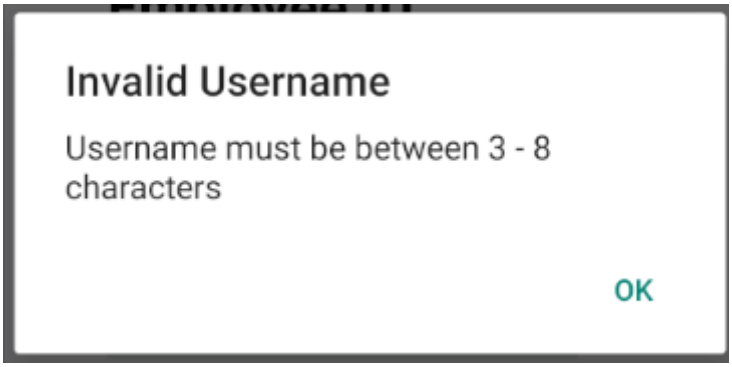


Figure 6.42: Error Message When Updating – Invalid Input.

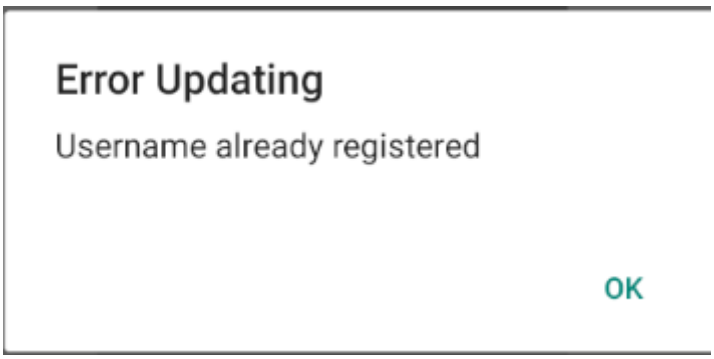


Figure 6.43: Error Message When Updating – Record Existed.

6.6.7 Dashboard Screen

This dashboard screen showcases the information related to the steel product so that the user can gain insights into it and make better business decisions. For example, the overview of the steel processing predicted by the trained CNN model, the information concluded from the Excel file data (‘data’ table in database) which represents the steel processing evaluated by the experts, different kinds of charts and so on.

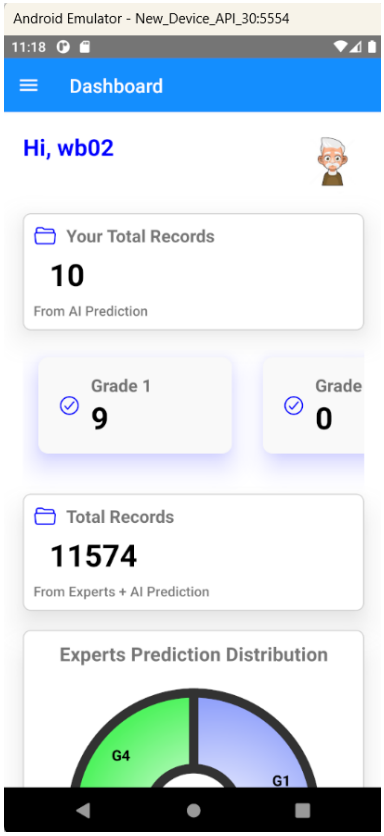


Figure 6.44: Example Of Dashboard Screen.

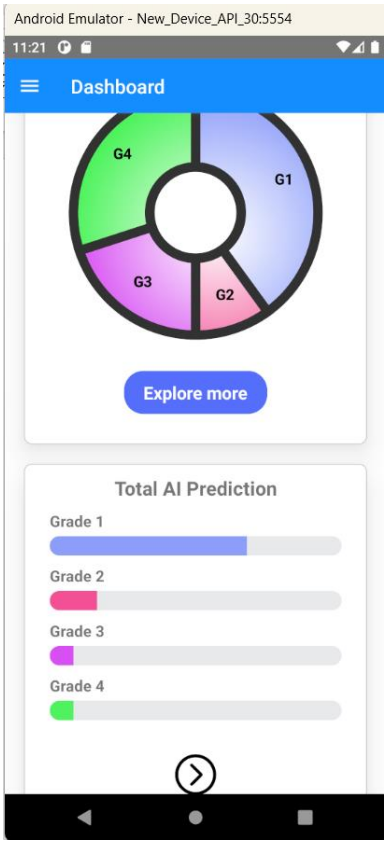


Figure 6.45: Example Of Dashboard Screen.

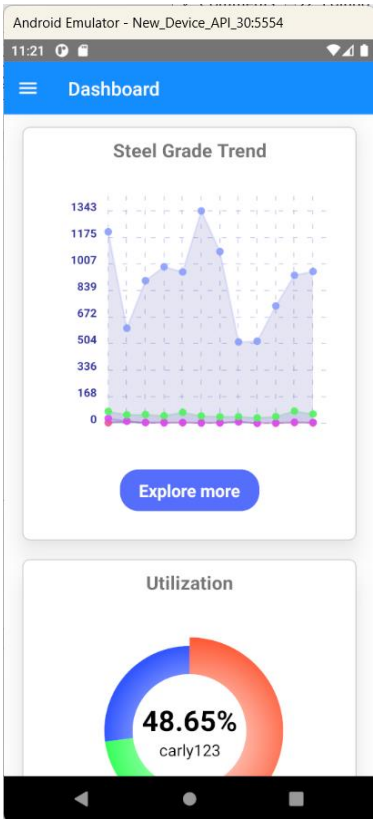


Figure 6.46: Example Of Dashboard Screen.

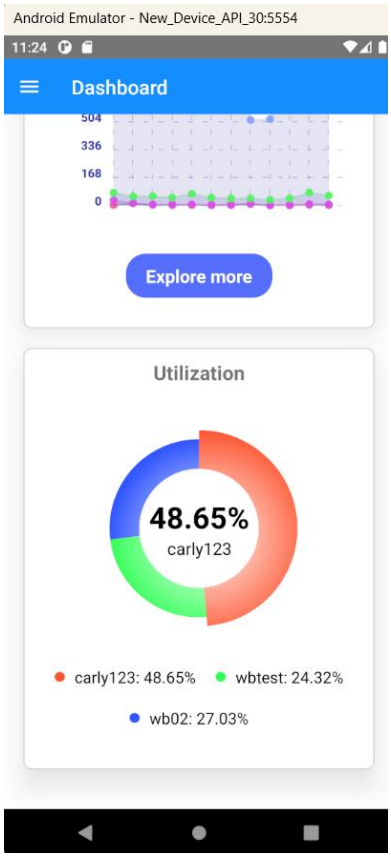


Figure 6.47: Example Of Dashboard Screen.

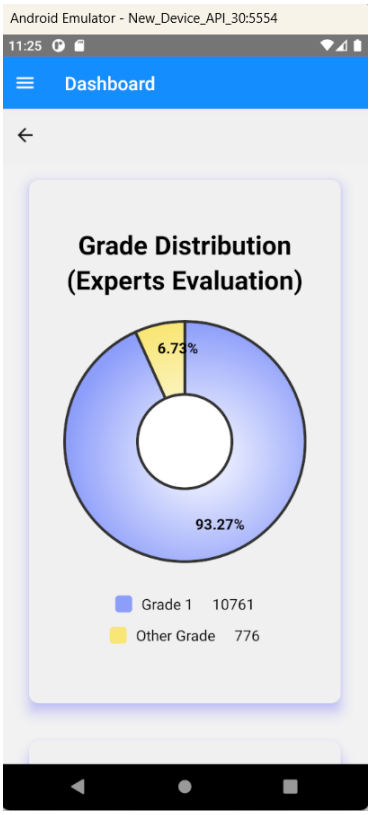


Figure 6.48: Example Of Dashboard Screen.

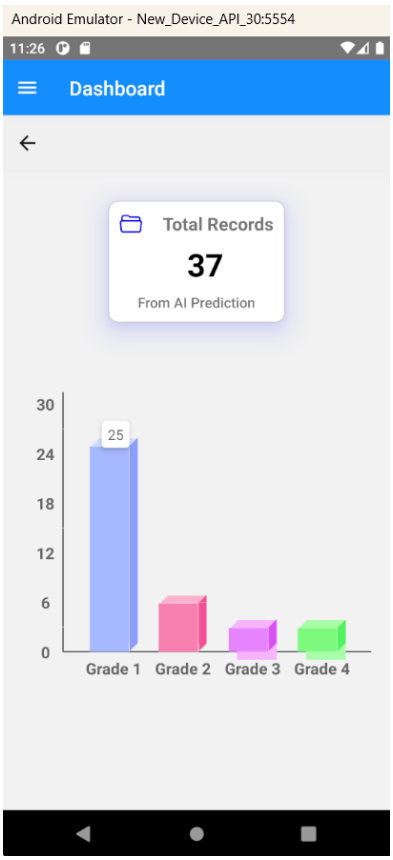


Figure 6.49: Example Of Dashboard Screen.

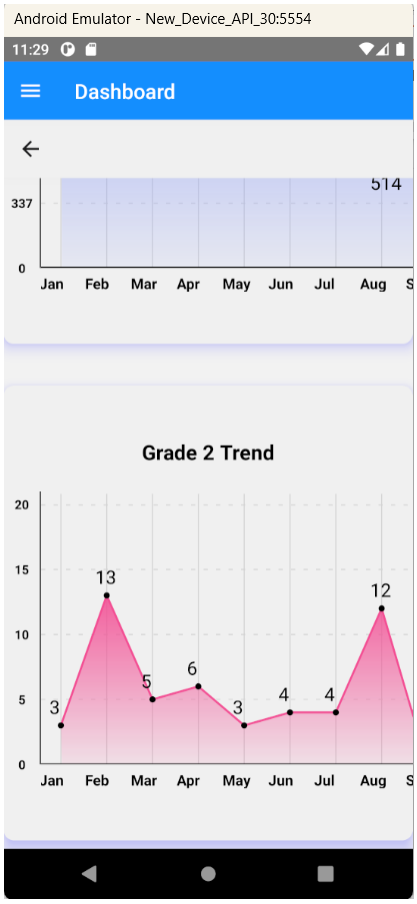


Figure 6.50: Example Of Dashboard Screen.

6.6.8 Steel Grading Processing Screen

This screen provides the interface that lets the user evaluate the grade of the steel product by the help of the trained CNN model. The user can select the input parameter values from the dropdown menu, fill in the necessary text field, and click the Predict button to assess the grade of the steel product. If the text field is empty or the text field is filled by invalid value, the application prompts the user to re-enter.

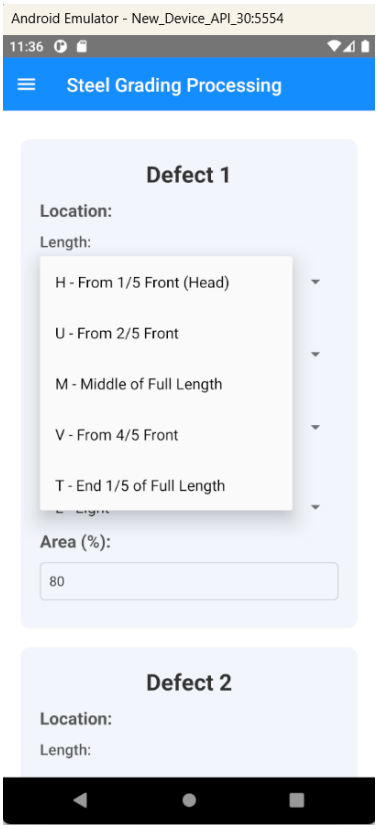


Figure 6.51: Steel Grading Processing Screen.

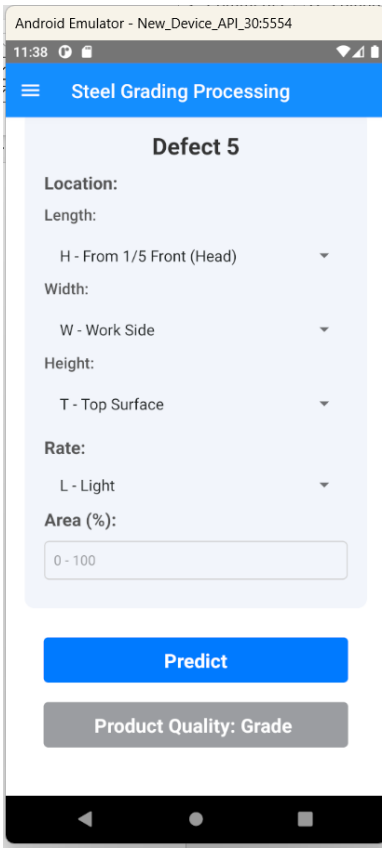


Figure 6.52: Steel Grading Processing Screen – Predict.

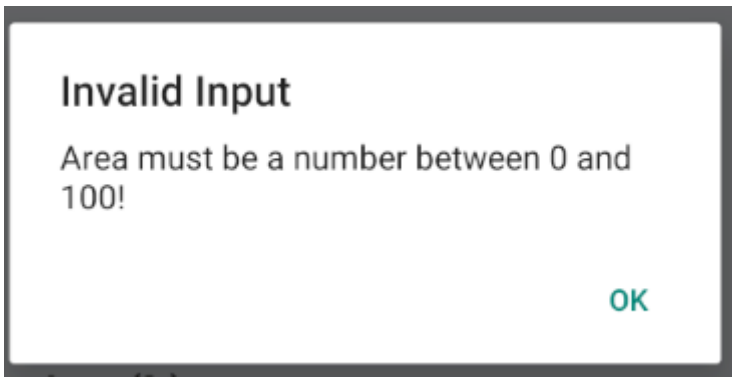


Figure 6.53: Error Message In Steel Grading Processing – Invalid Input.

6.6.9 Summary Screen

This screen gives an overview to the user about the records on the steel product grading processing. Once there is a steel grading processing, there will be a record inserted into this summary.

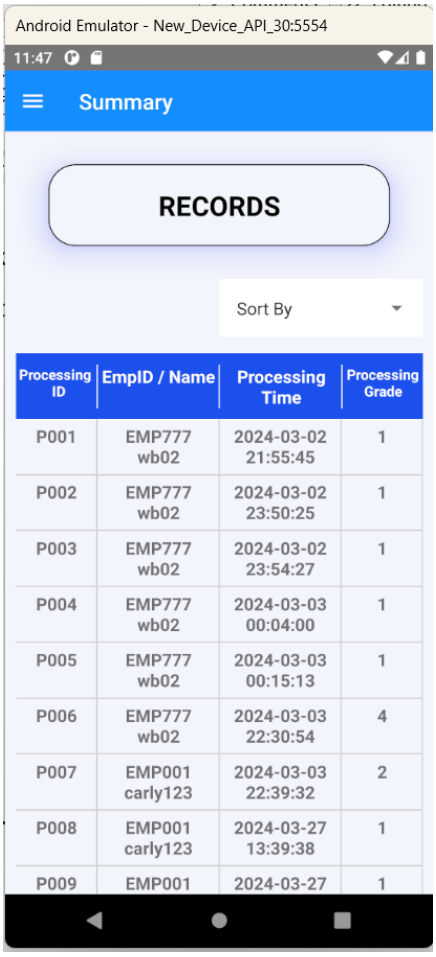


Figure 6.54: Summary Screen.

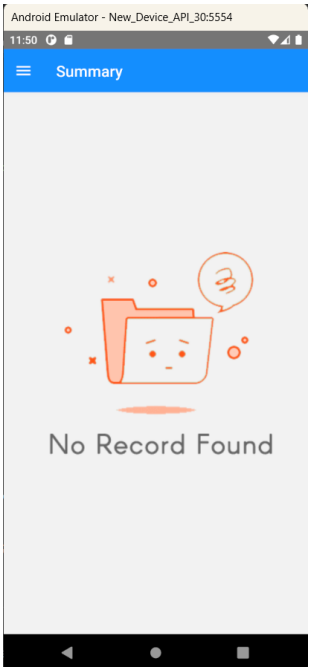


Figure 6.55: Summary Screen When No Records Yet.

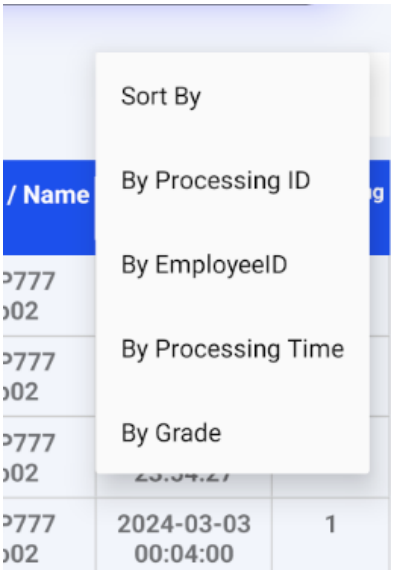


Figure 6.56: Sorting DropDown Menu In Summary Screen.

6.6.10 Get Help Screen

This screen gives some brief information and ideas to the new user so that the user can get familiar with the application as soon as possible. It consists of the application users' information as well as what each core functionality in this application does and how each of it works.

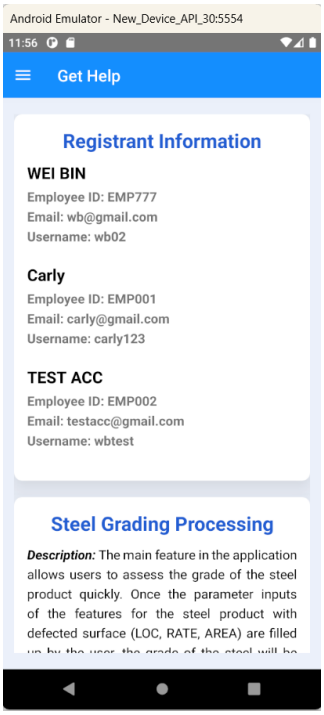


Figure 6.57: Example Of Get Help Screen.

6.6.11 Logout Screen

This screen lets the user logout the application after the user has completed tasks using this application. Once logged out, the user will be redirected back to the welcome screen.

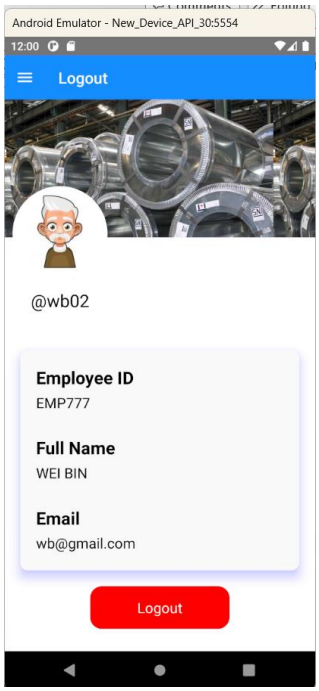


Figure 6.58: Logout Screen.

6.7 Summary

All in all, this chapter had discussed and highlighted the development of the mobile application in the aspect of both frontend and backend. For the frontend part, the UI had been developed using the React Native components as well as the well-known frontend programming language, JavaScript. As for the backend part, the CNN model had been trained, the SQLite database had been created, and the Flask application had been developed to serve as the robust backend server, handling the business logic and the requests coming from the frontend part. Lastly, the API endpoints had also been designed to facilitate the communication between frontend and backend.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

This chapter emphasizes the system testing part after the system implementation has been completed. It assures the mobile application always aligns with the functional requirements and specifications, and can work as expected without any errors and issues. For this mobile application, different types of testing such as unit testing, API testing, system usability testing and user acceptance testing (UAT) are performed to ensure the mobile application is robust and bug-free. The testing phase is starting with the test entry that the development of the application is completed and the testing environment is ready. On the other hand, the testing will be stopped or the exit criteria is that all the testing has been carried out and the passing rate of all the tests achieves at least 95%. Lastly, the traceability matrix is constructed to connect the use cases, functional requirements, unit testing test cases, API testing test cases and user acceptance testing test cases so that none of the functionality in the mobile application will be missed out for testing.

7.2 Unit Testing

7.2.1 Sign Up Account – Invalid Inputs

Table 7.1: Unit Test Case For Sign Up Account – Invalid Inputs.

Test Case ID	UT001				
Test Case Name	Unit Test Case for Sign Up Account using Invalid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Enter invalid format employeeID in the text field	1. Enter invalid format employeeID 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “test123” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that employeeID is invalid	Error message alerting that employeeID is invalid	PASS
Empty the employeeID text field	1. Empty the employeeID text field 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456”	Error message alerting that employeeID is required	Error message alerting that employeeID is required	PASS

		confirmPassword = “123456”			
Enter invalid format username in the text field	1. Enter invalid format username 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “longlongname” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that username is invalid	Error message alerting that username is invalid	PASS
Empty the username text field	1. Empty the username text field 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that username is required	Error message alerting that username is required	PASS
Empty the full name text field	1. Empty the full name text field 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “” email = “weibin@gmail.com” password = “123456”	Error message alerting that full name is required	Error message alerting that full name is required	PASS

		confirmPassword = “123456”			
Enter invalid format email in the text field	1. Enter invalid format email 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibinGmail” password = “123456” confirmPassword = “123456”	Error message alerting that email is invalid	Error message alerting that email is invalid	PASS
Empty the email text field	1. Empty the email text field 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “” password = “123456” confirmPassword = “123456”	Error message alerting that email is required	Error message alerting that email is required	PASS
Empty the password text field	1. Empty the password text field 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “”	Error message alerting that password is required	Error message alerting that password is required	PASS

		confirmPassword = “123456”			
Enter the confirm password text field with the value that different in the password text field	1. Fill in all text fields with valid input 2. Enter the password 3. Enter the confirm password 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “654321”	Error message alerting that passwords do not match	Error message alerting that passwords do not match	PASS

7.2.2 Sign Up Account – Duplicate Records

Table 7.2: Unit Test Case For Sign Up Account – Duplicate Records.

Test Case ID	UT002				
Test Case Name	Unit Test Case for Sign Up Account using Existing Records in Database				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Enter registered employeeID in the text field	1. Enter registered employeeID 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that employeeID has already registered	Error message alerting that employeeID has already registered	PASS
Enter registered username in the text field	1. Enter registered username 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that username has already registered	Error message alerting that username has already registered	PASS

Enter registered email in the text field	1. Enter registered email 2. Fill in other text fields with valid input 3. Click Sign Up button	employeeID = “EMP001” username = “wb02” fullname = “WEIBIN” email = “weibin@gmail.com” password = “123456” confirmPassword = “123456”	Error message alerting that email has already registered	Error message alerting that email has already registered	PASS

7.2.3 Sign Up Account – Valid Inputs and Not Duplicate Records

Table 7.3: Unit Test Case For Sign Up Account – Valid Inputs and Not Duplicate Records.

Test Case ID	UT003				
Test Case Name	Unit Test Case for Sign Up Account using Valid Inputs and Not Duplicate Records				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Fill in all the necessary text fields with valid inputs and not duplicate records	1. Fill in all text fields with valid input 2. Click Sign Up button	employeeID = “EMP007” username = “newtest” fullname = “TESTING” email = “test@gmail.com” password = “123456” confirmPassword = “123456”	Successful sign up message and redirect to login screen	Successful sign up message and redirect to login screen	PASS

7.2.4 Login Account – Invalid Inputs

Table 7.4: Unit Test Case For Login Account – Invalid Inputs.

Test Case ID	UT004				
Test Case Name	Unit Test Case for Login Account using Invalid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Enter invalid format employeeID in the text field	1. Enter invalid format employeeID 2. Fill in other text fields with valid input 3. Click Login button	employeeID = “test123” password = “123456”	Error message alerting that employeeID is invalid	Error message alerting that employeeID is invalid	PASS
Empty the employeeID text field	1. Empty the employeeID text field 2. Fill in other text fields with valid input 3. Click Login button	employeeID = “” password = “123456”	Error message alerting that employeeID is required	Error message alerting that employeeID is required	PASS

Empty the password text field	1. Empty the password text field 2. Fill in other text fields with valid input 3. Click Login button	employeeID = “EMP001” password = “”	Error message alerting that password is required	Error message alerting that password is required	PASS
Enter the wrong employeeID or password that do not have the matching record in database	1. Fill in all text fields with valid input 2. Click Login button	employeeID = “EMP001” password = “654321”	Error message alerting that employeeID or password is wrong	Error message alerting that employeeID or password is wrong	PASS

7.2.5 Login Account – Valid Inputs

Table 7.5: Unit Test Case For Login Account – Valid Inputs.

Test Case ID	UT005				
Test Case Name	Unit Test Case for Login Account using Valid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Enter valid and correct employeeID and password	1. Fill in all text fields with valid input 2. Click Login button	employeeID = “EMP001” password = “123456”	Redirect to Profile Screen	Redirect to Profile Screen	PASS

7.2.6 Edit Profile – Invalid Inputs

Table 7.6: Unit Test Case For Edit Profile – Invalid Inputs.

Test Case ID	UT006				
Test Case Name	Unit Test Case for Edit Profile using Invalid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Edit the username using invalid format username when updating profile	1. Enter invalid format username 2. Fill in other text fields with valid input 3. Click 'tick' button	username = "longlongname" fullname = "WEIBIN" email = "weibin@gmail.com"	Error message alerting that username is invalid	Error message alerting that username is invalid	PASS
Edit the username with empty username when updating profile	1. Empty the username text field 2. Fill in other text fields with valid input 3. Click 'tick' button	username = "" fullname = "WEIBIN" email = "weibin@gmail.com"	Error message alerting that username is required	Error message alerting that username is required	PASS

Edit the full name with empty full name when updating profile	1. Empty the full name text field 2. Fill in other text fields with valid input 3. Click ‘tick’ button	username = “wb02” fullname = “” email = “weibin@gmail.com”	Error message alerting that full name is required	Error message alerting that full name is required	PASS
Edit the email using invalid format email when updating profile	1. Enter invalid format email 2. Fill in other text fields with valid input 3. Click ‘tick’ button	username = “wb02” fullname = “” email = “weibin.com”	Error message alerting that email is invalid	Error message alerting that email is invalid	PASS
Edit the email with empty email when updating profile	1. Empty the email text field 2. Fill in other text fields with valid input 3. Click ‘tick’ button	username = “wb02” fullname = “” email = “”	Error message alerting that email is required	Error message alerting that email is required	PASS

7.2.7 Edit Profile – Duplicate Records

Table 7.7: Unit Test Case For Edit Profile – Duplicate Records.

Test Case ID	UT007				
Test Case Name	Unit Test Case for Edit Profile using Existing Records in Database				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Edit username to registered username in the text field	1. Edit the username to registered username 2. Fill in other text fields with valid input 3. Click 'tick' button	username = "carly123" fullname = "WEI BIN" email = "weibin@gmail.com"	Error message alerting that username has already registered	Error message alerting that username has already registered	PASS
Edit email to registered email in the text field	1. Edit the email to registered email 2. Fill in other text fields with valid input 3. Click 'tick' button	username = "wb02" fullname = "WEI BIN" email = "carly@gmail.com"	Error message alerting that email has already registered	Error message alerting that email has already registered	PASS

7.2.8 Edit Profile – Valid Inputs and Not Duplicate Records

Table 7.8: Unit Test Case For Edit Profile – Valid Inputs and Not Duplicate Records.

Test Case ID	UT008				
Test Case Name	Unit Test Case for Edit Profile using Valid Inputs and Not Duplicate Records				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Edit the desired text fields with valid inputs and not duplicate records	1. Edit all the desired text fields with valid inputs and not duplicate records 2. Click 'tick' button	username = "wbwb" fullname = "EDITED" email = "edited@gmail.com"	Update successful and redirect to Profile screen	Update successful and redirect to Profile screen	PASS

7.2.9 View Dashboard

Table 7.9: Unit Test Case For View Dashboard.

Test Case ID	UT009				
Test Case Name	Unit Test Case for View Dashboard				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
View the dashboard within the mobile application	1. Open the drawer navigator 2. Click the 'Dashboard' tab	-	Redirect to Dashboard screen	Redirect to Dashboard screen	PASS

7.2.10 View Brief Information

Table 7.10: Unit Test Case For View Brief Information.

Test Case ID	UT010				
Test Case Name	Unit Test Case for View Brief Information				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
View the brief information related to the application within the mobile application	1. Open the drawer navigator 2. Click the 'Get Help' tab	-	Redirect to Get Help screen and display the information	Redirect to Get Help screen and display the information	PASS

7.2.11 View Application Usage Summary

Table 7.11: Unit Test Case For View Application Usage Summary.

Test Case ID	UT011				
Test Case Name	Unit Test Case for View Application Usage Summary				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
View the steel processing records summary within the mobile application	1. Open the drawer navigator 2. Click the 'Summary' tab 3. Click the 'Sort By' dropdown menu to sort the records if necessary	-	Redirect to Summary screen and display the records summary based on the sorting option	Redirect to Summary screen and display the records summary based on the sorting option	PASS

7.2.12 Process Steel Grading – Invalid Inputs

Table 7.12: Unit Test Case For Process Steel Grading – Invalid Inputs.

Test Case ID	UT012				
Test Case Name	Unit Test Case for Process Steel Grading using Invalid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Empty the area text field	1. Empty the area text field 2. Fill in other necessary input parameters by selecting the value from the dropdown menu 3. Click 'Predict' button	area = ""	Error message alerting that there is invalid input	Error message alerting that there is invalid input	PASS
Enter the invalid format area in the text field	1. Enter the invalid format area in the text field 2. Fill in other necessary input parameters by selecting the value from the dropdown menu 3. Click 'Predict' button	area = "-50" OR area = "150"	Error message alerting that there is invalid input	Error message alerting that there is invalid input	PASS

7.2.13 Process Steel Grading – Valid Inputs

Table 7.13: Unit Test Case For Process Steel Grading – Valid Inputs.

Test Case ID	UT013				
Test Case Name	Unit Test Case for Process Steel Grading using Valid Inputs				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Fill in all the necessary input parameters with valid inputs	1. Fill in all necessary input parameters by selecting the value from the dropdown menu and key in the valid format area in the text field 2. Click ‘Predict’ button	area = “80”	Grade is predicted and displayed on the screen	Grade is predicted and displayed on the screen	PASS

7.2.14 Logout Account

Table 7.14: Unit Test Case For Logout Account.

Test Case ID	UT014				
Test Case Name	Unit Test Case for Logout Account				
Test Case Description	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Logout the account after had completed tasks using the application	1. Open the drawer navigator 2. Click the 'Logout' tab 3. Click the 'Logout' button to logout the current account from the application	-	Successful logout and redirect to the welcome screen	Successful logout and redirect to the welcome screen	PASS

7.3 API Testing

API testing is also carried out to validate the reliability, functionality, security and performance of the API endpoints. This API testing ensures that the data transmitted between the client (frontend part) and the server (backend part) via the API requests and responses is accurate, consistent and secure. To test and verify the API endpoints constructed at the Flask server, Postman is adopted in this testing phase.

Table 7.15: API Test Case For Initialise Database.

Test Case ID		AT001		
Test Case Name		API Test Case for Initialise Database		
Test Case Description		To verify if the database can be initialized successfully		
API Endpoint		http://192.168.0.135:5000/initialize		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**
-	-	JSON response with message 'Database Initialized' and status code 200	Database with necessary table is created	PASS

Table 7.16: API Test Case For Get All Employee Information.

Test Case ID		AT002		
Test Case Name		API Test Case for Get All Employee Information		
Test Case Description		To verify if all the employees' information can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/employees		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**
-	-	JSON response with an array of employee data and status code 200	{ "employees": [{ "email": "wb@gmail.com", "employeeID": "EMP777", "fullname": "WEI BIN", "selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSgOEdgA9z6JfuAGPbEOaBQeF6QShIIqFTsBw&usqp=CAU", "username": "wb02" }, { "email": "carly@gmail.com", "employeeID": "EMP001", "fullname": "Carly",	PASS

			"selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTun6bB9mPzGhmCIuMhMv00hc9qgDQClQ1YI4SVbmG5flylB8tZBMZ6y5xwrHXPvIfDOcs&usqp=CAU",
			"username": "carly123"
			},
			{
			"email": "testacc@gmail.com",
			"employeeID": "EMP002",
			"fullname": "TEST ACC",
			"selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRksES88c6suZ8F_YP0erB8tMAhAvpsBKHRIQ&usqp=CAU",
			"username": "wbtest"
			}
			]	

Table 7.17: API Test Case For Sign Up Account.

Test Case ID		AT003		
Test Case Name		API Test Case for Sign Up Account		
Test Case Description		To verify if user can successfully sign up a new account		
API Endpoint		http://192.168.0.135:5000/signup		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	{
	"employeeID": "EMP123",
	"username": "john",
	"fullname": "John Doe",
	"email": "john@example.com",
	"password": "123456"
	}
		JSON response with message 'Account Created Successfully' and status code 200	New account is created successfully and the data is stored inside the database	PASS

Table 7.18: API Test Case For Login Account.

Test Case ID		AT004		
Test Case Name		API Test Case for Login Account		
Test Case Description		To verify if user can successfully login the existing account		
API Endpoint		http://192.168.0.135:5000/login		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	{
	"employeeID": "EMP123",
	"password": "123456"
	}
		JSON response with message 'Login Successfully' and status code 200	The user can login successfully	PASS

Table 7.19: API Test Case For Get Specific Employee Information.

Test Case ID		AT005		
Test Case Name		API Test Case for Get Specific Employee Information		
Test Case Description		To verify if specific employee information can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/employee/<employeeID>		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

employeeID = “EMP777”	-	JSON response with a JSON object representing the specific employee and status code 200	{
			"email": "wb@gmail.com",
			"employeeID": "EMP777",
			"fullname": "WEI BIN",
			"selectedAvatar": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSgOEdgA9z6JfuAGPbEOaBQeF6QShIIqFTsBw&usqp=CAU",
			"username": "wb02"
			}
				PASS

Table 7.20: API Test Case For Update Profile.

Test Case ID		AT006		
Test Case Name		API Test Case for Update Profile		
Test Case Description		To verify if the user can update the profile account successfully		
API Endpoint		http://192.168.0.135:5000/update_profile		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	{
	“employeeID”: “EMP777”,
	“username”: “wbwb”,
	“fullname”: “YWB”,
	“email”: “wbwb@gmail.com”,
	“selectedAvatar”:
	"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRksES88c6suZ8F_YP0erB8tMAhAvpsBKHRIQ&usqp=CAU"
	}
		JSON response with message ‘Profile Update Successfully’ and status code 200	The profile is updated successfully and the edited information is stored into the database	PASS

Table 7.21: API Test Case For Predict the Steel Grade.

Test Case ID		AT007		
Test Case Name		API Test Case for For Predict the Steel Grade		
Test Case Description		To verify if the user can predict the steel grade by the help of backend trained CNN model		
API Endpoint		http://192.168.0.135:5000/predict		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	<pre>{
	  "defects": [
	    { "id": 1, "length": "H", "width": "W", "height": "T", "rate": 0, "area": "40" },
	    { "id": 2, "length": "H", "width": "W", "height": "T", "rate": 0, "area": "50" },
	    { "id": 3, "length": "H", "width": "W", "height": "T", "rate": 0, "area": "60" },
	    { "id": 4, "length": "H", "width": "W", "height": "T", "rate": 0, "area": "70" },
	    { "id": 5, "length": "H", "width": "W", "height": "T", "rate": 0, "area": "80" }
	  ],
	  "employeeID": "EMP777"
	}</pre>
		JSON response with a ‘prediction’ object and status code 200	The steel grade is predicted with the result ‘1’ is returned	PASS

Table 7.22: API Test Case For Get the Processing Summary Data.

Test Case ID		AT008		
Test Case Name		API Test Case for Get the Processing Summary Data		
Test Case Description		To verify if the processing summary data can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/processing_summary		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with an array of processing summary data and status code 200	{
			"processing_summary": [
			{
			"employeeID": "EMP777",
			"processing_grade": "1",
			"processing_id": "P001",
			"processing_time": "2024-03-02 21:55:45",
			"username": "wb02"
			},
			{
			"employeeID": "EMP777",
			"processing_grade": "1",
			"processing_id": "P002",
			"processing_time": "2024-03-02 23:50:25",
			"username": "wb02"
			}
			]
			}
				PASS

Table 7.23: API Test Case For Get the Processing Summary Data Used In Dashboard.

Test Case ID		AT009		
Test Case Name		API Test Case for Get the Processing Summary Data Used in Dashboard		
Test Case Description		To verify if the processing summary data which is used in the dashboard visualization can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/dashboard_processing_summary		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with a 'total_processing' object and a 'grade_counts' object as well as status code 200	{
			"grade_counts": {
			"1": 25,
			"2": 6,
			"3": 3,
			"4": 3
			},
			"total_processing": 37
			}
				PASS

Table 7.24: API Test Case For Convert Excel File Data To SQLite Table.

Test Case ID		AT010		
Test Case Name		API Test Case for Convert Excel File Data to SQLite Table		
Test Case Description		To verify if the Excel file data can be converted successfully into the SQLite table		
API Endpoint		http://192.168.0.135:5000/convert_data		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**
-	-	JSON response with message 'Data Converted Successfully' and status code 200	A SQLite table named 'data' is created with the table columns and records similar to the Excel file data	PASS

Table 7.25: API Test Case For Get the Total Records In ‘data’ Table.

Test Case ID		AT011		
Test Case Name		API Test Case for Get the Total Records in ‘data’ Table		
Test Case Description		To verify if the total records in the ‘data’ table inside the database can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/total_records		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with a ‘total_count’ object and status code 200	{
			"total_count": 11537
			}
				PASS

Table 7.26: API Test Case For Get the Total Records Of Each Grade In ‘data’ Table.

Test Case ID		AT012		
Test Case Name		API Test Case for Get the Total Records Of Each Grade In ‘data’ Table		
Test Case Description		To verify if the total records of each grade in the ‘data’ table inside the database can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/total_by_grade_in_experts_prediction		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with a ‘total_by_grade_dict’ object and status code 200	{
			"1": 10761,
			"2": 56,
			"3": 81,
			"4": 639
			}
				PASS

Table 7.27: API Test Case For Get Each Grade Count By Month In ‘data’ Table.

Test Case ID		AT013		
Test Case Name		API Test Case for Get Each Grade Count By Month In ‘data’ Table		
Test Case Description		To verify if each grade count by month in the ‘data’ table inside the database can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/grade_count_by_month		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with a ‘grade_count_dict’ object and status code 200	{
			"Apr 2022": {
			"1": 990,
			"2": 6,
			"3": 2,
			"4": 47
			},
			"Aug 2022": {
			"1": 514,
			"2": 12,
			"3": 8,
			"4": 42
			},
			"Dec 2022": {
			"1": 960,
			"2": 2,
			"3": 6,
			"4": 58
			},
			"Feb 2022": {
				PASS

			"1": 601,
			"2": 13,
			"3": 13,
			"4": 53
			},
			"Jan 2022": {
			"1": 1211,
			"2": 3,
			"3": 31,
			"4": 75
			},
			"July 2022": {
			"1": 1086,
			"2": 4,
			"3": 2,
			"4": 41
			},
			"Jun 2022": {
			"1": 1343,
			"2": 4,
			"3": 1,
			"4": 46
			},
			"Mar 2022": {
			"1": 902,
			"2": 5,
			"3": 4,
			"4": 55
			},
			"May 2022": {
			"1": 957,
			"2": 3,
			"3": 5,	

			"4": 69
			},
			"Nov 2022": {
			"1": 936,
			"2": 4,
			"3": 7,
			"4": 77
			},
			"Oct 2022": {
			"1": 742,
			"2": 0,
			"3": 2,
			"4": 42
			},
			"Sep 2022": {
			"1": 519,
			"2": 0,
			"3": 0,
			"4": 34
			}
			}	

Table 7.28: API Test Case For Get Processing Summary Data Used In Dashboard Regarding Specific Employee.

Test Case ID		AT014		
Test Case Name		API Test Case for Get Processing Summary Data Used In Dashboard Regarding Specific Employee		
Test Case Description		To verify if the specific employee's processing summary data which is used in the dashboard visualization can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/AIPredictionByEmployee/<employeeID>		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

employeeID = "EMP777"	-	JSON response with a 'total_processing' object and a 'grade_counts' object as well as status code 200	{
			"grade_counts": {
			"1": 9,
			"2": 0,
			"3": 0,
			"4": 1
			},
			"total_processing": 10
			}
				PASS

Table 7.29: API Test Case For Get Total Processing For Each Employee In Application.

Test Case ID		AT015		
Test Case Name		API Test Case for Get Total Processing For Each Employee In Application		
Test Case Description		To verify if the total processing records by each employee in application can be obtained and returned		
API Endpoint		http://192.168.0.135:5000/total_processing_for_each_employee		
Test Data	**Request Body**	**Expected Result**	**Actual Result**	**Status**

-	-	JSON response with a 'total_processing_by_employee' object and status code 200	{
			"total_processing_by_employee": [
			{
			"total_processing": 18,
			"username": "carly123"
			},
			{
			"total_processing": 9,
			"username": "wbtest"
			},
			{
			"total_processing": 10,
			"username": "wb02"
			}
			]
			}
				PASS

7.4 System Usability Testing

After that, the system usability testing is conducted to ensure the application meets the needs and expectations of end-users, resulting in improved user satisfaction, productivity and the success of the application. Some potential usability issues that the user might suffer to successfully complete the task can also be found out and rectified thoroughly. Since this application aims to replace the manual steel grade evaluation and not relying experts for the grading process, five novice users are selected as the participants to make sure the application can be adopted by any user who is not even familiar with the business operations. Each participant involved is required to conduct the usability testing together with the user acceptance testing (UAT). The testing is conducted physically in which the participants are required to do the testing in the testing environment set up. Furthermore, the consent form and test scenario are provided to make sure the testing can be run smoothly. During the testing, all the participants' actions and interactions with the application are observed and recorded for further analysis. Lastly, all the participants are required to fill up the User Satisfaction Survey form so that their satisfaction towards the application can be analysed. **Appendix C** shows the informed consent form signed by the participants.

7.4.1 Test Scenario

Table below demonstrates the test scenarios that the participants should carry out using the mobile application during the system usability testing.

Table 7.30: Test Scenarios.

SteelGrade App: Mobile App For Steel Product Quality Assessment
Scenario 1 – Sign Up New Account
Imagine you are a new user to this mobile application and you want to sign up a new profile account so that you can utilize the functionalities within the application. What would you do?
Scenario 2 – Login Account

Imagine you had signed up a new account for this application in the earlier stage. Now, you wish to login your account to do something using the application. What would you do?

Scenario 3 – Edit Profile

You found that your personal information displayed on the Profile screen has some errors due to your wrong input in the sign up stage. Let's say you wish to edit your username and email address to a new value. What would you do?

Scenario 4 – View Brief Information

Assume you are a new user to this application and do not have any idea about what this application provides. You wish to get familiar with this application as soon as possible. What would you do?

Scenario 5 – Process Steel Grading

After familiarising yourself with the application, you have three sets of input parameters and wish to assess the grade of the steel products. What would you do to evaluate the steel product grading with the following input parameters?

Defect No	**Location**			**Rate**	**Area**
	Length	**Width**	**Height**		
1	H	C	D	L	100
2	H	C	D	L	100
3	H	A	D	L	100
4	H	F	D	L	100
5	H	F	T	L	100

Defect No	**Location**			**Rate**	**Area**
	Length	**Width**	**Height**		
1	U	W	D	H	60
2	U	W	D	H	60
3	H	F	D	L	100
4	H	F	T	L	100
5	H	F	T	L	100

Defect No	Location			Rate	Area
	Length	Width	Height		
1	H	D	D	H	10
2	H	A	D	L	100
3	H	C	D	L	100
4	H	F	D	L	100
5	H	F	T	L	100

Scenario 6 – View Application Usage Summary
You wish to know the records summary of this application. For example, you want to know who has done the steel grading processing, what time he / she has done and what is the predicted result of that steel grading processing. What would you do? If you want to sort the records for better analysis, what would you do?
Scenario 7 – View Dashboard
Imagine you are a data analyst and you wish to analyse the data related to the steel product grading within the application to gain some decisive information for business decision making. What would you do? If you want to further explore the dashboard in a detailed view, what would you do?
Scenario 8 – Logout Account
Imagine you have completed all your tasks by utilizing the application. Now, you want to logout your current account and wish to login as another identity. What would you do?

7.4.2 Usability Testing Result

After conducting the system usability testing and collecting the participants' user satisfaction survey form, the System Usability Scale (SUS) score is calculated to reflect the usability and user satisfaction of the application. To compute the SUS score, a few steps are required. Firstly, subtract one from the user response for the odd-items whereas subtract the user response from 5 for the even-numbered items. This subtraction scales all the values from 0 to 4.

Lastly, add up the converted responses for each user and multiply the total by 2.5. Now, the range of possible values ranges from 0 to 100 and that value represents the SUS score (Sauro, 2011). For the SUS score interpretation, Alathas (2018) emphasized a general guideline to rate the application or system developed as table below.

Table 7.31: General Guideline on SUS Score Interpretation (Alathas, 2018).

SUS Score	Grade	Adjective Rating
>80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
<51	F	Awful

Table 7.32: SUS Score Result.

Participants	Usability Score										Total	Percentage (%)
	1	2	3	4	5	6	7	8	9	10		
1	3	1	5	1	4	2	5	2	3	1	33	82.5
2	4	2	3	2	4	1	3	2	4	1	30	75.0
3	5	1	4	3	5	2	5	1	4	1	35	87.5
4	3	1	4	1	3	1	4	1	5	2	33	82.5
5	5	1	5	1	5	2	4	1	4	2	36	90.0
Average SUS Score												83.5

Based on the SUS Score collected and computed, the grade of the mobile application is A and it has the Excellent adjective rating in term of the usability and user satisfaction. The result of the user satisfaction survey is displayed in the **Appendix D**.

7.5 User Acceptance Testing (UAT)

At the same time, the UAT is conducted to verify whether the application can be accepted by users, all the functionalities can be operated by users comfortably, and lastly move to the production environment. The UAT is carried out by the 5 participants who were involved in the system usability testing and the result of the UAT is displayed in the Appendix E.

7.6 Traceability Matrix

Table 7.33: Traceability Matrix.

Use Case ID	Functional Requirement ID	Unit Testing Test Case ID	API Testing Test Case ID	UAT Test Case ID
UC001	FR01	UT001, UT002, UT003	AT001, AT003, AT010	UAT001
UC002	FR02	UT004, UT005	AT004	UAT002
UC003	FR03	UT006, UT007, UT008	AT005, AT006	UAT003
UC004	FR05	UT012, UT013	AT007	UAT005
UC005	FR06	UT009	AT009, AT011, AT012, AT013, AT014, AT015	UAT006, UAT007
UC006	FR07	UT011	AT008	UAT008, UAT009
UC007	FR04	UT010	AT002	UAT004
UC008	FR08	UT014	-	UAT010

7.7 Summary

In conclusion, this chapter had discussed various testings such as unit testing, API testing, system usability testing and user acceptance testing which were conducted to make sure the application is bug-free and can achieve high user satisfaction. Also, the traceability matrix was constructed to map the user cases, functional requirements, unit test cases, API test cases and UAT test cases.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Introduction

In a nutshell, this project was initiated, worked, and lasted for 6 months following the standard SDLC practices which are planning, analysis, design, implementation, testing and continuous improvement based on feedback. Upon the meeting of the exit criteria in the testing phase, and obtaining good results indicating the completeness and correctness of the application, this project was successfully completed. Also, all of the project objectives were achieved in the aspects of scope, cost, time as well as quality:

1. Conduct a comprehensive review of existing quality assessment applications and the CNN algorithms to inform the development of the mobile application by 30 July 2023.
2. Implement and validate the CNN algorithm to automate the steel product grading process, ensuring the mobile application's accuracy and reliability by April 2024.
3. Integrate dashboard functionality into the mobile application, enabling informed business decision-making within the CSCS before April 2024.
4. Evaluate the mobile application's effectiveness in streamlining steel product quality assessment and its impact on decision-making within the CSCS through user feedback and user acceptance testing by achieving a 95% passing rate before April 2024.

Now, this SteelGrade App (mobile application for steel product quality assessment) developed can become the alternative approach for the CSCS employee to do the steel product grading. This application is trusted to become an effective way to carry out the steel product grading as it successfully replaces the manual evaluation by the expert engineers. It leverages the innovative AI-based solution (CNN model) to revolutionise the steel product grading method.

8.2 Limitations

Even though the application is good enough to serve its purpose for the users to complete their tasks, there are also few limitations that restrict the application from unleash to maximum. Here can be the limitations:

1. **Limitation to Local Users**

   Until this moment, the application is developed and hosted in the local which means that every user is required to first up the backend server, modify the corresponding IP address for API endpoints at frontend, and launch the emulator every time they want to use this application to carry out their daily tasks. This might be very tedious and troublesome, increasing the risk that user deprecates this application. In other words, the user cannot directly utilize the application by just clicking and launching the application. Some steps are still required to be implemented in advance beforehand.

2. **Low Support for Class 2 and Class 3 in Dataset**

   Although the overall accuracy of the trained model is excellent and this represents the trained model can perform well for the new unseen data, the support or the actual occurrences for Class 2 and Class 3 in the data that used to train the model are quite low. This might somehow reduce the accuracy of the model in classifying the specific instances or records that belong to Class 2 or Class 3.

3. **Lack of Admin Role and Corresponding Features Implementation to Manage the Application**

   The application lacks the admin who is responsible for the application update and management. For example, if the employee who has ever registered for the application but resigned now, the admin should take the responsibility to delete his / her account in the system. Furthermore, if the dashboard's

content needs to be changed from 2022-based data to 2023-based data, there is no any feature in the application to do so for now (still need rely on backend – manually change the Excel file to the desired one).

8.3 Future Work

Some improvements also can be made in the future so that the application will become more and more robust and more and more widely accepted by users:

1. **Deployment Work for Both Frontend and Backend**

   The backend Flask server can be hosted to the Cloud employing some cloud provider platforms such as Heroku and AWS. In this way, the backend server will always be in the up status and the static IP address of the server will be generated. Using the static IP address, the IP address of the API endpoints at the frontend is not required to change from time to time, place to place as now the server is hosted at cloud, not locally. Furthermore, the frontend React Native application can also be deployed into an APK file (for Android) or an IPA file (for iOS). Using these files, users can easily install the application into their device and use it instead of relying on the emulator. This kind of deployment makes the application more easy and convenient to use.

2. **Find more Comprehensive Dataset**

   Although the accuracy of the trained model now is acceptable, a more robust model who has high accuracy towards all class labels is invincible. In this case, more and more data records for Class 2 and Class 3 can be identified and added into the dataset to achieve the balanced dataset in which all different types of class label records are distributed evenly. With the balanced nature of the dataset, a higher accuracy model can be trained and obtained.

3. **Develop the Functionalities For Admin**

Admin-related features can be developed for the application content updating and managing purpose. Since some of the charts inside the dashboard are using the static data in Excel based on one year, the Excel file might need to be changed or updated in some situations. Thus, the development of admin role and related-function can bring the application's completeness to the next level.

4. **Integrate IoT Devices With the Application**

To streamline the manual necessary parameters input for the steel grading processing, IoT devices can be developed and integrated with the application. Using the IoT devices, the necessary parameters can be detected, captured, received and flowed to the application automatically to assess the grading. This development and implementation can further replace the tedious manual key in process.

REFERENCES

Alathas, H., 2018. *How to Measure Product Usability with the System Usability Scale (SUS) Score.* [Online] Available at <https://uxplanet.org/how-to-measure-product-usability-with-the-systemusability-scale-sus-score-69f3875b858f> [Accessed 2 April 2024].

Ali, M.A.H. and Thai, K.W., 2017. Automated fruit grading system. *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation, ROMA 2017*, 2017-Decem, pp.1–6. https://doi.org/10.1109/ROMA.2017.8231734.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. [online] *Journal of Big Data*, Springer International Publishing. https://doi.org/10.1186/s40537-021-00444-8.

Anderson, M., 2021. *What Are ASTM Standards.* MEAD METALS, INC.. [online] Available at: <https://www.meadmetals.com/blog/what-are-astm-standards#:~:text=ASTM%20standards%20are%20crucial%20to,proper%20pr> [Accessed 8 August 2023].

Bach, B., Freeman, E., Abdul-Rahman, A., Turkay, C., Khan, S., Fan, Y. and Chen, M., 2023. Dashboard Design Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 29(1), pp.342–352. https://doi.org/10.1109/TVCG.2022.3209448.

Beskopylny, A., Lyapin, A., Anysz, H., Meskhi, B., Veremeenko, A. and Mozgovoy, A., 2020. Artificial neural networks in classification of steel grades based on non-destructive tests. *Materials*, 13(11). https://doi.org/10.3390/ma13112445.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K. and Ghayvat, H., 2021. CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. Electronics, [online] 10(20), p.2470. https://doi.org/10.3390/electronics10202470.

Casallo, L., Lucero, E., Maradiegue, F. and Alvarez, J.C., 2021. Proposal of Work Standardization to Improve a Metal-mechanical Process. In: *2021 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2021*. Institute of Electrical and Electronics Engineers Inc. pp.668–672. https://doi.org/10.1109/IEEM50564.2021.9673052.

Chandra, V., 2015. Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 131(9), pp.7–10. https://doi.org/10.5120/ijca2015907294.

CSC Steel Holdings Berhad, 2022. *Products CSC Steel.* CSC Steel Holdings Berhad. [online] Available at: < https://cscmalaysia.com/products/> [Accessed 8 July 2023].

Dennis, A., Wixom, B. and Tegarden, D., 2015. Systems analysis and design: An object-oriented approach with UML. John wiley & sons.

Drozd, R. and Wolniak, R., 2021. Systematic assessment of product quality. *Journal of Open Innovation: Technology, Market, and Complexity*, [online] 7(4), p.235. https://doi.org/10.3390/joitmc7040235.

Federal Steel Supply, 2023. How Technology is Shaping the Future of the Steel Industry. [online] (13 February) Available at: < https://www.fedsteel.com/industry-trends/how-technology-is-shaping-thefuture-of-the-steel-industry/> [Accessed 22 July 2023].

Hastings Machine Company, n.d.. SAE Steel Properties Designation System. Hastings Machine Company. [online] Available at: < https://hastingsmachine.com/wp-content/uploads/2014/04/SAE-Steel-Properties-Designation-System.pdf> [Accessed 8 August 2023].

Kyeremeh, K., 2021. Overview Of System Development Life Cycle Models. *Journal of Management and Science*, 11(1), pp.12–22. https://doi.org/10.26524/jms.11.3.

Lai, W. L., 2023. *Current On Line QC Judging Workflow.* [Image] (AI綫上品質判定專家系統).

Precise Pipe, 2022. *Steel Pipe Comparison: Differences Of A53 PIPE Vs. A106.* Precise Pipe. [online] Available at: < https://precisepipe.com/steel-pipecomparison-differences-of-a53-pipe-vsa106/#:~:text=Mechanical%20Properties%20Of%20A53%20Pipes,B%20pipe s%20is%2060%2C000%20psi.> [Accessed 8 August 2023].

Reliance Foundry, n.d.. *Steel Grading: Chemistry and Properties.* Reliance Foundry. [online] Available at: < https://www.reliance-foundry.com/blog/steel-grades#:~:text=The%20AISI%2FSAE%20numbering%20system,%25%20Chr> [Accessed 8 August 2023].

Satyendra, 2014. *Standards, Standardization Process, and Standardized Work in Industry.* [online] (25 January) Available at: < https://www.ispatguru.com/standardization-process-in-steel-industry/> [Accessed 22 July 2023].

Sauro, J., 2011. *Measuring Usability with the System Usability Scale (SUS)*. [Online] Available at: https://measuringu.com/sus/#:~:text=What%20is%20a%20Good%20SUS,thro ugh% 20a%20process%20called%20normalizing. [Accessed 2 April 2024].

Tu, Q., Li, D., Xie, Q., Dai, L. and Wang, J., 2022. Automated Scrap Steel Grading via a Hierarchical Learning-Based Framework. *IEEE Transactions on Instrumentation and Measurement*, 71, pp.1–13. https://doi.org/10.1109/TIM.2022.3206816.

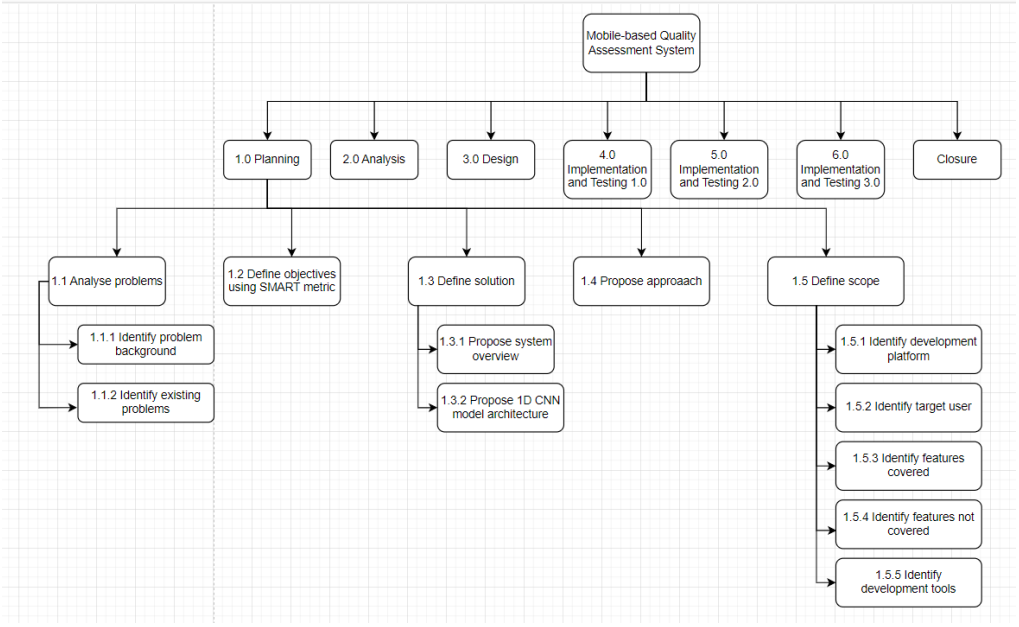
Varghese, R.R., Mathew Jacob, P., Sooraj, S., Ranjan, D.M., Cherian Varughese, J. and Raju, H., 2021. Detection and Grading of Multiple Fruits and Vegetables Using Machine Vision. *2021 8th International Conference on Smart Computing and Communications: Artificial Intelligence, AI Driven Applications for a Smart World, ICSCC 2021*, pp.85–89. https://doi.org/10.1109/ICSCC51209.2021.9528165.

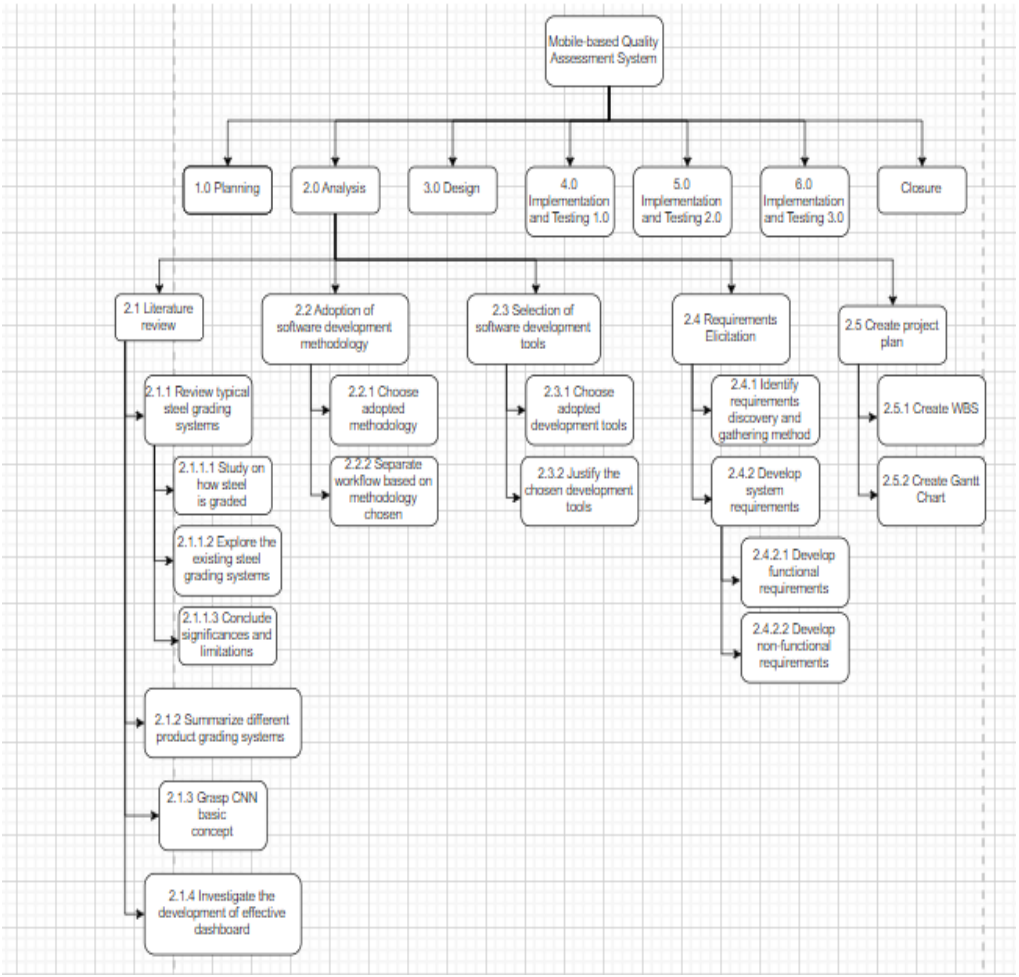
Velasquez Villagran, N., Estevez, E., Pesado, P. and De Juanes Marquez, J., 2019. Standardization: A key factor of industry 4.0. *2019 6th International Conference on eDemocracy and eGovernment, ICEDEG 2019*, pp.350–354. https://doi.org/10.1109/ICEDEG.2019.8734339.

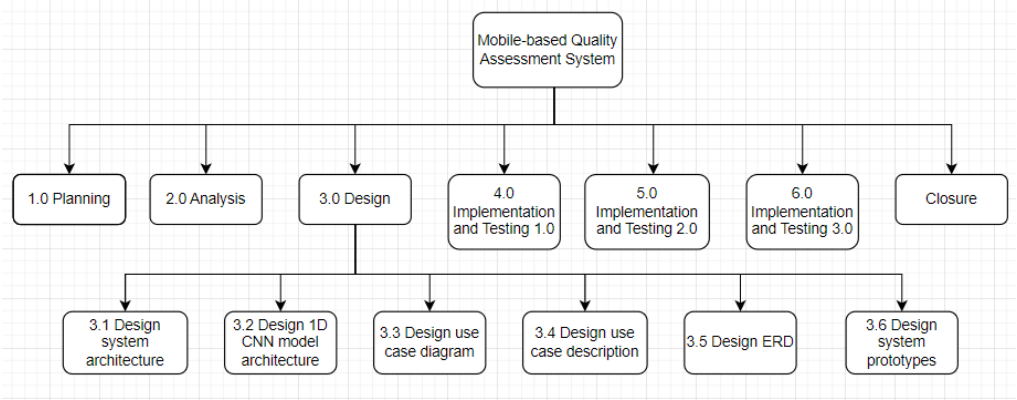
Yusianto, R., Suprijono, H., Islahudin, N. and Nugroho, D.S., 2022. Smart Potato Grading using Image Processing and Fuzzy Grading System. *2022 International Seminar on Application for Technology of Information and Communication: Technology 4.0 for Smart Ecosystem: A New Way of Doing Digital Business, iSemantic 2022*, pp.333–337. https://doi.org/10.1109/iSemantic55962.2022.9920382.

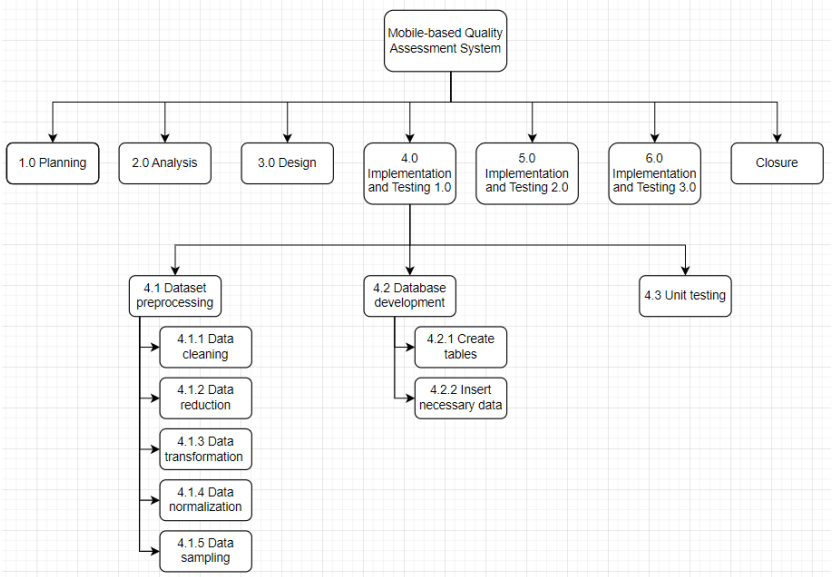
APPENDICES

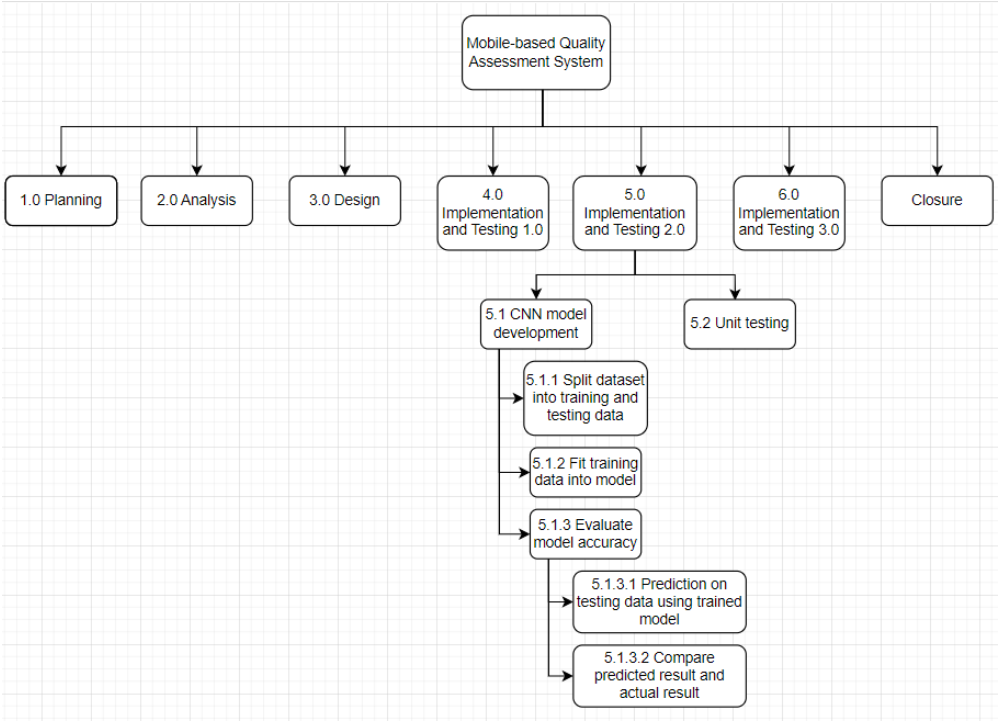
Appendix A: WBS Diagram

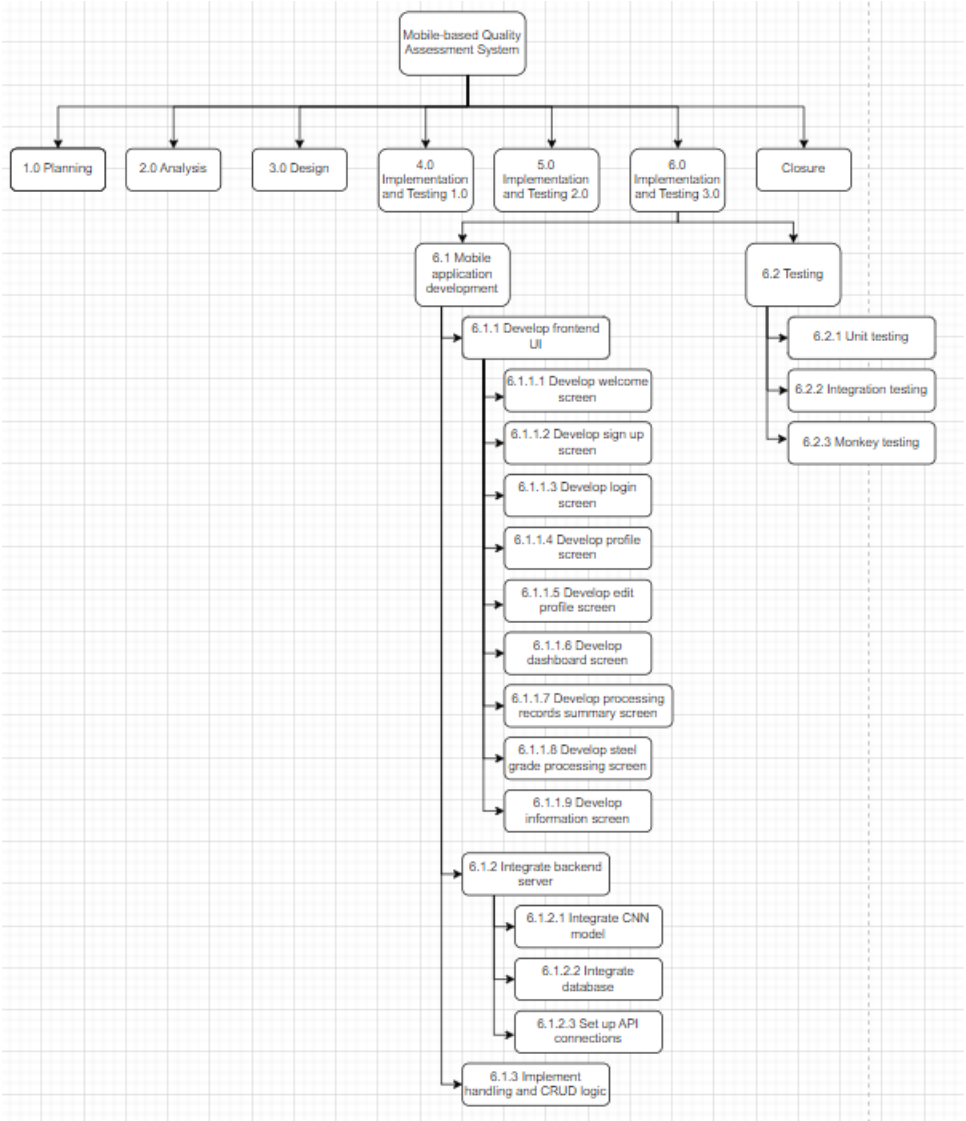


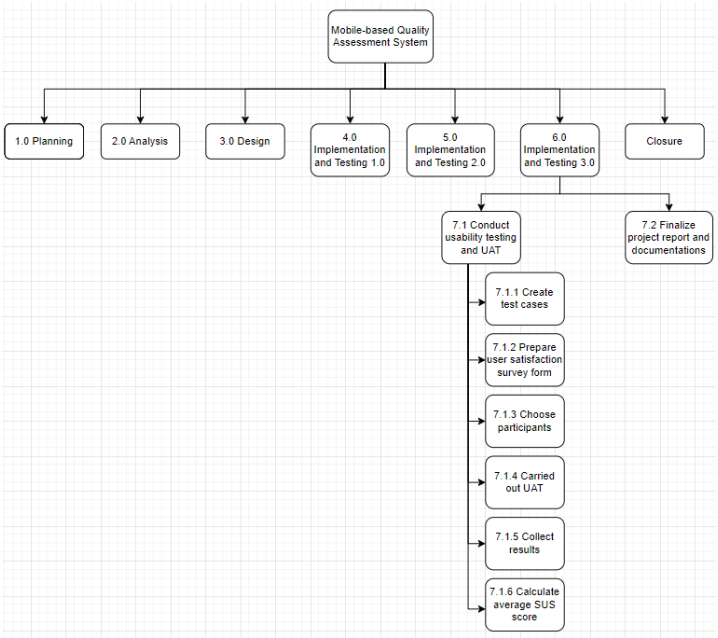



Mobile-based Quality Assessment System
1.0 Planning
2.0 Analysis
3.0 Design
4.0 Implementation and Testing 1.0
5.0 Implementation and Testing 2.0
6.0 Implementation and Testing 3.0
Closure
3.1 Design system architecture
3.2 Design 1D CNN model architecture
3.3 Design use case diagram
3.4 Design use case description
3.5 Design ERD
3.6 Design system prototypes


Mobile-based Quality Assessment System
1.0 Planning
2.0 Analysis
3.0 Design
4.0 Implementation and Testing 1.0
5.0 Implementation and Testing 2.0
6.0 Implementation and Testing 3.0
Closure
4.1 Dataset preprocessing
4.1.1 Data cleaning
4.1.2 Data reduction
4.1.3 Data transformation
4.1.4 Data normalization
4.1.5 Data sampling
4.2 Database development
4.2.1 Create tables
4.2.2 Insert necessary data
4.3 Unit testing


Mobile-based Quality Assessment System
1.0 Planning
2.0 Analysis
3.0 Design
4.0 Implementation and Testing 1.0
5.0 Implementation and Testing 2.0
6.0 Implementation and Testing 3.0
Closure
5.1 CNN model development
5.2 Unit testing
5.1.1 Split dataset into training and testing data
5.1.2 Fit training data into model
5.1.3 Evaluate model accuracy
5.1.3.1 Prediction on testing data using trained model
5.1.3.2 Compare predicted result and actual result


Mobile-based Quality Assessment System
1.0 Planning
2.0 Analysis
3.0 Design
4.0 Implementation and Testing 1.0
5.0 Implementation and Testing 2.0
6.0 Implementation and Testing 3.0
Closure
6.1 Mobile application development
6.2 Testing
6.1.1 Develop frontend UI
6.1.1.1 Develop welcome screen
6.1.1.2 Develop sign up screen
6.1.1.3 Develop login screen
6.1.1.4 Develop profile screen
6.1.1.5 Develop edit profile screen
6.1.1.6 Develop dashboard screen
6.1.1.7 Develop processing records summary screen
6.1.1.8 Develop steel grade processing screen
6.1.1.9 Develop information screen
6.1.2 Integrate backend server
6.1.2.1 Integrate CNN model
6.1.2.2 Integrate database
6.1.2.3 Set up API connections
6.1.3 Implement handling and CRUD logic
6.2.1 Unit testing
6.2.2 Integration testing
6.2.3 Monkey testing


Mobile-based Quality Assessment System
1.0 Planning
2.0 Analysis
3.0 Design
4.0 Implementation and Testing 1.0
5.0 Implementation and Testing 2.0
6.0 Implementation and Testing 3.0
Closure
7.1 Conduct usability testing and UAT
7.2 Finalize project report and documentations
7.1.1 Create test cases
7.1.2 Prepare user satisfaction survey form
7.1.3 Choose participants
7.1.4 Carried out UAT
7.1.5 Collect results
7.1.6 Calculate average SUS score

Appendix B: Gantt Chart

TASK	START DATE	DUE DATE	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
1.0 Planning	19/6/2023	16/7/2023												
1.1 Analyse problems	19/6/2023	25/6/2023												
1.1.1 Identify problem background	19/6/2023	21/6/2023												
1.1.2 Identify existing problems	21/6/2023	25/6/2023												
1.2 Define objectives using SMART metric	26/6/2023	2/7/2023												
1.3 Propose solution	29/6/2023	9/7/2023												
1.3.1 Propose system overview	29/6/2023	6/7/2023												
1.3.2 Propose 1D CNN model architecture	3/7/2023	9/7/2023												
1.4 Propose approach	6/7/2023	10/7/2023												
1.5 Define scope	9/7/2023	16/7/2023												
1.5.1 Identify development platform	9/7/2023	9/7/2023												
1.5.2 Identify target user	10/7/2023	11/7/2023												
1.5.3 Identify features covered	12/7/2023	13/7/2023												
1.5.4 Identify features not covered	14/7/2023	14/7/2023												
1.5.5 Identify development tools	15/7/2023	16/7/2023												

TASK	START DATE	DUE DATE	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
2.0 Analysis	17/7/2023	13/8/2023												
2.1 Literature review	17/7/2023	30/7/2023												
2.1.1 Review typical steel grading systems	17/7/2023	20/7/2023												
2.1.1.1 Study on how steel is graded	17/7/2023	17/7/2023												
2.1.1.2 Explore the existing steel grading systems	18/7/2023	19/7/2023												
2.1.1.3 Conclude significances and limitations	20/7/2023	20/7/2023												
2.1.2 Summarize different product grading systems	21/7/2023	23/7/2023												
2.1.3 Grasp CNN basic concept	24/7/2023	26/7/2023												
2.1.4 Investigate the development of effective dashboard	27/7/2023	30/7/2023												
2.2 Adoption of software development methodology	24/7/2023	30/7/2023												
2.2.1 Choose adopted methodology	24/7/2023	26/7/2023												
2.2.2 Separate workflow based on methodology chosen	27/7/2023	30/7/2023												
2.3 Selection of software development tools	31/7/2023	6/8/2023												
2.3.1 Choose adopted development tools	31/7/2023	2/8/2023												
2.3.2 Justify the chosen development tools	3/8/2023	6/8/2023												
2.4 Requirements Elicitation	31/7/2023	13/8/2023												
2.4.1 Identify requirements discovery and gathering method	31/7/2023	4/8/2023												
2.4.2 Develop system requirements	5/8/2023	13/8/2023												
2.4.2.1 Develop functional requirements	5/8/2023	9/8/2023												
2.4.2.2 Develop non-functional requirements	10/8/2023	13/8/2023												
2.5 Create project plan	7/8/2023	13/8/2023												
2.5.1 Create WBS	7/8/2023	10/8/2023												
2.5.2 Create Gantt Chart	11/8/2023	13/8/2023												

TASK	START DATE	DUE DATE	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
3.0 Design	14/8/2023	10/9/2023												
3.1 Design system architecture	14/8/2023	16/8/2023												
3.2 Design 1D CNN model architecture	17/8/2023	19/8/2023												
3.3 Design use case diagram	20/8/2023	21/8/2023												
3.4 Design use case description	22/8/2023	24/8/2023												
3.5 Design ERD	25/8/2023	26/8/2023												
3.6 Design system prototypes	27/8/2023	10/9/2023												

TASK	START DATE	DUE DATE	WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17	WEEK 18	WEEK 19	WEEK 20	WEEK 21	WEEK 22	WEEK 23	WEEK 24
4.0 Implementation and Testing 1.0	22/1/2024	18/2/2024												
4.1 Dataset preprocessing	22/1/2024	4/2/2024												
4.1.1 Data cleaning	22/1/2024	23/1/2024												
4.1.2 Data reduction	24/1/2024	26/1/2024												
4.1.3 Data transformation	27/1/2024	29/1/2024												
4.1.4 Data normalization	30/1/2024	31/1/2024												
4.1.5 Data sampling	1/2/2024	4/2/2024												
4.2 Database development	5/2/2024	11/2/2024												
4.2.1 Create tables	5/2/2024	8/2/2024												
4.2.2 Insert necessary data	9/2/2024	11/2/2024												
4.3 Unit Testing	12/2/2024	18/2/2024												

TASK	START DATE	DUE DATE	WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17	WEEK 18	WEEK 19	WEEK 20	WEEK 21	WEEK 22	WEEK 23	WEEK 24
5.0 Implementation and Testing 2.0	19/2/2024	3/3/2024												
5.1 CNN model development	19/2/2024	25/2/2024												
5.1.1 Split dataset into training and testing data	19/2/2024	21/2/2024												
5.1.2 Fit training data into model	22/2/2024	23/2/2023												
5.1.3 Evaluate model accuracy	24/2/2024	25/2/2024												
5.1.3.1 Prediction on testing data using trained model	24/2/2024	26/2/2024												
5.1.3.2 Compare predicted result and actual result	24/2/2024	27/2/2024												
5.2 Unit testing	26/2/2024	3/3/2024												

TASK	START DATE	DUE DATE	WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17	WEEK 18	WEEK 19	WEEK 20	WEEK 21	WEEK 22	WEEK 23	WEEK 24
6.0 Implementation and Testing 3.0	4/3/2024	7/4/2024												
6.1 Mobile application development	4/3/2024	31/3/2024												
6.1.1 Develop frontend UI	4/3/2024	17/3/2024												
6.1.1.1 Develop welcome screen	4/3/2024	4/3/2024												
6.1.1.2 Develop sign up screen	5/3/2024	5/3/2024												
6.1.1.3 Develop login screen	6/3/2024	6/3/2024												
6.1.1.4 Develop profile screen	7/3/2024	7/3/2024												
6.1.1.5 Develop edit profile screen	8/3/2024	8/3/2024												
6.1.1.6 Develop dashboard screen	9/3/2024	12/3/2024												
6.1.1.7 Develop processing records summary screen	13/3/2024	14/3/2024												
6.1.1.8 Develop steel grade processing screen	15/3/2024	16/3/2024												
6.1.1.9 Develop information screen	17/3/2024	17/3/2024												
6.1.2 Integrate backend server	18/3/2024	31/3/2024												
6.1.2.1 Integrate CNN model	18/3/2023	22/3/2024												
6.1.2.2 Integrate database	23/3/2024	26/3/2024												
6.1.2.3 Set up API connections	27/3/2024	31/3/2024												
6.1.3 Implement handling and CRUD logic	18/3/2024	31/3/2024												
6.2 Testing	1/4/2024	7/4/2024												
6.2.1 Unit testing	1/4/2024	2/4/2024												
6.2.2 Integration testing	3/4/2024	5/4/2024												
6.2.3 Monkey testing	6/4/2024	7/4/2024												

TASK	START DATE	DUE DATE	WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17	WEEK 18	WEEK 19	WEEK 20	WEEK 21	WEEK 22	WEEK 23	WEEK 24
7.0 Closure	8/4/2024	14/4/2024												
7.1 Conduct usability testing and UAT	8/4/2024	11/4/2024												
7.1.1 Create test cases	8/4/2024	8/4/2024												
7.1.2 Prepare user satisfaction survey form	8/4/2024	8/4/2024												
7.1.3 Choose participants	8/4/2024	8/4/2024												
7.1.4 Carried out UAT	9/4/2024	10/4/2024												
7.1.5 Collect results	11/4/2024	11/4/2024												
7.1.6 Calculate average SUS score	11/4/2024	11/4/2024												
7.2 Finalize project report and documentations	12/4/2024	14/4/2024												

Appendix C: Consent Form

Participant 1:

Consent & Recording Release Form

I agree to participate in the study conducted and recorded by the MoB.

I understand and consent to the use and release of the recording by MoB. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by MoB without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24 / 3 /2024

Please write your name: Jeremy Ng Yao Jie

Please sign your name: _____

Thank You.

We appreciate your participation.

Participant 2:

Consent & Recording Release Form

I agree to participate in the study conducted and recorded by the MoB.

I understand and consent to the use and release of the recording by MoB. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by MoB without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24 / 3 /2024

Please write your name: Wong Zhi Xin

Please sign your name: _____

Thank You.

We appreciate your participation.

Participant 3:

Consent & Recording Release Form

I agree to participate in the study conducted and recorded by the MoB.

I understand and consent to the use and release of the recording by MoB. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by MoB without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 28 / 3 /2024

Please write your name: Yap Chia Hau

Please sign your name: _____

Thank You.

We appreciate your participation.

Participant 4:

Consent & Recording Release Form

I agree to participate in the study conducted and recorded by the MoB.

I understand and consent to the use and release of the recording by MoB. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by MoB without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 28 / 3 /2024

Please write your name: Tham Kar Weng

Please sign your name: _____

Thank You.

We appreciate your participation.

Participant 5:

Consent & Recording Release Form

I agree to participate in the study conducted and recorded by the MoB.

I understand and consent to the use and release of the recording by MoB. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by MoB without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24 / 3 /2024

Please write your name: Susan Chiew

susan

Please sign your name: _____

Thank You.

We appreciate your participation.

Appendix D: User Satisfaction Survey Result

Participant #1: Jeremy Ng Yao Jie

User Satisfaction Survey (adapted from System Usability Scale, Brooke, J. (1986))

1 = Stronly Diagree 5-Strongly Agree	**1**	**2**	**3**	**4**	**5**
1. I think that I would like to use this application for steel grading related matter.			✓		
2. I found the application unnecessarily complex.	✓				
3. I thought the application was easy to use.					✓
4. I think I would need the support of a technical person to be able to use the application.	✓				
5. I found this application was easily moved through without a lot of backtracking or data re-entry.				✓	
6. I thought there was too much inconsistency in this application.		✓			
7. I would imagine that most people would learn to use this application very quickly.					✓
8. I found the application very awkward to use.		✓			
9. I felt very confident using the application.			✓		
10. I needed to learn a lot of things before I could get going with this application.	✓				

Please leave your comments (if any)	
What did you like best about the site?	The clean layout of dashboard
What did you like least about the site?	Sometimes need wait to wait the data to load
If you were to describe this site to a colleague in a sentence or two, what would you say?	Can be used to complete task
Do you have any other final comments or questions?	Include more information into dashboard using domain knowledge

Participant #2: Wong Zhi Xin

User Satisfaction Survey (adapted from System Usability Scale, Brooke, J. (1986))

1 = Stronly Diagree 5-Strongly Agree	**1**	**2**	**3**	**4**	**5**
1. I think that I would like to use this application for steel grading related matter.				✓	
2. I found the application unnecessarily complex.		✓			
3. I thought the application was easy to use.			✓		
4. I think I would need the support of a technical person to be able to use the application.		✓			
5. I found this application was easily moved through without a lot of backtracking or data re-entry.				✓	
6. I thought there was too much inconsistency in this application.	✓				
7. I would imagine that most people would learn to use this application very quickly.			✓		
8. I found the application very awkward to use.		✓			
9. I felt very confident using the application.				✓	
10. I needed to learn a lot of things before I could get going with this application.	✓				

Please leave your comments (if any)	
What did you like best about the site?	Easy to understand the UI
What did you like least about the site?	-
If you were to describe this site to a colleague in a sentence or two, what would you say?	-
Do you have any other final comments or questions?	What is the use of the dashboard to me if I am not the data analyst

Participant #3: Yap Chia Hau

User Satisfaction Survey (adapted from System Usability Scale, Brooke, J. (1986))

1 = Stronly Diagree 5-Strongly Agree	**1**	**2**	**3**	**4**	**5**
1. I think that I would like to use this application for steel grading related matter.					✓
2. I found the application unnecessarily complex.	✓				
3. I thought the application was easy to use.				✓	
4. I think I would need the support of a technical person to be able to use the application.			✓		
5. I found this application was easily moved through without a lot of backtracking or data re-entry.					✓
6. I thought there was too much inconsistency in this application.		✓			
7. I would imagine that most people would learn to use this application very quickly.					✓
8. I found the application very awkward to use.	✓				
9. I felt very confident using the application.				✓	
10. I needed to learn a lot of things before I could get going with this application.	✓				

Please leave your comments (if any)	
What did you like best about the site?	Simple and easy to use
What did you like least about the site?	Face some laggy when using maybe because the laptop performance
If you were to describe this site to a colleague in a sentence or two, what would you say?	An easy-to-use application to handle the specific task
Do you have any other final comments or questions?	-

Participant #4: Tham Kar Weng

User Satisfaction Survey (adapted from System Usability Scale, Brooke, J. (1986))

1 = Stronly Diagree 5-Strongly Agree	**1**	**2**	**3**	**4**	**5**
1. I think that I would like to use this application for steel grading related matter.			✓		
2. I found the application unnecessarily complex.	✓				
3. I thought the application was easy to use.				✓	
4. I think I would need the support of a technical person to be able to use the application.	✓				
5. I found this application was easily moved through without a lot of backtracking or data re-entry.			✓		
6. I thought there was too much inconsistency in this application.	✓				
7. I would imagine that most people would learn to use this application very quickly.				✓	
8. I found the application very awkward to use.	✓				
9. I felt very confident using the application.					✓
10. I needed to learn a lot of things before I could get going with this application.		✓			

Please leave your comments (if any)	
What did you like best about the site?	Simple layout not confusing user
What did you like least about the site?	Need to key in the parameters one by one manually when do the steel grading processing
If you were to describe this site to a colleague in a sentence or two, what would you say?	Can get familiar very fast
Do you have any other final comments or questions?	No

Participant #5: Susan Chiew

User Satisfaction Survey (adapted from System Usability Scale, Brooke, J. (1986))

1 = Stronly Diagree 5-Strongly Agree	**1**	**2**	**3**	**4**	**5**
1. I think that I would like to use this application for steel grading related matter.					✓
2. I found the application unnecessarily complex.	✓				
3. I thought the application was easy to use.					✓
4. I think I would need the support of a technical person to be able to use the application.	✓				
5. I found this application was easily moved through without a lot of backtracking or data re-entry.					✓
6. I thought there was too much inconsistency in this application.		✓			
7. I would imagine that most people would learn to use this application very quickly.				✓	
8. I found the application very awkward to use.	✓				
9. I felt very confident using the application.				✓	
10. I needed to learn a lot of things before I could get going with this application.		✓			

Please leave your comments (if any)	
What did you like best about the site?	Straight-to-point function and user-friendly UI
What did you like least about the site?	-
If you were to describe this site to a colleague in a sentence or two, what would you say?	This application can help us to evaluate the steel grade automatically
Do you have any other final comments or questions?	Maybe can use other ways to input the parameters for the grading part

Appendix E: UAT Results.

Participant #1:

Testing Date	**24 / 3 /2024**		
Testing Start Time	**10.00am**	**Testing End Time**	**10.25am**
Tester Name	**Jeremy Ng Yao Jie**		

Test Case ID	**Test Scenario**	**Status**
UAT001	Able to sign up a new profile account	PASS
UAT002	Able to login to the existing account	PASS
UAT003	Able to edit the profile information	PASS
UAT004	Able to get the brief information to get familiar with the application	PASS
UAT005	Able to process the steel grading at least three time	PASS
UAT006	Able to obtain the dashboard information to do data analysis	PASS
UAT007	Able to further explore the charts in dashboard	PASS
UAT008	Able to view the application usage summary after processing the steel grading	PASS
UAT009	Able to use the sorting function to view the application usage summary	PASS
UAT010	Able to logout the current account in the application	PASS

Participant #2:

Testing Date	**24 / 3 /2024**		
Testing Start Time	**11:00am**	**Testing End Time**	**11.30am**
Tester Name	**Wong Zhi Xin**		

Test Case ID	**Test Scenario**	**Status**
UAT001	Able to sign up a new profile account	PASS
UAT002	Able to login to the existing account	PASS
UAT003	Able to edit the profile information	PASS
UAT004	Able to get the brief information to get familiar with the application	PASS
UAT005	Able to process the steel grading at least three time	PASS
UAT006	Able to obtain the dashboard information to do data analysis	PASS
UAT007	Able to further explore the charts in dashboard	PASS
UAT008	Able to view the application usage summary after processing the steel grading	PASS
UAT009	Able to use the sorting function to view the application usage summary	PASS
UAT010	Able to logout the current account in the application	PASS

Participant #3:

Testing Date	**28 / 3 /2024**		
Testing Start Time	**2:30pm**	**Testing End Time**	**2:50pm**
Tester Name	**Yap Chia Hau**		

Test Case ID	**Test Scenario**	**Status**
UAT001	Able to sign up a new profile account	PASS
UAT002	Able to login to the existing account	PASS
UAT003	Able to edit the profile information	PASS
UAT004	Able to get the brief information to get familiar with the application	PASS
UAT005	Able to process the steel grading at least three time	PASS
UAT006	Able to obtain the dashboard information to do data analysis	PASS
UAT007	Able to further explore the charts in dashboard	PASS
UAT008	Able to view the application usage summary after processing the steel grading	PASS
UAT009	Able to use the sorting function to view the application usage summary	PASS
UAT010	Able to logout the current account in the application	PASS

Participant #4:

Testing Date	**28 / 3 / 2024**		
Testing Start Time	**3:15pm**	**Testing End Time**	**3:45pm**
Tester Name	**Tham Kar Weng**		

Test Case ID	**Test Scenario**	**Status**
UAT001	Able to sign up a new profile account	PASS
UAT002	Able to login to the existing account	PASS
UAT003	Able to edit the profile information	PASS
UAT004	Able to get the brief information to get familiar with the application	PASS
UAT005	Able to process the steel grading at least three time	PASS
UAT006	Able to obtain the dashboard information to do data analysis	PASS
UAT007	Able to further explore the charts in dashboard	PASS
UAT008	Able to view the application usage summary after processing the steel grading	PASS
UAT009	Able to use the sorting function to view the application usage summary	PASS
UAT010	Able to logout the current account in the application	PASS

Participant #5:

Testing Date	**24 / 3 / 2024**		
Testing Start Time	**12:00pm**	**Testing End Time**	**12:30pm**
Tester Name	**Susan Chiew**		

Test Case ID	**Test Scenario**	**Status**
UAT001	Able to sign up a new profile account	PASS
UAT002	Able to login to the existing account	PASS
UAT003	Able to edit the profile information	PASS
UAT004	Able to get the brief information to get familiar with the application	PASS
UAT005	Able to process the steel grading at least three time	PASS
UAT006	Able to obtain the dashboard information to do data analysis	PASS
UAT007	Able to further explore the charts in dashboard	PASS
UAT008	Able to view the application usage summary after processing the steel grading	PASS
UAT009	Able to use the sorting function to view the application usage summary	PASS
UAT010	Able to logout the current account in the application	PASS